Feature generation and dimensionality reduction for connectome classification

Master's thesis

Dmitry Petrov

to.dmitry.petrov@gmail.com

National Research University
Higher School of Economics (Moscow)

June 14, 2016

Contents

Overview

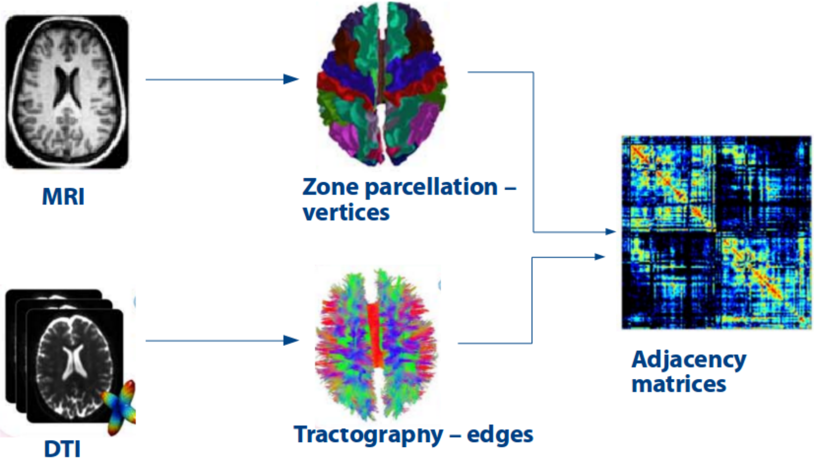
What has been done

- We considered binary classification task for Autism Spectrum Disorders and Typically Developing Groups (94 subjects, 51 ASD and 43 TD) based on structural connectomes
- We generated about 500 different sets of features, using different combinations of connectome weights, normalizations and graph metrics
- We found new simple features and classification model that yield 0.8 ROC AUC score on evaluation which is comparable to published studies

Problem statement

What is connectome

Connectome is graph which represents structural or functional connections between anatomically distinct brain areas.



Problem statement

Connectome classification challenge

We consider binary classification task – norm vs pathology on connectomes. It has several major challenges:

- High dimensionality(~1000 features),
- Small samples (~100 subjects)
- Features are highly correlated
- Most of the machine learning algorithms deal with feature vectors, not matrices

Dataset

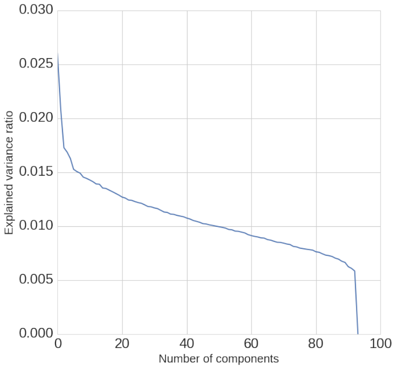
ASD vs TD classification

- UCLA high functional autism dataset, available at http://umcd.humanconnectomeproject.org/
- 94 subjects, 264×264 adjacency matrices
- ASD – 51 subjects (6 females), age $13{,}0 \pm 2{,}8$ years
- TD – 43 subjects (7 females), age $13{,}1 \pm 2{,}4$ years
- For each zone center there are also 3D-coordinates

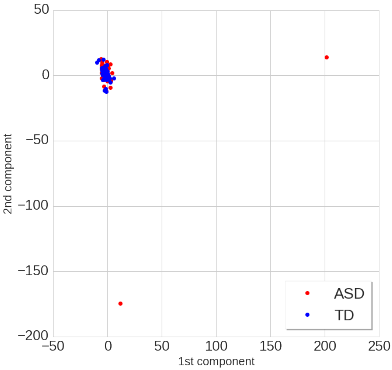
Dimensionality reduction

Standard methods just doesn't show simple structure in our data

Classification performance is very poor – 0.6 ROC AUC at best



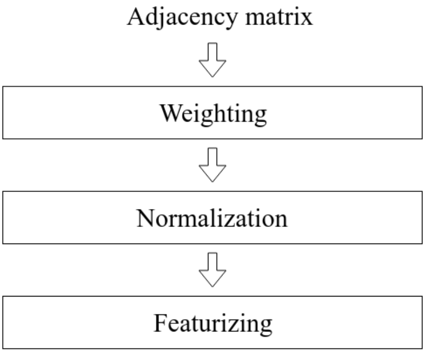
PCA – explained variance ratio



PCA – first two components

Feature generation

Adjacency matrix processing pipeline



Example: $a_{ij} \Rightarrow w_{ij} = \frac{a_{ij}}{d_{ij}^2} \Rightarrow w'_{ij} = \frac{w_{ij}}{\sqrt{deg(i)deg(j)}} \Rightarrow$ node degrees

Feature generation

Weights estimate connections importances

- Original: a_{ij}
- Binary: $b_{ij} = 1$, if $a_{ij} > 0$, 0 – else.
- **Binary normalized by distance:** $\frac{b_{ij}}{l_{ij}^2}$
- **Rooted:** $\sqrt{a_{ij}}$.
- **Original divided by square distance:** $\frac{a_{ij}}{l_{ij}^2}$.
- **Rooted weights divided by distance:** $\frac{\sqrt{a_{ij}}}{l_{ij}}$.

Notations: a_{ij} – original matrix weights, l_{ij} – distances between zone centers. **Bold** – proposed by me.

Feature generation

Normalizations turn different subjects to one scale and also provide connection importances

We used the following normalizations:

- No normalization at all.
- By sum: $\frac{a_{ij}}{\sum_{i,j} a_{ij}}$.
- By max: $\frac{a_{ij}}{\max_{i,j} a_{ij}}$
- **By sum + by max**.
- Spectral: $\frac{a_{ij}}{\sqrt{deg(i)deg(j)}}$.
- **Spectral + by max**.
- Random walk: $p_{ij} = \frac{a_{ij}}{\sum_j a_{ij}}$

Notations: a_{ij} – original matrix weights, $deg(i)$ – degree of node i.
Bold – proposed by me.

Most simple way – unfold matrix into vector. We used the following weights for bags of edges:

- Weights after weighting/normalization
- Shortest path matrix weights
- Random walk matrix weights:

$$W_\alpha = (I - \alpha P)^{-1}.$$

Notations: I – identity matrix, P – random walk matrix obtained by random walk normalization. α were given one of three values 0.2, 0.5 or 0.8.

Each metric provides 264 values (by number of nodes):

- Weighted node degrees
- Average neighbourhood node degrees
- Closeness centrality
- Betweenness centrality
- Eigenvector centralities
- Number of triangles around node
- Clustering coefficient
- Eccentricity
- Characteristic path length
- Efficiency
- Local efficiency

Feature generation

Global undirected metrics

Each metric represent global aspect of the graph:

- Averaged local metrics from previous slide
- Density
- Transitivity
- Degree assortativity
- Graph weighted assortativity
- Largest clique size
- Mean/sum of weights in largest clique
- Radius
- Diameter
- Number of graph centers
- Index of graph center (if there is one)
- Algebraic connectivity
- Freeman centralization: degree, betweenness, closeness, eigenvector

Feature generation
Common directed metrics

For matrices obtained by random walk normalization we also calculated common directed metrics

- Node in-degrees
- PageRank
- Hubs
- Authorities
- Stationary distribution vector

We considered negative logarithms of random walk probabilities:

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

On these matrices we calculated shortest path matrix and then calculated:

- In- and -out efficiencies
- In- and out- degrees
- In- and out- characteristic path lenghts
- In- and out- eccentricities
- In- and out- closeness centralities.

To produce an outer baseline, we calculate six global network metrics used by the authors of the dataset. These features were computed for binarized networks:

- Weighted clustering coefficient
- Characteristic path length
- Normalized clustering coefficient
- Normalized characteristic path length
- Small-worldness
- Modularity

We considered classifiers with feature selection – regularized linear and tree ensembles:

- **Linear classifiers**: Logistic regression, SVM and SGD with modified Huber loss. For all of them elastic net regularization was applied.
- **Tree based classifiers**: Adaboost on trees, gradient boosted decision trees (xgboost library), random forest.

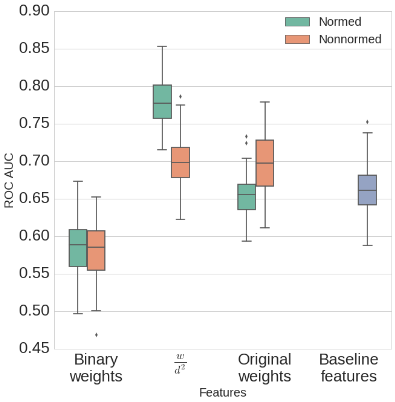
- **Performance metric.** ROC AUC.
- **Hyperparameter optimization.** Grid search based on 10-fold cross-validation (CV) with fixed random state.
- **Best models evaluation.** 50 iterations of 10-fold CVs with fixed random states.
- **Selected models additional evaluation.** 100 iterations of 10-fold CVs with fixed random states. Combination of predictions for different folds were combined to produce prediction for all sample

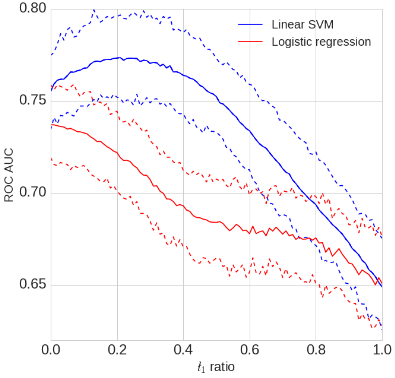
Proposed custom weights provided best results after evaluation on different features!

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
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	no norm	directed node 'in efficency'	LR	0.76	0.17
wbysqdist	sum	undirected node degree	XGB	0.76	0.16
wbysqdist	spect	directed node pagerank	SVC	0.75	0.17

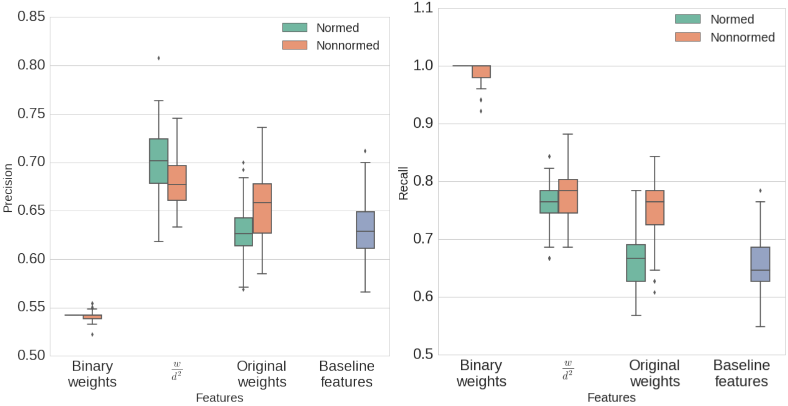
Results

ROC AUC boxplots and elastic net l_1-ratio





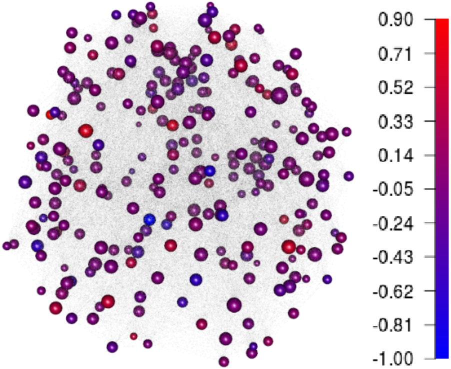
Precision and recall boxplots



Results

Most important zones visualiztion

Most important nodes according to SVM weights. Size reflects node degree.



Conclusion and discussion

- Dimensionality reduction didn't show simple structure in data
- We found that combination of weighting by square distance and spectral normalization gives 0.8 ROC AUC score for linear SVM
- It is comparable to published studies and these features with model are simple and interpretable (which is nice)
- Due to small sample size and large feature set it is possible that we overfitted. Additional confirmation is needed

Thank you for your attention!

Feature generation and dimensionality reduction for connectome classification

Dmitry Petrov

to.dmitry.petrov@gmail.com

National Research University
Higher School of Economics (Moscow)

June 14, 2016

Weighted degree

$$k_i^W = \sum_{j \in V} w_{ij}. \tag{1}$$

Average weighted neighborhood degree

$$k_{nn,i}^W = \frac{\sum_{j \in V} w_{ij} k_j^W}{k_i^W}. \tag{2}$$

Closeness centrality Inverse of average weighted distance to other nodes:

$$(l_i^W)^{-1} = \frac{n-1}{\sum_{j \in V, j \neq i} d_{ij}^W}, \tag{3}$$

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 [?]). We use the weighted version with shortest paths being computed for the weighted graph:

$$b_i = \frac{2}{(n-1)(n-2)} \sum_{\substack{h,j \in V \\ h \neq j, h \neq i, j \neq i}} \frac{\rho_{hj}(i)}{\rho_{hj}}, \quad (4)$$

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):

$$ec_i = v_i, \tag{5}$$

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. 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

$$t_i^W = \frac{1}{2} \sum_{j,h \in V} (\hat{w}_{ij} \hat{w}_{ih} \hat{w}_{jh})^{\frac{1}{3}}. \tag{6}$$

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):

$$c_i^W = \frac{2t_i}{k_i(k_i - 1)}, \tag{7}$$

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:

$$ecc_i^W = \max_{j \in V, j \neq i} d_{ij}^W. \quad (8)$$

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

$$l_i^W = \frac{\sum_{j \in V, j \neq i} d_{ij}^W}{n-1}, \quad (9)$$

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:

$$e_i^W = \frac{\sum_{j \in V, j \neq i} (d_{ij}^W)^{-1}}{n-1}, \quad (10)$$

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:

$$(e_{loc})_i^W = \frac{\sum_{(j,h)\in E_i} (d_{jh}^W(G_i))^{-1}}{k_i(k_i - 1)}, \tag{11}$$

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:

$$(e_{loc2})_i^W = \frac{\sum_{(j,h)\in E_i}(w_{ij}w_{ih}(d_{jh}^W(G_i)^{-1})^{1/3}}{k_i(k_i-1)}. \quad (12)$$

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

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

$$L^W = \frac{1}{n} \sum_{i \in V} l_i^W \tag{13}$$

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

$$E_{global}^W = \frac{1}{n} \sum_{i \in V} e_i^W \tag{14}$$

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

$$E^W_{local} = \frac{1}{n} \sum_{i \in V} (e_{loc})^W_i \quad (15)$$

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

$$C^W = \frac{1}{n} \sum_{i \in V} c^W_i \quad (16)$$

Graph transitivity

Weighted graph-level transitivity is defined by:

$$T^W = \frac{\sum_{i \in V} 2t_i^W}{\sum_{i \in V} k_i(k_i - 1)} \tag{17}$$

Graph density

Weighted graph density is defined by:

$$D^W = \frac{\sum_{i,j \in V} w_{ij}}{n(n-1)} \tag{18}$$

Graph assortativity by weighted degree

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

$$r = \frac{|E|^{-1} \sum_{(i,j)\in E} k_i^W k_j^W - \left[|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}(k_i^W + k_j^W) \right]^2}{|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}((k_i^W)^2 + (k_j^W)^2) - \left[|E|^{-1} \sum_{(i,j)\in E} \frac{1}{2}(k_i^W + k_j^W) \right]^2}. \quad (19)$$

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):

$$r = \frac{|E|^{-1} \sum_{(i,j)} \hat{w}_{ij} k_i^W k_j^W - \left[|E|^{-1} \sum_{(i,j)} \frac{1}{2} \hat{w}_{ij} (k_i^W + k_j^W) \right]^2}{|E|^{-1} \sum_{(i,j)} \frac{1}{2} \hat{w}_{ij} ((k_i^W)^2 + (k_j^W)^2) - \left[|E|^{-1} \sum_{(i,j)} \frac{1}{2} \hat{w}_{ij} (k_i^W + k_j^W) \right]^2}. \tag{20}$$

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:

$$CL^W_{max} = \max_{G'_k} \sum_{i,j \in V'_k} w_{ij}. \quad (21)$$

Mean sum of weights of the largest clique

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

$$CL^W_{mean} = \frac{\sum_{G'_k} \sum_{i,j \in V'_k} w_{ij}}{m}. \quad (22)$$

Graph diameter This is the value of the greatest eccentricity:

$$diam^W = \max_{i \in V} ecc^W_i. \quad (23)$$

Graph radius

This is the value of the smallest eccentricity:

$$rad^W = \min_{i \in V} ecc_i^W \tag{24}$$

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

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:

$$Laplacian_{ij}^{W} = \begin{cases} -w_{ij} & \text{if } i \neq j, \\ k_i^{W} & \text{if } i = j. \end{cases} \tag{25}$$

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:

$$CF = \frac{\sum_{i \in V} \max_{i \in V} centrality_i - centrality_i}{\max_G \sum_{i \in V} \max_{i \in V} centrality_i - centrality_i}. \qquad (26)$$

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

Weighted clustering coefficient

$$CC = \frac{1}{n} \sum_{i \in V} \frac{2t_i}{d_i(d_i - 1)}, \tag{27}$$

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

Characteristic path length

$$CPL = \frac{1}{n} \sum_{i \in V} \frac{\sum_{j \in V, j \neq i} g_{ij}}{n - 1}, \tag{28}$$

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

Normalized CC

$$\lambda = \frac{CC}{CC_{rand}}, \tag{29}$$

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

$$\gamma = \frac{CPL}{CPL_{rand}}, \tag{30}$$

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

Small-worldness

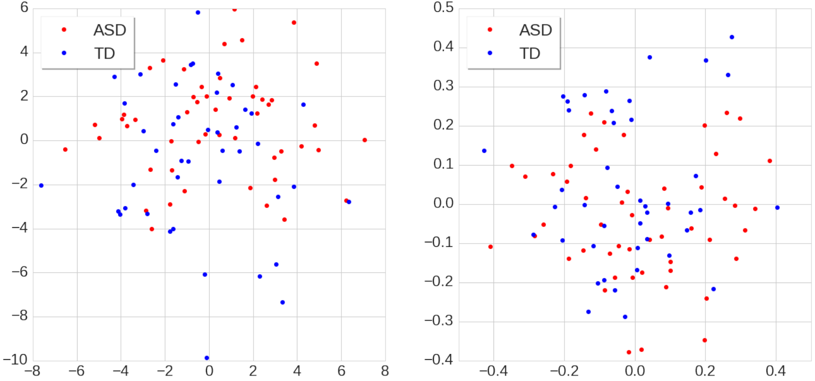
$$\sigma = \frac{\lambda}{\gamma}. \tag{31}$$

Modularity

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{d_i d_j}{2m}]\delta(c_i c_j), \tag{32}$$

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.

There is no clear structure in our data



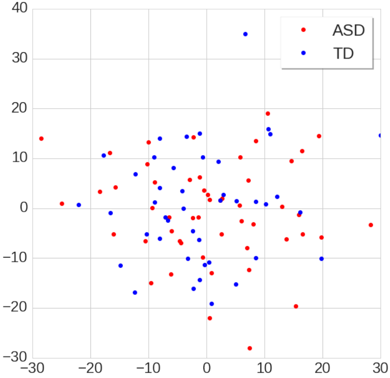
PCA

Kernel PCA (cosine kernel)

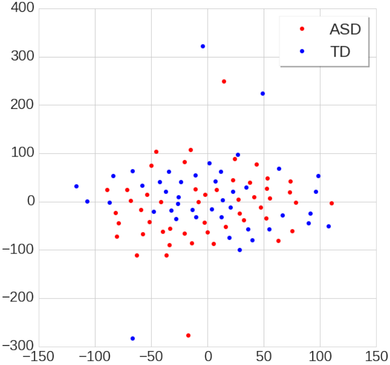
Results

Isomap and TSNE dimensionality reduction on weighted degrees

Again, there is no clear structure in our data



Isomap



TSNE

Appendix
Autism ML results overview

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)

Appendix
Neuroimage ML accuracy