Классификация коннектомов на основе локальных метрик на стохастических матрицах

Александр Иванов
Национальный исследовательский университет
«Высшая Школа Экономики»
alexander.radievich@gmail.com

Дмитрий Петров
Национальный исследовательский университет
«Высшая Школа Экономики»
to.dmitry.petrov@gmail.com

Аннотация

Графовые метрики – популярный подход для классификации структурных коннектомов, графов описывающих структурные связи между различными участками мозга. В нашей работе мы предлагаем считать эти метрики на стохастических матрицах случайных блужданий этих графов. При этом часть этих метрик мы предлагаем считать на логарифмах элементов матриц, чтобы сохранить физический смысл вероятностей перехода между вершинами (поскольку вероятности переходов перемножаются, а не складываются). Используя этот прием, мы сгенерировали признаки на основе метрик, использующих расстояние, для задачи классификации нормы и людей с расстройствами аутистического спектра методами машинного обучения. В итоге на новых признаках были получены результаты (площадь под ROC-кривой – 0.75) на уровне ранее опубликованных работ по этой теме.

1. Введение

Коннектом – дискретная математическая модель, описывающая структурные и функциональные связи между различными зонами мозга [1]. Быстрый рост популярности анализа коннектомов обусловлен предположением, что структурные свойства данной модели мозга могут дать понимание природы изменений (в том числе связанных с различными заболеваниями) в структуре мозга и механизмах его функционирования. При этом диагностика психиатрических заболеваний и нейродегенеративных расстройств на основе данных МРТ и дифузионно-взвешенного МРТ еще не достаточно точна ([2], [3]). Кроме того, большинство исследований используют выборки относительно малых размеров (порядка 10-100 человек), что может приводить к тому, что результаты будут недействительны для выборок больших размеров.

В данной работе мы рассматриваем новые признаки для классификации коннектомов на основе стохастических матриц. Для их генерации мы используем норму случайного блуждания (см. раздел 4.2.2), в результате которой получается стохастическая матрица. На этих матрицах мы вычисляем два типа признаков: те, которые используют полученные вероятности без изменений (входящие степени вершин, хабы и авторитеты, pagerank), а также те, которые используют кратчайшие расстояния между вершинами. Чтобы во втором случае получались интерпретируемые признаки, мы считаем метрики, основанные на расстояниях на отрицательных логарифмах вероятностей перехода стохастической матрицы (см. раздел 3), ведь вероятности переходов осмысленно нужно перемножать, а не складывать.

Для верификации этой методики мы использовали выборку из 94 коннектомов (норма и люди с расстройствами аутистического спектра) UCLA. Используя предлагаемую методику и различные виды взвешивания исходных данных, мы сгенерировали более 700 различных наборов признаков для этих данных. Для каждого из них мы определили оптимальные параметры и выявили комбинации, дающие лучшие результаты. Чтобы избежать переобучения под выборку, мы проверяли результаты, используя 50 различных разбиений на обучающую и тестовую подвыборки. Главной целью тестирования являлось выявление моделей, которые переобучились и не способны предсказывать с хорошими результатами на новых данных.

Признаками, показавшими наилучшую метрику качества, оказались pagerank, out-efficiency, in-efficiency и in-degrees. На этих признаках получается классификация, у которой средняя площадь под ROC-кривой на 50 тестовых подвыборках составляет 0.75, что на уровне опубликованных работ по этой теме. Таким образом, мы получили интерпретируемые признаки, которые дают результаты, сравнимые с существующими. Это является важным моментом, потому что многие работы сосредоточены на достижении наивысшего результата, никак его не интерпретируя.

2. Постановка задачи

Пусть X — множество коннектомов, Y — множество меток классов. Множество коннектомов представляют собой множество симметричных взвешенных матриц смежности. За счет комбинирования взвешиваний (см. раздел 4.2.1), нормировок (см. раздел 4.2.2) и вариантов построения признаков (см. раздел 5.1) мы можем построить отображение $\alpha : X \to X_M$, где X_M — множество вариантов векторов признаков коннектомов X, причем $X_m \in X_M$ — один из этих варинтов признаков. Требуется построить алгоритмы $f_m : X_m \to Y$, каждый из которых будет способен классифицировать произвольный объект $x \in X_m$. Также требуется выбрать варианты, при которых достигается наивысшая метрика качества, и сравнить с результатами в работе [6].

3. Классификация графов на основе стохастических матриц

В нашей работе мы предлагаем классифицировать коннектомы над основе стохастических матриц случайных блужданий. У этого подхода есть интерпретируемая физическая интуиция – вероятность перехода между вершинами графа представляет собой вероятность перехода сигнала между зонами. Мы не знаем, насколько эта интерпретация осмысленна с нейрофизиологической точки зрения, поэтому в нашей работе мы верифицируем этот подход с помощью качества бинарной классификации на признаках на основе этих матриц.

Использование графовых метрик для классификации коннектомов – хорошо изученная область [7]. Эти метрики делятся на два типа. Во-первых, те, которые считаются на весах графов, например, входящие степени верши, хабы и авторитеты, pagerank. Эти метрики хорошо изучены в различных областях и в нашей работе мы используем канонические алгоритмы для их вычисления.

Второй тип метрик (например closeness, betweenness и local efficiency центральности) основан на предположении, что веса графа представляют расстояния между вершинами, которые можно складывать. Но для матриц случайных блужданий, которые мы предлагаем использовать, это не так, ведь веса их ребер представляют собой вероятности прохода между вершинами. Например, если мы хотим вычислить вероятность перехода из вершины i в вершину j через вершину k, то вероятность этого перехода будет $p_{ij} = p_{ik}p_{kj}$. Это означает, если мы хотим использовать метрики, основанные на расстоянии, мы должны использовать аддитивные веса. Мы решили эту проблему использованием отрицательных логарифмов весов ребер:

$$w^p_{ij} = -\ln p_{ij}. \tag{1}$$

Интуиция здесь простая: в этом случае расстояния могут быть корректно складываться и кратчайшие пути будут иметь корректную интерпретацию, как пути с наибольшей вероятностью прохождения сигнала от вершины i в вершину j. Использование именно отрицательных логарифмов обуславливается необходимостью неотрицательных весов ребер для алгоритма Дейкстры. Таким образом, мы использовали отрицательные логарифмы для метрик, основанных на длинах кратчайших путей (Раздел 5.1.4). В этом заключается новизна нашего подхода, поскольку у этих метрик в случае стохастических матриц появляется некоторая физическая интуиция.

4. Данные

4.1. Набор данных

Были использованы публично доступные данные по расстройствам аутистического спектра (UCLA [4], [5]). Данные включают матрицы 51 коннектомов с расстройством аутистического спектра (6 представителей женского пола) и 43 коннектомов нормы (7 представителей женского пола). Средний возраст (стандартное отклонение) был 13 (2.8) для группы расстройства аутистического спектра и 13.1 (2.4) для группы нормы. Все детали об участниках и составлению матриц коннектомов могут быть найдены в оригинальной работе [7].

4.2. Предварительная обработка

Путем комбинирования 6 методов взвешивания и 2 видов нормировок мы получаем 12 вариантов множества коннектомов.

4.2.1. Взвешивания. Здесь l_{ij} — расстояние между зонами i и j.

Мы использовали 6 различных схем взвешивания. Во-первых, мы использовали оригинальные веса коннектомов.

$$a^{origw}_{ij}. \tag{2}$$

Во-вторых, мы использовали бинарные веса:

$$a_{ij}^{binar} = \begin{cases} 1, & \text{если } a_{ij} > 0 \\ 0, & \text{иначе.} \end{cases} \quad (3)$$

В-третих, взвешивание на квадрат расстояния между вершинами:

$$a_{ij}^{wbysqdist} = \frac{a_{ij}}{l_{ij}^2}. \quad (4)$$

В-четвертых, корень от предыдущего взвешенного результата:

$$a_{ij}^{rootwbydist} = \frac{\sqrt{a_{ij}}}{l_{ij}}. \quad (5)$$

В-пятых, корень оригинальных весов

$$a_{ij}^{sqrtw} = \sqrt{a_{ij}}. \quad (6)$$

В-шестых, обратные веса:

$$a_{ij}^{invdist} = \frac{1}{l_{ij}}. \quad (7)$$

4.2.2. Нормировки. Спектральная нормировка:

$$w_{ij}^{spectral} = \frac{a_{ij}}{\sqrt{d_i d_j}}, \quad (8)$$

где a_{ij} — веса ребер, соединяющих вершины i и j; d_i — взвешенная степень вершины i.

Нормировка случайного блуждания:

$$w_{ij}^{rwalk} = \frac{a_{ij}}{\sum_j a_{ij}}. \quad (9)$$

В вычислениях использовалась просто нормировка случайного блуждания и спектральная нормировка вместе с нормировкой случайного блуждания, примененные последовательно.

5. Процедура классификации

5.1. Построение признаков

Имея на входе 12 комбинаций предварительной обработки, мы составляем дальше всевозможные комбинации с помощью 12 видов метрик на графах. Получаем 144 вида комбинаций, для которых используем 5 моделей классификации (Раздел 5.2). Дальше мы еще рассматриваем мешок ребер ($6 \times 2 \times 1 \times 5 = 60$ комбинаций) и признаки базового уровня, для которых мы рассматриваем те же 5 моделей классификации ($1 \times 1 \times 1 \times 5 = 5$ комбинаций). В итоге мы получаем $720 + 60 + 5 = 785$ комбинаций моделей с наборами признаков, для которых мы подбираем оптимальные параметры.

5.1.1. Признаки базового уровня. В качестве признаков базового уровня мы взяли 6 метрик, используемых авторами набора данных [7]. Заметим, что эти признаки посчитанны для бинаризованных сетей, т.е. для базового уровня будут только невзвешенные ребра.

Взвешенный коэффициент кластеризации

$$CC = \frac{1}{n} \sum_{i \in V} \frac{2t_i}{d_i(d_i - 1)}, \quad (10)$$

где t_i — количество треугольников для вершины i.

Характеристическая длина пути

$$CPL = \frac{1}{n} \sum_{i \in V} \frac{\sum_{j \in V, j \neq i} g_{ij}}{n - 1}, \quad (11)$$

где g_{ij} — длина кратчайшего пути между вершинами i и j.

Нормализованный коэффициент кластеризации

$$\lambda = \frac{CC}{CC_{rand}}, \quad (12)$$

где CC_{rand} — средний взвешенный коэффициент кластеризации CC из сгенерированных случайных графов. Эти графы мы получаем, случайным образом меняя ребра (5 изменений в среднем на одно ребро), оставляя таким образом степень вершины, но меняя шаблон соединения вершин. 100 таких случайных графов генерируется для каждого вычисления.

Нормализованная характеристическая длина пути

$$\gamma = \frac{CPL}{CPL_{rand}}, \quad (13)$$

где CPL_{rand} — средняя характерстическая длина пути CPL таких же случайных графов.

Small-worldness

$$\sigma = \frac{\lambda}{\gamma}. \quad (14)$$

Модулярность

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{am} \right] \delta(c_i c_j). \quad (15)$$

5.1.2. Мешок ребер. Простейший способ генерации признаков — рассматривать матрицу, как вектор. Каждое взвешенное ребро интерпретируется как признак и никакие взаимодействия между ими не учитываются. Для матрицы 264x264 этот метод производит 34716 признаков (потому что матрицы симметричны и с нулевой диагональю). Для нормализации случайного блуждания величина признаков будет в два раза больше, ибо матрица получится уже не симметричной. Отметим, что для рассмотрения

такого большого количества признаков мы понизили размер подвыборки на дерево в градиентном бустинге.

5.1.3. Общие метрики. Входящие степени вершин

$$k_i^{in} = \sum_{j \in V} p_{ji}. \tag{16}$$

Исходящие степени вершин не рассчитывались в силу того, что все они равны 1.

PageRank

Pagerank приблизительно рассчитывает вероятность, что некто, случайно переходящий по ссылкам в интернете, перейдет к определенной странице (вершине). Использовалась резализация igraph для вычисления.

Хабы и авторитеты (Hubs and authorities)

Это иной алгоритм оценки важности вершин в направленных графах: Авторитеты оценивают вершины в терминах входящих ребер, и хабы оценивают вершины в плане соединений к другим вершинам [8]. Использована реализация из igraph.

5.1.4. Метрики на длинах кратчайших путей. На матрице W мы вычисляем матрицу кратчайших путей $D = d_{ij}$, используя алгоритм Дейкстры. Здесь d_{ij}^W — взвешенная длина кратчайшего пути между вершинами i и j. Далее, в скобках указаны аббревиатуры для соответствующих метрик, которые будут использоваться при представлении результатов на графиках и в таблицах. Каждая метрика легко обобщается на случай входящих и исходящих ребер.

Эффективность (efficiency)

$$e_i^W = \frac{\sum_{j \in V, j \neq i} (d_{ij}^W)^{(-1)}}{n-1}. \tag{17}$$

Характеристическая длина пути (characteristic path length)

$$l_i^W = \frac{\sum_{j \in W, j \neq i} d_{ij}^W}{n-1}. \tag{18}$$

Эксцентриситет (eccentricity)

$$ecc_i^W = \max_{j \in V, j \neq i} d_{ij}^W. \tag{19}$$

Центральность близости (closeness centrality)

$$l_i^{W(-1)} = \frac{n-1}{\sum_{j \in V, j \neq i} d_{ij}^W}. \tag{20}$$

Промежуточная центральность (betweenness centrality)

Показывает число раз, когда кратчайший путь между двумя другими вершинами пролегает через данную. Мы используем взвешенную версию с кратчайшими путями, вычисленными для взвешенного графа.

$$b_i = \frac{2}{(n-1)(n-2)} \sum_{h,j \in V, h \neq j, h \neq i, j \neq i} \frac{\rho_{hj}(i)}{\rho_{hj}}, \tag{21}$$

где ρ_{hj} — число взвешенных кратчайших путей между h и j, и $\rho_{hj}(i)$ — число взвешенных кратчайших путей междоу h и j, которые проходят через i.

Итак, нами были использованы эффективность на входящих и исходящих ребрах (in- и out- efficiency), характеристические длины путей на входящих и исходящих ребрах (in- и out- characteristic path length), эксцентриситеты на входящих и исходящих ребрах (in- и out- eccentricity), центральности близости на входящих и исходящих ребрах (in- и - out closeness centrality) и промежуточные центральности на входящих и исходящих ребрах (in- и out- betweenness centrality). Точные формулы могу быть найдены в [9].

5.2. Классификаторы

Мы использовали 5 классификаторов: логистическую регрессию (Logit Model), метод опорных векторов (SVM) с линейным ядром, случайный лес (RF), стохастический зеркальный спуск (SGD huber) и градиентный бустинг деревьев (GBT). Для всех методов мы масштабировали признаки с min-max масштабированием:

$$\mathbf{x}^{scaled} = \frac{\mathbf{x} - x_{min}}{x_{max} - x_{min}}. \tag{22}$$

Для линейной классификации мы также использовали elastic-net регуляризацию:

$$\hat{\omega} = argmin_\omega \left(l(\mathbf{y}, \mathbf{x}, \omega) + \alpha \left(\frac{1-\rho}{2} ||\omega||_2^2 + \rho ||\omega||_1 \right) \right), \tag{23}$$

где $l(\mathbf{y}, \mathbf{x}, \omega)$ — классификационная функция потерь (кусочно-линейная или логистическая), α — коэффициент регуляризации и ρ — l_1 коэффициент. Мы вычисляли коэффициенты регуляризации с помощью стохастического зеркального спуска.

Реализации классификаторов брались из пакетов sklearn и xgboost. Описания параметров и все параметры по умолчанию приведены в документации к этим пакетам. Мы же, в свою очередь, приведем параметры, которые мы подбирали.

Логистическая регрессия

- Регуляризация: l_1, l_2, elastic-net.

  Для l_1 и l_2 регуляризаций:

- Параметр C: 0.01, 0.05, 0.1, ... , 0.9, 0.95.

Для elastic-net регуляризации:

- α: 0.001, 0.01, 0.1, 0.5, 1.0.
- $l_1 ratio$: 0, 0.2, 0.4, 0.6, 0.8, 1.0.
- Количество итераций: 50, 100, 200.

Для регуляризации l_1 и l_2 максимальное количество итераций составляло 50.

SVM

Без регуляризации:

- Параметр C: 0.0005, 0.001, 0.005, 0.01, ... , 0.4, 0.45.
- Ядро: линейное, полиномиальное, RBF, сигмоидное.
- Степень полинома: 2, 3, 4.
- γ: 0.01, 0.001, 0.05, 0.1, 0.2.

Максимальное количество итераций здесь было равно 100.

Для elastic-net регуляризации:

- α: 0.001, 0.01, 0.1, 0.5, 1.0.
- l_1 коэффициент: 0, 0.2, 0.4, 0.6, 0.8, 1.0.
- Количество итераций: 50, 100, 200.

SGD Huber

- α: 0.001, 0.01, 0.1, 0.5, 1.0.
- l_1 коэффициент: 0, 0.2, 0.4, 0.6, 0.8, 1.0.
- Количество итераций: 50, 100, 200.

Градиентный бустинг деревьев

- Максимальная глубина: 2, 3, 5.
- Темп обучения: 0.01, 0.05, 0.1, 0.5.
- Количество деревьев: 100 и 200.
- Размер подвыборки: 1.0, 0.7, 0.5.
- Размер подвыборки на дерево: 0.01, 0.05, 0.1, 0.3, 0.5 (для мешка ребер бралось 0.01, 0.05 и 0.1).

Случайный лес

- Количество деревьев: 100 и 200
- Критерий: Коэффициент Джинни и энтропия
- Максимальная глубина: 3, 5, 7
- Максимальное количество признаков: корень количества признаков и логарифм количества признаков по основанию два

5.3. Оценка качества

Сравнение моделей происходило в два этапа. На первом этапе подбирались оптимальные параметры. На втором этапе производилось тестирование моделей на полученных оптимальных параметрах. Качество алгоритмов измерялось с помощью ROC AUC — площади под ROC-кривой.

В подборе оптимальных параметров использовалась 10-fold кросс-валидация. Данные разбивались на 10 частей случайным образом. Для каждой части производилось обучение модели на 9 частях, а предсказание — на оставшейся. Таким образом, мы получали предсказания на всем наборе данных и получали оценку качества.

Тестирование полученных параметров производилось следующим образом: мы выбрали случайным образом 50 различных разбиений на обучающую и тестовую выборку и каждую модель тестировали на этих разбиениях. Обучающая выборка составляла пятую часть всей выборки. Для каждого разбиения вычислялась ROC AUC метрика на тестовой выборке. Для этих значений метрики качества вычислялись среднее $E(ROC\ AUC)$ и стандартное отклонение $\sigma(ROC\ AUC)$.

Данная процедура была сделана во избежание попадания в лучшие результаты моделей, которые переобучились на определенных разбиениях. Во всех таблицах использовалось 100 случайным образом выбранных разбиений вместо 50.

5.4. Инструменты

Для всех вычислений мы использовали язык Python. Вычисление матриц, графовых метрик и численных расчетов производились в NumPy [10], SciPy, NetworkX [11], igraph и louvain библиотеках. Также была использована scikit-learn библиотека [12]. Из нее брались все реализации классификаторов, кроме градиентного бустинга деревьев. Реализация градиентного бустинга была взята из xgboost пакета, приемущественно из-за скорости. Графики были простроены в matplotlib и seaborn.

6. Результаты и обсуждение

Мы уже писали аналогичную работу [6] на этих же данных, но там мы перебирали различные нормировки и взвешивания, а в качестве признаков использовали только степени вершин. В той работе лучшим классификатором оказался SVM с линейным ядром и elastic-net регуляризацией. Отметим, что в данной работе этот результат тоже присутствует. Вы можете увидеть его на втором месте в таблице 3. Здесь же мы уже решили исследовать эту задачу на предмет наличия других признаков, дающих та-

	Взвешивание	Спектральная нормировка	Признаки	Классификатор	$E(ROC\ AUC)$	$\sigma(ROC\ AUC)$
1	binar	-	базовый уровень	GBT	0.622	0.103
2	binar	-	базовый уровень	RF	0.602	0.124
3	binar	-	базовый уровень	SGD huber	0.543	0.101
4	binar	-	базовый уровень	Логит-регрессия	0.521	0.127
5	binar	-	базовый уровень	SVM	0.519	0.139

Таблица 1. Топ 5 базового уровня

	Взвешивание	Спектральная нормировка	Признаки	Классификатор	$E(ROC\ AUC)$	$\sigma(ROC\ AUC)$
1	rootwbydist	—	мешок ребер	Логит-регрессия	0.669	0.127
2	rootwbydist	+	мешок ребер	Логит-регрессия	0.659	0.113
3	rootwbydist	+	мешок ребер	Логит-регрессия	0.657	0.109
4	invdist	—	мешок ребер	SGD huber	0.644	0.158
5	sqrtw	+	мешок ребер	SVM	0.628	0.112

Таблица 2. Топ 5 мешка ребер

	Взвешивание	Спектральная нормировка	Признаки	Классификатор	$E(ROC\ AUC)$	$\sigma(ROC\ AUC)$
1	wbysqdist	—	in_efficiency	SVM	0.752	0.117
2	wbysqdist	—	in_degrees	SVM	0.752	0.122
3	wbysqdist	+	pagerank	SVM	0.748	0.128
4	wbysqdist	—	in_degrees	SVM	0.746	0.123
5	wbysqdist	—	in_efficiency	SVM	0.743	0.113
6	wbysqdist	+	pagerank	SVM	0.738	0.121
7	wbysqdist	—	in_efficiency	Логит-регрессия	0.728	0.129
8	wbysqdist	—	pagerank	SVM	0.727	0.112
9	wbysqdist	—	in_efficiency	GBT	0.722	0.114
10	wbysqdist	—	in_degrees	Логит-регрессия	0.715	0.127
11	origw	+	in_efficiency	SVM	0.698	0.113
12	wbysqdist	+	pagerank	Логит-регрессия	0.687	0.125
13	rootwbydist	+	out_efficiency	GBT	0.687	0.136
14	wbysqdist	—	pagerank	SVM	0.686	0.112
15	invdist	+	hubs	SVM	0.685	0.118
16	origw	+	in_efficiency	Логит-регрессия	0.685	0.119
17	wbysqdist	+	in_efficiency	SVM	0.682	0.107
18	wbysqdist	+	in_degrees	SVM	0.681	0.123
19	wbysqdist	+	in_efficiency	SVM	0.680	0.112
20	wbysqdist	+	in_degrees	SVM	0.678	0.123

Таблица 3. Топ 20 всех комбинаций

Обозначения к взвешиваниям: binar — бинарные веса, wbysqdist — взвешивание на квадрат расстояния между вершинами, rootwbydist — корень от wbysqdist, sqrtw — корень оригинальных весов, invdist — обратные веса.

Спектральная нормировка: наличие или отсутствие спектральной нормировки (см. раздел 4.2.2).

Обозначения к признакам: базовый уровень — признаки, используемые авторами данных, мешек ребер — см. раздел 5.1.2, pagerank и hubs — см. раздел 5.1.3, in-degrees — входящие степени вершин, in- и -out efficiency — эффективность на входящих и исходящих ребрах.

Обозначения к классификаторам: GBT — градиентный бустинг деревьев, RF — случайный лес, SGD huber — стохастический градиентный спуск, SVM — метод опорных векторов.

$E(ROC\ AUC)$ — среднее значение ROC AUC на 50 случайных разбиениях.

$\sigma(ROC\ AUC)$ — стандартное отклонение ROC AUC на этих же 50 случайных разбиениях.

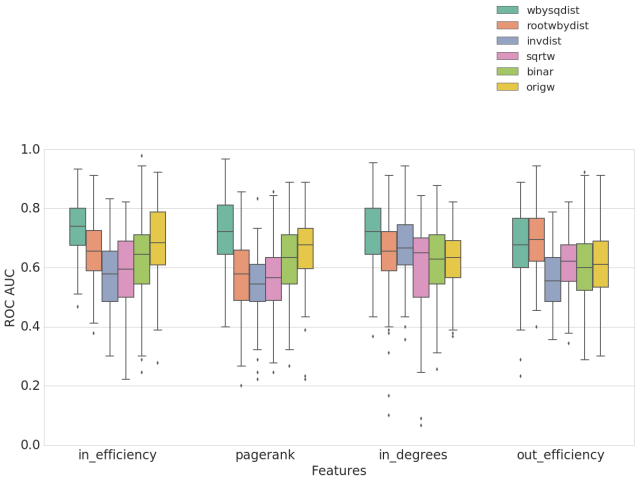

Рис. 1. Результаты для выбранных признаков (features) в зависимости от взвешиваний (разным цветом)

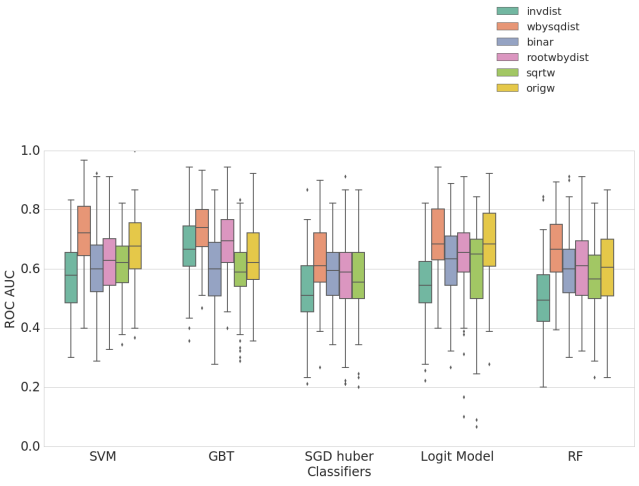

Рис. 2. Результаты для классификаторов в зависимости от взвешиваний

кие же хорошие результаты. Отметим, что результаты несколько иные в силу различия метода тестирования. Из-за этого различия будем ориентироваться на полученные нами результаты (см. таблицу 3).

В итоге на данных по расстройствам аутистического спектра были опробованы 785 комбинаций взвешиваний, нормировок, признаков и моделей. Для каждой комбинации были найдены оптимальные параметры моделей.

Как мы видим из таблиц 1, 2 и 3, показывающих результаты на этапе тестирования, признаки базового уровня дают лучший результат 0.622 ±0.103 ROC AUC, мешок ребер позволяет достигнуть 0.669 ±0.127 ROC AUC. Из полученных метрик качества на мешке ребер и признаках базового уровня становится ясным, что в качестве признаков лучше работают метрики, основанные на расстоянии. Лучше всего они проявили себя при взвешивании на квадрат расстояния между вершинами (wbysqdist). Лучший результат для этих метрик составляет 0.752 ±0.117 ROC AUC, что в среднем на 0.130 лучше признаков базового уровня и на 0.083 лучше мешка ребер. Не известно точно, насколько это эффект нормировки, а насколько это эффект признаков. Однако ясно, что лучше всего на этих признаках показали себя линейные классификаторы.

Кроме того, нашей целью было исследование других подходящих признаков. Благодаря отрицательным логарифмам (см. 3), стало возможным использование признаков, основанных на длинах кратчайших путей, для нормировки случайного блуждания с корректной интерпретацией. В итоге признаки, которые дали лучшие результаты на целевой метрике, оказались in-degrees, in-efficiency, pagerank и out-efficiency.

7. Заключение

В качестве продолжения нашего исследования [6] мы поставили перед собой задачу нахождения большего набора интерпретируемых признаков, нежели степени вершин. Поэтому в данной работе мы предложили метод генерации признаков для классификации коннектомов на основе стохастических матриц случайных блужданий. Мы проверили этот способ в задаче бинарной классификации на выборке из 94 человек (норма и люди с расстройствами аутистического спектра). Всего мы сгенерировали порядка 700 признаков, используя различные комбинации взвешиваний, нормировок, признаков и моделей. Помимо мешка ребер и базовых признаков были опробованы два типа признаков: общие метрики на графах и метрики, требующие вычисления кратчайших путей.

Мы выяснили, что признаками, показавшими наилучшую метрику качества, оказались pagerank,

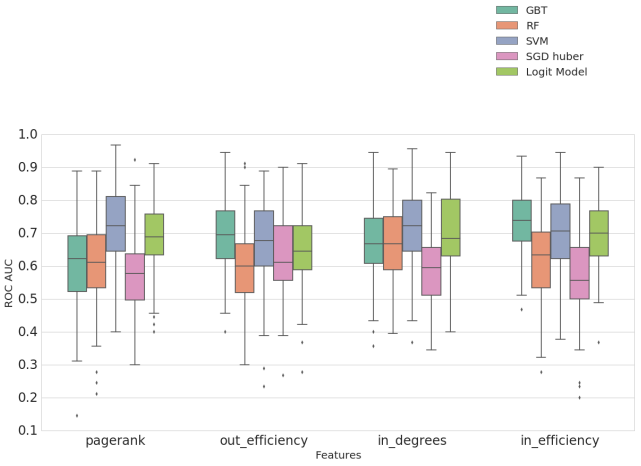

Рис. 3. Результаты для выбранных признаков в зависимости от классификатора

out-efficiency, in-efficiency и in-degrees. Для этих признаков получены результаты на уровне работ на эти данные в районе 0.75. Таким образом, мы получили интерпретируемые признаки, которые дают результаты не хуже существующих.

У нашей работы есть несколько важных ограничений. Во-первых, мы опробовали наши признаки только на одном наборе данных. Этого недостаточно, чтобы говорить о широкой применимости этих признаков. Их стоит опробовать на более крупной выборке для той же задачи классификации, желательно с применением других алгоритмов получения коннектомов. Во-вторых, неясно, насколько осмысленны наши признаки с точки зрения нейрофизиологии, несмотря на то, что за ними стоит некоторая физическая интуиция. Наконец, стоит опробовать эти признаки на других задачах классификации, чтобы понять, ловят ли они сигнал специфичный для задачи различения нормы и аутизма, или это более общие признаки.

Благодарности

Выражается благодарность авторам набора данных UCLA за предоставление возможности для исследования. Также хотелось бы поблагодарить коллег из ИППИ РАН за плодотворные дискуссии касательно работы и предоставление вычислительных ресурсов.

Статья подготовлена при проведения исследования (No 16-05-0050) в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2016 г. и в рамках государственной поддержки ведущих университетов Российской Федерации "5-100".

Список литературы

[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] 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] 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)

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

[6] Petrov, D. and Dodonova, Y. and Zhukov, L. and Belyaev, M. *Boosting Connectome Classification via Combination of Geometric and Topological Normalizations* Proc. of 6th intern. workshop on Pattern Recognition in Neuroimaging, 2016 p. 4

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

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

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

[11] 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)

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vander- plas, 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)