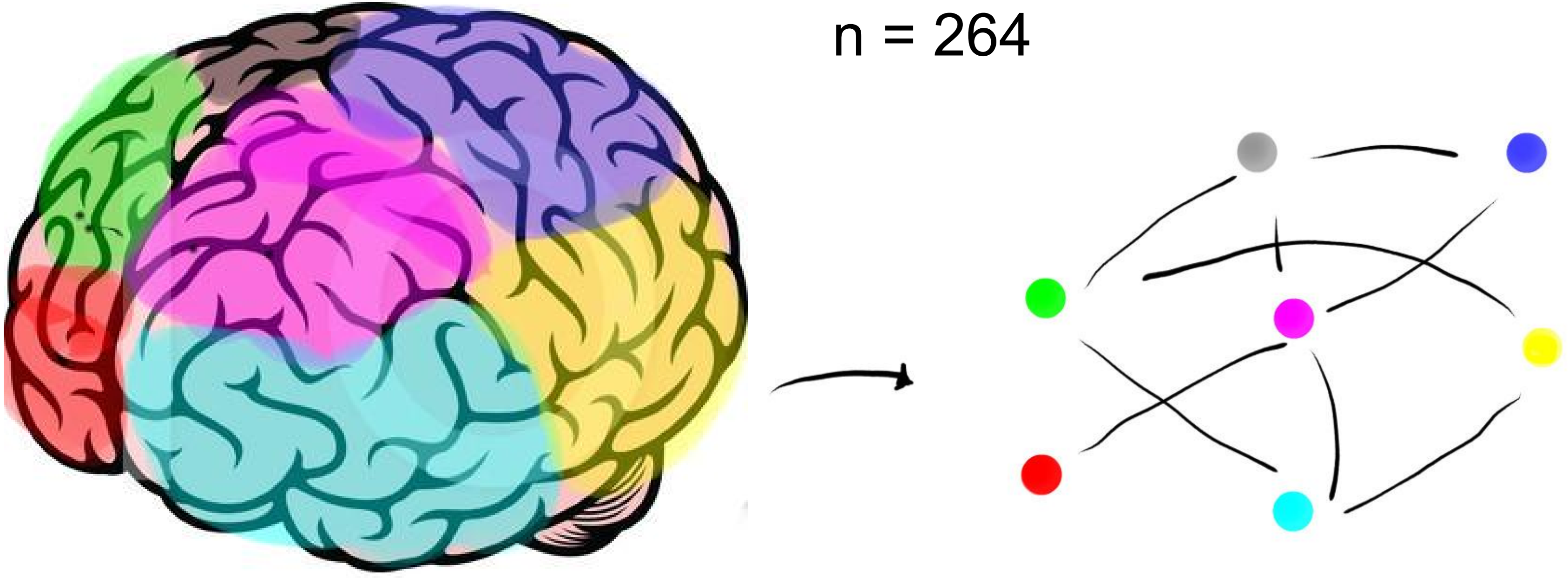
Classification of Brain Networks using Dirichlet Distribution of Graph Spectra

Anna Tkachev, Yulia Dodonova

September 2016

What is a Connectome (Brain Network)?

n = 264



Data Description

Publicly available dataset[1], includes 94 precomputed connectomes

51 high-functioning **autism spectrum disorder** (ASD) subjects
43 **typically developing** (TD) subjects

(age and sex balanced in two classes)

1. 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)

Machine Learning on Graphs?

How to use matrices for classification tasks using machine learning techniques?

- Graph spectra?

In previous works, we studied **adjacency**, **Laplacian** and **normalized Laplacian** matrices spectra. **In this work I focused on the normalized Laplacian.**

Why Normalized Laplacian?

- Important and well know matrix in graph theory

- Arises in random walk/diffusion equations.
  It is analogous to the Laplace operator, but for graphs.

Normalized Laplacian Matrix

A - adjacency matrix (non-binary), entries a_{ij}.

D - diagonal matrix of node degrees, entries: $d_i = \sum_j a_{ij}$.

$\mathcal{L}$ - Nomalized Laplacian, $\mathcal{L} = D^{-1/2}(D - A)D^{-1/2}$.

All matrices nxn

Eigenvalues of Normalized Laplacian

Eigenvalue: $\mathcal{L}v = \lambda v$

(Here i *is the sample number)*

Sorted in ascending order : $\lambda_{i,0} \leq ... \leq \lambda_{i,(n-1)}$

Properties of Normalized Laplacian spectrum:

- $0 \leq \lambda_{i,j} \leq 2$
- $\lambda_{i,0} = 0$ ⇒ use n-1 eigenvalues (263 for this dataset)

Normalized Laplacian Spectra as Feature-Vectors

The feature vector for the i -th sample: $(\lambda_{i,1}, \ldots \lambda_{i,(n-1)})$

Task: classify ASD versus TD

Different linear and tree-based models tested in our previous works, as well as models based on distances between empirical densities of spectra.

Dirichlet Distribution Model?

Most important (for this paper) **property** of graph Laplacian spectrum:

$$\sum_{j=0}^{n-1} \lambda_{ij} = n$$

⇒ *Assume* $\left(\frac{\lambda_{i,1}}{n}, \ldots \frac{\lambda_{i,(n-1)}}{n}\right)$ *has Dirichlet distribution*

Dirichlet Distribution

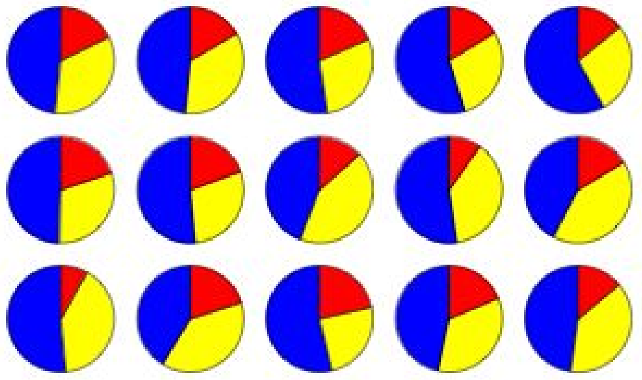
Let $(p_1,..p_r)$ denote a random vector, $p_i > 0$, $\sum_j p_j$=1. Then the probability density of the Dirichlet distribution with vector of parameters $(\alpha_1,..\alpha_r)$ is given by:

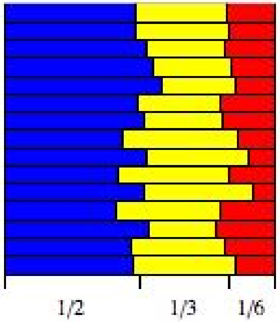
$$\frac{\Gamma(\sum_{j=1}^r \alpha_j)}{\prod_{j=1}^r \Gamma(\alpha_j)} \prod_{j=1}^r p_j^{\alpha_j - 1},$$

where Γ is the gamma function.

Important parameter: $\alpha_0 = \sum_j \alpha_j$

Dirichlet Distribution: Intuition





In this example: $$(\frac{\alpha_1}{\alpha_0}, \ldots \frac{\alpha_r}{\alpha_0}) = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$$

α_0 is related to the variance of the lengths of the pieces

*link: wikipedia

Dirichlet-Based Classifier

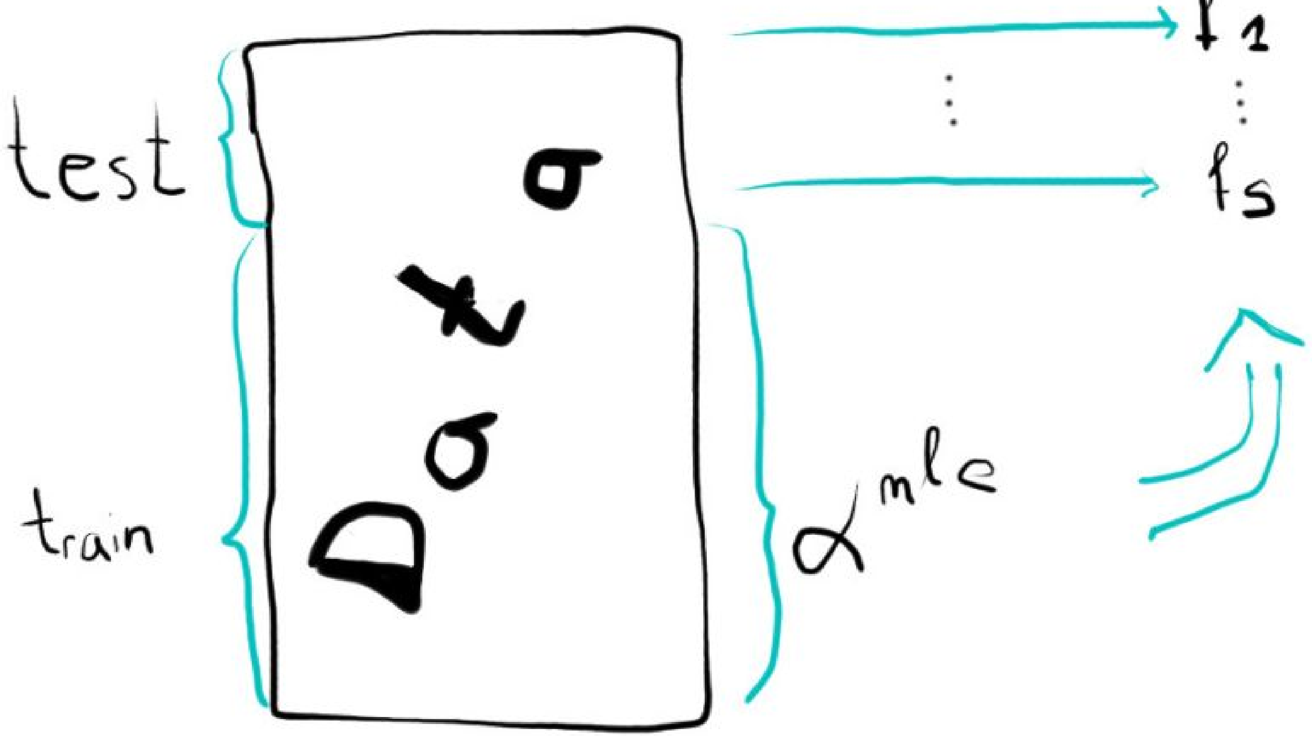
Tools: Python, scikit-learn and Python library: https://github.com/ericsuh/dirichlet

Approach: Construct single feature, then linear classifier with this feature

- split data into **5 folds**, assume data to have a **Dirichlet distribution**,
- **estimate parameter vector** $(\alpha_1, \ldots \alpha_r)$ from the data (using only the train data), in this problem $r = n - 1 = 263$
- calculate the **loglikelihood** that each (test) sample is generated by the Dirichlet distribution with parameter $(\alpha_1, \ldots \alpha_r)$
- the value of loglikelihood is the **single feature** for this sample

Two functions used from the dirichlet library: *mle* and *loglikelihood* functions

Dirichlet-Based Classifier



Evaluation and Results

We use 10 runs of 5-fold cross validation

Metric: ROC AUC (out-of-fold)

Result: **0.666** AUC mean, 0.004 std

Logistic regression with l2 regularization (baseline)

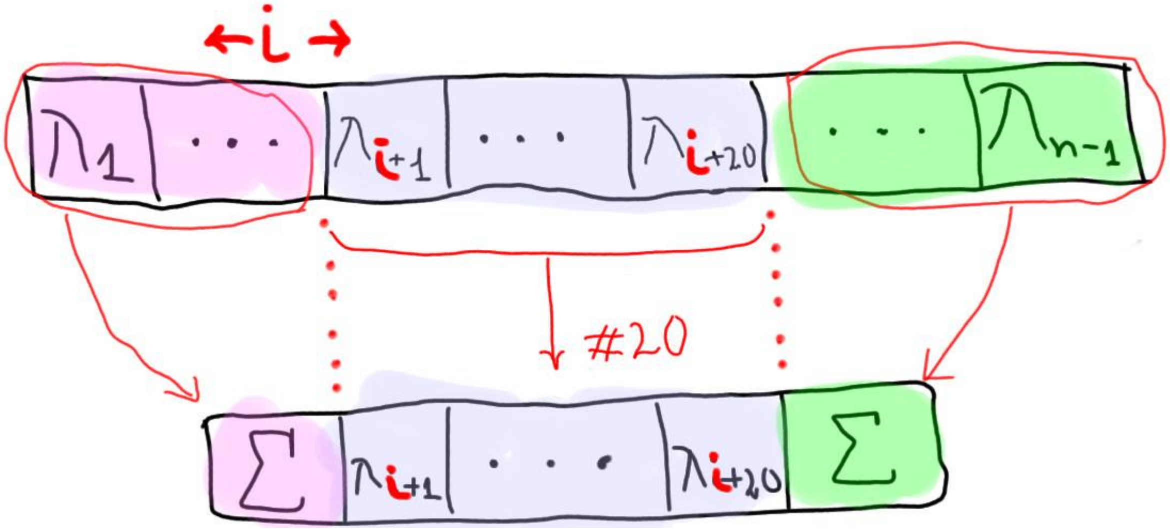
Result: **0.643** AUC mean, 0.046 std

Not so good, modification?

“Sliding Window”

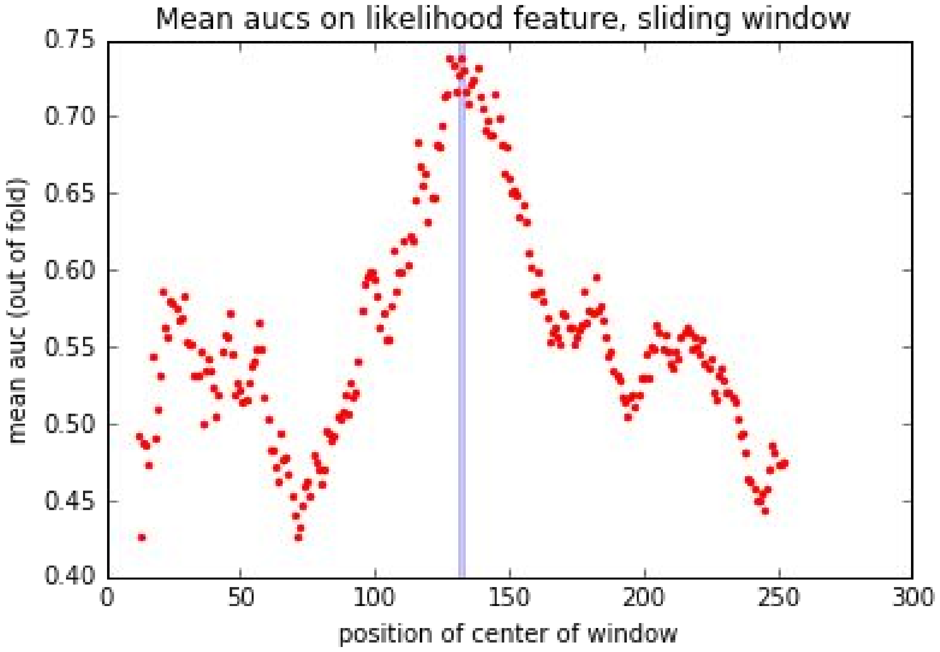
Problem(?): high dimensionality (r = n-1 = 263)

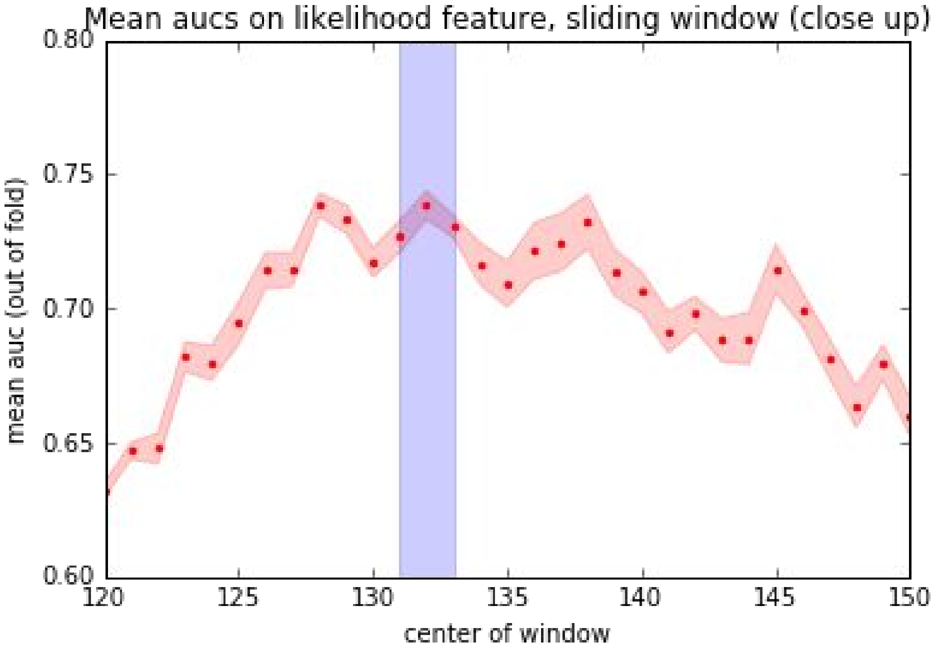
Solution: new features, but conserving “sum up to n”



Sliding Window: Results (with same approach on new features)

0.739 AUC mean, 0.003 std





Results: Comparisons

No baseline for comparisons.
In our previous works, combinations of methods (linear, tree-based) on different features (spectra of different matrices, bag-of-edges, node degrees)

Best result obtained: **0.77** AUC mean (linear SVM on node degrees)

Result of **this work**: **0.739** AUC mean, 0.003 std
Why it may be relevant? Only **single feature** in the model

Results: Comparisons

Does the “sliding window” give good results for other methods?

Other methods:

- **Euclidean distance**
- **Cosine distance**
- **Chi-squared distance**

Similar: calculate distance to mean vector of eigenvalues.

Results: Comparisons

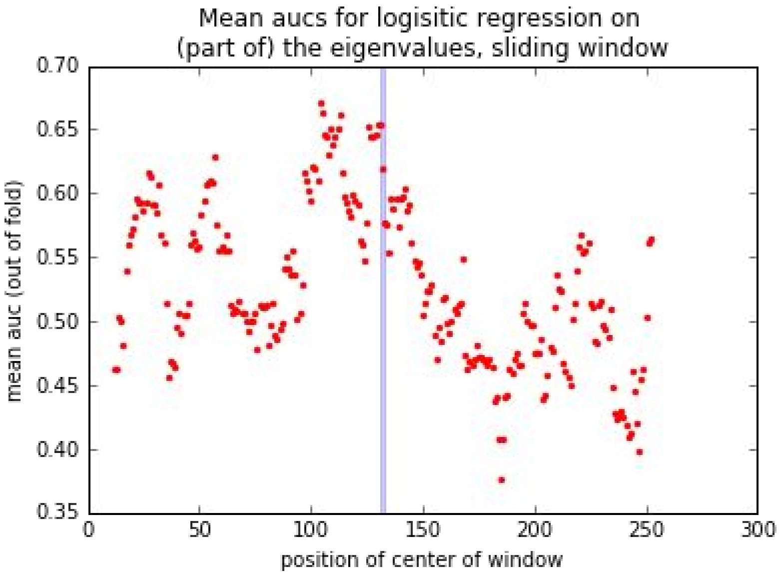
Does the "sliding window" give good results for other methods?

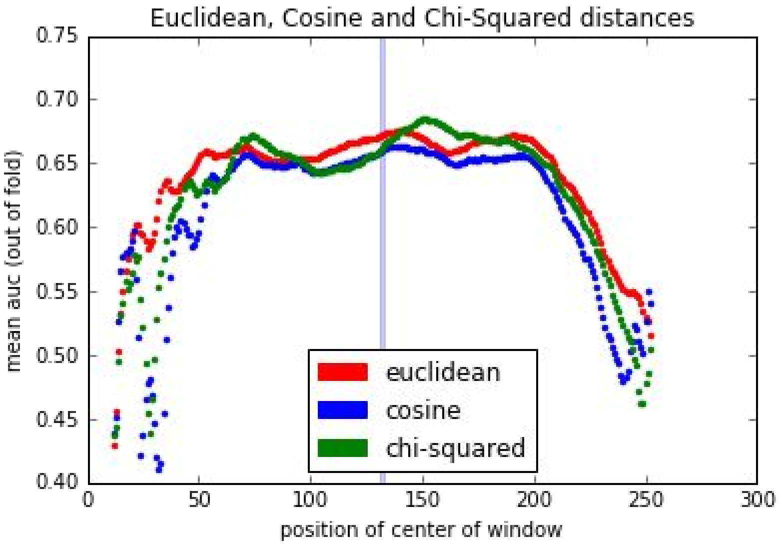
All with sliding window

Classifiers	AUC mean	AUC std
Eigenvalues, logistic regression	0.643	0.046
Dirichlet likelihood, all eigenvalues	0.666	0.004
Dirichlet likelihood, sliding window	0.739	0.003
Eigenvalues, logistic regression, sliding window	0.671	0.024
Euclidean distance, sliding window	0.676	0.012
Cosine distance, sliding window	0.664	0.016
Chi-squared distance, sliding window	0.685	0.010

Results: Comparisons

“Sliding window” added to some methods: plot





Additional Discussion: Low Variance?

0.739 AUC mean, 0.003 std
Low variance of presented classifier: **Why?**

The answer: variability in prediction comes from **partition of data into folds** for cross-validation.

- Single feature in model
- For each sample, this feature **depends only on the *mle*** calculated on the train data of the cross-validation

Turns out the ***mle* is almost always the same** for any subset of samples.

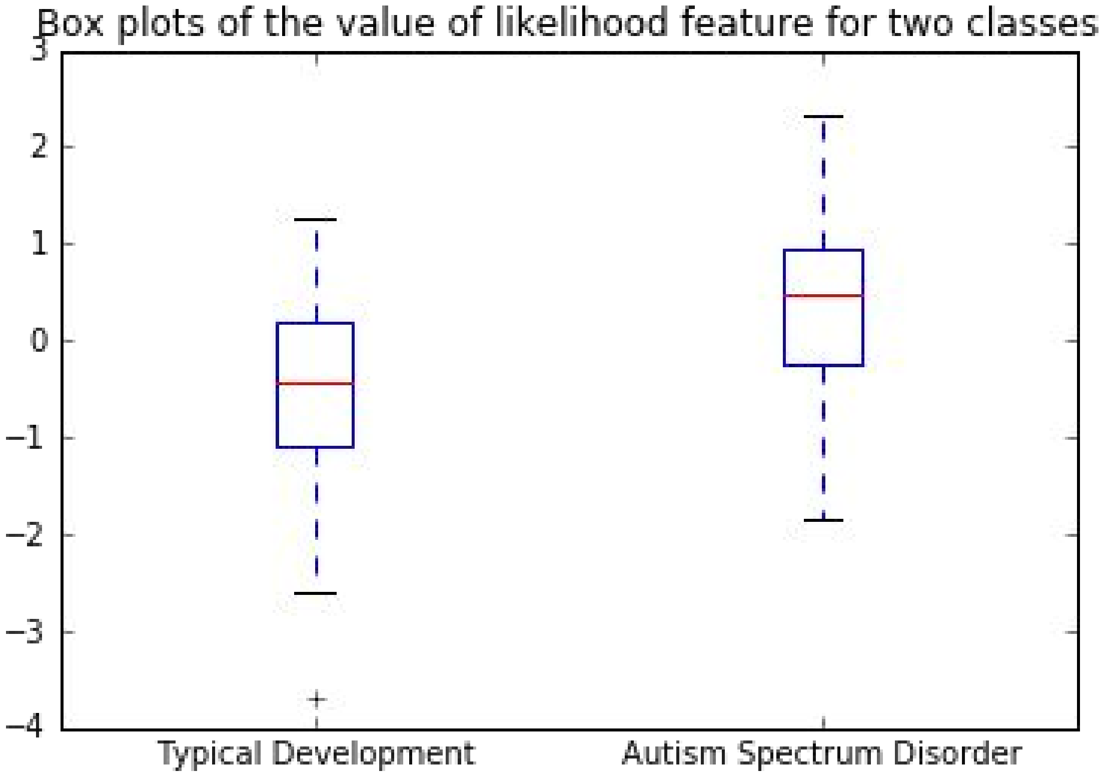
Additional Discussion: Low Variance?

Therefore:

For the two groups(ASD and TD), the “mean” spectra are the same. The classifier seems to catch the difference in the variability of the spectra between the two groups.

Autists are less diverse ?

The likelihood feature



Conclusions

- We used brain networks (connectomes) for disease classification task
- We used normalized Laplacian spectra for network representation
- Applied Dirichlet distribution to model these spectra
- Simple classifier, (relatively) good results

Questions?

annatkachev42@gmail.com

Thank you!