National Research University
"Higher School of Economics"

Faculty of Computer Science
Master's programme in Data Science
Department of Complex Systems Modelling Technologies

Master's thesis

Feature engineering and dimensionality reduction for structural connectome classification

Student: Dmitry Petrov

Group: m14NoD TMSS

Scientific advisor: Leonid Zhukov

Moscow
2016

Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Магистерская программа «Науки о данных»
Кафедра «Технологии моделирования сложных систем»

Выпускная квалификационная работа

Генерация признаков и снижение размерности в задаче классификации структурных коннектом

Студент: Дмитрий Петров

Группа: м14НоД ТМСС

Научный руководитель: Леонид Жуков

Москва
2016

Abstract

In this work we investigated Autism Spectrum Disorder vs Typically Developing classification task based on structural connectomes. Using combination of different weighting schemes, topological normalizations and graph metrics we constructed about 500 feature sets and tested them using selected classifiers and cross-validation techniques. We found features obtained with combination of weighting by distance and topolgical normalization which achieved 0.8 ROC AUC score. It is comparable with results described in recent studies. We also tried dimensionality reduction on the best obtained features, but didn't find simple geometry in our data.

Аннотация

В нашей работе мы решали задачу различения аутизма и нормы на основе структурных коннектом. Используя различные схемы взвешения, топологической нормализации коннектом и различных графовых метрик, мы сконструировали около 500 различных наборов признаков. Мы проверили их для выбранных классификаторов (линейных и на основе решающих деревьев) с помощью процедур перекрестного контроля. Мы обнаружили набор признаков, полученный с помощью комбинации нормировки на расстояние и топологической нормировки, который существенно улучшает качество классификации, измеренное как площадь под ROC-кривой. Лучшая модель на лучших признаках дала ROC AUC около 0.8, что на уровне опубликованных работ по этой теме. Мы также попробовали общепринятые методы снижения размерности на лучших признаках, но не обнаружили простой геометрии в наших данных.

Contents

1 Introduction **1**

2 Connectome machine learning overview **3**

3 Problem statement and data **4**

4 Connectome preprocessing **6**
4.1 Weights 7
4.2 Normalizations 7

5 Connectome featurizing **9**
5.1 Bag of edges 9
5.2 Undirected graph metricss 10
5.2.1 Local 10
5.2.2 Global 12
5.3 Directed graph metrics 15
5.3.1 Common metrics 15
5.3.2 Custom metrics based on random walk logarithms 15
5.4 Baseline features 16

6 Machine learning pipeline **17**
6.1 Classifiers 17
6.2 Performance metric and hyperparameters grid search 18
6.3 Dimensionality reduction 18
6.4 Programming tools 19

7 Results **19**
7.1 Best features and classifiers 19
7.2 Weighting and normalization effects 20
7.3 Dimensionality reduction 25

8 Conclusion and discussion **26**

9 Acknowledgements **27**

10 Bibliography **28**

1 Introduction

Connectome graphs are discrete mathematical models which represent structural or functional connections between anatomically distinct brain areas [1]. A recent rapid growth of connectome analysis is driven by the assumption, that structural properties of brain networks captured by connectomes can provide new insight into the nature of disease-related (or treatment-caused) changes in brain structure and functioning. However, most of the studies employ group-based comparison of network features (e.g., disease versus norm). In this case even when group differences in network features are found, they might not be predictive for individual connectomes.

To date diagnostic of psychiatric disorders and neurodegenerative disorders based on neuroimaging data is far from being accurate (for a recent reviews on this topic, see [2] and [3]). This is true for many conditions, including autism spectrum disorders considered in our study. Most of the studies reported to date are based on relatively small samples and mostly incorporate the logics of group-based comparison without any cross-validation procedures (for review of findings specific to autism spectrum disorders, see [4], [5]). Even if the use machine learning techniques they suffer from overfitting and poor choice of performance metrics and hyperparameters.

In our study we used machine learning algorithms with cross-validation procedures to investigate structural differences between typically developing (TD) and autism spectrum disorder (ASD) subjects. In each step we build our models on train datasets and then validate them on unknown test data. This provided an insight of how well our models will behave on newly coming observations, while predictive power of the features based on whole-group analysis remains unclear [6].

We generated about 500 different datasets for ASD vs TD connectome classification task, using combinations seven connectome weighting schemes, six topological normalizations, and set of graph metrics (common undirected, common directed and custom directed). Pipeline scheme (excluding dimensionality reduction) is shown on Figure 1.

On each of these datasets we performed hyperparameter grid search for chosen classifiers: Logistic Regression, Linear SVM, Stochastic Gradien Descent with modified Huber loss, Random Forest, Adaboost and Boosted Decision Trees. For the 50 best combinations of classifiers and features we reported mean and std of ROC AUC distributions across 50 different 10-fold splits with fixed random states.

We further investigated effects of combined normalization scheme incorporating geometric and topological normalization of structural connectomes. We test our approach by classifying autism spectrum disorder versus typical development connectomes with linear and tree-based classification methods. We showed that neither geometric nor topological normalization alone improve classification performance. However, a significant performance increase is achieved using their combination. We further improve leave-one-out cross-validation results and report relative zone importance by adjusting l_1-regularization ratio

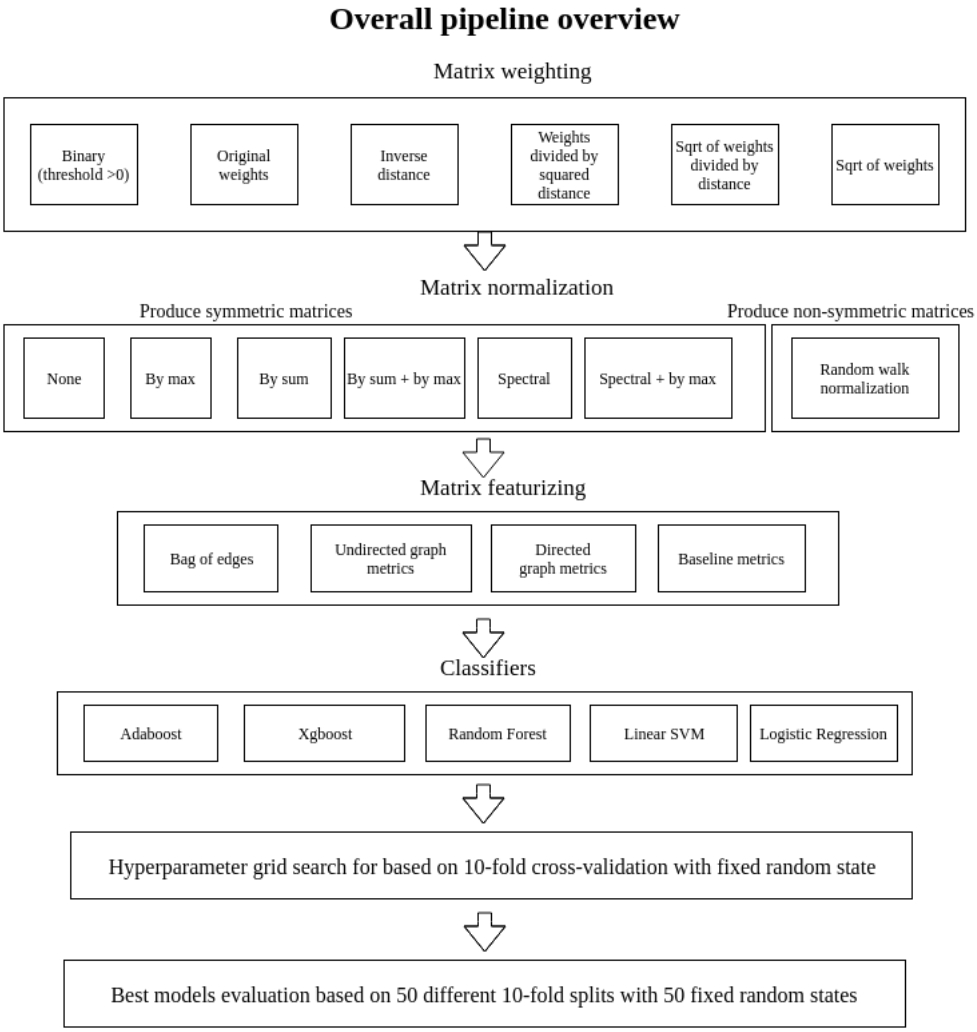


Figure 1: Machine learning pipeline scheme

for linear SVM.

We also tried all common dimensionality reduction methods on the best feature set – Principal Component Analysis with linear and with radial basis function, sigmoid, polynomial (degrees 2-5) and cosine kernels, Multidimensional Scaling, Spectral Embedding, Isomap and t-distributed Stochastic Neighbor Embedding (TSNE). It didn't yield any simple geometry in our data.

There are several limitations of this study. First of all, sample size is quite small for the results to be conclusive. Although the groups of ASD and TD subjects are relatively large compared to similar studies published to date, it is highly desirable to replicate the analyses on larger samples. This is primarily due to high dimensionality of the task at hand. For example, analysis based on the bag of edges involved tens of thousands of features with only 94 observations. We employed statistical techniques of feature selection and cross-validation recommended for such situations, but larger sample size would be the best recipe to improve the analysis.

Second, there are certain methodological aspects of the data that should be noted. For example, parcelling brain volume into nodes was quite unusual for structural network analysis. Usualy DTI-based networks are constructed on atlas-based zones or their partitions. Thus, results obtained for this functional-connectivity-based parcellation scheme need to be reproduced on networks with alternative parcellation of brain zones.

Finally, we intentionally left aside any neurological interpretation of our findings. Our study is purely exploratory in nature, and our analysis is blind to substantial meaning of the observed differences. We output the labels of brain zones for which significant differences in local network characteristics between ASD and TD groups are found, but do not go any further in interpreting our results. Further investigation on additional datasets and classification tasks is required to generalize our conclusions.

2 Connectome machine learning overview

Machine learning is relatively new in the neuroimaging data analysis. Very comprehensive recent overview on the topic was done by Ababshirani et al. [2] Authors examined more than 200 studies focused on differentiation of various mental and neurodegenerative disorders such as Alzheimers' disease, schizophrenia, depressive disorders, autism spectrum disorders and attention-deficit hyperactivity disorder. For each of them they gave brief feature description, classification methods and performance.

It turned out, that there are several common pitfalls of machine learning in neuroimaging. First, overfitting. For example, studies often select features with statistic tests on the whole sample which can lead to inflated results. Figure 2 shows screenshot fromm Ababshirani et al. review with scatter plot of accuracy vs. sample size in 200 reviewed studies. We see that classification performance tend to drop with the sample size increase which can indicate possible overfit in low sample models.

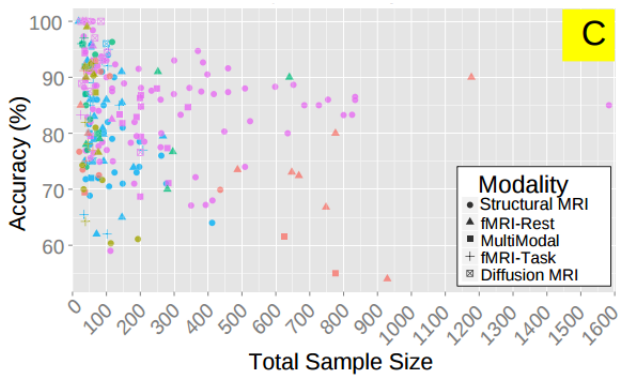


Figure 2: Screenshot from review by Arbabshirani et al. [2]. It shows overall accuracy vs. overall sample size scatter plot of 200 machine learning on neuroimaging data studies.

Second pitfall is a poor choice of classification performance metrics. For example, most studies report only overall accuracy, even for imbalanced datasets. Third, poor hyperparameter optimization. Some of the reviewed studies used only default parameters of implemented models. We tried to avoid all these pitfalls in our study (see 'Machine learning pipeline' section).

ASD vs TD classification studies has all pitfalls mentioned above. Figure 3 shows screenshot of a table from Arbabshirani review with results of 20 studies concerning ASD vs TD classification task. We see that only two of them use metric that differs from accuracy and also that bigger sample size studies tend to have lower performance which agian can indicate possible overfitting on smaller samples.

3 Problem statement and data

We consider the two classes 'typically developing' (TD) and 'autism spectrum disorder' (ASD) based on diagnosis, and classify individual brain networks based on constructed features described in the following chapters.

We use UCLA autism dataset publicly available for download at the UCLA Multimodal Connectivity Database [8], [9]. The dataset includes DTI-based connectivity matrices of 51 ASD subjects (6 females) and 43 TD subjects (7 females). Average age (age standard deviation) were 13.0 (2.8) for ASD group and 13.1 (2.4) for TD group. To control for possible confounding effects, we included both age and sex as features in all analyses. Details on participants recruitment, DTI scans acquisition and construction of connectivity matrices can be found in the paper by Rudie et al. [7]. In this section, we only focus on some key aspects of the pipeline.

DTI scans were acquired on a Siemens 3T Trio. The DTI sequence consisted of 32 scans with different diffusion-weighted directions (b=1000 s/mm^2), three

Table 5
Summary of 20 MRI-based ASD classification studies. N/A indicates information was not available or could not be found.

Modality	Disorder	Features	# of features	Classifier	Number of subjects	Overall accuracy	Reference
dMRI	ASD	FA and MD of selected ROIs	18	SVM	TDC = 30, ASD = 45, Total = 75	80%	Ingalhalikar et al. (2011)
fMRI (social interaction task)	ASD	Activation of selected voxels processed by factor analysis	4 factors	Gaussian naïve Bayes	HC = 17, TDC = 17, Total = 34	97%	Just et al. (2014)
fMRI (two language tasks and a Theory-of-Mind task)	ASD	AG, MPFC and PCC based FC maps	N/A	Logistic regression	TD = 14, ASD = 13, Total = 27	96.0%	Murdaugh et al. (2012)
rsfMRI	ASD	ICA components of rsfMRI	10 components	Logistic regression	TDC = 20, ASD = 20, Total = 40	78.0%	Uddin et al. (2013)
rsfMRI	ASD	FC among ROIs	Variable	Logistic regression and SVM (best results)	TD1 = 59, TD2 = 89 ASD1 = 59, ASD2 = 89, Total = 296	76.7%	Plitt et al. (2015)
rsfMRI	ASD	FC among 90 ROIs	4005	Probabilistic neural network	TDC = 328, ASD = 312, Total = 640	90%	Iidaka (2015)
rsfMRI	ASD	Functional connectivity among 220 ROIs	24,090	Random forest	TDC = 126, ASD = 126, Total = 252	91%	Chen et al. (2015)
rsfMRI	ASD	FC among ROIs	26,393,745	Thresholding	TD = 40, ASD = 40, Total = 80	79.0%	Anderson et al. (2011)
sMRI	ASD	Thickness and volumetric of ROIs along with interregional features	N/A	Multi-kernel SVM	HC = 59, ASD = 58, Total = 117	96.3%	Wee et al. (2014)
sMRI	ASD	Voxel-wise GM and WM maps	N/A	SVM	TD = 24, ASD = 24, Total = 48	92.0%	Uddin et al. (2011)
sMRI	ASD	GM volume map	N/A	SVM	HC = 40, ASD = 52, ASD-Sib = 40	80.0–85.0%	Segovia et al. (2014)
sMRI	ASD	Regional thickness measurements extracted from SBM	7	Logistic model trees	HC = 16, ASD = 22, Total = 38	87%	Jiao et al. (2010)
sMRI	ASD	Morphometric features of selected ROIs	314	SVM	HC = 20, ASD = 21, Total = 41	74% (AUC)	Gori et al. (2015)
sMRI	ASD	GM and WM maps	>10,000	SVM	HC = 22, ASD = 22, Total = 44	77%	Ecker et al. (2010b)
sMRI	ASD	Volumetric and geometric features of selected cortical locations	5 features from each ROI	SVM	HC = 20, ASD = 20, Total-40	85%	Ecker et al. (2010a)
sMRI	ASD	Gray maps from VBM-DARTEL	200	SVM	TDC = 38, ASD = 30, Total = 76	80.0% (AUC)	Calderoni et al. (2012)
sMRI	ASD	Volumetric measures and cerebellar vermis area	9	Discriminant function analysis	TDC = 15, ASD = 52, Total = 67	92.3–95.8%	Akshoomoff et al. (2004)
fMRI-Task and dMRI	ASD	Causal connectivity weights, FC values and FA values	19	SVM	TDC = 15, ASD = 15, Total = 30	95.9%	Deshpande et al. (2013)
sMRI, dMRI and MRS	ASD	Cortical thickness, FA and neurochemical concentration	3	Decision tree	TD = 18, ASD = 19, Total = 37	91.9%	Libero et al. (2015)
sMRI and rsfMRI	ASD	Volume of selected subcortical regions, fALFF, number of voxels and Z-values of selected regions and global VMHC voxel number	22	Random tree classifier	TDC = 153, ASD = 127, Total = 280	70.0%	Zhou et al. (2014)

Figure 3: Screenshot from review by Arbabshirani et al. (NeuroImage, 2016). It shows summary of 20 machine learning studies of ASD vs TD classification based on neuroimaging data.

scans with no diffusion sensitization at b=0, and six scans at b=50 s/mm^2. An in-plane voxel dimension was 2×2mm with 2-mm thick axial slices, and total scan time was 8 min 1 s. Subjects with excessive motion artifacts were not included in the final sample. Mean and maximum relative motion did not differ in ASD and TD groups. Motion and eddy current correction was performed on the diffusion-weighted images using 'eddy correct' in FMRIB's Diffusion Toolbox.

Whole brain deterministic tractography was performed on voxelwise fractional anisotropy (FA) values using the fiber assignment by continuous tracking (FACT) algorithm [10] in Diffusion Toolkit [11]. Tractography was carried out with relaxed constraints: maximum turn angle was set at 50^o, and no FA cutoff was applied. This means that the algorithm implied somewhat boosted likelihood of detecting longer fibers between spatially distant areas. Fibers were smoothed using a spline filter; fibers shorter than 5 mm were excluded from connectivity count.

As mentioned in the previous section, the choice of brain volume parcelling scheme is an important step in connectivity matrix construction. It determines the number and location of the vertices of brain networks and thus the structure of the graph to be analysed. For this dataset, definition of nodes was somewhat unusual for DTI-based networks that commonly use atlas-based or voxel-wise parcellation approaches. Instead, connectivity matrices in this dataset were created using parcellation scheme recently proposed by Power et al. [12] based on a large meta-analysis of fMRI studies combined with whole brain functional connectivity mapping. This approach produced 264 brain regions and thus 264×264 connectivity matrices. For the purposes of this study, we take this parcellation scheme as is and do not discuss its potential benefits and caveats.

The number of streamlines connecting each pair of regions was used to set the respective edge weights. Thus, the resulting adjacency matrices were symmetric and weighted, with larger weights indicating more streamlines detected between the respective brain regions. Following recommendations by Jones et al. [13] we prefer not to use the term 'fiber count' because the number of streamlines detected by tractography algorithm does not necessarily correspond to the number of actual white matter fibers.

4 Connectome preprocessing

Connectome datasets usually include non-normalized DTI connectivity matrices. But number of detected streamlines is known to vary from individual to individual and can also be affected by fiber tract length, volume of cortical regions and other factors. Normalization of connectivity matrices is highly recommended prior to any analyses (e.g., see [14] and [15]).

There is no consensus on how to do it. There seems to be two major approaches to it. The first approach directly involves geometric measures such as volume of the cortical regions or physical path lengths between the regions [14], [15]. The second requires purely topological normalizations (e.g., see [16] and [17]). We used several approaches in this study.

4.1 Weights

We used six different weighting schemes. First, we used weights a_{ij} from original connectomes that are proportional to the number of streamlines detected by tractography.

Second, we obtained binary weights:

$$a_{ij}^{b} = 1 \text{ if } a_{ij} > 0, 0 \text{ else.} \tag{1}$$

The motivation here is following: maybe particular weights are not important and the only important thing is existence of link between nodes.

Third, we used weighting by squared distance between nodes:

$$a_{ij}^{weighted} = \frac{a_{ij}}{l_{ij}^2}. \tag{2}$$

where l_{ij} is the Euclidean distance between centers of regions i and j. We used MNI coordinates of zone centers provided by the authors of the dataset to obtain the reasonable proxy of the distances between brain regions; these distances are the same for all subjects. This weighting scheme has some physical intuition behind: if we consider original weights a_{ij} as 'area of the tract wire', than this weighting scheme provides us 'resistance of the wire', which proportional to area and inversely proportional to distance. Distances were squared to produce non-dimensional quantity.

Fourth, we used square root of previous weights:

$$a_{ij}^{rootweighted} = \frac{\sqrt{a_{ij}}}{l_{ij}}. \tag{3}$$

This weighting scheme was introduced as another version of previous weighting scheme.

Fifth, we used square root of original weights.

$$a'_{ij} = \sqrt{a_{ij}}. \tag{4}$$

Sixth, we used square root of original weights:

$$a_{ij}^{invdist} = \frac{1}{l_{ij}}. \tag{5}$$

Weightings 5-6 were introduced to test alone effects of root weights/inverse distances.

4.2 Normalizations

Number of detected streamlines in connectomes is known to vary from individual to individual and can also be affected by fiber tract length, volume of cortical regions and other factors. Normalization of connectivity matrices is highly recommended prior to any analyses (e.g., see [14] and [15]).

There is no consensus on how to normalize the streamline count. There seems to be two major approaches to it. The first approach directly involves geometric measures such as volume of the cortical regions or physical path lengths between the regions [14], [15]. The second requires purely topological normalizations (e.g., see [16] and [17]).

Topological normalizations themselves can differ in what effects they aim to eliminate. In the simplest case, streamline count for each pair of regions is normalized by the total number of streamlines in the entire brain, thus reducing variability among the connectivity matrices due to differences in the total number of detected streamlines. More sophisticated procedures involve weighting each edge by the arithmetic mean or geometric mean of the total number of streamlines leaving its adjacent regions. Yet another approach aims to interpret weights as probabilities of coming from one region to another and thus produces non-symmetric matrices as a result of normalization.

Alongside with pure weights (non normalization at all) we used six topological normalizations schemes: First, by sum of all matrix elements

$$w_{ij}^{by\ sum} = \frac{a_{ij}}{\sum_{ij} a_{ij}}. \tag{6}$$

Second, this normalization was modified by dividing each normalized matrix by its maximum value, as recommended by [17]. This further reduces differences between different connectivity matrices and allows comparison based on purely topological characteristics.

$$w_{ij}^{sum\ by\ max} = \frac{w_{ij}^{sum}}{\max_{i,j} w_{ij}^{sum}}. \tag{7}$$

Third, we used normalization by maximum alone.

$$w_{ij}^{by\ max} = \frac{a_{ij}}{\max_{i,j} a_{ij}}. \tag{8}$$

Fourth, we applied weighted communicability normalization [**?**] to each of the three weighted sets of connectomes:

$$w_{ij}^{spectral} = \frac{a_{ij}}{\sqrt{d_i d_j}}, \tag{9}$$

where a_{ij} is weight of edge between nodes i and j; d_i is weighted degree of node i.

This normalization scheme produces matrices closely related to normalized graph Laplacians. Indeed, Eq. (2) in a matrix form is $W^{normed} = D^{-1/2}AD^{-1/2}$, where A is a matrix of edge weights and D is a diagonal matrix of node degrees, while the normalized graph Laplacian is obtained by $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$ with I being an identity matrix. In particular, this means that matrices normalized by the geometric mean of the adjacent degrees have the same spectral properties as normalized Laplacians.

Fifth, we combined this normalization with normalization by maximum:

$$w_{ij}^{spectral\ by\ max} = \frac{w_{ij}^{spectral}}{\max_{i,j} w_{ij}^{spectral}}. \tag{10}$$

Finally, we used random walk normalization:

$$p_{ij} = \frac{a_{ij}}{\sum_j a_{ij}}. \tag{11}$$

This normalization produces from symmetric graph adjacency matrix non-symmetric random walk on graph matrix. Physical intuition here is simple: maybe we should look on probabilities of signal transmission in the network determined by network connections. This consideration helped us to make some of the most successful features (see part about custom directed graph metrics).

5 Connectome featurizing

Here we describe features generated to implement supervised machine learning techniques. Here and forth V is the set of all nodes in our network G and $n = |V|$ the number of nodes. E is the set of all edges in the network, and $m = |E|$ is number of them. Edges (i, j) have weights w_{ij}, which are normalized $(0 \leq w_{ij} \leq 1)$. a_{ij} is the connection status between nodes i and j. $A^W = w_{ij}$ and $A = a_{ij}$ are weighted and unweighted adjacency matrices.

Please note: here and forth by 'weights' we mean weights obtained through combination of weighting schemes and topological normalizations, mentioned above.

5.1 Bag of edges

The simplest method to produce features is to treat matrix as a vector. Each weighted edge acts as a feature, and no relationships between them are taken into account. For 264×264 connectivity matrices this method produces 34,716 features (because DTI connectivity matrices are symmetric with zero diagonal). Please note, that for random walk normalization bag of edges are twice as large, because it produces non-symmetric random matrix from symmetric graph adjacency matrix.

In addition to bag of edges on weights, we also used bag of edges of shortest path matrices. For random walk matrices shortest paths were calculated using negative logarithm of probabilities:

$$w_{ij}^{\ln} = -\ln p_{ij}. \tag{12}$$

Intuition here is simple: in this case distances can be correctly added and shortest paths have meaningful interpretation as paths with highest probability of signal transmission from node i to j. Logarithm was negative because we used

Dijkstra shortest path algorithm for calculations which requires non-negative graph weights.

Finally, we used random walk matrices, which we calculated as follows:

$$W_\alpha = (I - \alpha A)^{-1}. \tag{13}$$

Intuition here that maybe signal decays with each transmission with coefficient α, so matrix W is limit all such 'decaying' random walks.

5.2 Undirected graph metricss

To capture properties of overall network structure, we also compute local node-based and global graph metrics. We use weighted metrics whenever possible and unweighted when there is no ready-made solution (see Programming tools section). For a discussion of possible metrics in brain connectivity analysis we recommend [18].

5.2.1 Local

Weighted degree

$$k_i^W = \sum_{j \in V} w_{ij}. \tag{14}$$

Average weighted neighborhood degree

$$k_{nn,i}^W = \frac{\sum_{j \in V} w_{ij} k_j^W}{k_i^W}. \tag{15}$$

Closeness centrality Inverse of average weighted distance to other nodes:

$$(l_i^W)^{-1} = \frac{n-1}{\sum_{j \in V, j \neq i} d_{ij}^W}, \tag{16}$$

where d_{ij}^W is weighted shortest path length between nodes i and j. Note that because we deal with weighted networks, normalization by $(n-1)$ does not guarantee maximum centrality value of 1.

Betweenness centrality Quantifies the number of times a node acts as a bridge along the shortest path between two other nodes (Freeman [19]). We use the weighted version with shortest paths being computed for the weighted graph:

$$b_i = \frac{2}{(n-1)(n-2)} \sum_{\substack{h,j \in V \\ h \neq j, h \neq i, j \neq i}} \frac{\rho_{hj}(i)}{\rho_{hj}}, \tag{17}$$

where ρ_{hj} is the number of weighted shortest paths between h and j, and $\rho_{hj}(i)$ is the number of weighted shortest paths between h and j that pass through i. Again, note that because we deal with weighted networks, normalization by $\frac{2}{(n-1)(n-2)}$ does not guarantee maximum centrality value of 1.

Eigenvector centrality Gives high values to vertices that are connected to many other well-connected vertices (Bonacich, 1986):

$$ec_i = v_i, \tag{18}$$

where v is eigenvector, corresponding to the largest eigenvalue of A^W.

Due to the theorem of Perron–Frobenius, there exists an eigenvector of the maximal eigenvalue with only nonnegative (positive) entries.

The only caveat here is how the entries of the eigenvector are normalized. Most commonly, they are simply divided by the maximal value so that the maximal eigenvector centrality within each graph is always 1. However, this is somewhat contrintuitive (maximum should ideally be reached for certain graph configuration). Hence, strictly speaking, eigenvector centrality gives a kind of 'relative centrality' within the graph rather than an absolute value with respect to what is possible.

Weighted number of triangles

$$t_i^W = \frac{1}{2} \sum_{j,h \in V} (\hat{w}_{ij} \hat{w}_{ih} \hat{w}_{jh})^{\frac{1}{3}}. \tag{19}$$

Important. $\hat{w}_{ij}$ stands for normalized weights here: all weights are divided by the maximum weight.

Clustering coefficient The problem here is that there are different possible generalizations of the clustering cooefficient to weighted graphs. The one used here is described in Saramäki et al. (2007). This is the formula implemented in NetworkX, and also the one given in Rubinov and Sporns (2010):

$$c_i^W = \frac{2t_i}{k_i(k_i - 1)}, \tag{20}$$

where t_i^W is the weighted number of triangles for the node i.

Eccentricity The eccentricity $ecc(i)$ of node i is the greatest weighted shortest path length from node i to any other node:

$$ecc_i^W = \max_{j \in V, j \neq i} d_{ij}^W. \tag{21}$$

Characteristic path length Average distance between node i and all other nodes:

$$l_i^W = \frac{\sum_{j \in V, j \neq i} d_{ij}^W}{n - 1}, \tag{22}$$

where d_{ij}^W is weighted shortest path length between nodes i and j. Note that this is the inverse of closeness centrality (and vice versa).

Efficiency Weighted node efficiency is computed as the mean inverse shortest path length from node i to all other nodes:

$$e_i^W = \frac{\sum_{j \in V, j \neq i} (d_{ij}^W)^{-1}}{n-1}, \tag{23}$$

where d_{ij}^W is weighted shortest path length between nodes i and j.

Local efficiency Local efficiency was introduced by Latora and Marchiori (2001) as a measure that reveals how much the system is fault tolerant, i.e. how efficient the communication is between the first neighbors of i when i is removed. Hence, they define the local efficiency as the average efficiency of the local subgraphs induced by the first neighbors of i. Latora and Marchiori state that this definition is valid both for unweighted and for weighted graphs. Thus, the proposed metrics seems to be:

$$(e_{loc})_i^W = \frac{\sum_{(j,h) \in E_i} (d_{jh}^W (G_i)^{-1}}{k_i(k_i - 1)}, \tag{24}$$

where G_i is a subgraph induced by the first neighbors of i, E_i is the set of edges of this subgraph. However, Rubinov and Sporns (2010) propose another generalization of local efficiency to weighted graphs:

$$(e_{loc2})_i^W = \frac{\sum_{(j,h) \in E_i} (w_{ij} w_{ih} (d_{jh}^W (G_i)^{-1})^{1/3}}{k_i(k_i - 1)}. \tag{25}$$

This second generalization, however, looks contrintuitive. Why should weights w_{ij} and w_{ih} contribute to the estimate of how how efficient the communication is between the first neighbors of i when i is removed? Still, both versions of local efficiency were implemented.

5.2.2 Global

Graph characteristic path length This is the average node-level characteristic path length:

$$L^W = \frac{1}{n} \sum_{i \in V} l_i^W \tag{26}$$

Graph global efficiency This is the average node-level efficiency:

$$E_{global}^W = \frac{1}{n} \sum_{i \in V} e_i^W \tag{27}$$

Graph local efficiency This is the average node-level local efficiency (recall that there are two versions of them):

$$E_{local}^{W} = \frac{1}{n}\sum_{i \in V}(e_{loc})_i^W \tag{28}$$

Graph clustering coefficient This is the average node-level clustering coefficient:

$$C^W = \frac{1}{n}\sum_{i \in V} c_i^W \tag{29}$$

Graph transitivity Weighted graph-level transitivity is defined by:

$$T^W = \frac{\sum_{i \in V} 2t_i^W}{\sum_{i \in V} k_i(k_i - 1)} \tag{30}$$

Graph density Weighted graph density is defined by:

$$D^W = \frac{\sum_{i,j \in V} w_{ij}}{n(n-1)} \tag{31}$$

Graph assortativity by weighted degree This is Pearson correlation coefficient of weighted degrees between pairs of connected nodes (Newman, 2003):

$$r = \frac{|E|^{-1}\sum_{(i,j)\in E} k_i^W k_j^W - \left[|E|^{-1}\sum_{(i,j)\in E}\frac{1}{2}(k_i^W + k_j^W)\right]^2}{|E|^{-1}\sum_{(i,j)\in E}\frac{1}{2}((k_i^W)^2 + (k_j^W)^2) - \left[|E|^{-1}\sum_{(i,j)\in E}\frac{1}{2}(k_i^W + k_j^W)\right]^2}. \tag{32}$$

Graph weighted assortativity as in Rubinov and Sporns (2010) This is another generalization of the assortativity coefficient to weighted networks, described by Rubinov and Sporns (2010). They refer to it as a modification from Leung and Chau (2007):

$$r = \frac{|E|^{-1}\sum_{(i,j)} \hat{w}_{ij} k_i^W k_j^W - \left[|E|^{-1}\sum_{(i,j)}\frac{1}{2}\hat{w}_{ij}(k_i^W + k_j^W)\right]^2}{|E|^{-1}\sum_{(i,j)}\frac{1}{2}\hat{w}_{ij}((k_i^W)^2 + (k_j^W)^2) - \left[|E|^{-1}\sum_{(i,j)}\frac{1}{2}\hat{w}_{ij}(k_i^W + k_j^W)\right]^2}. \tag{33}$$

Note that normalized weights (divided by the maximum weight in the network) are used here.

Maximal sum of weights of the largest clique Let G' be a set of the largest complete subgraphs of the network, $m = |G'|$, V'_k the set of nodes of G'_k. Maximal sum of weights of the largest clique is defined by:

$$CL^W_{max} = \max_{G'_k} \sum_{i,j \in V'_k} w_{ij}. \tag{34}$$

Mean sum of weights of the largest clique Mean sum of weights of the largest clique is defined by:

$$CL^W_{mean} = \frac{\sum_{G'_k} \sum_{i,j \in V'_k} w_{ij}}{m}. \tag{35}$$

Graph diameter This is the value of the greatest eccentricity:

$$diam^W = \max_{i \in V} ecc^W_i. \tag{36}$$

Graph radius This is the value of the smallest eccentricity:

$$rad^W = \min_{i \in V} ecc^W_i \tag{37}$$

Number of graph centers This is the number of nodes i such that $rad^W = ecc^W_i$.

Index of graph center (if a single vertex) If the number of graph centers equals 1, the index i is returned (note that indexes start with 0). Else, NaN is returned. Note that this is the only feature that intentionally includes $NaNs$.

Graph algebraic connectivity The algebraic connectivity of a graph G is the second-smallest eigenvalue of the Laplacian matrix of G, where the elements of the Laplacian are given by:

$$Laplacian^W_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j, \\ k^W_i & \text{if } i = j. \end{cases} \tag{38}$$

Freeman centralization: degree, betweenness, closeness, eigenvector The centralization of any network is a measure of how central its most central node is in relation to how central all the other nodes are. Centralization measure then (a) calculates the sum in differences in centrality between the most central node in a network and all other nodes, and (b) divides this quantity by the theoretically largest such sum of differences in any network of the same size:

$$CF = \frac{\sum_{i \in V} \max_{i \in V} centrality_i - centrality_i}{\max_G \sum_{i \in V} \max_{i \in V} centrality_i - centrality_i}. \tag{39}$$

5.3 Directed graph metrics

Since random walk normalization produces directed random walk graph, we've also calculated directed metrics. In addition to common metrics like PageRank, hubs and authorities score, we've calculated some additional custom metrics based on random walk intuition.

5.3.1 Common metrics

Node in-degree

$$k_i^{in} = \sum_{j \in V} p_{ji}. \tag{40}$$

We didn't calculate out-degrees, because they all are equal to 1.

PageRank score Introduced in 1998 by Brin and Page and roughly estimates probability that a person randomly clicking on links in the network will arrive at particular node. It is closely related to random walks on graph and we are fully aware that it is somewhat tautological feature but we used it nonetheless. We used igraph implementation to calculate PageRank score.

Hubs and authorities scores Introduced by Kleinberg in 1999 [26]. It is another algorithmic way to estimate importance of nodes in directed graphs: authority score shows the value of the node in terms of incoming edges, and hub estimates value of node links to other nodes. We calculated it using igraph library implementation.

Stationary distribution vector We also used as features coordinates left eigenvector π of random walk matrix P corresponding to eigenvalue 1:

$$\pi^T P = \pi. \tag{41}$$

5.3.2 Custom metrics based on random walk logarithms

Many graph metrics are based on assumption that weights of graph edges represent some kind of distance between nodes which can be added. For the random walk matrix, where edge weight represents probability of walk between nodes, it may not be true. For example, if we want to calculate probability of going from node i to j through node k, the resulting probability will be $p_{ij} = p_{ik} * p_{kj}$. It means that if we want to use distance based graph metrics, we should use something which is additive. We solved this problem with the use of negative logarithms of edge weights:

$$w_{ij}^p = -\ln p_{ij}. \tag{42}$$

Then on this new matrix W^p we calculated shortest path matrix $D^p = \{d_{ij}^p\}$ using Dijkstra algorithm (hence negative logarithm) for which we calculated: in- and -out efficiencies, in- and out- degrees, in- and out- characteristic path lenghts, in- and out- eccentricities, in- and out- closeness centralities. Exact formulas can be found in [18].

5.4 Baseline features

To produce an outer baseline, we calculate six global network metrics used by the authors of the dataset [7]. Note that these features are computed for binarized networks, hence only non-weighted edges a_{ij} appear below:

Weighted clustering coefficient

$$CC = \frac{1}{n} \sum_{i \in V} \frac{2t_i}{d_i(d_i - 1)}, \tag{43}$$

where t_i is the number of triangles for the node i.

Characteristic path length

$$CPL = \frac{1}{n} \sum_{i \in V} \frac{\sum_{j \in V, j \neq i} g_{ij}}{n - 1}, \tag{44}$$

where g_{ij} is the length of the shortest path (geodesic) between the vertices i and j.

Normalized CC

$$\lambda = \frac{CC}{CC_{rand}}, \tag{45}$$

where CC_{rand} is the average CC from simulated random networks. We randomize network by swapping edges between random pairs of vertices (five swaps on average for each edge), thus preserving each vertex degree, but changing connectivity pattern. One hundred of such random networks is produced for each subject.

Normalized CPL

$$\gamma = \frac{CPL}{CPL_{rand}}, \tag{46}$$

where CPL_{rand} is the average CPL from the same random networks.

Small-worldness

$$\sigma = \frac{\lambda}{\gamma}. \tag{47}$$

Modularity

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{d_i d_j}{2m}] \delta(c_i c_j), \tag{48}$$

where m is the sum of weighted edges in the network, and c is the community. Hence, Q values represent the proportion of within-module edges in the network minus the expected proportion from a similar random network. We follow the

authors of the original paper and produce one reference graph partition based on the group average network with Louvain modularity algorithm and compute modularity values with respect to this partition. Since modularity Q values can vary based on random differences in module assignments from run to run, Q values are averaged over 100 iterations of the algorithm. Note that the entire sample is used to produce the group average network. In terms of machine learning algorithms this means that we incorporate information about both train and test to compute this particular feature. We do this intentionally to maintain correspondence between our features and those used by the authors of the dataset.

There are three important differences in how we computed the six metrics used by the authors of the original dataset. First, Rudie et al. [7] used the Brain Connectivity Toolbox (BCT) for Matlab, and we wrote custom scripts in Python. Second, the authors produced networks with 5% to 8.5% sparsity in 0.5% increments, binarized them and computed average values for each metric. To obtain networks with exact sparsity level, the BCT algorithm sorts all edges and preserves the needed exact number of strongest ones. This means that multiplicity of edges at threshold level is ignored and hence some edges with threshold-level weight are preserved, while others are dropped. We modify this procedure so that all edges with threshold-level weight are preserved; hence, actual sparsity of the networks produced by our algorithm is slightly higher than the nominal level. Even with this modification, sparsity levels below 7% lead to a disconnected network in at least one participant; sparsity level 8.5% is above the minimum sparsity level in the online dataset. Hence, we only used three sparsity levels (7%, 7.5%. and 8%). With these sparsity levels, the procedure remains the same: we produce thresholded networks, binarize them, compute six metrics as described above and average the obtained values across sparsity levels.

6 Machine learning pipeline

We consider the two classes 'typically developing' (TD) and 'autism spectrum disorder' (ASD) based on diagnosis, and classify individual brain networks based on constructed features described above.

6.1 Classifiers

We used the following classifiers: logistic regression (LR); SVM with linear kernel [20]; random forest (RF), boosted decision trees (BDT) and stochastic gradient descent with modified Huber loss. It is important that all of these methods perform feature selection, either in terms of weighting or in terms of selecting best features for splitting at tree nodes, so we are able to report most important nodes for this classification task.

For linear classification we also applied elastic-net regularization:

$$\hat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}}(l(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\omega}) + \alpha \left(\frac{(1-\rho)}{2} \|\boldsymbol{\omega}\|_2^2 + \rho \|\boldsymbol{\omega}\|_1) \right), \tag{49}$$

where $l(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\omega})$ – classification loss (hinge, logistic or modified Huber), α – regularization coefficient and ρ is l_1-ratio. We calculated regularization coefficients using stochastic gradient descent.

For linear methods we scaled features with min-max scaling:

$$\boldsymbol{x}^{scaled} = \frac{\boldsymbol{x} - x_{min}}{x_{max} - x_{min}}. \tag{50}$$

We also excluded zero-variance features from our analysis. Both of these steps were based on train folds and did not incorporate information from test folds.

6.2 Performance metric and hyperparameters grid search

We measured the performance (prediction quality) of our algorithms with ROC AUC – area under the receiver operating characteristic curve. It is a common used metric for binary classification which calculates the area under the True Positive Rate/False Positive Rate curve obtained by changing the classification threshold. It provides more complete picture of classification performance then accuracy and like accuracy can be summarized by one number.We also report precision and recall for the best models on the best features.

We compare models through two-step procedure. First, for each dataset, we perform hyperparameters grid search based on 10-fold cross-validation with fixed random state for reproducibility.

Second, for each dataset we evaluate best parameters for each of the four models on 50 10-fold splits with fixed different random states. We didn't use a holdout set due to small sample size. It is one of the shortcomings of our study, but it was important for us to use all the data available due to high dimensionality of the problem.

We also tested several chosen models with on 100 10-fold splits. Predictions on each test fold are combined to make a prediction on the entire sample due to a relatively small sample size. Then 100 iterations of 10-fold cross-validation give 100 ROC AUC values based on the entire sample instead of 1,000 ROC AUC values based on one tenths of the sample.

We confirmed the results for the fine-tuned model on the best feature set using the leave-one-out cross-validation procedure. We also extract the relative feature importance according to the best classification model.

6.3 Dimensionality reduction

On the best feature sets we did dimensionality reduction on two components to check if there is simple geometry in our data. Namely, we did Principal Component Analysis with linear and with radial basis function, sigmoid, polynomial (degrees 2-5) and cosine kernels, Multidimensional Scaling, Spectral Embedding,

Isomap and t-distributed Stochastic Neighbour Embedding (TSNE). Parameters for each method were chosen manually. Before each dimensionality reduction we did standard scaling:

$$\boldsymbol{x}^{scaled} = \frac{\boldsymbol{x} - \mu_x}{\sigma_x}, \tag{51}$$

where μ_x and σ_x are mean and standard deviation of feature x.

6.4 Programming tools

We used Python and IPython notebooks platform [21]. We did matrix calculations, graph metrics computation and numerical analysis in NumPy [22], SciPy, NetworkX [23], igraph and louvain libraries. We also employed scikit-learn library [24] for all dimensionality reduction methods and classifiers except BDT for which we used xgboost library. We plotted 2D figures with matplotlib[25] and seaborn in Python; 3D plots were produced with igraph and rgl libraries in R.

7 Results

We were able to found features and models which showed ROC AUC on the level of published studies to date (about 0.8). Namely, it was Linear SVM on node degrees of the connectomes weighted by squared Euclidean distances and normalized by the geometric mean of adjacent node pairs. We investigated this phenomena further and were able to achieve 0.84 ROC AUC on leave-one-out cross-validation. We also tried all common dimensionality reduction methods (PCA, kernel PCA, Isomap, TSNE, MDS, Spectral Embedding) on the best features. None of them showed clear geometry in our data, which may indicate possible overfit.

7.1 Best features and classifiers

Table 1 shows 50 best combinations of features and classifiers in terms of mean ROC AUC across 500 test folds (produced by 50 iterations of 10-fold cross-validations with different fixed random states). We see that most winning models and graph metrics combination are based on weights divided by square distance. It is interesting that even baseline features performed better on squared original weights.

Table 1: 50 best combinations of classifiers and features. Mean and std ROC AUC values were calculated on 50 10-fold splits with diffrent fixed random states, thus producing 500 ROC AUC values obtained on one tenth of the sample.

Weighting	Norm	Features	Classifier	Mean AUC	Std AUC
rootwbydist	sum	undirected node local efficiency	XGB	0.79	0.17
wbysqdist	no norm	directed local 'in efficiency'	XGB	0.77	0.15
wbysqdist	spect	directed local pagerank_node	SVC	0.77	0.17
wbysqdist	spect	directed local pagerank_node	SVC	0.77	0.17
sqrtw	spect + max	undirected node Barrat neighborhood degree	XGB	0.76	0.15
sqrtw	spect	undirected node Barrat neighborhood degree	XGB	0.76	0.15
wbysqdist	spect	undirected node degree	SVC	0.76	0.18
wbysqdist	spect	undirected node degree	SVC	0.76	0.18
wbysqdist	no norm	directed node 'in efficency'	LR	0.76	0.17
wbysqdist	sum	undirected node degree	XGB	0.76	0.16
binar	spect	undirected closeness node (by random)	XGB	0.76	0.16
wbysqdist	spect	undirected node degree	SVC	0.76	0.18
wbysqdist	spect	undirected node degree	SVC	0.76	0.18
wbysqdist	sum	undirected node degree	XGB	0.75	0.15
wbysqdist	spect	directed node pagerank	SVC	0.75	0.17
wbysqdist	no norm	directed node in-degree	SVC	0.75	0.17
wbysqdist	no norm	directed node 'in efficiency'	SVC	0.75	0.17
origw	spect + max	undirected all metrics	XGB	0.75	0.16
binar	sum	undirected global with randomized	XGB	0.74	0.18
rootwbydist	sum	undirected global modularities	RF	0.74	0.16
wbysqdist	spect	directed node pagerank	SVC	0.74	0.18
binar	sum	shortest path edgevector	LR	0.74	0.15
wbysqdist	no norm	directed node in-degree	SVC	0.74	0.17
wbysqdist	sum	undirected node degree normalized by random	Huber	0.74	0.15
wbysqdist	max	undirected node degree normalized by random	LR	0.74	0.16
wbysqdist	no norm	undirected node degree normalized by random	LR	0.74	0.16
wbysqdist	sum	undirected node degree	XGB	0.74	0.16
wbysqdist	spect	directed node pagerank	SVC	0.74	0.17
wbysqdist	max	undirected node degree normalized by random	Huber	0.74	0.16
rootwbydist	no norm	directed node 'in closeness'	SVC	0.74	0.17
binar	spect + max	undirected node neighborhood degrees	XGB	0.74	0.17
wbysqdist	sum	undirected node degree normalized by random	Huber	0.74	0.15
wbysqdist	spect	directed node pagerank	SVC	0.74	0.18
wbysqdist	spect	undirected node local efficiency	XGB	0.74	0.16
sqrtw	sum + max	shortest paths edgevector	LR	0.73	0.17
sqrtw	spect + max	undirected node clustering coefficient	XGB	0.73	0.18
rootwbydist	sum	undirected global modularities	XGB	0.73	0.17
wbysqdist	no norm	directed node 'in efficiency'	LR	0.73	0.17
sqrtw	no norm	baseline metrics	RF	0.73	0.18
binar	spect	undirected node local efficiency	XGB	0.73	0.17
rootwbydist	sum	undirected node local efficiency	XGB	0.73	0.18
sqrtw	spect	undirected node clustering coefficient	XGB	0.73	0.18
wbysqdist	no norm	undirected node degree normalized by random	Huber	0.73	0.16
sqrtw	max	paths edgevector	LR	0.73	0.17
invdist	no norm	random walks edgevector (0.8)	Huber	0.73	0.16
wbysqdist	no norm	directed node 'out-efficiency'	SVC	0.73	0.17
origw	no norm	undirected node local efficiency	XGB	0.73	0.18
binar	spect + max	undirected CPL normalized by random	XGB	0.73	0.17
invdist	spect	random walks edgevector (0.8)	Huber	0.73	0.16
invdist	no norm	random walks edgevector (0.5)	SVC	0.73	0.16

Notifications. Weighting: origw – original weights, rootwbydist – root weights by distance, wbysqdist – weights by squared distance, sqrtw – square root of original weights, binar – binary, invdist – binary weghts divided by distances. Classifiers: XGB – Boosted Decision Trees (xgboost implementation), SVC – linear SVM, LR – Logistic Regression, RF – Random Forest, Huber – Stochastic Gradient Descent with modified Huber loss. Norm: sum – by matrix weights sum, spect – by geometric mean of the degrees (9), max – by maximum connectome weight, spect + max – combination spectral and sum normalizations (order matters).

7.2 Weighting and normalization effects

Since most of the best results included weights normalized by distance and spectral topological normalization, we've decided to envestigate this phenomena further. We choose three weighting schemes: original weights, binary, $\frac{w}{d^2}$ weights;

and two normalization schemes: without and with spectral normalization. For each of six combinations of weights and normalization we used weighted degrees as features. We also used baseline metrics calculated on original weights as seventh dataset.

To avoid overfitting we tested models with the best parameters on 100 10-fold splits. This was done for each of the seven datasets and four classifiers (Logistic Regression, Linear SVM, Random Forest, BDT), with the same random states of 10-fold split to ensure comparability of results. Predictions on each test fold are combined to make a prediction on the entire sample due to a relatively small sample size. Then 100 iterations of 10-fold cross-validation give 100 ROC AUC values based on the entire sample instead of 1,000 ROC AUC values based on one tenths of the sample.

Figures 4-6 show boxplots of the ROC AUC, precision and recall values for the models on selected feature sets. The best classification performance is 0.77 mean ROC AUC across 100 iterations. This best model runs on node degrees of the connectomes weighted by squared Euclidean distances and normalized by the geometric mean of adjacent node pairs. It performs significantly better ($p = 7.8 \times 10^{-18}$, Wilcoxon test with Bonferroni post-hoc correction) than the best model on the baseline feature set with 0.66 mean ROC AUC.

Models on the datasets with other combinations of weighting and normalization schemes perform similarly to the baseline or worse. All pairwise differences between the results on our datasets and the baseline are significant (Wilcoxon test on ROC AUC values with Bonferroni post-hoc correction has p-values less than 10^{-8}, except the difference between the baseline and the degrees on normalized original weights with $p = 0.028$).

The best model for weighted normalized node degrees is obtained by SVM with linear kernel. Figure 7 compares the results obtained by using different classifiers on this dataset. Five of seven best models are linear. BDT perform better only on baseline features and the node degrees on the original non-normed weights.

We also adjusted elastic-net regularization parameters for the best SVM and LR models on the best feature set (node degrees on the weighted by distance and normalized connectomes). Results are shown in Figure 8. Elastic-net regularization slightly improves SVM classification performance compared to l_1- and l_2-regularizations. Finally, we test the best model on the best feature set using leave-one-out cross-validation. This yields ROC AUC of 0.83. We average SVM coefficients for each of 94 SVM models and select ones with absolute weight greater than 0.5. Results are presented in Table 1. The importance of all node degrees is visualized in Figure 9.

Neither normalization by distance nor topological normalization alone boosts classification performance. Weighting by distance could provide some kind of 'regularization' penalizing weak long connections which could be prone to tractography errors. If it was the case we would see performance improvement on all feature sets based on distance-weighted connectomes. The same concerns the topological normalization procedure. It is also possible that with this particular combination of normalizations we found a pattern specifically for this particular classification task (ASD vs TD). Short connections may be more important for

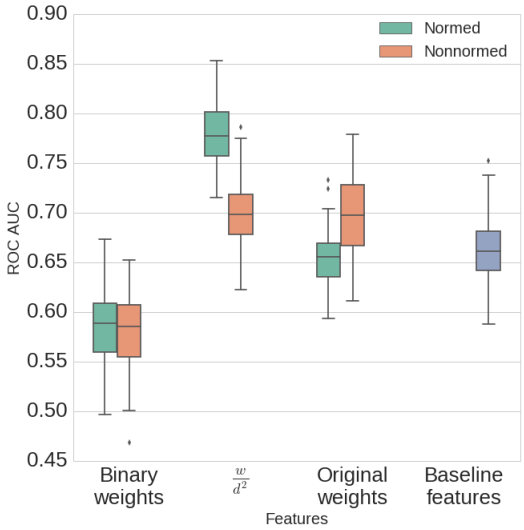


Figure 4: ROC AUC distributions for best models on seven feature sets. Each distribution is based on 100 predictions on the whole sample based on different 10-fold splits with fixed random states (described in Methods).

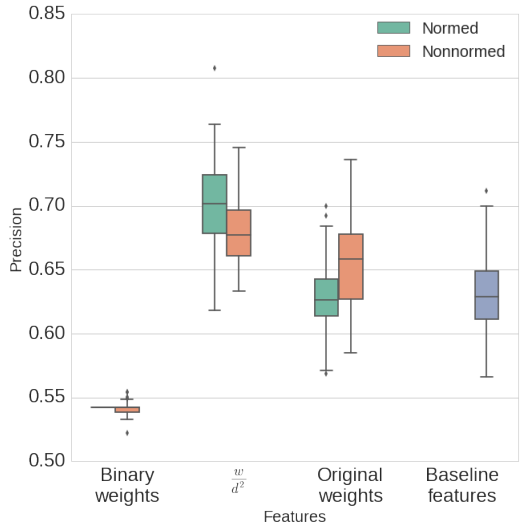


Figure 5: Precision distributions for best models on seven feature sets. Each distribution is based on 100 predictions on the whole sample based on different 10-fold splits with fixed random states (described in 'Classification' pipeline section).

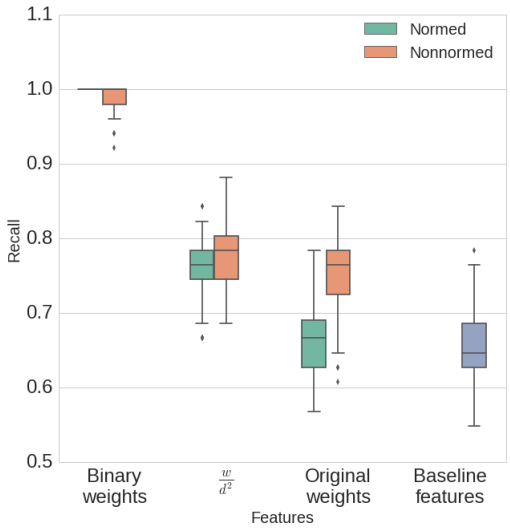


Figure 6: Recall distributions for best models on seven feature sets. Each distribution is based on 100 predictions on the whole sample based on different 10-fold splits with fixed random states (described in 'Classification' pipeline section).

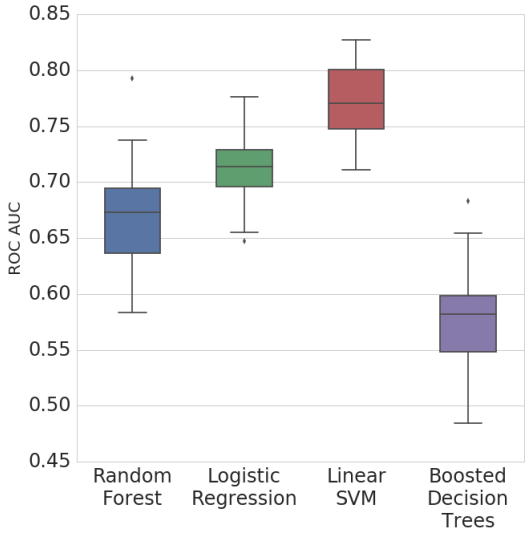


Figure 7: ROC AUC distributions for the best model in each class on the best feature set (node degrees on the weighted by distance normalized connectomes). Each distribution is based on 100 predictions on the whole sample based on different 10-fold splits with fixed random states (described in Methods).

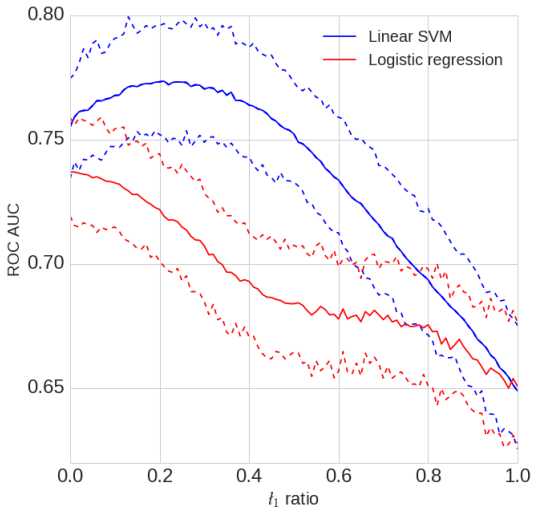


Figure 8: ROC AUC values for linear SVM and LR for 101 l_1-ratios from 0 to 1 in 0.01 increments with α's fixed at 0.01 for SVM and 0.0008 for LR. Each ROC AUC mean value is based on 100 predictions on the whole sample based on different 10-fold splits (described in Methods). Upper and lower dashed lines for each color represent 75-th and 25-th quantile of ROC AUCs distribution (consistently with the boxplots).

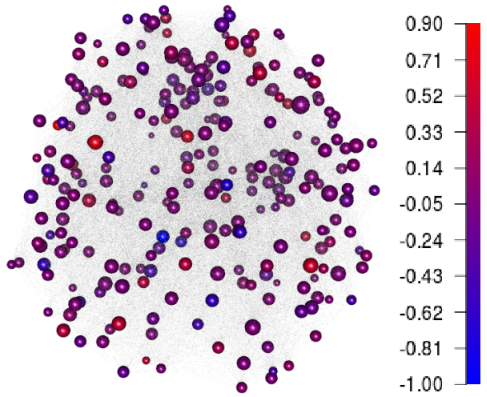


Figure 9: Zone centers in their physical coordinates (axial view). Node size is proportional to group average node degrees obtained on matrices weighted by Euclidean distances and normalized by the geometric mean of degrees of the adjacent nodes (minimal node degree is 0.584, maximal node degree is 1.419). Node color represents mean absolute SVM weight of the respective node (see the colorbar).

Table 2: Zones with the highest absolute mean SVM weights across 94 leave-one-out folds

Rank	Name of region (mean weight ± std)
1	Left Precuneous Cortex (-0.96 ± 0.04)
2	Left Precentral Gyrus (0.91 ± 0.04)
3	Right Precuneous Cortex (-0.71 ± 0.04)
4	Left Superior Parietal Lobule (0.69 ± 0.04)
5	Left Cingulate Gyrus posterior division (0.62 ± 0.04)
6	Right Lateral Occipital Cortex inferior division (0.58 ± 0.04)
7	Right Occipital Pole (0.56 ± 0.04)
8	Right Intracalcarine Cortex (0.56 ± 0.05)
9	Brain-Stem (0.55 ± 0.05)
10	Left Middle Temporal Gyrus posterior division (0.54 ± 0.04)
11	Right Frontal Pole (0.54 ± 0.04)
12	Left Lateral Occipital Cortex inferior division (0.51 ± 0.04)
13	Right Temporal Pole (0.50 ± 0.04)

ASD patients, see for example [27].

7.3 Dimensionality reduction

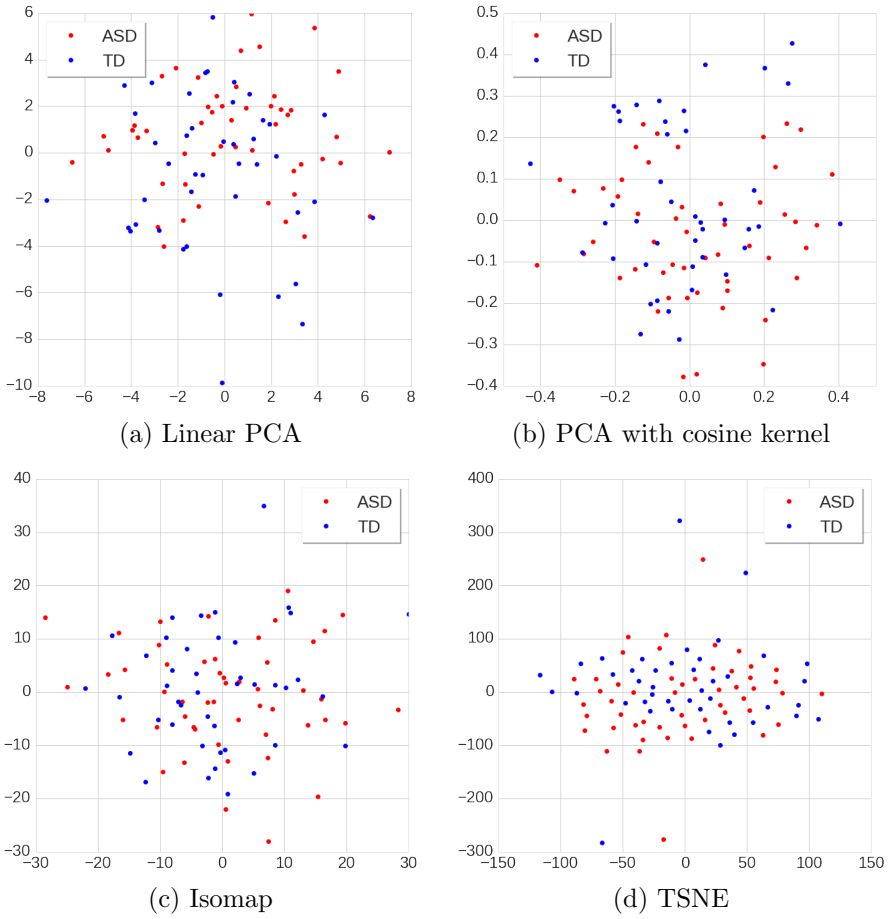
Finally, we wanted to validate our results and check whether there is some intrinsic geometry in our data. We performed dimensionality reduction methods for one of the best feature sets – degrees of the connectomes weighted by squared Euclidean distances and normalized by the geometric mean of adjacent node pairs.

Namely, we did Principal Component Analysis with linear and with radial basis function, sigmoid, polynomial (degrees 2-5) and cosine kernels, Multidimensional Scaling, Spectral Embedding, Isomap and t-distributed Stochastic Neighbor Embedding (TSNE). Parameters for each dimensionality reduction method were chosen manually.

Figure 5 shows results for PCA with linear and cosine kernels, Isomap and TSNE (figures for other dimensionality reduction methods are pretty much the same). We see that there is no visible geometry in our data. For example, explained variance ratios for linear PCA are 0.033 and 0.031 for the first two components respectively.

Classification based on dimensionality reduction features doesn't give ROC AUC better than 0.65. It means that there is no simple geometry in our data which can explain phenotype differences between ASD and TD subjects.

Also, fail of linear and non-linear dimensionality reduction may also indicate overfit to current dataset. Due to small sample size and high dimensionality of the task it is possible that among our features there are some which are



(a) Linear PCA

(b) PCA with cosine kernel

(c) Isomap

(d) TSNE

Figure 10: Dimensionality reduction caption

highly correlated with our target. After all, we generated hundreds of thousands features for the sample size of about one hundred observations.

8 Conclusion and discussion

We generated about 500 hundred different datasets for ASD vs TD connectome classification task, using combinations seven connectome weighting schemes, six topological normalizations, and set of graph metrics (common undirected, common directed and custom directed).

On each of these datasets we performed hyperparameter grid search for chosen classifiers: Logistic Regression, Linear SVM, Stochastic Gradien Descent with modified Huber loss, Random Forest, Adaboost and Boosted Decision

Trees. We reported results for 50 best combinations of classifiers and features.

We further investigated effects of combined normalization scheme incorporating geometric and topological normalization of structural connectomes. We test our approach by classifying autism spectrum disorder versus typical development connectomes with linear and tree-based classification methods. We showed that neither geometric nor topological normalization alone improve classification performance. However, a significant performance increase is achieved using their combination. We further improve leave-one-out cross-validation results and report relative zone importance by adjusting l_1-regularization ratio for linear SVM.

We also tried all common dimensionality reduction methods on the best feature set – Principal Component Analysis with linear and with radial basis function, sigmoid, polynomial (degrees 2-5) and cosine kernels, Multidimensional Scaling, Spectral Embedding, Isomap and t-distributed Stochastic Neighbor Embedding (TSNE). It didn't yield any meaningful results which may indicate possible overfit to dataset.

There are several limitations of this study. First of all, sample size is quite small for the results to be conclusive. Although the groups of ASD and TD subjects are relatively large compared to similar studies published to date, it is highly desirable to replicate the analyses on larger samples. This is primarily due to high dimensionality of the task at hand. For example, analysis based on the bag of edges involved tens of thousands of features with only 94 observations. We employed statistical techniques of feature selection and cross-validation recommended for such situations, but larger sample size would be the best recipe to improve the analysis.

Second, there are certain methodological aspects of the data that should be noted. For example, parcelling brain volume into nodes was quite unusual for structural network analysis. Usualy DTI-based networks are constructed on atlas-based zones or their partitions. Thus, results obtained for this functional-connectivity-based parcellation scheme need to be reproduced on networks with alternative parcellation of brain zones.

Finally, we intentionally left aside any neurological interpretation of our findings. Our study is purely exploratory in nature, and our analysis is blind to substantial meaning of the observed differences. We output the labels of brain zones for which significant differences in local network characteristics between ASD and TD groups are found, but do not go any further in interpreting our results. Further investigation on additional datasets and classification tasks is required to generalize our conclusions.

9 Acknowledgements

I gratefully acknowledge UCLA Multimodal Connectivity Database for making the dataset available to the research and education community. I also thank my colleagues from Institute for Information Transmission Problems for fruitful discussions concerning my research and provided computing resources.

10 Bibliography

References

[1] Craddock, R.C., Jbabdi, S., Yan, C.G., Vogelstein, J.T. Imaging human connectomes at the macroscale. Nature Methods 10, 6, 524–539 (2013)

[2] Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls MR Arbabshirani, M.R., Plis S., Sui J., Calhoun V.D.:, Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. Neuroimage (2016, In press, available online).

[3] First, M. et al. Consensus Report of the APA Work Group on Neuroimaging Markers of Psychiatric Disorders (2012)

[4] Kana, R.K, Uddin, L.Q., Kenet, T., Chugani, D., Müller, R.A. Brain connectivity in autism. Frontiers in human neuroscience 8, 349 (2014)

[5] Hernandez, L.M., Rudie, J.D., Green, S.A., Bookheimer, S., Dapretto, M. Neural signatures of autism spectrum disorders: insights into brain network dynamics. Neuropsychopharmacology 40, 171–189 (2015)

[6] Hastie, T., Tibshirani, R., Friedman, J. The elements of statistical learning. Springer (2001)

[7] Rudie, J.D., Brown, J.A., Beck-Pancer, D., Hernandez, L.M., Dennis, E.L., Thompson, P.M., et al. Altered functional and structural brain network organization in autism. Neuroimage Clin 2, 79–94 (2013)

[8] Brown, J.A., Rudie, J.D., Bandrowski, A., Van Horn, J.D., Bookheimer, S.Y. The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. Frontiers in Neuroinformatics 6, 28 (2012)

[9] Available online at: http://umcd.humanconnectomeproject.org

[10] Mori, S., van Zijl, P.C. Fiber tracking: principles and strategies — a technical re- view. NMR in Biomedicine 15, 468–480 (2002)

[11] Available online at: http://trackvis.org/dtk

[12] Power, J.D., Cohen, A.L., Nelson, S.M., Wig, G.S., Barnes, K.A., Church, J.A., Vogel, A.C., Laumann, T.O., Miezin, F.M., Schlaggar, B.L., Petersen, S.E. Functional net- work organization of the human brain. Neuron 72, 665–678 (2011)

[13] Jones, D.K., Knösche, T.R., Turner R. White matter integrity, fiber count, and other fallacies: the do's and don'ts of diffusion MRI. Neuroimage 73, 239–254 (2013)

[14] Bassett, D.S., Brown, J.A., Deshpande, V., Carlson, J.M., Grafton, S., Conserved and variable architecture of human white matter connectivity. Neuroimage 54, 2, 1262–1279 (2011)

[15] Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V.J., Meuli, R., Thiran, J.-T. Mapping human whole-brain structural networks with diffusion MRI. PLoS One 2, 7, e597 (2007)

[16] Gong, G., Rosa-Neto, P., Carbonell, F., Chen, Z.J., He, Y., Evans, A.C. Age- and gender-related differences in the cortical anatomical network. J. Neurosci. 29, 50, 15684–15693 (2009)

[17] Duarte-Carvajalino, J.M., Jahanshad, N., Lenglet, C., McMahon, K.L., de Zubicaray, G.I., Martin, N.G., Wright, M.J., Thompson, P.M., Sapiro, G.Hierarchical topological network analysis of anatomical human brain connectivity and differences related to sex and kinship. Neuroimage 59, 4, 3784–3804 (2012)

[18] Rubinov, M., Sporns, O., Complex network measures of brain connectivity: uses and interpretations. Neuroimage 52, 3, 1059–1069 (2010)

[19] Freeman, L.C. Centrality in social networks: conceptual clarification. Soc. Netw. 1, 215–239 (1978)

[20] Vapnik, V.N. Statistical Learning Theory. Wiley-Interscience (1998)

[21] Pérez, F., Granger, B. E.: IPython: A System for Interactive Scientific Computing. Computing in Science & Engineering, 9, 21–29 (2007)

[22] van der Walt, S., Colbert, S. C., Varoquaux, G: The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13, 22–30 (2011)

[23] Hagberg, A. A., Schult, D. A., Swart, P. J.: Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science Conference 11–15 (2008)

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830 (2011)

[25] Hunter, J. D. Matplotlib: A 2D graphics environment. Computing In Science Engineering 9, 3. 90–95 (2007)

[26] Kleinberg, J.: Authoritative sources in a hyperlinked environment, Journal of the ACM 46 (5): 604–632 (1999).

[27] Herbert M.R., Ziegler D.A., Deutsch C.K. et al. Dissociations of cerebral cortex, subcortical and cerebral white matter volumes in autistic boys. Brain, 126, 1182–1192 (2003)