ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том ?? **20??** **Вып. на рецензировании**

УДК 621.391 : 519.218.5

 В.А. Любецкий, К.Ю. Горбунов, С.А. Пирогов, Г.А. Хазиев, А.И. Агламазова

КЛАСТЕРИЗАЦИЯ ТОЧЕК МНОГОМЕРНОГО ПРОСТРАНСТВА НА ОСНОВЕ ИДЕЛОЛОГИИ SEURAT[1]

В статье на математическом уровне обсуждаются современные прикладные задачи дискретной оптимизации, и для одной из них, кластеризации точек в многомерном вещественном пространстве, подробно рассмотрен алгоритм её решения. Характерной особенностью этих задач является большой размер исходных данных, часто задаваемых матрицами с десятками тысяч строк и десятками тысяч столбцов. Это приводит к проблемам обработки больших данных в условиях сложного вычислительного алгоритма; при этом нередко требуется быстро получить достаточно точное решение задачи, поэтому вопрос о сложности алгоритма имеет принципиальное значение. Решение упомянутой задачи кластеризации приводит к трудным математическим проблемам: удалению скрытых параметров; выделению значимых признаков; переходу к оптимальным и информационно значимым координатам точек; специфическом представлении (картами многообразия) вершин нагруженного графа; выбор функции, зависящей от текущей кластеризации вершин, максимизация которой приводит к искомой кластеризации; для каждого кластера выбор признаков, которые индивидуально характеризуют его; и, наконец, понижения размерности исходных данных.

Ключевые слова: оптимальное преобразование графов, эволюция строки вдоль дерева, оптимальная дискретная кластеризация.

DOI: 10.31857/S05552923??, **EDN:** ??

§ 1. Введение

В этой статье на математическом уровне изложены постановки ряда современных (но уже ставших классическими) прикладных задач и для одной из них, кластеризации множества точек в многомерном вещественном пространстве, описан алгоритм

[1] Работа выполнена в рамках государственного задания ИППИ РАН, утвержденного Минобрнауки России.

решения на основе идеологии Seurat. Этот алгоритм представляет собой пример современной обработки данных, направленный на решение сложной вычислительной задачи. Решения поставленных задач широко востребованы и используются в разных прикладных областях, но мы не касаемся здесь собственно прикладных аспектов. Хотя эвристические алгоритмы для решения рассмотренной задачи кластеризации широко применяются, авторам неизвестны цельные математические изложения её решения; в какой-то мере алгоритмы могут быть извлечены из прикладных работ, однако, там изложения строятся в контексте соответствующих прикладных областей, что затруднит математически ориентированного читателя.

Для таких задач кроме собственно математического исследования требуется найти эффективный алгоритм и доказательство того, что алгоритм действительно находит заранее математически описанное решение; кроме того, нужно найти оценку времени работы (сложность) алгоритма; и, в равной мере, найти эффективную компьютерную реализацию этого алгоритма. Поскольку эти задачи возникают в прикладном аспекте, не всегда осознаётся, что компьютерной программе предшествует математическая постановка и, по меньшей мере, математическое осмысление задачи. Иными словами, случается, что прикладник пишет эвристическую компьютерную программу до математических постановки и исследования задачи.

В заключение заметим, что наша научно-методическая статья ориентирована на широкий круг читателей; хотя лучше, если они знакомы с материалом первого курса математического факультета.

Наконец, общим местом является замечание, что алгоритмическая и компьютерная обработка больших данных – универсальное направление исследований и метод работы буквально во всех областях естественных, инженерных и гуманитарных наук. Такая обработка опирается на методы современной математики (от алгоритмов – алгебры – анализа до геометрии) и также опирается на современные приёмы эффективного программирования.

§ 2. Примеры задач

Начнём с трёх примеров математических задач, хотя методы решения первых двух из них здесь не обсуждаются (иначе пришлось бы значительно увеличить размер текста, и возникла тематическая разбросанность), но их математическое обсуждение доступно по ссылкам.

1. Даны ориентированные графы с именами рёбер, без повторения имён или с их повторениями. Такие графы иногда называют *структурами*; в сущности, даны две картинки, показанные на рис. 1. Фиксированы шесть *операций*, преобразующие одну структуру в другую; каждой операции сопоставлено строго положительное рациональное (в общем случае, вещественное) число, которое называется *ценой* данной операции. Цена показывает, сколь редко используется данная операция в

процессе, который сам описывается как минимум суммарной цены всех используемых в нём операций (с их повторениями): высокая частота использования означает низкую цену операции, а низкая частота использования – её высокую цену. Таким образом, обеспечивается адекватность этого процесса. Упомянутые шесть операций над структурами хорошо известны и не приводятся здесь, см. например, [1]. Итак, задача состоит в том, чтобы найти алгоритм, который по двум данным структурам a и b и данным ценам операций выдаёт одну из цепочек операций, которые последовательно преобразуют a в b, и имеет минимальную суммарную цену операций.

В [2] получен алгоритм, по времени работы линейный от суммарного числа рёбер в данных структурах a и b, который находит такую, минимальную при данных ценах, цепочку операций. Вообще, линейная сложность алгоритма в реальных прикладных задачах – большая редкость.

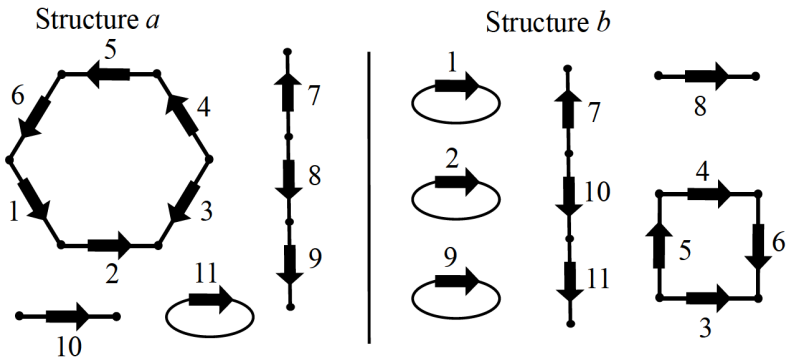

Рис. 1. Даны две структуры, нагруженные ориентированные графы a и b, и также цены каждой из шести операций над структурами. Требуется найти цепочку (последовательность) операций с их повторениями, суммарная цена которой минимальна по сравнению со всеми возможными цепочками, начинающимися с a и заканчивающимися на b.

2. В этом разделе речь идёт, скорее, о классе задач, для которых известно много эвристических алгоритмов решения. Дано корневое дерево (иногда, ациклическая сеть), для которого заданы длины рёбер, они соответствуют «дискретному времени» перехода от «предка» в начале ребра к «потомку» в конце ребра («начало» расположено ближе к корню дерева, а «конец» – дальше от него). В листьях даны современные однотипные данные из предметной области. Нелистовым вершинам однозначно сопоставлены объекты того же типа, какой представлен в листьях, – такое сопоставление называется *расстановкой* объектов по нелистовым вершинам; поскольку листьям объекты заранее сопоставлены, расстановка определена на всех вершинах. В пространстве всех расстановок задан функционал, который

зависит от переменной (в нелистовых вершинах) расстановки и от длин рёбер. Задача состоит в минимизации функционала. Иными словами, задача состоит в описании дискретной эволюции «прапредка» – объекта в корне дерева вдоль всего дерева вплоть до современных данных в листьях. Примерами объектов служат последовательности в фиксированном алфавите, которыми, как известно, можно кодировать любой конечный объект; часто графы заданного типа. Иногда задача ставится так, что и само дерево является аргументом функционала или, наоборот, дано дерево и все рёбра считаются одинаковой длины, т.е. длины отсутствуют. Нами найдены эффективные компьютерные решения ряда задач такого рода и доказаны соответствующие теоремы, хотя в целом этот важный круг задач далёк от математического освещения, см., например, [3].

Один из важных подходов к определению такой эволюции состоит в следующем. Для данных точек в n-мерном пространстве задано ещё поле их скоростей, которое интерпретируется как потенциал развития точки-клетки (см. начало следующего пункта 3). Поле скоростей преобразуется в марковскую (или близкую к ней) цепь переходных вероятностей развития точек-клеток. По цепи образуются нечёткие множества клеток, которые образуют начальные, промежуточные и финальные макросостояния, и определяются вероятности перехода из одного макросостояния в другое. Отсюда находится эволюция макросостояний, от начального в финальные, которые интерпретируются как кластеры клеток [4].

3. Перейдём к задаче кластеризации точек в многомерном вещественном пространстве, решение которой составит оставшуюся часть статьи. Дана числовая матрица, строки которой называются *признаками* (features), а столбцы – *клетками*, не имея в виду именно биологическую клетку, а любую «изолированную часть мира». Далее столбцы будут рассматриваться как точки соответствующего пространства. Элемент матрицы – вещественное число, которое называется *экспрессией* (выраженностью) признака-строки i в клетке-столбце j. *Кластеризацией* столбцов матрицы называется разбиение её столбцов на непересекающиеся множества, каждое из которых называется *кластером*. Требуется найти кластеризацию, для которой столбцы внутри любого кластера наиболее похожи друг на друга в сравнении с похожестью столбцов между разными кластерами. «Похожесть» двух столбцов определяется функционалом, который не так просто сформулировать; это будет сделано по ходу решения задачи, что, увы, отступает от принципа, который пропагандировался во Введении.

   Многочисленные эвристические решения этой задачи весьма сложны, в качестве примера мы сошлёмся на её решение в контексте прикладной биоинформатики, [5]. Мы изложим ниже в некоторых пунктах оригинальное решение, для которого разработали используемую нами эффективную компьютерную программу.

В заключение отметим, что в разделах 3-10 мы следуем общему плану метода Seurat-Louvain-Leiden. Наша цель – привлечь внимание математически ориентированного

читателя к многочисленным математическим проблемам, которые возникают в этом методе, далёком от математического не только обоснования, но даже отчасти и понимания. Это также относится и к задачам, сформулированным в разделах 1-2.

Следующий раздел 3 подробно излагается в разделах 4-10; читателю менее знакомому с этим материалом можно вначале пропустить раздел 3.

§ 3. План решение задачи кластеризации столбцов числовой матрицы.

Итак, клетка – точка в N-мерном пространстве вещественных чисел $\mathbb{R}^N$. Иными словами, *координатами* клетки j называется столбец чисел в данной матрице, который удобно представить себе "расположенным под клеткой" j. Число столбцов в этой и последующих матрицах не меняется и равно M. Приведём план решения более или менее общий для разных подходов к этой задаче. Математическое обоснование и оценка сложности предлагаемого ниже алгоритма далеки от решения, хотя активно изучаются. При желании можно прочесть пункты этого плана после остальных разделов статьи.

1. В данной числовой матрице $X = (x_{ij})$, размера $N \times M$, каждый её элемент x_{ij} заменим его долей относительно всего столбца j, так полученную матрицу обозначим также X. Для каждой строки i матрицы X вычислим среднее x_i и выборочную дисперсию σ_i^2 , и поэлементно преобразуем матрицу X в новую матрицу Y, в которой σ_i^2 уже слабо зависит от x_i.

   Напомним, что для любой матрицы размера $N \times M$ среднее i-ой строки равно

   $$x_i = \frac{1}{M} \sum_{j=1}^{M} x_{ij},$$

   а дисперсия строки равна

   $$\sigma_i^2 = \frac{1}{M-1} \sum_{j=1}^{M} (x_{ij} - x_i)^2$$

   и стандартное отклонение строки равно σ_i.
2. По матрице Y образуем новую матрицу Z. Для этого по каждой строке $i \in Y$ построим регрессию f для множества $\{(y_l, \sigma_l^2)\}$ точек на плоскости: точка – дисперсия σ_l^2 вместе со средним y_l, строки l, где y_l берётся только из заданной окрестности I_i среднего y_i. Тогда матрица Z определяется как

   $$Z = \frac{y_{ij} - y_i}{f(y_i)}.$$

   В Z выберем заданное число строк с наибольшей новой дисперсией $\delta = \frac{\sigma_i^2}{f(i)}$. Обозначим $\widetilde{Z}$ часть матрицы Z, содержащую только эти строки.

3. Для столбцов матрицы $\widetilde{Z}$ найдём лучшие координаты – координаты по базису из левых сингулярных векторов этой матрицы. Затем спроектируем столбцы этой матрицы на подпространство из n наиболее информативных таких векторов, где информативность зависит от соответствующих сингулярных чисел. Полученную матрицу обозначим Z^*, её размер $n \times M$. Само n определяется по доле необъяснённой дисперсии для $\widetilde{Z}$. Заметим, что квадраты сингулярных чисел и левые сингулярные векторы совпадают с собственными числами и собственными векторами, соответственно, ковариационной матрицы $\frac{1}{M-1}\widetilde{Z}^T \cdot \widetilde{Z}$ признаков, размера $M \times M$.
4. Столбцы матрицы Z^* одновременно являются точками в $\mathbb{R}^n$ (столбцов-точек M штук) в n-ом вещественном пространстве $\mathbb{R}^n$ и одновременно ещё они будут *вершинами* следующего графа G. Для каждой из этих точек j по евклидову расстоянию до других точек, найдём список из k ближайших соседей точки j (этот j-список начинается с самой j), и рассмотрим *ранги* точек из j-списка по возрастанию расстояния от j.
5. В графе G проведём ребро между вершинами j и l, если их списки имеют общего k-соседа. Рёбрам припишем веса, что закончит определение нагруженного неориентированного графа G. Выберем некоторую начальную кластеризацию $\mathcal{C}_0$ вершин в G.
6. Итеративно найдём *максимум функции модулярности*, аргументом которой является текущая кластеризация $\mathcal{C}$ вершин в G; в результате найдём финальную кластеризацию вершин в G (вершины – столбцы в матрице Z^*).
7. Вернёмся к матрице Y, которая имеет n и M исходных строк и столбцов, но теперь ещё и кластеризацию её клеток (назовём её *матрицей кластеризации*). В каждом кластере найдём *дифференциально-экспрессированные строки* (DE-строки или, то же самое, *DE-признаки*) с помощью Mann-Whitney U-теста и процедуры Benjamini-Hochberg.
8. Граф G (с координатами в n-ом пространстве) можно приблизить другим графом G_1, вершины которого находятся в K-ом пространстве, где K много меньше, чем n, например, $K = 2$. Это делается с помощью нелинейной функции, называемой *UMAP*; *граф* G_1 в K-ом пространстве находится как минимум *дивергенции Кульбака-Лейблера*. Решаемая здесь задача понижения размерности данных (то есть размерности n до размерности K) сама по себе – одна из центральных при работе с большими данными. Вообще говоря, такое понижение размерности можно выполнить от любой размерности, например, от N к K.

§ 4. лог-Преобразование матрицы X

Часто дисперсия строки зависит от её среднего. Например, пусть $X = \alpha + \mu e^{\eta} + \varepsilon$, где X – измеряемая случайная величина и μ – её точное значение, а ε и η – аддитивная и мультипликативная погрешности. Пусть параметры подобраны так, что

каждая строка является выборкой случайной величины X со средним u и дисперсией v. Тогда $u = \alpha + \mu \cdot \mathbf{E}(e^\eta)$, и $v = \mu^2 \mathrm{Var}(e^\eta) + \mathrm{Var}(\varepsilon)$. Отсюда дисперсия v зависит от среднего u; а именно, $v = \frac{\mathrm{Var}(e^\eta)}{\mathbf{E}(e^\eta)^2} \cdot (u - \alpha)^2 + \mathrm{Var}(\varepsilon) = (c_1^2 u + c_2)^2 + c_3$ с тремя параметрами c_1, c_2, c_3. Здесь, например, $c_3 = \mathrm{Var}(\varepsilon) + \alpha(\alpha - \frac{\mathrm{Var}(e^\eta)}{\mathbf{E}(e^\eta)})$.

Мы хотим избавиться от «скрытого параметра» u с тем, чтобы после преобразования матриц, $Y = g(X)$, дисперсия $\mathrm{Var}(Y)$ слабо зависела от математического ожидания $\mathbf{E}(Y)$, то есть в существенном зависела только от самой строки.

Для этого разложим Y по степеням $X - u$:

$$Y = g(u) + g'(u)(X - u) + g''(u)(X - u)^2 + \dots.$$

В разложении оставим только аффинную часть и получим $\mathbf{E}(Y) \approx g(u)$ и $\mathrm{Var}(Y) \approx g'(u)^2 \cdot v$.

Пусть $x_{i1}, \dots, x_{ij}, \dots, x_{iM}$ — строка i в матрице X со средним $x_i = u$, а $y_{i1} = g(x_{i1}), \dots, y_{ij} = g(x_{ij}), \dots, y_{iM} = g(x_{iM})$ — значения функции g, строка со средним $y_i = \mathbf{E}(Y) \approx g(u)$. Более того, $g(x_i) \approx g(u) \approx y_i$ и $\mathrm{Var}(Y) = \sigma_i^2(Y) \approx v(u) g'(u)^2$.

Положим $v(u)g'(u)^2 = 1$. Отсюда $g(u) = \int\limits_0^u \frac{du}{\sqrt{v(u)}} + C$. Интегрирование приводит к гиперболо-арсинусному преобразованию, указанному ниже.

Итак, применим функцию g к каждой ячейке в X, учитывая $\sigma_i^2(Y) \approx 1$, а векторы $(1, \dots, 1)$ и $(\dots, y_i, \dots)$ можно рассматривать как слабо зависящие друг от друга.

Таким образом, в нашем алгоритме для каждого элемента x_{ij} исходной матрицы X сначала вычислим его долю в столбце j и затем применим гиперболо-арсинусное преобразование:

$$x_{ij} \to y_{ij} = \begin{cases} \frac{1}{c_1} \cdot \mathrm{arsinh}(\frac{c_2}{\sqrt{c_3}} + \frac{c_1}{\sqrt{c_3}} \cdot x_{ij}), c_3 > 0 \\ \frac{1}{c_1} \cdot \ln(c_2 + c_1 \cdot x_{ij}), c_3 = 0, \end{cases}$$

где гиперболический арсинус определяется как

$$\mathrm{arsinh}(z) = \ln(z + \sqrt{z^2 + 1}),$$

а константы c_1, c_2, c_3 подбираются по исходным данным.

§ 5. Выделение строк с большой дисперсией

Для матрицы Y и каждой её строки i вычислим среднее y_i и дисперсию σ_i^2. Построим полиномиальную низкой степени регрессию $f(\cdot)$ для множества точек $\{(y_i, \sigma_l^2)\}$, где y_i в середине интервала $\mathcal{O}_\rangle$, а интервалы $\{\mathcal{O}_\rangle\}$ выбраны так, что число точек в каждом из них примерно одинаково, рис. 2. Положим $f(i) = f(y_i)$ – значение регрессии $f(\cdot)$ в y_i. Регрессия относится к интервалу $\mathcal{O}_\rangle$, и в этом смысле называется

локальной, а число $f(i)$ можно назвать *регрессионной дисперсией* строки i (иногда её называют средне-дисперсионным отношением). Она характеризует «дисперсию» строки i без учёта её индивидуальности. Иными словами, нас будут интересовать строки, для которых дисперсия σ_i^2 не подчиняется общему правилу, заданному регрессией $f(\cdot)$, то есть σ_i^2 и $f(i)$ значительно отличаются, $f(i)$ существенно меньше σ_i^2.

Образуем матрицу $Z = \frac{y_{ij}-y_i}{f(i)}$. Для её строк i среднее равно $z_i = 0$, а дисперсия равна $\delta_i = \frac{\sigma_i^2}{f(i)}$. В Z слишком большие дисперсии рассматриваются как артефакт исходных данных; поэтому строки, у которых $\delta_i^2 > \sqrt{M}$, удаляются из Z. В так полученной матрице Z оставим только строки с наибольшей регрессионной дисперсией δ_i в заданном количестве. Обозначим $\widetilde{Z}$ часть матрицы Z, содержащую только выбранные строки (например, 2000); тогда её размер $2000 \times M$.

Итак, строки матрицы $\widetilde{Z}$ слабо зависят от среднего и не подчиняются регрессионному правилу, имея большую дисперсию. Это позволяет отчётливо различать столбцы по их координатам.

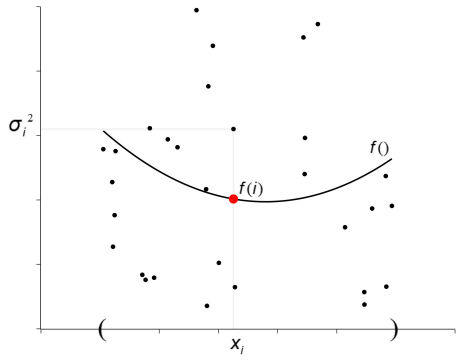

Рис. 2. Показана квадратичная регрессия, локальная на интервале $\mathcal{O}_\rangle$ с серединой y_i, которая определяет регрессионную дисперсию $f(i)$ строки i матрицы Y. В этом интервале точки плоскости определяются как среднее y_l (абсцисса) строки l и её дисперсия σ_l^2 (ордината).

В результате каждая клетка-столбец получает 2000 координат, представляющих точку в 2000-ом пространстве вещественных чисел $\mathbb{R}^{2000}$. Получим множество $\mathcal{M}$ из M точек в этом пространстве.

§ 6. Выбор лучших координат для точек из множества $\mathcal{M}$

Выберем ещё лучшие, *новые координаты* клетки j, которые будут линейными комбинациями её старых координат, столбцов матрицы $\widetilde{Z}$. Это выполняется в два шага.

1-ый шаг. Для матрицы $\widetilde{Z}$ вычислим сингулярные числа и левые сингулярные векторы u_1, u_2, ...единичной длины, и перейдём к *ортонормированному* базису в $\mathbb{R}^{2000}$, состоящему из этих векторов, рис. 3. Удобно считать, что сингулярные числа расположены по убыванию $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{2000}$, как и соответствующие им строки матрицы и новые координатные оси. Новые координаты клетки j (её старые координаты – элементы столбца $\widetilde{Z}_j$) равны $(u_k, \widetilde{Z}_j)$, где указана k-я координата, $1 \leq k \leq 2000$. Получен результат 1-го шага – матрица размера $2000 \times M$, которую обозначим $\widehat{Z}$.

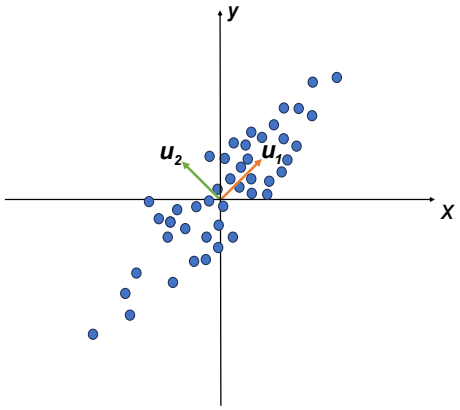

Рис. 3. Для клеток j показан переход от их старых координат – столбцов матрицы $\widetilde{Z}$ к их новым координатам – столбцам матрицы $\widetilde{Z}'$.

2-ой шаг. Оставим в $\widetilde{Z}'$ только n новых и наиболее информативных координат, которые назовём *сокращёнными* (или наиболее информативными); они образуют матрицу Z^* размера $n \times M$, которая получена из матрицы $\widetilde{Z}'$ 1-го шага размера $2000 \times M$.

Выбор числа n основан на следующем соображении информативности новых координат. *Для краткости* будем далее опускать знаки $\widetilde{\ }'$. Замена старых координат Z_j клетки j на её проекцию на первые n координат $Z_j \to Z_j^* = \sum_{k=1}^{n} (u_k, Z_j) \cdot u_k$ вносит *средний квадрат ошибки* в расчёте на одну клетку, равный $\frac{1}{M} \sum_{j=1}^{M} \|Z_j - \widetilde{Z_j}'\|^2 = \sum_{k=n+1}^{N} \lambda_k$, где $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{2000}$. Эта сумма называется *остаточной дисперсией*.

Величина $\frac{1}{M}\sum_{j=1}^{M}\|\sum_{k=1}^{n}(u_k, Z_j)^2 \cdot u_k\|^2 = \frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{n}(u_k, Z_j)^2 = \sum_{k=1}^{n}\lambda_k$ называется *объяснённой дисперсией. Относительная ошибка* δ_n определяется как отношение остаточной дисперсии к выборочной дисперсии; эта доля называется *необъяснённой дисперсией*: $\delta_n = \frac{\lambda_{n+1}+\cdots+\lambda_{2000}}{\lambda_1+\lambda_2+\cdots+\lambda_{2000}}$.

Выберем n из условия $\delta_n \le \varepsilon_0$, где ε_0 – заданный порог.

§ 7. Кластеризация клеток по их новым сокращённым координатам (столбцам в матрице Z^*)

Образуем граф G: его вершины – клетки-точки j с их n сокращёнными и новыми (наиболее информативными) координатами – столбцами Z_j^* матрицы Z^*. Этот граф называется kNN-графом, а координаты клетки-точки-вершины j называются её *PCA-координатами.*

Для каждой вершины j по евклидову расстоянию между точками в n-мерном пространстве составим список k ближайших к ней вершин (k-соседей, сама j первая в этом j-списке, где k параметр). *j-Окрестностью* в G называется j-список вместе со всеми рёбрами, соединяющими любые две вершины из него. Заметим, что таким образом мы переходим к новым окрестностям для всех вершин j: к картам, которые представлены одномерными симплексами на многообразии всех клеток.

Список нумеруем подряд натуральными числами от 1 (которые называют *рангами*) по возрастанию евклидова расстояния. Обозначим $\mathrm{rank}(v, j)$ – ранг вершины v в j-списке, который начинается с вершины j. Вершины, которые находятся на одинаковом расстоянии от j, нумеруются дробными числами: например, вершины, находящиеся на расстоянии 1, 2, 7, 7, 8, ... имеют ранги 1, 2, 3,5, 3,5, 4, ...

Вершины j и l соединим ребром, если $j \ne l$, а l является некоторым соседом в j-списке и j является некоторым соседом в l-списке.

Вес A_{jl} ребра $e = (j, l)$ положим равным максимуму по всем общим k-соседям v для j и l: $A_{jl} = \max\{k - 0,5(\mathrm{rank}(v, j) + \mathrm{rank}(v, l))|v\}$.

Найдём некоторую начальную кластеризацию $\mathcal{C}_0$ вершин графа G (разбиение его вершин). Выбор $\mathcal{C}_0$ важен и существенно влияет на окончательный результат кластеризации, известно несколько способов такого выбора. Полученные в результате кластеры могут допускать рёбра между вершинами из разных кластеров; о таких рёбрах говорят, что они "*между кластерами*"; наоборот, о рёбрах, у которых оба края принадлежат одном кластеру, говорят, что они "*в кластере*".

Начиная с начальной кластеризации $\mathcal{C}_0$, итеративно найдём локальный максимум функции $H(\mathcal{C})$, которая называется *модулярностью* кластеризации $\mathcal{C} = \mathcal{C}(\mathcal{G})$ графа G и характеризует качество разбиения вершин в G. Функция модулярности

определятся как

$$H(\mathcal{C}) = \frac{1}{2m} \sum_{j,l} \left(A_{jl}(e) - \gamma \cdot \frac{k_j \cdot k_l}{2m} \right) \to \max .$$

Здесь A_{jl} – вес ребра e, которое пробегает *все рёбра* во всех кластерах текущего разбиения $\mathcal{C} = \mathcal{C}(G)$, а m – сумма весов всех рёбер в G, и k_j сумма весов всех рёбер в G, инцидентных вершине j; γ – параметр, называемый *гранулярностью*: минимум суммарного веса рёбер в кластере (относительно суммарного веса всех рёбер) по всем кластерам, делённый на максимум аналогичной доли суммарного веса рёбер между всеми парами разных кластеров.

О параметре γ заметим следующее. При его увеличении число отрицательных рёбер e может увеличиться, и становится выгодным разбить кластер на несколько кластеров, чтобы отрицательные рёбра сосредоточились между кластерами и, тем самым, были исключены из H. Тогда число кластеров увеличится, а сами они уменьшатся. Наоборот, при уменьшении γ число положительных рёбер может увеличиться, поэтому становится выгодным объединить несколько кластеров в один, чтобы включить в H межкластерные рёбра, ставшие положительными. Тогда число кластеров уменьшится, и они увеличатся.

Нетривиальный выбор вида функции $H(\mathcal{C}(G))$ принципиально влияет на результат кластеризации; отметим следующие соображения по поводу выбора и обоснования функции модулярности. Обозначим $r \in \mathcal{C}$ – кластер данной кластеризации $\mathcal{C}$. Веса A_{jl} рёбер в G можно интерпретировать как мультиграф без весов: вершины j и l соединяются рёбрами в количестве A_{jl}. Обозначим той же буквой A_{jl} симметричную матрицу инцидентности полученного *мультиграфа* без петель, который так же обозначим G (на диагонали находятся нули, остальные числа равны весу $\frac{A_{jl}}{2}$; как обычно, у петли вес удваивается). Тогда $A_{jl} = A_{lj}$ и суммирование по всем рёбрам $j, l \in r$ приводит к удвоенной сумме числа рёбер в r. В формуле для H перейдём от суммирования по рёбрам $e = (j, l) \in G$ к суммированию по кластерам $r \in G$ и получим

$$H(\mathcal{C}(G)) = \frac{1}{m} \sum_{r} \left[(\frac{1}{2} \cdot \sum_{(j.l) \in r} A(j, l)) = \frac{k(r)^2}{4m} \right],$$

где m – число рёбер в G и $\sum_{(j.l) \in r} A(j, l)$ – удвоенное число рёбер внутри кластера r. Здесь $k(j)$ – степень инцидентности вершины j в графе G, вычисляемая по матрице инцидентности A_{jl}, а $k(r) = \sum_{j \in r} k(j)$ – степень инцидентности кластера r. Легко видеть, что $\sum_{j \in G} k(j) = 2m$. Итак, уменьшаемое – число рёбер в $r \in \mathcal{C}(G)$, и мы хотим интерпретировать вычитаемое.

Это можно сделать как среднее число рёбер в семействе случайных графов с теми же вершинами (и параметрами), что у G и той же кластеризацией этих вершин. Однако приведём более простое комбинаторное пояснение. На множестве вершин мультиграфа G с его кластеризацией рассмотрим "идеальный" (не мульти, обычный) граф G' с той же кластеризацией и теми же степенями инцидентности $k(j)$, что у вершин мультиграфа G. Напомним, что в G и G' степень инцидентности определяется как сумма весов всех инцидентных ей рёбер. А именно, G' полный с петлями и в нём вес ребра (j,l) полагаем равным $\frac{k(j)k(l)}{2m}$, если $j \neq l$, и $\frac{k(j)k(l)}{4m}$, если $j = l$. В G' для любой вершины j степень инцидентности равна $\sum_{l\neq j} \frac{k(j)k(l)}{2m} + \frac{2k(j)^2}{4m} = k(j)$. И в G' сумма весов всех рёбер в r равна $\sum_{j\neq l,\{j,l\}\in r} \frac{k(j)k(l)}{2m} + \sum_j \frac{k(j)^2}{4m} = \frac{k(r)^2}{2m}$. Последнее проверяется прямым раскрытием скобок в $k(r)$. Повторим, что графы G и G' имеют в качестве общих параметров саму кластеризацию $\mathcal{C}$, количества вершин и рёбер, и степени инцидентности $\{k(j)\}$ для всех вершин j. Величины $k(j)$ можно рассматривать как характеристики неоднородности графа, которые одинаковы у G и G': у них в одних и тех же областях плотность рёбер повышена или понижена по сравнению со средней плотностью.

7.1. Итеративный алгоритм кластеризации Leiden. Изложенный выше алгоритм уточняется следующим образом. Для исходного графа G выбирается начальная кластеризация $\mathcal{C}_0$ и максимизируется функция модулярности $H(\mathcal{C}(G))$. По полученному графу строим новый граф G_1, в котором все вершины одного кластера в G представлены одной новой вершиной в G_1. Все рёбра внутри кластера в G заменяются петлёй в G_1 у этой новой вершины, а все рёбра между двумя разными кластерами в G заменяются одним новым ребром в G_1 между соответствующими новыми вершинами. Вес петли полагаем равным суммарному весу исходных соответствующих внутрикластерных рёбер, а вес нового ребра – суммарному весу исходных соответствующих межкластерных рёбер. После этого максимизируем функцию $H(\mathcal{C}(G_1))$. Таким образом процесс продолжается, пока максимум $H(\mathcal{C}(G_k))$ на очередном графе G_k увеличивается, рис. 4. Затем последовательными объединениями вернёмся к исходному графу G. А именно, если вершины в G_{k-1} входят в один и тот же кластер в G_k, то соответствующие им кластеры объединяются в один кластер в G_{k-1}. И обратным ходом придём к кластеризации исходного графа G. Получим кластеризацию вершин в исходном G. При каждом переходе от $k-1$ к k качество соответствующей кластеризации графа G_k не уменьшается.

7.2. К обоснованию итеративного алгоритма кластеризации Louvain-Leiden. В качестве начальной кластеризации можно выбрать *синглетонную кластеризацию*, которая состоит из всех вершин графа G_k, рассматриваемых как одноэлементные множества ("синглетоны"). Но обычно в качестве начальной кластеризации берут более крупную кластеризацию, называемую *окрестностной*; обозначим её C_0. Граф

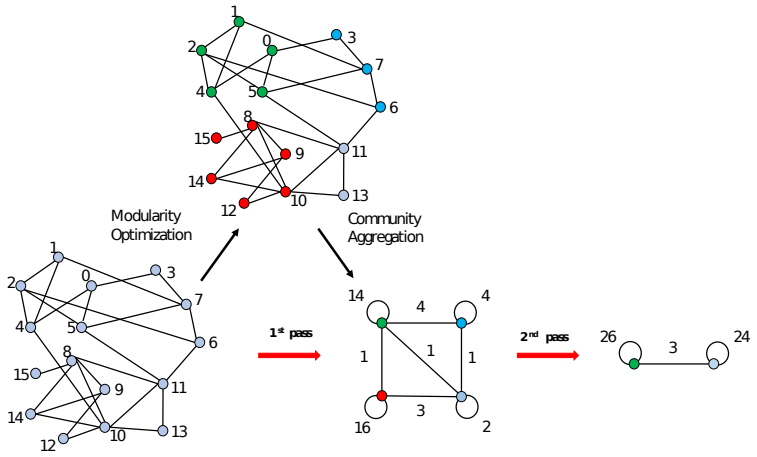

Рис. 4. Веса всех рёбер единичные. Показаны шаги итеративного алгоритма Leiden. Пример взят из [6].

G_k далее обозначим G, например, это граф G, определённый в начале этого раздела, с которого и начинает работу итеративный алгоритм. Для построения этой C_0 используются описанные в начале этого раздела j-окрестности всех вершин в G, и веса рёбер графа G. Опишем кластер $c \in C_0$, каждый кластер образуется по некоторой вершине $j \in G$ и в этом смысле обозначается c_j (кластер с "центром" в j). Кластеры c_j образуются итеративно, каждый начиная с j-окрестности. В подграфе c_j обозначим $d(v)$ сумму весов рёбер в c_j, инцидентных v; это – степень вершины v, отнесённая к c_j. Проредим каждое c_j, удаляя из него вершины $v \in c_j$ вместе со всеми инцидентными им рёбрами, если $\frac{d(v)}{|c_j|} < r$, где $r \in (0, 1]$ – порог и $|c_j|$ – сумма весов всех рёбер в c_j. Удаление таких вершин из c_j зависит от порядка просмотра вершин в c_j; продолжаем удаление, пока такая вершина находится. Удаления выполняются независимо для каждого j. Если $|c_j| = 0$, то вершину j сохраним; если v удаляется из всех j-окрестностей в G, то оставим v в качестве синглетона.

После этого в полученном наборе подграфов c_j устраним их пересечения по вершинам: пусть v принадлежит нескольким c_j. Для каждого c_j вычислим *средний вес* рёбер, соединяющих v со всеми вершинами в c_j. Вершину v сохраним в той c_j, для которой средний вес максимален, а из остальных c_j удалим её. Итоговые c_j образуют кластеры начальной кластеризации C_0 в графе G.

Наметим доказательство того, что при каждом переходе от k к $k + 1$ качество соответствующей кластеризации графа G_k не уменьшается.

Для краткости докажем здесь более высокое качество кластеризации C_0 по сравнению с синглетонной в упрощённом случае, когда учитываются только k рёбер при каждой вершине, а вершины из окрестностей не удаляются и ближайший сосед двух вершин из j-окрестности – сама точка j, и ранги – целые попарно различные числа. Пусть G – полный граф с петлями: так будет, если отсутствующие рёбра положить

с весом 0. Напомним, что значение H на любой кластеризации в G (с точностью до мультипликативной константы) равно сумме разностей весов рёбер, лежащих внутри кластера в G и в идеальном графе G'. Петли лежат внутри кластера при любой кластеризации и, следовательно, приносят одинаковый вклад в H для синглетонной и окрестностной кластеризаций. Поэтому нам требуется оценить суммы весов в G и G' рёбер, которые лежат внутри кластера в окрестностной кластеризации, но не в синглетонной. Это все рёбра, не являющиеся петлями, в кластерах $c_j \in C_0$.

Поэтому в G сумма весов рёбер, которые не петли, из $c_j \in C_0$ равна $\frac{k^2(k-1)}{2} - \frac{1}{2}\sum_{i\neq l}(i+l) = \frac{k^2(k-1)}{2} - \frac{(k-1)k(k+1)}{4} = \frac{k(k-1)^2}{4}$, здесь i и l – ранги вершин в c_j, которые принимают значения от 1 до k. Оценим сумму k_i весов рёбер, инцидентных вершине ранга i из $c_j \in C_0$:

$$k_i = k(k-1) - \frac{1}{2}\sum_{l\neq i}(i+l) = \frac{3k^2 - 5k - 2ik + 4k}{4} \leqslant \frac{3k(k-1)}{4}.$$

При $\gamma = 1$ в идеальном графе G' сумма весов рёбер из c_j равна $\frac{1}{2m}\sum_{i\neq l\in c_j} k_i k_j \leqslant \frac{1}{2m}\left(\frac{3}{4}k(k-1)\right)^2 \frac{k(k-1)}{2} = \frac{9}{64m}k^3(k-1)^3$. Итак, качество окрестностной кластеризации больше качества синглетонной, если $\frac{k(k-1)^2}{4} > \frac{9}{64m}k^3(k-1)^3$, то есть $m > \frac{9}{16}k^2(k-1)$. Обычно обрабатываются графы, у которых $m > 15000$, а k не больше 30, так что это неравенство выполняется.

Теорема 1. *Пусть граф G полный с петлями и C_k – итоговая кластеризация k-го уровня, а C_{k+1} – кластеризация $(k+1)$-го уровня, в которой все кластеры – синглетоны. Тогда $H(C_k) \leqslant H(C_{k+1})$.*

Доказательство. Назовём кластеры кластеризации в C_{k+1} $(k+1)$-го уровня крупными, кластеры кластеризации в C_k k-го уровня – средними, кластеры кластеризации в C_{k-1} $(k-1)$-го уровня – мелкими. Аналогично назовём и сами кластеризации в этих графах.

Из определения веса ребра между кластерами следует, что суммарный вес m всех рёбер между кластерами не меняется при переходе к кластеризации следующего уровня (независимо от того, являются ли кластеры следующего уровня синглетонами). И также следует: если крупные кластеры – синглетоны, то суммарный вес всех рёбер, лежащих внутри такого синглетона (а это вес петли при соответствующем среднем кластере) равен суммарному весу рёбер между мелкими кластерами, лежащими внутри среднего. Суммируя этот вес по синглетонам, получим, что сумма $\sum A_{jl}$ из формулы для H для крупной кластеризации равна этой сумме для средней.

Осталось рассмотреть сумму произведений $k_j \cdot k_l$ для пар (j, l) вершин, лежащих внутри кластера данной кластеризации. Любой крупный кластер-синглетон a состоит из одного среднего кластера, который обозначим a'. Для крупной кластеризации слагаемое в упомянутой сумме – квадрат веса петли при a', назовём её первой сум-

мой. Для средней кластеризации это – сумма по всем парам (j, l) мелких кластеров (в a') произведений $k_j \cdot k_l$; назовём её второй суммой.

Первая сумма не больше второй. Действительно, в первой сумме при раскрытии скобок возникают произведения $w(e_1) \cdot w(e_2)$ весов рёбер e_1 и e_2 между мелкими кластерами, лежащими внутри a', причём каждое произведение встретится два раза, если $e_1 \neq e_2$ и один раз, если иначе. Во второй сумме при раскрытии скобок также возникнут произведения $w(e_1) \cdot w(e_2)$ весов рёбер e_1 и e_2, уже не обязательно лежащих внутри a'. Однако каждое такое произведение для рёбер, лежащих внутри a', встретится не меньше двух раз, если $e_1 \neq e_2$ и не меньше одного раза, если иначе. Поэтому часть суммы $\sum k_j k_l$ из формулы для H, относящаяся к одному кластеру-синглетону a, для крупной кластеризации не превосходит аналогичную часть суммы (относящуюся к a') для средней кластеризации. Суммируя по всем синглетонам и учитывая, что сумма $\sum k_j k_l$ входит в H со знаком минус, получим утверждение. ▲

Однако нет гарантии, что функция H не уменьшится на обратном ходе процедуры. Поэтому, мы начинали этот ход с итоговой кластеризации каждого, не обязательно заключительного, уровня и из всех таким образом полученных кластеризаций в G выбирали кластеризацию с максимальным значением H.

§ 8. Дифференциально-экспрессируемые признаки.

Уже получена кластеризация (разбиение) $\{G_s\}$ клеток-столбцов исходной матрицы Y размера $N \times M$, которая прошла log-нормирование, но не шкалирование-центрирование; матрица состоит из старых координат, до перехода к PCA-координатам и сокращения менее информативных из них; здесь s пробегает все кластеры. Для каждого кластера G клеток и каждой строки i определим дифференциально экспрессируемые признаки-features. Такие признаки-строки сокращённо называются *DE-признаками* (или *DE-строками*). Говоря интуитивно, признак-строка i называется *дифференциально-экспрессируемой*, если экспрессии в клетках $j \in G$ статистически значимо отличаются от экспрессий во всех остальных клетках $l \notin G$ той же строки i. Это понимание *DE-строк* для данного кластера G ниже уточняется: сначала с помощью Mann-Whitney U-теста [7–9], затем с помощью процедуры Benjamini-Hochberg [10], а также ещё ряда условий.

Итак, фиксирована строка i (её имя и соответствующий индекс можно не упоминать) и фиксирован кластер G, который индуцирует разбиение строки на два множества, которые будем обозначать теми же буквами G и $\neg G$, по-прежнему называя их кластерами. Расположим все экспрессии строки i в порядке их возрастания; и присвоим числам в полученной последовательности экспрессий *ранги* – натуральные числа также по их возрастанию, начиная с 1. Одинаковым числам присвоим средний по ним ранг. Пусть R_1 – сумма рангов экспрессий в данном кластере G, и R_2 – сумма рангов экспрессий вне этого кластера, т.е. в $\neg G$; пусть m_1 и m_2 – мощно-

сти множеств G и $\overline{G}$, соответственно; $m_1 + m_2 = M$. Поскольку $R_1 + R_2 = M \cdot \frac{1+M}{2}$, можно оперировать только с R_1 или только с R_2.

Предполагается, что G и $\overline{G}$ – *независимые* выборки из двух непрерывных распределений $\mathfrak{F}$ и $\mathfrak{L}$, которые *совпадают* или *не совпадают*. Для экспериментально-математических выборок G и $\neg G$ вопрос об их независимости трудно разрешим, так что он остаётся на усмотрение вычислителя; в нашей ситуации это является предположением. Если $\mathfrak{F} = \mathfrak{L}$, то для экспрессий $y_j \in G$ и $y_l \in \neg G$ выполняется $\mathbf{P}(y_j < y_l) = 1/2$. Это свойство называется отсутствием "*доминирования*" G и $\neg G$ друг над другом. Обратное неверно: например, если $\mathfrak{F}$ и $\mathfrak{L}$ – два нормальных распределения с одинаковыми математическими ожиданиями, но разными дисперсиями, то доминирование отсутствует: $\mathbf{P}(y_j - y_l < 0) = \frac{1}{2}$, так как разность этих распределений имеет нулевое математическое ожидание.

Итак, мы хотим проверить, выполняется ли *0-ая гипотеза*, которая состоит в "отсутствии доминирования G и $\neg G$ друг над другом". В случае отрицания (говорят: отвержения) 0-ой гипотезы, строка i называется *DE-признаком*. Если 0-ая гипотеза отвергается, то на том же уровне достоверности выполняется $\mathfrak{F} \neq \mathfrak{L}$.

8.1. Критерий Манна-Уитни-Вилкоксона (Wilcoxon rank-sum test, Mann-Whitney U test и определение *p*-value; сокращённо МУВ). По строке i матрицы Y образуем случайную величину $U = R_2 - \frac{m_2(m_2+1)}{2} = \sum_{j=1}^{m_1} \sum_{l=1}^{m_2} I(x_j < x_l)$, где x_j и x_l – экспрессии в кластере G и вне него, то есть в $\neg G$, соответственно; а $I(\cdot)$ – характеристическая функция условия, которое приведено в скобках. Если 0-ая гипотеза *выполняется*, то её математическое ожидание и дисперсия равны $\mathbf{E}(U) = \frac{m_1 \cdot m_2}{2}$ и $\sigma(U)^2 = m_1 \cdot m_2 \frac{M+1}{12}$.

В разделе 6 выполнено центрирование и нормирование матрицы Y. Аналогично этот общий приём применяется и к U, чтобы получить статистику (случайную величину) $Z = \frac{U - m_1 m_2/2}{\sqrt{m_1 m_2 (M+1)/12}}$. Если в последовательности рангов наблюдается асимметрия (относительно линейного порядка чисел) экспрессий в пользу G или, наоборот, в пользу $\neg G$, то Z *отклоняется* вправо или влево от нуля, больше или меньше. Если это отклонение Z превосходит заданный порог Z_0, то строку-признак i считаем DE; уточним этот подход.

Повторим, пусть G и $\neg G$ – выборки двух некоторых случайных величин. Повторим, 0-я гипотеза состоит в том, что $\mathbf{P}(y_j < y_l) = \frac{1}{2}$ для $y_j \in G$ и $y_l \in \overline{G}$ (говорят: «y_j не доминирует y_l, а y_l не доминирует y_j»). Эта гипотеза выражает "отсутствие порядковой асимметрии" между G и $\overline{G}$. Заметим, что распределение для Z задаётся мерой на $\mathbb{R}$, которая определяется как Z^{-1} от меры на M-ом пространстве, индуцированной распределениями $\mathfrak{F}$ и $\mathfrak{L}$.

При выполнении 0-й гипотезы и возрастающих $m_1, m_2 \to \infty$, распределение статистики Z стремится к нормальному 0-1 распределению, обозначаемому $N(0,1)$. В

силу этого приближённо случайная величина Z имеет распределение $N(0,1)$, то есть вместо обычно неизвестных распределений $\mathfrak{F}$ и $\mathfrak{L}$ вероятность, далее везде обозначаемую $\mathbf{P}$, будем вычислять по нормальному распределению $N(0,1)$.

Хотя это дальше не используется, заметим: если экспрессия в G доминирует экспрессию в $\neg G$, то есть $\mathbf{P}(y_j < y_l) > 1/2$, то Z стремится к $+\infty$, а если экспрессия в G в том же смысле доминирует экспрессию в $\neg G$, то Z стремится к $-\infty$ при также возрастающих m_1, m_2.

Выберем Z_0 из условия $2\mathbf{P}(x > Z_0) = \alpha$, тогда Z_0 называется *квантилем уровня значимости* α; само α задаётся вычислителем, например, $\alpha = 0,1$.

Значение p-value для данной строки i, обозначаемое $p_{\mathrm{val}}(i)$, определяется по распределению $N(0,1)$ как удвоенная площадь под распределением правее числа $|Z|$, то есть $p_{\mathrm{val}}(i) = 2\mathbf{P}(x > |Z|)$; а эмпирическое значение $Z = Z(i)$ вычисляется по данной строке i.

Заметим $|Z(i)| > Z_0 \Leftrightarrow p_{\mathrm{val}}(i) < \alpha$. Число $p_{\mathrm{val}}(i)$ называется *текущим уровнем значимости* с уровнем значимости α; например, $\alpha = 0,05$.

Если $p_{\mathrm{val}}(i) < \alpha$, то строка-признак i включается в *предварительный список DE*-признаков данного кластера G клеток (и 0-я гипотеза отвергается). Иначе для i принимается 0-я гипотеза.

Дальнейшее сокращение *предварительного списка DE*-признаков связано с возможной случайностью самого значения $p_{\mathrm{val}}(i)$.

8.2. Дополнение МУВ процедурой Беньямини-Хохберга. Определение p_{adj}-значения. Переставим строки-features i, $1 \leq i \leq N$, в матрице Y по возрастанию самих чисел $p_{\mathrm{val}}(i)$: $p_{\mathrm{val}}(1) \leqslant \cdots \leqslant p_{\mathrm{val}}(n)$, где бывшая строка i получает какой-то свой номер k.

По определению:

$$p_{\mathrm{adj}}(k) = \min\left\{\frac{n}{k} \cdot p_{\mathrm{val}}(k), p_{\mathrm{adj}}(k+1)\right\}, p_{\mathrm{adj}}(n) = p_{\mathrm{val}}(k).$$

Тогда $p_{\mathrm{adj}}(k) \leqslant p_{\mathrm{adj}}(k+1)$ и $0 < p_{\mathrm{val}}(k) \leqslant p_{\mathrm{adj}}(k) \leqslant 1$.

Итак, *DE-признаком* ещё раз предварительно называется строка i (с новым номером k), которая удовлетворяет в качестве 1-го шага в пункте 8.1 условию $p_{\mathrm{adj}}(k) < \alpha$. Очень слабым вариантом является $\alpha = 0,1$.

Обратной индукцией легко проверить: для строки k, если выполняется $p_{\mathrm{val}}(k) > \alpha$ (то есть строка k не-DE-признак относительно p_{val}), то $p_{\mathrm{adj}}(k) > \alpha$ (эта же строка – не-DE-признак относительно p_{adj}).

На так отобранные DE-признаки накладываются ещё три дополнительных условия соответственно с параметрами q, q_1, q_2 (например, $q = 2$, $q_1 = q_2 = 0,25$ или очень слабый вариант параметров $q = 0,1$, $q_1 = 0,01$, $q_2 = 0$).

Первое условие на DE-признаки:

$$\log_2 \mathrm{FC}(i) = \left| \log_2 \left(\frac{1}{m_1} \cdot \sum_{j \in G} x_j \right) - \log_2 \left(\frac{1}{m_2} \cdot \sum_{l \notin G} x_l \right) \right| > q.$$

Второе и третье условия на DE-признаки: для признака i обозначим pct.1 и pct.2 доли клеток с ненулевой экспрессией в кластере G и в его дополнении $\neg G$, соответственно; и положим: $\max\{\mathrm{pct.1}, \mathrm{pct.2}\} \geqslant q_1$, $|\mathrm{pct.2} - \mathrm{pct.1}| \geqslant q_2$.

Окончательное определение DE-признака состоит в выполнении всех перечисленных условий.

Маркером называется DE-признак, для которого $p_{\mathrm{adj}}(i) < \beta$ с особо малым значением $\delta \ll \alpha$ (например, $\beta = 10^{-20}$ или меньше); значение β подбирается по данным.

§ 9. Пояснение к процедуре Беньямини-Хохберга.

До конца этого пункта предположим 0-ую гипотезу, обозначив её $(*)$.

Напомним, что распределение для Z лишь приближается к нормальному распределению; а $\mathbf{P}(x)$ здесь удобнее понимать как $\mathbf{P}(x) = \int\limits_x^\infty e^{-\frac{t^2}{2\pi}}\, dt$.

Тогда вероятность *отвержения* 0-ой гипотезы равна $\mathbf{P}(p_{\mathrm{val}}(i) < \alpha) = \mathbf{P}(x > \mathbf{P}^{-1}(\alpha/2))$, где $\mathbf{P}^{-1}(\alpha/2) = Z_0$ и $\mathbf{P}^{-1}$ – обратная функция к $\mathbf{P}$. Итак, если 0-ая гипотеза верна, то $\mathbf{P}(p_{\mathrm{val}}(i) < \alpha) = \alpha$. Матрица Y имеет N строк, поэтому $\mathbf{P}$("для всех строк 0-ая гипотеза принимается") $= (1-\alpha)^N$, а $\mathbf{P}$("для некоторых строк 0-ая гипотеза отвергается") $= 1 - (1-\alpha)^N$, что при большом N кажется парадоксальным в предположении $(*)$. Упомянутая в пункте 8.2 перенумерация строк (от i на k) не влияет на эту кажущуюся парадоксальность.

Для процедуры Беньямини-Хохберга доказано (здесь не приводится), что при замене условия $p_{\mathrm{val}}(i) < \alpha$ на условие $p_{\mathrm{adj}}(i) < \alpha$ получим: $\mathbf{P}(\{k | p_{\mathrm{adj}}(i) < \alpha\} = \varnothing) \geqslant 1 - \alpha$ что уменьшает соединение маловероятного события с предположением $(*)$. Заметим, что замена p_{val} на p_{adj}, с одной стороны, удаляет ложно дифференциально-экспрессируемые признаки ("ложно-положительные"), а с другой стороны, может привести к потере реально дифференциально-экспрессируемых признаков ("ложно-отрицательных").

§ 10. Понижение размерности данных.

Если кластеризованное множество точек (данные) в пространстве $\mathbb{R}^n$ большой размерности n отобразить в пространство $\mathbb{R}^K$, где $K < n$, например, на плоскость $(K = 2)$, обычной (линейной) проекцией, то проекции кластеров в $\mathbb{R}^n$, т.е. кластеры в $\mathbb{R}^K$, могут перекрывать друг друга или находиться в проекции ближе, чем по евклидову расстоянию в $\mathbb{R}^n$. Такой неадекватной ситуации стремится избежать нели-

нейная «проекция», называемая UMAP, рис. 5. Обычно данные в $\mathbb{R}^n$ и $\mathbb{R}^K$ представляются графами G и G_1, вершины которых однозначно соответствуют друг другу и точкам соответствующих пространств. Итак, задача понижения размерности состоит в переходе от данных в $\mathbb{R}^n$ к соответствующим данным в $\mathbb{R}^K$. А точнее, от графа G к графу G_1, у которых одинаковые клетки-вершины.

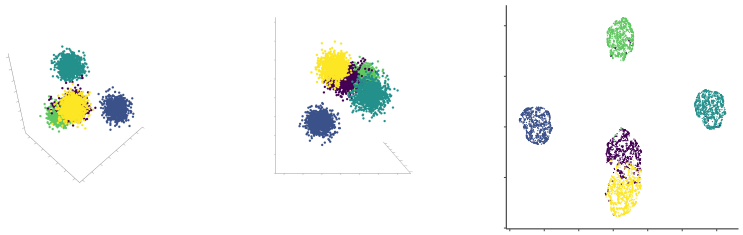

Рис. 5. Показан пример нелинейной проекции UMAP данных (здесь не приводятся) в большой размерности на плоскость ($K = 2$).

Повторим точнее, даны клетки-точки j в n-ом пространстве с их PCA-координатами (и одновременно j – вершины графа G). Как и в пункте 7, для каждой точки-клетки j строится список k-соседей и $\rho(j,l)$ – евклидово расстояние от j до её k-соседа l. Пусть ρ_j – наименьшее ненулевое расстояние от j до её ближайшего k-соседа. Для данного j вычислим нормировочное число σ_j, заданное уравнением

$$\sum_l \exp\left(-\frac{\rho(j,l)-\rho_j}{\sigma_j}\right) = \log_2 k,$$

где l пробегает всех k-соседей точки-клетки j.

Строится новый граф G_1, который имеет такие же вершины-клетки, рёбра и кластеризацию, какие у графа G; но вершины у G_1 однозначно соответствуют точкам K-мерного пространства. Тем самым, точки-вершины у G переходят в точки-вершины у G_1; это отображение и называется UMAP, [11].

Перейдём к построению *весов* рёбер и *координат* вершин у графа G_1. Но сначала изменим веса в графе G, и полученный граф обозначим G_{01}. В нём вершины-клетки j и l, соединяются неориентированным ребром, если они являются k-соседями друг друга, как и было в пункте 7. Но вес ребра $e = (j,l)$ в G_{01} положим равным $W(j,l) = w(j \to l) + w(l \to j) - w(j \to l) \cdot w(l \to j), 0 < W(j,l) < 1$, где вспомогательный односторонний вес равен

$$w(j \to l) = \exp\left(-\frac{\rho(j,l)-\rho_j}{\sigma_j}\right).$$

Вес $W(j,l)$ можно представить себе как вероятность существования хотя бы одного ребра между вершинами j, l.

Таким образом, получен новый в части весов неориентированный нагруженный граф G_{01} в n-мерном пространстве "большой размерности", где те же самые клетки j являются вершинами с теми же рёбрами e и новыми весами $W(e)$. Пусть для краткости $K = 2$.

Перейдём к построению *самого графа* G_1 на плоскости: он имеет те же вершины j и рёбра e, что у графов G и G_{01}, но опять новые веса. Обозначим $x(j)$ координаты на плоскости клетки-вершины $j \in G_1$. Определим в G_1 вес того же, что в G_{01}, ребра $e = (j,l)$, полагая $S_{jl} = \left(1 + a\|x(j) - x(l)\|_2^{2b}\right)^{-1}, 0 < S_{jl} < 1$, где a и b – параметры ("UMAP-веса"). Веса S_{jl} в G_1 близки к расстоянию, обратному к евклидову между $x(j)$ и $x(l)$.

Осталось вычислить координаты $x(j)$ всех вершин j так, чтобы G_{01} и G_1 были в некотором смысле близки друг к другу. Задача UMAP состоит в том, чтобы найти все координаты $x(j)$ из условия близости двух распределений на одних и тех же рёбрах в G_{01} и G_1. А именно, найти их из условия минимума следующей функции, называемой дивергенцией Кульбака-Лейблера: $L(x) = \sum_e L(x(j_1), \ldots, x(j_M)) = \sum_e \left[W(e) \log\left(\frac{W(e)}{S(e)}\right) + (1 - W(e)) \log\left(\frac{1-W(e)}{1-S(e)}\right)\right] \to \min$, минимизируя по координатам $x = x_1, x_2; \ldots; x_{M-1}, x_M$ всех вершин в G_1.

Результат минимизации определит искомый 2-мерный граф G_1 на плоскости. Повторим: на него переносится кластеризация с графа G в n-мерном пространстве.

Как обычно, численная минимизация существенно зависит от выбора начальной точки. Начальными координатами вершин графа G_1 можно выбрать две первые, наиболее информативные координаты точек в n-мерном пространстве.

Минимизируемый функционал перепишем:

$$L(x) = \sum_j \sum_{l \neq j} \left(W_{jl} \cdot \log \frac{W_{jl}}{S_{jl}} + (1 - W_{jl}) \cdot \log \frac{1 - W_{jl}}{1 - S_{jl}} \right) \to \min$$

Получим задачу минимизации:

$$\min_x L(x) = \min_x (- \sum_j \sum_{i \neq j} (W_{jl} \cdot \log S_{jl} + (1 - W_{jl}) \cdot \log(1 - S_{jl}))) = \min_{x,y} \sum_j \sum_{l \neq j} (W_{jl} \cdot A_{jl}(S) + (1 - W_{jl}) \cdot R_{jl}(S),$$

где

$$A_{jl} = -\log S_{jl}, R_{jl} = -\log(1 - S_{jl}).$$

Для минимизации по $x(j)$ можно использовать алгоритмы градиентного спуска или стохастического градиентного спуска, в которых итерации заканчиваются, если минимизируемая функция начинает мало меняться.

СПИСОК ЛИТЕРАТУРЫ

1. *Gorbunov K. Yu, Lyubetsky V.A.* An almost exact linear complexity algorithm of the shortest transformation of chain-cycle graphs //Eprint, Apr 29 2020. arXiv:2004.14351 [math.CO]. https://doi.org/10.48550/arXiv.2004.14351
2. *Gorbunov K. Yu, Lyubetsky V.A.* Linear time additively exact algorithm for transformation of chain-cycle graphs for arbitrary costs of deletions and insertions // Mathematics 2020, V.8. N.11. Art. 2001. P. 1–28. https://doi.org/10.3390/math8112001
3. *Gorbunov K. Yu, Lyubetsky V.A.* Algorithms for the reconstruction of genomic structures with proofs of their low polynomial complexity and high exactness // Mathematics. 2024. V. 12. N. 6, Art. 817. P. 1–26. https://doi.org/10.3390/math12060817
4. *Weiler P., Lange M., Klein M., Pe'er D. Theis F.* CellRank 2: unified fate mapping in multiview single-cell data //Nat. Methods. 2024. V. 21. Art. 1196–1205. P. 1–32 https://doi.org/10.1038/s41592-024-02303-9
5. Satija Lab. Seurat - Guided Clustering Tutorial. https://satijalab.org/seurat/articles/pbmc3k_tutorial.html, 2023 (дата обращения: 13 июня 2025).
6. *Blondel V. D., Guillaume J.-L., Lambiotte R., Lefebvre E.* Fast unfolding of communities in large networks //Journal of Statistical Mechanics: Theory and Experiment, 2008. V.10. P10008. P.1–12. https://doi.org/10.1088/1742-5468/2008/10/P10008.
7. *Боровков А. А.* Математическая статистика. Дополнительные главы. М.: Наука 1984. – 472 с.
8. *Гаек Я., Шидак З.* Теория ранговых критериев. М.: Наука 1971. – 376 с.
9. *Чибисов Д. М.* Лекции по асимптотической теории ранговых критериев. Вып. 14 М.: МИАН, 2009. – 186 с.
10. *Benjamini Y., Hochberg Y.* Controlling the false discovery rate: a practical and powerful approach to multiple testing //Journal of the Royal Statistical Society. Series B (Methodological), 1995. V.57. I.1. P.289–300. https://doi.org/10.2307/2346101
11. *Hardle W. K., Simar L., Fengler M. R.* Applied Multivariate Statistical Analysis. 6th Edition. – Springer, 2024. – 637 p.

Любецкий Василий Александрович
Горбунов Константин Юрьевич
Пирогов Сергей Анатольевич
Хазиев Георгий Андреевич
Агламазова Анастасия Ильинична
Институт проблем передачи информации
им. А.А. Харкевича Российской академии наук, Москва
lyubetsk@iitp.ru
gorbunov@iitp.ru
pirogov@iitp.ru
khaziev@iitp.ru
aglamazova.an@iitp.ru

Поступила в редакцию
09.09.2025
После доработки
26.09.2023
Принята к публикации
20.10.2023