Predicting and Comparing Brain Networks

Andrew Murwin
School of Computing and Augmented Intelligence
Arizona State University
Tempe, Arizona USA
amurwin@asu.edu

1 INTRODUCTION

The human brain remains one of the biggest mysteries of the world, acting as the source for all knowledge we have. Each person's brain represents their knowledge, skills, and memories. However, while it serves as the host of these accomplishments, it can also host disorders and disabilities for unfortunate individuals. The biggest issue lies in the fact that we know these illnesses exist in the brain but have yet to come up with an accurate way to identify them. Our motivation for this research was to improve currently held knowledge about the brain through furthering our current understanding of brain networks and how human characteristics may represent themselves in topological differences, as well as provide a basis on which to understand how mental illnesses may affect the brain.

The purpose of this project is to classify individual brain networks based on three label sets: sex, high-math capability, and creativity. From a topical level, there are two standard ways of approaching a multiple classification problem. The first option is the significantly more complex one of creating a single classification model that functions for all three categories. While a binary classifier already runs into issues with the multiple comparisons problem (MCP), which occurs when multiple hypotheses are tested simultaneously, this dataset provides further difficulty with both the math capability and creativity labels being continuous data. The second route is to generate an individual model for each label set. This both prevented MCP and allowed for individual, but comparable, accuracy calculations to be made for each label set. Further scaling and normalization techniques were used to process the data into a consistent form. As we were focused on improving all three label predictions, a method of normalization proposed in [1] was used which was known to mitigate factors known to positively influence predicting the sex of a brain network. As a result of our decision, we were able to generate highly accurate models for both creative and math capability but were unable to generate a solid model for sex prediction.

As generating more useful visualizations was more subjective in nature, we focused more on efficient and easily interpretable visualizations. There are two issues with the dataset that make this challenge significantly more difficult. The first is that the brain networks have no easily interpretable meaning. While the graphs represent connections in the brain, those connections don't directly correlate to anything. As such, general visualizations were produced. These visualizations show their value either when used in conjunction with predictions or other network graphs to highlight patterns. Much of the foundation of these comparisons are based on a previous work that focused on weighted graph comparisons [2].

The structure of the rest of the paper is as follows. Section 2 covers the proposed solution used in our research, including the dataset and methods performed. Section 3 covers the results of the experiments performed. Section 4 concludes the paper, and section 5 summarizes the individual contributions to the project.

2 PROPOSED SOLUTION

The dataset used in the analysis is the 114 brain network dataset (www.andrew.cmu.edu/user/lakoglu/courses/95828/S17/projectsources/brainnetworks.rar). In the dataset, each brain network consists of a 70x70 sparse matrix, with the values representing brain density between different areas of the brain [3]. Additionally, a metadata CSV was included that provided additional data, such as the FSIQ (math), CCI (creativity) and sex label for each brain network [4]. The areas are represented by 70 nodes, indicated by the rows of the graphs. The brain networks themselves are bi-directional, with the weights being the same both ways. As a result, only half of the matrix is used and storing them as sparse matrices saves a significant amount of space. To assist in data processing, each sparse matrix was converted to a CSV file for ease of use. For preprocessing, any data that had a NaN value for any of the three labels was removed, resulting in a final dataset of 113 brain networks.

The data was split into training and testing for use in the model generation. For this, we are using the train-test-split library from sklearn [5] and generate an 80-20 split. We also ensure reproducibility by utilizing a seed for each generation. Following this, we normalize the data using a formula noted in [1] that scaled each node by its column sum.

$$W_{ij} = \frac{a_{ij}}{\sum_{j=0}^{J} a_{ij}}$$

Equation 1: Brain Network Normalization

This equation creates a directed graph through the normalization. One important consideration of using this equation is that, although this scaling was found improve FISQ and CCI prediction, it was also found to reduce the effect of brain size

between men and women, a distinct component used in differentiating sexes based on brain graphs. Following this, we utilized StandardScaler, also from the sklearn library, to scale the data so that every feature was between 0 and 1. Finally, the matrix was flattened for use in feature selection.

Given that there were 4900 features for each network graph, feature selection was necessary to minimize the number of features used, especially given the number of zero features in each network. Principle Component Analysis (PCA) was used to select the features with the most variance in the dataset. PCA work by calculating the eigenvalues and vectors of the covariance matrix, sorting the values based on the eigenvalues, and the using the first N columns to transform the data, where N is the number of features selected. Based on the variance explained by different feature counts, 20 components were selected. Those 20 components explained 54% of the variance with only 0.004% of the data.

As briefly mentioned in the Introduction, we decided to generate individual models for each label set. For each of the label sets, multiple models, distance formulas, and accuracies are tested. Since KNN had the best results, it will serve as the focus for most of this paper. The model used with the most variations is a K-Nearest Neighbors model. For the KNN, the two distance formulas tested were Sum of Squared Distance (SSD) and Angle Between Vectors (ABV).

$$\sum_{n=1}^{N}(x_n - y_n)^2$$

Equation 2: Sum of Squared Distance

Sum of Squared Distance (SSD) calculates the distance as the sum of the squaring of the element-wise subtraction of two feature vectors. The formula is used in a similar way to the Euclidean distance between two vectors, with the difference being the lack of finding the square root of the final sum. Since the distances are in comparison to each other, rather than as values, this final step can be skipped to improve efficiency. SSD provides an accurate view of the distance, but with relatively small decimal scores due to being normalized, two points whose distance is far relative to their scores may appear to have a relatively small SSD. To compensate for this, the Angle Between Vectors is also tested as a distance calculation.

$$\cos^{-1}\left(\frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|}\right)$$

Equation 3: Sum of Squared Distance

Angle Between Vectors (ABV) finds the dot product of the two features vectors after they were converted to unit vectors. ABV provides the alternate view, where distances are proportional to their scores, but lose the overall scale due to normalization. For calculating the accuracy of the KNN, comparable calculations were necessary for comparison between the continuous and binary data.

$$\frac{\sum_{n=1}^{N}\left(1-\frac{|y_n-\hat{y}_n|}{y_n}\right)}{N}$$

Equation 4: Continuous Accuracy Score

For the continuous accuracy, we calculate how close the prediction is to the true value, rather than just if it was classified correctly or not. This decision was made to represent that idea that with continuous data, given a true label of 100, a label of 99 should score significantly higher than a label of 0. In contrast, the binary scoring system is necessary for binomial data. The accuracy is then averaged over the test data predictions.

$$\frac{\sum_{n=1}^{N}(1-|y_n-|\hat{y}_n||)}{N}$$

Equation 5: Binomial Accuracy Score

For the binomial accuracy, we follow a similar pattern to the continuous accuracy. First the prediction is converted into a decimal representing the accuracy for the individual data point. This is done by rounding the binomial prediction. Although the model predicts a decimal between 0 and 1, limiting the KNN to only odd K-values, combined with the nature of binomial data, means that rounding $\hat{y}_n$ will always give the closest binomial class. The accuracy is then averaged over all the predictions.

One of the biggest issues with using a dataset with as many features as this one is the excessive amount of data available. While additional features can lead to additional insights, they can also lead to increased runtime and overfitting [6]. The number of features used, twenty, was original selected based on Figure 1, twenty served as the approximate "elbow" in the graph.

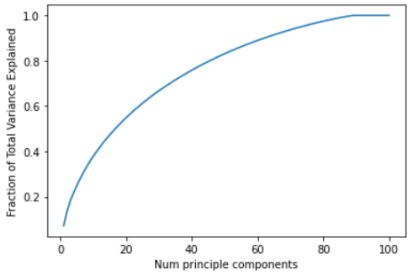

Figure 1: Total Variance Explained at Each Component

Although features were used in the predictions used in this paper is based on an approximate selection, additional testing was carried out into determining an optimal feature selection count. Similar to how all combinations are tested to determine the optimal distance type and K-Neighbors pair, each combination of distance type, K neighbors, and feature count is averaged over 10 runs to empirically determine the best result.

In addition to the KNN, we also experimented with a K-Means model. We used sklearns built in KMeans clustering functions to complete this [5]. The data was processed the same way as the KNN, and the optimal K-value was determined by the elbow method, in which the best K value is the distinct point were

increasing the K value any more results in diminishing returns on accuracy. The final K value used was K=6.

The last model evaluated was a linear regression model, used to identify distinct linear correlations present within the brain connections. We used sklearn's built in Linear Regression to model this [5]. The same data preprocessing and processing done in both the KNN and K-Means was used. The model was then fit to the training data and used to predict the test data. For the evaluation of the performance, the same accuracy calculations used for the KNN were used for their respective label sets. The root mean squared error was also calculated as an additional performance metric.

Three separate techniques were used to create the final visualizations that accompanied the predictions. The first was a standard matrix graph. The weights were converted to a white-red-green color scale and graphed. High values are green to contrast with the mostly red graph, creating distinctions between the densities. The second graph is a circular weighted node graph, with the weights represented as the opacity of the edges between the nodes. The final layout is a spring layout, where the weight is represented by the distance between the nodes, where a high weight corresponds to a short distance, effectively clumping highly connective nodes in the center of the graph. Each of the three graphs are examined to determine their viability of being used in network comparisons.

3 RESULTS

For all runs used in results, the PCA was used to select 20 features from the original brain networks. The accuracies for the individual models are calculated using the formulas described in the Proposed Solution above. For each combination of distance type and K Neighbors, the model was run ten times, and the accuracies were averaged over the ten runs.

Table 1: Math Capability Extrema Accuracies

Distance Type	K Neighbors	Accuracy
SSD	35	0.9033
	11	0.8991
ABV	31	0.9052
	9	0.8989

Overall, the accuracies for the math predictive models are relatively consistent across the 28 runs, with an average accuracy of 90.2% and peak accuracy of 90.52% using 31 neighbors and the ABV distance calculation. In predicting math capability given a brain network, both the distance type and the number of neighbors compared against had minimal impact on the accuracy of the prediction. While there was a definite correlation, comparing the extrema accuracies gives a difference of only 0.63%.

Table 2: Creativity Capability Extrema Accuracies

Distance Type	K Neighbors	Accuracy
SSD	35	0.9189
	9	0.9144
ABV	15	0.9193
	9	0.9167

Similar to the math capabilities, the creative capabilities remain relatively consistent across the 28 runs. Overall, the average accuracy stands at 91.8%, with a peak accuracy of 91.9% when calculating with 15 neighbors and the ABV distance formula. Also like the math capabilities, the accuracy of the creative capabilities is not significantly impacted by the distance type or number of neighbors used. When comparing the extrema, there is only a 0.49% difference in accuracy, even less than the difference in math capability. Although both the math and creativity showed strong correlations between their scores and the brain networks, the sex prediction did not.

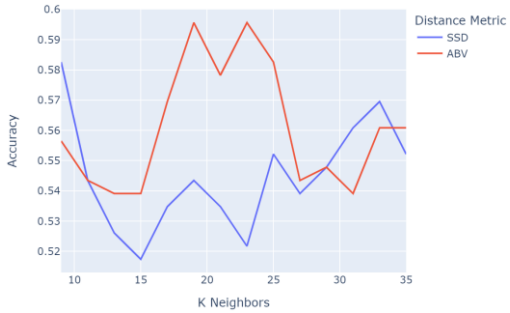

Figure 2: Sex Prediction Accuracies

With an average accuracy of only 55.3%, the peak accuracy was 59.6%, using 19 neighbors and the ABV distance calculation. In contrast to the other two label sets, the variance in the accuracy was significantly higher. Comparing the extrema shows a significant difference in accuracy of 7.85%. Interestingly though, while there was significant variance with both the distance type and the number of neighbors used, there was no consistent correlation between them. As seen in Figure 1, the ABV distance metric did perform best for intermediate range but performed worst than the SSD for both small and large K values. While there exists some correlation as the accuracy is still better than a random guess, the correlation is not strong enough to be reliably predicable. Ultimately this result was not surprising because, as discussed previously, the normalization equation used was known to mitigate a prevailing factor in determining sex based on brain networks [1].

After the completion of the main research, additional research was completed to determine the optimal PCA selection for each label set. To accurately determine the optimal combinations for each label, 2800 models were generated. The distribution was each of the ten PCA options, with both distance types, run with every odd

number of neighbors from 9 to 35, averaged over ten independent seeded runs.

Table 3: Best Average PCA per Label

Label	PCA	Accuracy
Creativity	40	0.9200
Math	20	0.9021
Sex	10	0.5896

In Table 3, each PCA was run over the label set to produce an average accuracy and provide a holistic look as to the best PCA for each set. For the creativity, a PCA of 40 features was the best, although the average only performed 0.2% better than the average at PCA 20, which can be considered negligible. The math label showed a PCA of 20 be the best on average, which was the same feature selection used in the experimentation. The sex label had the most interesting result out of the three. On average, the PCA of 10 features had a 3.7% improvement over the PCA of 20.

Table 4: Best Individual PCA per Label

Label	PCA	Dist. Type	K Neighbors	Accuracy
Creativity	40	ABV	15	0.9213
Math	20	ABV	31	0.9052
Sex	90	ABV	29	0.6261

For each label/PCA combination, the best result was also found. For the creativity, the PCA selection was 40 features, resulting in an accuracy of 92.13%. In comparison to the best accuracy with a PCA of 20, there was only a 0.2% improvement. Overall, this level of improvement can be considered relatively negligible. For the math prediction, a PCA of 20 was also determined to be the optimal selection, so there is no comparison necessary. The sex prediction had a significantly higher difference in comparison, with a best accuracy of 62.61% and difference of 3.01%. Something important to note is that, while there is a larger difference, its level of difference is inconsistent due to the fluctuations in accuracy as shown in Figure 2.

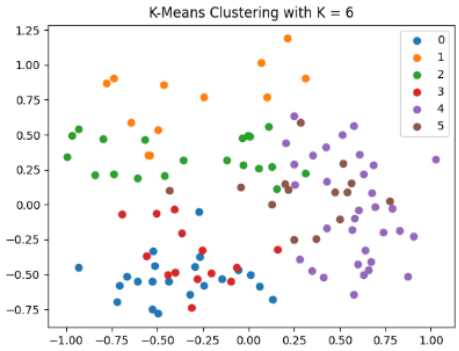

Figure 3: Sample K-Means Clustering

Using K=6, we calculated the K-Means for the dataset, clustering on the 20 features selected with the PCA. While the clusters don't appear to be distinct, much of the overlap between the clusters can be explained by the nature of flattening 20-dimensional points down to 2D space for the plot. Unfortunately, K-Means being a clustering technique, combined with the high dimensionality of the data, means that the clusterings are difficult to classify.

Table 5: Linear Regression Performance Metrics

Label	Accuracy	RMSE
Creativity	0.8917	16.73
Math	0.8711	24.56
Sex	0.5652	0.89

Like the KNN, the creativity and math predictions produced a high accuracy while the sex prediction did not. In addition to the accuracy, the Linear Regression also uses the root mean squared error (RSME) to estimate analyze the estimate of the standard deviation in error. The math prediction model scored roughly 25 points off on average, while creativity performed significantly better at roughly 17 points off. Due to sex being a binary classification rather than a continuous label set, the RSME is less useful.

The three separate visualizations described in the Proposed Solution were created both for the purpose of individual graph analysis and graph comparison.

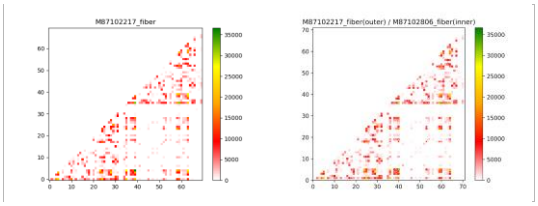

Figure 4: Weighted Density Matrix

For comparing densities of connections within the graphs, the matrix graph serves as the most apparent way. In Figure 4a, the graph shows the density of a sample brain network. Although half of the graph remains unused, there is a clear distinction between it and the node connections. The color scheme used also makes dense connections immediately prevalent. It also has the advantage of the nodes retaining their numbered labels for easy identification. Figure 4b shows a comparison between two graphs. While trying to distinguish between the two graphs can be difficult, it still shows overall trends similar between the two graphs. Due to its condensed nature, the matrix graph serves as a solid way to compare similar nodes.

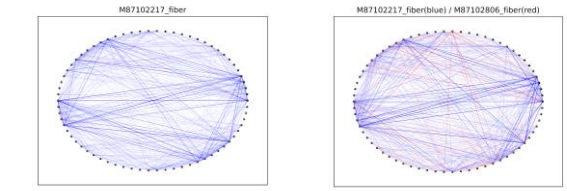

Figure 5: Circular Node Graphs

For comparing node connections between graphs, the circular node graph serves as the most apparent way. On the node graphs, each line symbolized a connection between two nodes, with the thickness of the connection representing the density of the connection in the brain. For the circular node graphs, the isolated graphs hold little meaning as it lacks any labels or quantitative measurements. When comparing two graphs though, using the density of the connections means that each line connecting nodes ends up the color of the stronger connection between the two. With the addition of necessary node numbers, this serves as a solid way to compare two different nodes.

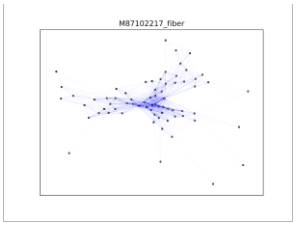

Figure 6: Spring Node Graph

The spring style of graph show in Figure 6 shows a different perspective for individual graphs. Through using the inverse connection densities to represent the distances, the graph represents corresponding nodes within a graph, showing the nodes that best represent the graph. Given the spring nature of the graph though, using it to compare two graphs is impossible because the distances between nodes differs between graphs, meaning the nodes are not in the same place and no longer have a way to be compared.

4 CONCLUSION

In conclusion, we were able to develop accurate KNN models to predict both the creativity and math capabilities of a brain networks. KNN was chose over K-Means and Linear Regression due to its improved accuracy and additional metrics available. Unfortunately, an accurate sex prediction model was unable to be created, likely due to the data transformation we performed during the preprocessing stage. Additionally, we were able to generate multiple visualizations that let us easily compare multiple brain networks. In my personal work, I was able to determine the optimal PCA for each of the three label sets to further improve accuracy.

5 CONTRIBUTIONS

This project was completed by a team of five people:

- Andrew Murwin (amurwin@asu.edu)
- Derek Deng (dwdeng@asu.edu)
- Benjamin Hay (bchay@asu.edu)
- Vincent Le (vtle5@asu.edu)
- Breanna Seitz (bdseitz1@asu.edu)

My contribution to the project were performing research, processing data conversion, and coding the models. I wrote the script for converting the MAT files to CSV for they would be in an easier to process format. To meet the requirements of manual implementation, I was responsible for coding and validating both the PCA feature selection function and the KNN model. I was also in charge of running all of the KNN models and performing analysis on the predictions.

For this portfolio project, I did an in-depth examination on the PCA, and the selection of features used. In the original paper, the number of features selected was twenty, and was selected based on a visual interpretation of the cumulative variance explained graph. Instead of doing a visual selection, I tested a series of feature counts to determine which one was the optimal one for each label set.

REFERENCES

[1] Duarte-Carvajalino, Julio M., et al. "Hierarchical Topological Network Analysis of Anatomical Human Brain Connectivity and Differences Related to Sex and Kinship." NeuroImage, vol. 59, no. 4, 15 Feb. 2012, pp. 3784–3804, www.sciencedirect.com/science/article/pii/S1053811911012687, 10.1016/j.neuroimage.2011.10.096. Accessed 15 Sept. 2022.

[2] Alper, Basak, et al. "Weighted Graph Comparison Techniques for Brain Connectivity Analysis." www.microsoft.com, 27 Apr. 2013, research.microsoft.com/en-us/um/people/nath/docs/brainvis_chi2013.pdf. Accessed 15 Sept. 2022.

[3] Kulkarni, Vivek, et al. "Sex Differences in the Human Connectome." Lecture Notes in Computer Science, 28 Nov. 2012, pp. 82–91., https://doi.org/10.1007/978-3-319-02753-1_9.

[4] "Index of /User/Lakoglu/Courses/95828/S17/Projectsources/brainnetworks.rar." www.andrew.cmu.edu, www.andrew.cmu.edu/user/lakoglu/courses/95828/S17/projectsources/brainnetworks.rar. Accessed 15 Sept. 2022.

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[6] Hawkins, D.M. 2003. The Problem of Overfitting. Journal of Chemical Information and Computer Sciences. American Chemical Society (ACS)., https://doi.org/10.1021/ci0342472