Санкт-Петербургский государственный университет

Прикладная математика и информатика

Статистическое моделирование

Зиннатулина Белла Раифовна

Статистические методы кластеризации в больших объемах данных

Выпускная квалификационная работа

Научный руководитель:

д. ф.-м. н., профессор М. С. Ермаков

Рецензент:

д. ф.-м. н., профессор Г. Л. Шевляков

Санкт-Петербург

2017

Saint Petersburg State University
Applied Mathematics and Computer Science
Statistical Modelling

Zinnatulina Bella

Statistical clustering methods with application to big data

Graduation Project

Scientific Supervisor:
Doctor of Physics and Mathematics,
Professor M. S. Ermakov

Reviewer:
Doctor of Physics and Mathematics,
Professor G. L. Shevlyakov

Saint Petersburg
2017

Оглавление

Введение

Современный мир — это мир цифровых технологий, которые порождают очень большое количество данных. С ростом объёмов информации растут и потребности в возможностях её хранения и обработки. В качестве примеров источников больших данных можно привести социальные сети, устройства видеорегистрации, метеорологические данные, и это только малая часть. Сами источники говорят об актуальности обратки поступающих данных.

Таким образом, возникает ряд проблем. Во-первых, способность порождать данные оказалась сильнее, чем способность их хранить и своевременно обрабатывать. Во-вторых, кроме количества поступающей информации изменился и её характер. Основной объем данных — неструктурированная, непрерывно поступающая информация.

В данной работе особый интерес представляет кластерный анализ. Как было сказано выше, большие данные имеют множество характеристик, которые ограничивают круг методов кластеризации, которые могут с ними работать. В том числе, так как большие данные являются порождением современных технологий, методы их анализа так же находятся на стадии активного развития.

Таким образом, целью работы является изучение статистических методов кластеризации больших объемов данных с их практическим применением.

Для решения поставленной цели в первичные задачи работы входило изучение существующей литературы, освещающей проблемы, связанные с анализом больших объемов данных, в том числе кластеризацией, и способы их решения. Исследуемые подходы описаны в Главе 1.

Также задачей являлось изучение наиболее интересных и эффективных методов кластеризации больших данных, отобранных благодаря изучению соответствующей литературы. В работе дан обзор иерархической кластеризации в Главе 4. Более детально изучены следующие методы: алгоритм кластеризации Маркова, описанный в Главе 5 и стохастическая блочная модель, которой посвящены Главы 2 и 3.

Стохастическая блочная модель представляет наибольший интерес в данной работе. В Главе 2 сформулированы теоретические основы метода, собранные и структурированные из большого количества источников, таких как [1], [2], [3], [4], [5].

Практическое применение алгоритма при различных параметрах модели с соответ-

ствующими результатами представлены в Главе 3. Реализация алгоритмов метода SBM отражена в Приложениях А, Б, В.

Глава 1

Методики анализа больших данных

1.1. Основные характеристики больших данных

Актуальности исследования больших данных посвящено множество статей, но общего формального определения тому, что же такое большие данные, дано не было. Например, в статье [6] понятие «большие данные» относят к операциям, которые можно выполнять исключительно в большом масштабе.

В таком случае, необходимо определить, какими параметрами должны обладать данные, чтобы их можно было отнести к категории больших данных. Согласно [7], большие данные имеют следующие определяющие характеристики: объем, скорость, многообразие и ценность.

- **Объем**. Стремительно растущее количество данных, поступающих отовсюду, предъявляет новые требования в отношении хранения и их обработки.
- **Скорость**. Важна как скорость поступления данных, так и скорость обработки и полученния результатов.
- **Многообразие**. Данные содержат неструктурированную информацию. Необходима возможность одновременной обработки различных типов структур данных.
- **Ценность**. Анализ больших данных способен повлиять на увеличение эффективности работы и конкурентоспособности компании, создание новых продуктов дает понимание происходящих процессов.

Таким образом, можно сформулировать следующее определение.

Большие данные — это технологии и архитектуры нового поколения для экономичного извлечения ценности из разноформатных данных большого объема, путем их быстрого захвата, обработки и анализа.

Следует отметить, что большие данные зачастую плохо структурированы, зашумлены, имеют большое количество пропусков, таким образом нуждаются в предварительной обработке.

1.2. Предварительная обработка данных

В связи с тем, что большие данные поступают непрекращающимся потоком, имеет смысл сохранять все данные, не беспокоясь о том, какая часть будет актуальна для последующего анализа и принятия решения. Недостатком является то, что для извлечения полезной информации требуется последующая обработка этих огромных массивов данных.

Можно выделить следующие способы предварительной обработки данных:

- **Предварительная нормировка и масштабирование данных.** Полезно при дальнейшем использовании нейронных сетей, в противном случае ошибки, обусловленные переменными, изменяющимися в широком диапазоне, будут сильнее влиять на обучение сети.

- **Редукция размерности.** Используется для сокращение числа переменных и определение структуры взаимосвязей между ними.

- **Кластеризация.** Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятие решений, применяя к каждому кластеру свой метод анализа.

Наибольший интерес в данной работе представляет именно кластеризация объектов. Стоит заметить, что кластеризация является в том числе элементом именно анализа данных, а не только этапом предварительной обработки.

1.3. Кластерный анализ

Определение 1. Кластерный анализ (англ. cluster analysis) — общий термин для целого ряда методов, используемых для группировки объектов в сравнительно однородные группы на основе сходства их характерных признаков.

Согласно [8], в качестве основных целей кластерного анализа выделяют

- **Понимание данных** путём выявления кластерной структуры. При получении группы кластеров становится возможным построение своей модели для анализа данных каждого кластера. Число кластеров резонно делать небольшим.

- **Редукция размерности данных**. Сокращение исходной выборки путём выделения наиболее типичного представителя каждого кластера. В данной ситуации важнее обеспечить высокую степень сходства объектов внутри каждого кластера.
- **Обнаружение новизны**. Выделяются нетипичные объекты, которые не удаётся присоединить ни к одному из кластеров.

Глава 2

Теоретические основы метода SBM

2.1. Стохастическая блочная модель

Стохастическая блочная модель (Stokhastic Block Model или SBM) — метод кластеризации, в основе которого лежит моделирование случайных графов. Особенностью метода является его вариативность в плане требований к восстановлению кластерной структуры, а именно слабое, сильное и точное восстановление. Также преимуществом алгоритма является возможность проверки условий разделимости графа. Не стоит забывать, что речь идет об анализе больших данных, поэтому немаловажную роль играет трудоёмкость алгоритма. Точность и временная сложность алгоритма SBM представлены в Теореме 2.1.4.

Основная задача кластеризации состоит в разделении вершин размеченного графа на кластеры, внутри которых находятся наиболее плотно связанные между собой вершины. То есть восстановление кластерной структуры опирается на следующую идею: ребра, связывающие вершины графа, больше распространены внутри кластеров, чем между ними, так как наличие ребра между вершинами является признаком «связи» между ними. Также речь может идти о гиперграфе, так как зачастую встречаются задачи с перекрывающимися кластерами.

2.1.1. Описание метода

Введем обозначения согласно [1].

Пусть V — множество вершин графа размера $n \in \mathbb{N}$ (по количеству объектов выборки), $k \in \mathbb{N}$ — количество кластеров, на которое предполагается разделить множество вершин V.

Введем вектор вероятностей $p = (p_1, \ldots, p_k)$, который будет описывать относительные размеры кластеров. То есть, если p_i имеет большое значение относительно остальных элементов p, то, скорее всего, соответствующий кластер велик относительно остальных.

Введем X — случайный n-мерный вектор «лейблов» (скрытых переменных, обозначающих принадлежность элемента кластеру), компоненты которого н.о.р. и являют-

ся элементами из $[k] := \{1, \cdots, k\}$ с вероятностью $p = (p_1, \ldots, p_k)$. То есть $X_i \in X$, $i \in 1, \cdots, n$ — лейбл вершины $V_i \in V$.

Пусть W — симметричная матрица размера $k \times k$, элементы которой лежат в $[0, 1]$ и обозначают вероятности связи вершин графа. Тогда пара вершин $(V_i, V_j) \in V \times V$, $(i, j \in 1, \cdots, n)$ связаны ребром с вероятностью $W_{X_i X_j} \in W$.

Наглядно проиллюстрируем на примере $k = 3$. Пусть у нас есть 3 кластера, тогда известно n — количество вершин, $p = \{p_1, p_2, p_3\}$. Тогда размеры кластеров будут соответственно np_1, np_2, np_3. Вероятность наличия ребра внутри $i-$го кластера равна W_{ii}, между кластерами i и j — W_{ij}, $i, j \in [k]$, что продемонстрировано на Рис. 2.1.

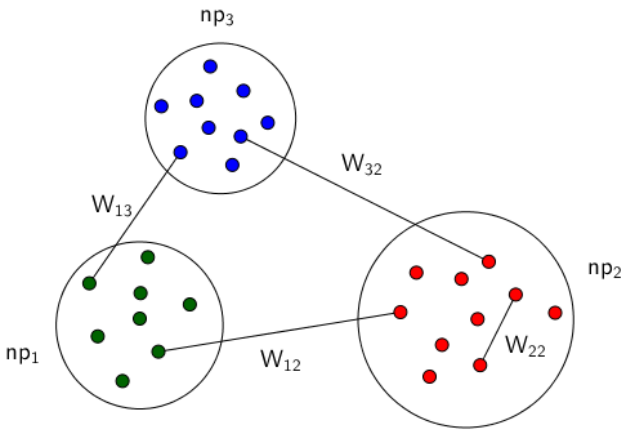

Рис. 2.1. Пример при $k = 3$.

Пара (X, G) строится с помощью SBM(n, p, W), где G — неориентированный граф из n вершин, где вершины i и j соединены ребром с вероятностью $W_{X_i X_j}$ и независимо от других пар вершин.

Целью обнаружения кластера является восстановление разметки X с некоторым уровнем точности путем наблюдения G. То есть мы получим k кластеров размера $np_1, np_2, \ldots, np_k$.

Согласие между двумя векторами $X, \hat{X}$ с элементами из $[k]$ получается путем минимизации расстояния Хэмминга между X и любым перемаркированным $\hat{X}$, где $\hat{X}$ — любое преобразование элементов вектора X фиксированной перестановки $[k]$.

Алгоритм обнаружения сообществ с точностью $\alpha \in [0, 1]$ принимает на вход G, полученный с помощью SBM(n, p, W), на выходе — преобразование $\hat{X}$ из X с уровнем согласия α с вероятностью $1 - o_n(1)$.

- *Точное* восстановление разрешимо в $\mathrm{SBM}(n, p, W)$, если существует алгоритм с точностью $\alpha = 1$.
- *Сильное* восстановление разрешимо в $\mathrm{SBM}(n, p, W)$, если существует алгоритм с точностью $\alpha = 1 - o_n(1)$.
- *Слабое* восстановление разрешимо в $\mathrm{SBM}(n, u, \hat{W})$, (где u — равномерно распределение на $[k]$, а $\hat{W}$ — матрица, с константой вне диагонали) если существует алгоритм с точностью $\alpha = \frac{1}{k} + \varepsilon, \varepsilon > 0$.

Другими словами, точное восстановление требует идеальной реконструкции кластеров, сильное — почти идеальной, а слабое требует улучшить случайный равновероятный выбор. Далее возникает вопрос, как выбрать оптимальную точность $\alpha \in [0, 1]$ относительно параметров p и W.

Прежде чем ответить на поставленный вопрос, заметим, что в условиях разрешимости слабого восстановления скрыта простейшая модель кластеризации на равновероятные (а значит и равные по размеру) кластеры. Согласно [1] назовем эту модель Symmetric Stochastic Block Model (SSBM).

2.1.2. Выбор α

Точное восстановление

Положим, что p, которая определяет относительные размеры классов, не зависит от n. В свою очередь от n зависит матрица W, которая опеределяет вероятности связи n вершин графа:

$$W = S_n \frac{Q}{n},$$

где $Q \in \mathrm{R}_+^{k \times k}$ — симметричная матрица, не зависящая от n, S_n — параметр интенсивности, определяющийся количеством вершин графа.

Определим величину SNR следующим образом:

$$\mathrm{SNR} = |\lambda_{min}|^2 / |\lambda_{max}|$$

,

где $|\lambda_{min}|$ и $|\lambda_{max}|$ наименьшее и наибольшее собственное число матрицы $\mathrm{diag}(p)Q$ соответственно.

Теорема 2.1.1 ([1]). *Точное восстановление разрешимо в* $SBM(n, p, \log(n)Q/n) \iff$
$J(p, Q) := \min_{1 \le i < j \le k} D_+((\mathrm{diag}(p)Q)_i || (\mathrm{diag}(p)Q)_j) \ge 1$, *где* D_+ *определено как*
$D_+(\mu||\nu) = \max_{t \in [0,1]} \sum_x \nu(x) f_t(\mu(x)/\nu(x)), \quad f_t(y) = 1 - t + ty - y^t.$

Более того, пороговое значение эффективно достижимо.

Таким образом, согласно Теореме 2.1.1, если столбцы $\mathrm{diag}(p)Q$ «достаточно различны», где разница определяется с помощью метрики D_+, то имеет смысл разделить соответствующие кластеры.

Замечание 1 ([1]). Обозначим через $\mathrm{SSBM}(n, k, \frac{a}{n}, \frac{b}{n})$ симметричный разреженный $\mathrm{SBM}(n, p, W)$, где p — равномерное распределение на $[k]$ и

$$W_{ij} = \begin{cases} a/n, & \text{если } i = j, \\ b/n, & \text{иначе.} \end{cases}$$

Определим SNR в случае k симметричных кластеров:

$$\mathrm{SNR} = \frac{(a-b)^2}{k(a + (k-1)b)}$$

, тогда

- независимо от k, если $\mathrm{SNR} > 1$ (порог Kesten-Stigum или KS), то задача распознавания кластеров разрешима за полиномиальное время;
- если $k \ge 5$, то задача разрешима для некоторого $\mathrm{SNR} < 1$.

Первые алгоритмы, которым удалось достичь порога KS при $k = 2$ основаны на подсчете количества случайных блужданий без самопересечения и взвешенного невозврата блуждания между вершинами.

Слабое восстановление

Теорема 2.1.2 ([1]). *Для* $k = 2$*:*

- *Слабое восстановление эффективно разрешимо, если* $SNR > 1$ *(т.е. порог KS эфективно достижим при* $k = 2$*).*
- *Слабое восстановление не разрешимо при* $SNR \le 1$.

Теорема 2.1.3 ([1]). *Замечание 2.1.2 справедливо для всех* $k \ge 2$. *В частности*

- *Задача слабого восстановления разрешима за* $O(n \log n)$*, если* $SNR > 1$.
- *Слабое восстановление информации теоретически разрешимо при* $SNR < 1$*, если* $k \geq 5$*, но алгоритм неэффективен.*

Теорема 2.1.4 ([4])**.** *Для* $\forall k \in \mathbb{Z}$*,* $p \in (0,1)^{\alpha}$*, где* $|p| = 1$*,* $Q \in \mathbb{R}^{k \times k}$ *— симметричная матрица, строки которой различны,* $\exists \varepsilon(c) = O(1/ln(c))$*, для* $\forall c >> 1$*, такая что алгоритм* $SBM(n, p, cQ/n)$ *выявляет кластерную структуру с точностью* $1 - e^{-\Omega(c)}$ *и временной сложностью* $O_n(n^{1+\varepsilon(c)})$.

Таким образом, алгоритм способен восстановить кластерную структуру за $O_n(n^{1+\varepsilon(c)})$, при учете выполнения условий разделимости.

Симметричная стохастическая блочная модель

В случае равновероятной кластеризации, где $p = \frac{1}{k}, \cdots, \frac{1}{k}$ очевидно, что ни у какого кластера нет предпочтения, а значит их размеры одинаковы и равны $\frac{n}{k}$, где n — число кластеризуемых объектов, k — количество кластеров.

В данной модели нет разницы между группами, поэтому вероятности связи между всеми кластерами равны, так же как и вероятности связи внутри кластеров. Тогда модель имеет следующие параметры: SSBM(n,p,a,b).

Матрица W будет иметь следующий вид:

$$W = \begin{pmatrix} a & b & \dots & b \\ b & \ddots & b & \vdots \\ \vdots & b & \ddots & b \\ b & \dots & b & a \end{pmatrix}$$

Определение 2. Для $\forall$ вершины v графа G пусть $N_r(v)$ — множество всех вершин, с кратчайшим путём из G в v длины r. Также определим $\hat{N}_r(v)$ — вектор,$i-$м элементом которого является количество вершин в $N_r(v)$, принадлежащих кластеру i, $i \in 1 \dots k$.

Определение 3. Для $\forall$ вершин v и v' графа G, r, $r' \in \mathbb{R}$ и E — множества ребер графа G определим $N_{r,r'[E]}(v \cdot v')$ как количество пар (v, v'), таких что $v_1 \in N_{r[G \setminus E]}(v)$, $v_2 \in N_{r'[G \setminus E]}(v')$, и $(v_1, v_2) \in E$.

Определение 4. Введем $I_{r,r'[E]}(v \cdot v')$ следующим образом:

$$I_{r,r'[E]}(v \cdot v') = N_{r+2,r'[E]}(v \cdot v') \cdot N_{r,r'[E]}(v \cdot v') - N^2_{r+1,r'[E]}(v \cdot v').$$

2.1.3. Теория обнаружения кластеров

Определение 5. Пусть $V = [n]$ — вершины, $E(G)$ — ребра, $G(V, E(v))$ — гиперграф с $N = |E(G)|$. Пусть $\mathcal{X}$ и $\mathcal{Y}$ — конечные множества, входный и выходной алфавит соответственно, $Q(\cdot|\cdot)$ — канал из $\mathcal{X}^k$ в $\mathcal{Y}$, называемый ядром. Для каждой вершины из V назначается вершинная переменная из $\mathcal{X}$, а для каждого ребра из $E(G)$ определяется реберная переменная из $\mathcal{Y}$. Пусть y_I — реберная переменная, прикрепленная к ребру I, а $x[I]$ — k узловых переменных, прикрепленных к I. Определим графический канал между графом G и ядром Q: $P(\cdot|\cdot) \equiv \prod_{I \in E(G)} Q(y_I|x[I]), x \in \mathcal{X}^V, y \in \mathcal{Y}^{E(G)}$. Схема работы канала продемонстрирована на Рис. 2.2.

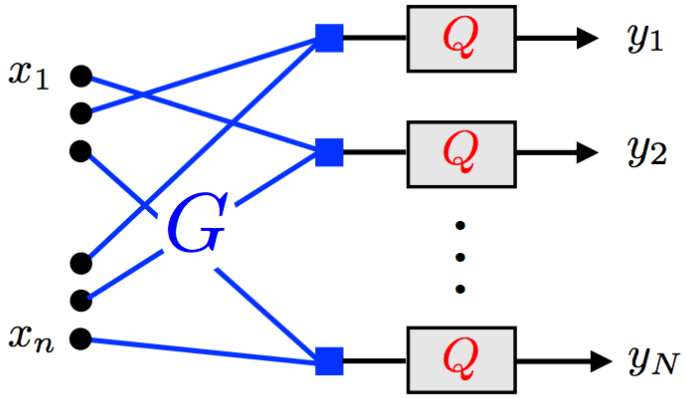

Рис. 2.2. Схема работы канала $P(\cdot|\cdot)$.

Поскольку теория обнаружения сообществ в SBM активно развивается и только в последние годы появились основополагающие ограничения, существует множество открытых проблем. Ниже представлены некоторые из них:

- Теорема 2.1.1 дает исчерпывающий результат для сообществ линейного размера, то есть когда элементы p и k не кратно n.
- Частичное восстановление, находящееся между слабым и точным, практически не изучено. Остаются открытыми вопросы точности восстановления в случае с несколькими асимметричными сообществами.
- Открыт вопрос нахождения точного порога слабого восстановления при $k \geq 3$.

2.1.4. Алгоритм обнаружения кластеров

Стоит отметить, что в изученных и структурированных статьях по методу SBM формальное описание работы алгоритма носило чисто теоретический характер. Далее представлено его описание с последующим применением на практике.

С учетом того, что необходимо было провести анализ программного кода и восстановить структуру алгоритма, данные, к которым он применялся, были смоделированы самостоятельно. Процесс моделирования представлен в Приложении А.

Моделирование происходило таким образом, что изначально были известны метки вершин, для последующей проверки точности работы алгоритма в восстановлении кластерной структуры. Также в моделировании учитывался характер матрицы W и вектора p. Была реализована проверка условий разделимости.

Ниже опишем принципы работы алгоритма.

На первом шаге необходимо получить оценки p и Q. Напомним, что вектор p содержит вероятности принадлежности объектов кластерам и, соответственно, определяет относительные размеры кластеров. Матрица W содержит вероятности связи вершин графа. Согласно Теореме 2.1.1 определим W следующим образом:

$$W = \log(n)Q/n,$$

где $Q \in \mathrm{R}_{+}^{k\times k}$ — симметричная матрица.

Далее стоит отметить, что на реальных данных нам необходимо получить оценки p и Q.

Положим, что при условии разметки вершин графа с погрешностью x, можно вычислить приближения p и Q с точностью до $O(x+\log(n)/\sqrt{n})$ с вероятностью $1-o(1)$.

На следующем шаге необходимо скорректировать приведенные на первом шаге оценки параметров для контроля вероятности ошибки.

Пусть p' и Q' — оценки параметров p и Q с погрешностью не более чем δ. На основании оценок параметров p' и Q' производим разметку вершин графа относительно предполагаемых кластеров так, что ошибка классификации соседних вершин, погрешность разметки которых составляет не больше чем δ, имеет частоту ошибки классификации в $\mathrm{e}^{O(\delta \log n)}$ раз выше, чем если бы p' и Q' были бы истинными оценками параметров и производили корректную классификацию.

Далее, напомним, что по Теореме 2.1.1 точное восстановление разрешимо в

$\mathrm{SBM}(n, p, \log(n)Q/n) \Longleftrightarrow$

$J(p,Q) := \min\limits_{1\le i<j\le k} D_+((\mathrm{diag}(p)Q)_i||(\mathrm{diag}(p)Q)_j) \ge 1$, где D_+ определено как

$D_+(\mu||\nu) = \max\limits_{t\in[0,1]} \sum\limits_x \nu(x) f_t(\mu(x)/\nu(x)), \quad f_t(y) = 1 - t + ty - y^t.$

На следующем шаге необходимо показать, что $D_+((PQ)_i, (PQ)_j)$ разделимы для $\forall i, j$, где $(PQ)_i-$ $i-$й столбец. То есть соответствующие вершины i и j находятся на достаточном для разделения расстоянии относительно метрики D_+.

Для доказательства эффективной достижимости границы $J(p,Q) > 1$ используется алгоритм, основанный на двухэтапной процедуре.

- На первом шаге исходный граф разбивается на два подграфа. Полученные подграфы не являются абсолютно независимыми кластерами, но существенно разделимы из-за разреженности исходного графа. Далее, на первом подграфе запускается алгоритм, получающий сильное восстановление.

- На втором шаге предварительная кластеризация усиливается с помощью использования второго подграфа, который «очищает» сильную кластеризацию первого. Если $D_+((\mathrm{diag}(p)Q)_i||(\mathrm{diag}(p)Q)_j) > 1 \ \forall i < j$, тогда вершины,которые были неправильно классифицированы на первом этапе, могут быть с большой вероятностью корректно переклассифицированы.

  Если алгоритм эффективен для на первом шаге, то весь алгоритм эффективен, т.к. второй шаг имеет общую сложность $O(n^{1+\varepsilon})$, то есть линеен по n.

Согласно обозначениям Определений 2, 3 и 4 определим $r = \frac{3}{4}\log n/\log d$. Пусть $k_{max} = 1/\delta$, где $\delta = \log(n)\log\log n$.

Алгоритм

Таким образом, работу алгоритма можно разделить на следующие этапы:

1. На вход алгоритму подается пара (G, γ), где G — исходный граф, $\alpha \in [0,1]$.

2. Определим подграф G' на множестве вершин $1 \dots n$, где каждое ребро из графа G выбирается независимо с вероятностью γ.

3. Подсчитаем $I_{r,r'[E]}(v_i \cdot v_j)$ для всех i и j графа G'.

4. Существует такое разбиение вершин i и j, что $I_{r,r'[E]}(v_i \cdot v_j) > 0 \Longleftrightarrow$ когда вершины i и j лежат в одном кластере. Далее неоходимо выбрать по одному из известных представителей кластеров $v[1], v[2], \ldots v[k]$.

5. Для каждой вершины v' определим тот кластер i, $i \in 1 \ldots k$, представитель которого v_i максимизирует величину $I_{r,r'[E]}(v[i] \cdot v')$, получим σ'.

6. На основе полученных на предыдущих шагах результатах оценим относительные размеры предполагаемых кластеров.

7. Для каждой вершины v определим наиболее вероятное сообщество, обозначим его через σ''_v, исходя из полученной σ'.

8. Исходя из полученных результатов, получим новые оценки p и Q.

9. Далее, для каждой вершины v определим наиболее вероятное сообщество, исходя из σ''.

10. Алгоритм выдает метку каждой вершины v.

Глава 3

Практическое применение метода SBM

В работе описаны различные методики анализа кластерной структуры данных. Наибольших интерес представили алгоритмы MCL и SBM. В частности, наибольшее внимание было уделено SBM по причине простоты реализации, которая находится в Приложении А. Также интерес представило обилие условий применимости алгоритма SBM, которые определяют степень восстановления кластерной структуры данных.

С учетом того, что метод SBM только развивается и статьи выходят в настоящее время, мною была проделана большая работа по восстановлению структуры метода, расшифровки обозначений и разбора алгоритмов.

Также была проведена реализация алгоритма точной кластеризации, представленная в Приложении А.

В Приложении Б реализован алгоритм, согласно Теореме 2.1.1. Для начала была смоделирована матрица инцидентности исходя из параметров p, W, n, k. По матрице построен и визуализирован граф при помощи библиотеки "iclust"в пакете R. На рис. 3.1 представлен граф, состоящий из 1000 вершин.

3.0.1. Результаты при $n = 1000$, $k = 5$

Матрица инцидентности была смоделирована исходя из следующих параметров: $n = 1000$, $k = 5$, $p = (0.2, 0.2, 0.2, 0.2, 0.2)$, матрица W имеет следующую структуру:

$$W = \begin{pmatrix} \frac{1}{50} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} \\ \frac{1}{1000} & \frac{1}{50} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} \\ \frac{1}{1000} & \frac{1}{1000} & \frac{1}{50} & \frac{1}{1000} & \frac{1}{1000} \\ \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{50} & \frac{1}{1000} \\ \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{1000} & \frac{1}{50} \end{pmatrix}$$

.

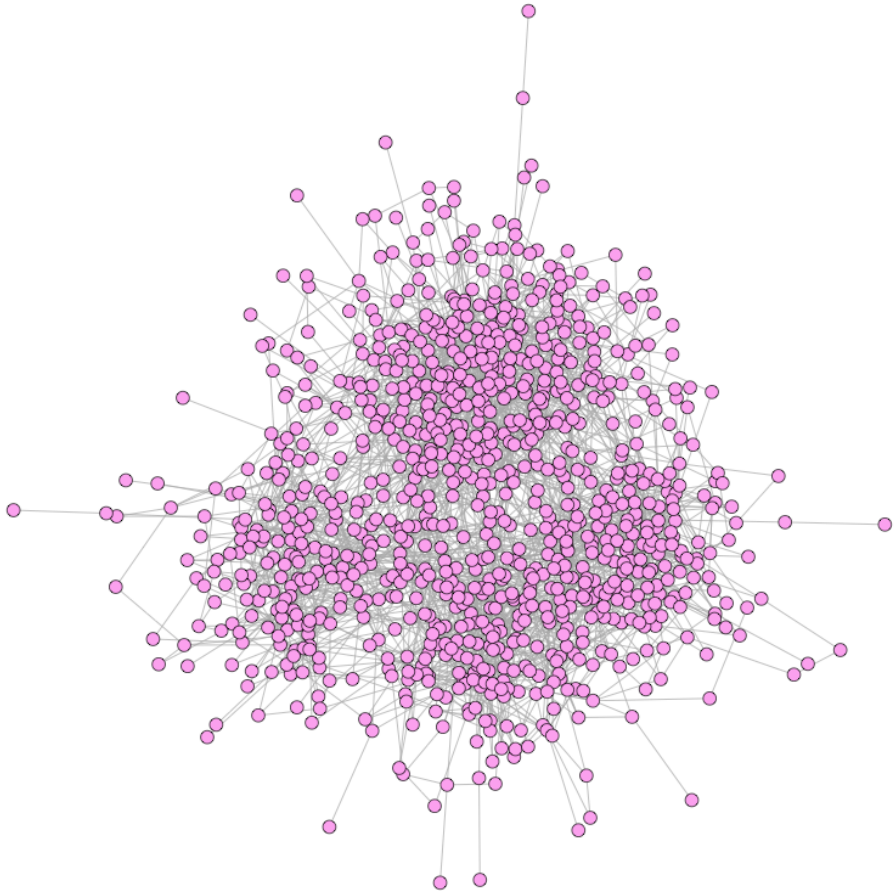

Рис. 3.1. Кластеризуемый граф, $n = 1000$.

Далее, к данному графу был применен алгоритм SSBM. В итоге, благодаря полученной разметке вершин графа на соответствующие кластеры, становится возможным визуализировать его кластерную структуру, представленную на рис. 3.2.

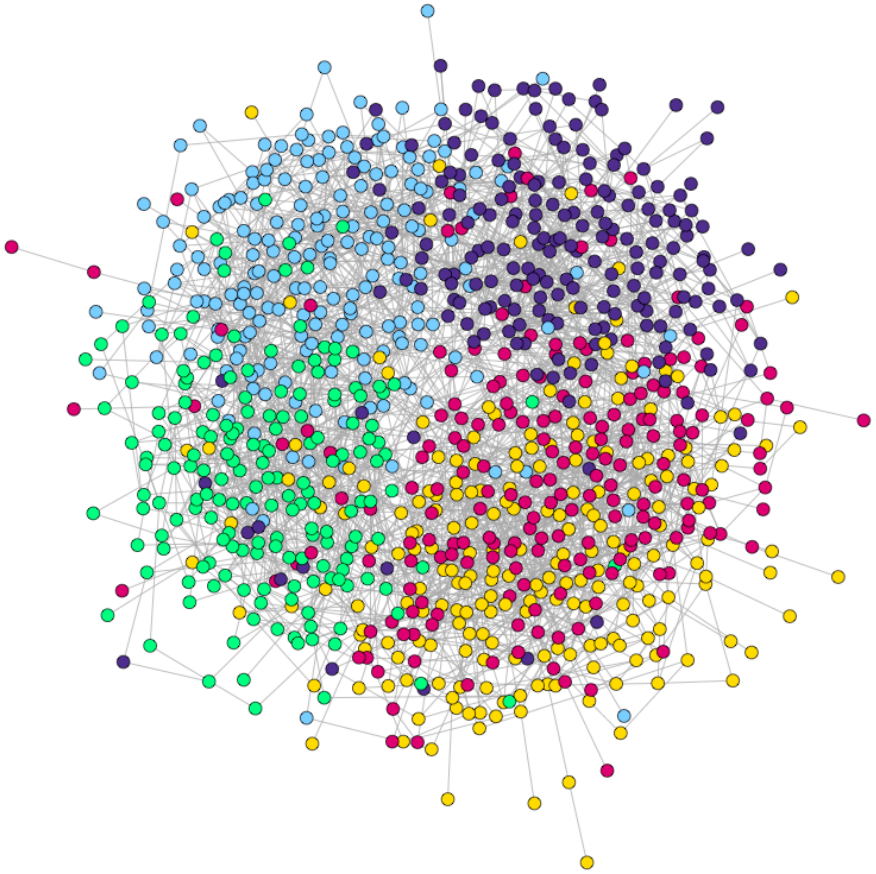

Рис. 3.2. Размеченный на k = 5 кластеров граф, $n = 1000$.

3.0.2. Результаты SSBM при n=12, k=2

Для более наглядной демонстрации работы алгоритма, структуры его параметров и результатов, ниже будет представлен пример разбиения графа, состоящего из $n = 12$ вершин на $k = 2$ кластера.

Также параметрами алгоритма является вектор $p = (0.5, 0.5)$, матрица W — симметричная и имеет следующую структуру:

$$W = \begin{pmatrix} 0.66 & 0.16 \\ 0.16 & 0.66 \end{pmatrix}$$

.

Проверяя условие разделимости с учетом того, что у нас симметричная модель, согласно Замечанию 1, получаем:

$$W_{ij} = \begin{cases} a/n, & \text{если } i = j, \\ b/n, & \text{иначе.} \end{cases}$$

, следовательно $a = 0.66 * n = 7.92$, $b = 0.16 * n = 1.92$. Тогда SNR $= \frac{(a-b)^2}{k(a+(k-1)b)} = 1.82 > 1$, значит задача разделения разрешима.

При малом размере выборки становится возможным продемонстрировать M — матрицу инцидентности графа.

$$M = \left(\begin{array}{cccccc|cccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}\right)$$

.

Стоит заметить, что матрица смоделирована таким образом, что она имеет блочную структуру, где каждый блок представляет отдельный кластер.

Согласно моделированию, описанному в Приложении Б, вершины имеют следующий вектор меток: $X = (2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1)$.

Визуализация графа по матрице инцидентности M представлена на рис. 3.3.

Далее, к матрице M применялся алгоритм SSBM, реализованный в Приложении В.

Результатом работы программы являются вероятности принадлежности точек кластерам:

	[,1]	[,2]
[1,]	0.008333333	0.991666667
[2,]	0.008333333	0.991666667

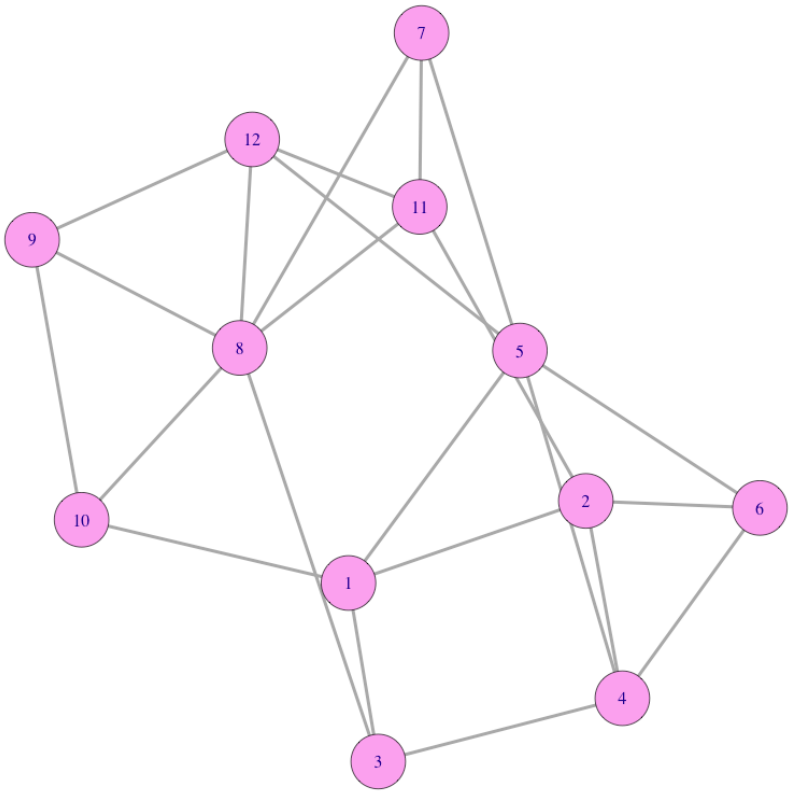

Рис. 3.3. Кластеризуемый граф, $n = 12$.

[3,] 0.008333333 0.991666667

[4,] 0.008333333 0.991666667

[5,] 0.008333333 0.991666667

[6,] 0.008333333 0.991666667

[7,] 0.991666667 0.008333333

[8,] 0.991666667 0.008333333

[9,] 0.991666667 0.008333333

[10,] 0.991666667 0.008333333

[11,] 0.991666667 0.008333333

[12,] 0.991666667 0.008333333

Таким образом разметка $\hat{X}$ вершин графа на $k = 2$ кластера выглядит следующим образом: $\hat{X} = (2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1)$, что совпадает с разметкой X, заданной при моделировании.

Визуализация кластерной структуры, согласно полученной разметке, представлена

на рис. 3.4.

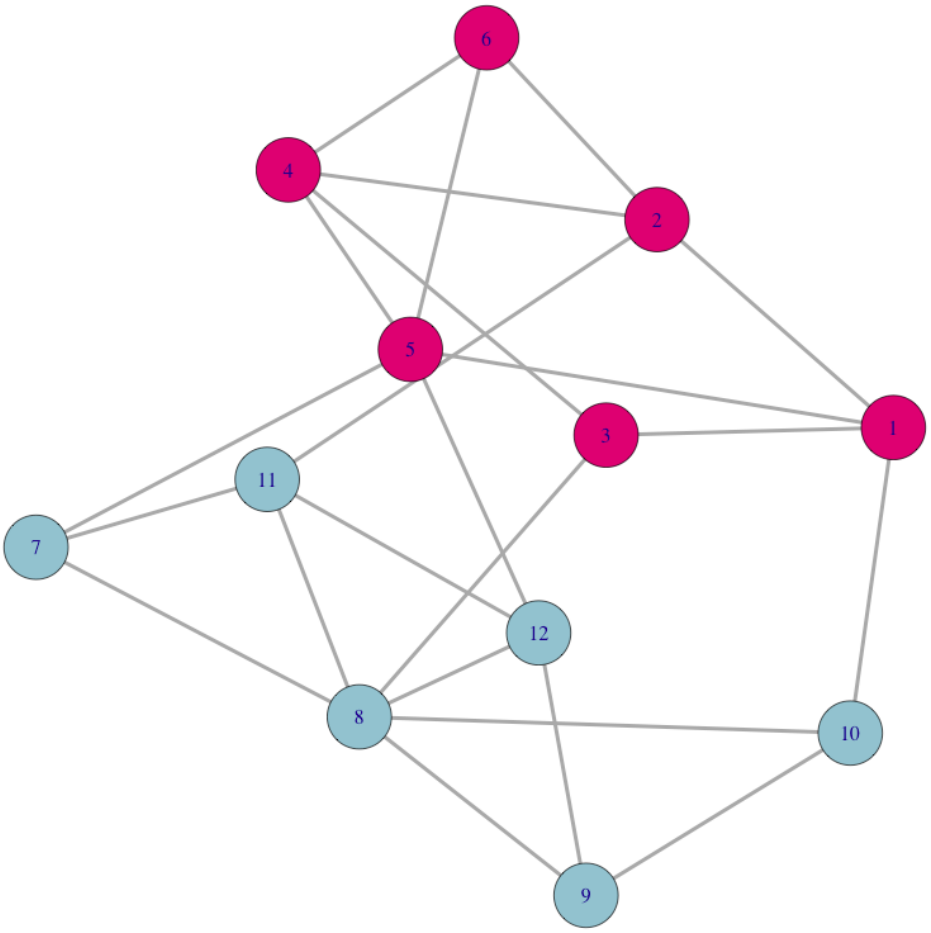

Рис. 3.4. Кластеризуемый граф, $n = 12$, $k = 2$.

3.0.3. Пример неразделимого кластера

Далее, для демонстрации значимости условий разделимости, будет приведен пример параметров, при которых, задача разделения на кластеры не разрешима.

Пусть так же $n = 12$, $k = 2$, $p = (0.5, 0.5)$, матрица W — симметричная и имеет следующую структуру:

$$W = \begin{pmatrix} 0.67 & 0.46 \\ 0.46 & 0.67 \end{pmatrix}$$

.

Проверяя условие разделимости, получим:

$$W_{ij} = \begin{cases} a/n, & \text{если } i = j, \\ b/n, & \text{иначе.} \end{cases}$$

, следовательно $a = 0.67 * n = 8.04$, $b = 0.46 * n = 5.52$. Тогда SNR $= \frac{(a-b)^2}{k(a+(k-1)b)} = 0.23 < 1$, значит задача разделения не разрешима.

Согласно моделированию разметка графа имеет следующий вид:

$$X = (2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1)$$

. Но алгоритм, при указанных выше параметрах, дает следующую разметку:

$$\hat{X} = (2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2)$$

.

Таким образом 7 из 12 точек, что составляет 58%, были не верно кластеризованы.

Глава 4

Иерархическая кластеризация

Иерархические алгоритмы кластеризации интересны тем, что они строят не одно разбиение выборки на кластеры, а систему вложенных разбиений. Результат обычно представляется в виде дерева кластеризации — дендрограммы. Алгоритмы иерархической кластеризации разделяются на *дивизимные* или нисходящие и *агломеративные* или восходящие.

Нисходящие алгоритмы разбивают выборку на всё более и более мелкие кластеры, а восходящие, которые более распространены, объединяют объекты во всё более крупные кластеры.

4.0.1. Постановка задачи

Разнообразие методов иерархической кластеризации связано с выбором меры подсчета расстояния между точками пространства и правилами определения расстояния между кластерами.

На первом шаге каждый объект представляет собой отдельный кластер, расстояния между ними определяются выбранной метрикой — мерой расстояния между объектами в пространстве переменных.

Согласно обозначениям [9], для одноэлементных кластеров определяется функция расстояния

$$R(\{x\},\{x'\}) = \rho(x,x').$$

От характера представленных данных зависят меры, используемые для измерения расстояний.

1. *Метрические шкалы*

   - *Евклидова метрика*
     $\rho(x,x') = \sqrt{\sum_{i=1}^{n}(x_i,x_i')^2}.$
   - *Квадрат евклидовой метрики*
     $\rho(x,x') = \sum_{i=1}^{n}(x_i,x_i')^2.$

Придается более серьезное значение большим расстояниям.

- *Манхэттенская метрика*
  $\rho(x, x') = \sum_{i=1}^{n} |x_i - x'_i|$.
  Следует отметить, что так как здесь не предусматривается возведение расстояний в квадрат, влияние отдельных больших разностей (выбросов) уменьшается.

- *Расстояние Минковского*
  $\rho(x, x') = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$.
  Изменяя параметр p, можно в различной степени придавать значение удаленным точкам по сравнению с относительно близкими.

- *Настроенная мера*
  $\rho(x, x') = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{r}}$.
  Является обобщением меры Минковского, здесь варьируются 2 параметра, p отвечает за взвешивания разностей по отдельным координатам, а r — между объектами.

2. *Номинальные шкалы*

Для использования номинальных шкал необходимо представление данных в виде таблицы сопряженности, где строки соответствуют категориям одной номинальной переменной, а столбцы таблицы — другой, на пересечении стоят частоты — число объектов, у которых имеется соответсвующее сочетание категорий.

Пусть f_{ij} — частота совместного появления $i-$го признака по строке и $j-$го по столбцу, f^o_{ij} — наблюдаемая частота (observed), f^e_{ij} — ожидаемая (expected).

- *Мера* χ^2
  $\rho(x, x') = \sqrt{\sum_{i,j} \frac{(f^o_{ij} - f^e_{ij})^2}{f^e_{ij}}}$.
  Назовем $\frac{(f^o_{ij} - f^e_{ij})^2}{f^e_{ij}}$ стандартизованным остатком. В рассчете расстояния между категориями x и x' участвуют стандартизованные остатки, принадлежащие этим категориям. Клетки с более высокими показателями вносят более весомый вклад в численное значение критерия χ^2, а следовательно, и в расстояние между двумя объектами x и x'.

- *Мера* φ^2

  $\rho(x, x') = \sqrt{\frac{\sum\limits_{i,j} \frac{(f^o_{ij} - f^e_{ij})^2}{f^e_{ij}}}{\sum\limits_{i,j} f^o_{ij}}}$.

  Является нормализованной мерой χ^2.

Формализуем процесс слияния кластеров. Пусть U и V — ближайший кластеры, обозначим $W := U \cup V$. Тогда расстояние от нового кластера W до любого другого кластера S вычисляется следующим образом

$$R(W, S) = \alpha_U R(U, S) + \alpha_V R(V, S) + \beta R(U, V) + \gamma \mid R(U, S) - R(V, S) \mid,$$

где α_U, α_V, β, γ — числовые параметры. Формула была предложена Лансом и Уильямсом в 1967 году [10] и является обобщением практически всех способов определения расстояния между кластерами.

На практике используются следующие способы вычисления расстояний $R(W,S)$:

- *Расстояние ближнего соседа*

$$R(W,S) = \min_{w \in W, s \in S} \rho(w,s), \text{ где } \qquad \alpha_U = \alpha_V = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}.$$

Метод хорошо работает, если кластеры имеют форму удлиненных цепочек.

- *Расстояние дальнего соседа*

$$R(W,S) = \max_{w \in W, s \in S} \rho(w,s), \text{ где } \quad \alpha_U = \alpha_V = \frac{1}{2}, \beta = 0, \gamma = \frac{1}{2}.$$

Метод хорошо работает, когда кластеры имеют форму сильно удаленных друг от друга скоплений точек.

- *Среднее расстояние*

$$R(W,S) = \frac{1}{\mid W \mid\mid S \mid} \sum_{w \in W} \sum_{s \in S} \rho(w,s), \text{ где } \quad \alpha_U = \frac{\mid U \mid}{\mid W \mid}, \alpha_V = \frac{\mid V \mid}{\mid W \mid}, \beta = -\alpha_U \alpha_V, \gamma = 0.$$

- *Расстояние между центрами*

$$R(W,S) = \rho^2 \left(\sum_{w \in W} \frac{w}{\mid W \mid}, \sum_{s \in S} \frac{s}{\mid S \mid} \right), \text{ где } \quad \alpha_U = \frac{\mid U \mid}{\mid W \mid}, \alpha_V = \frac{\mid V \mid}{\mid W \mid}, \beta = -\alpha_U \alpha_V, \gamma = 0.$$

Здесь большее значение придается крупным кластерам.

- *Расстояние Уорда*

$$R(W,S) = \frac{\mid S \mid\mid W \mid}{\mid S \mid + \mid W \mid} \rho^2 \left(\sum_{w \in W} \frac{w}{\mid W \mid}, \sum_{s \in S} \frac{s}{\mid S \mid} \right), \text{ где } \quad \alpha_U = \frac{\mid S \mid\mid U \mid}{\mid S \mid + \mid W \mid},$$

$$\alpha_V = \frac{\mid S \mid\mid V \mid}{\mid S \mid + \mid W \mid}, \beta = -\frac{\mid S \mid}{\mid S \mid + \mid W \mid}, \gamma = 0.$$

При использовании расстояний ближнего и дальнего соседа после включения объекта в кластер по-прежнему учитываются расстояния от данного объекта до других.

Среднее расстояние и расстояние между центрами определяют рассчет некоторой центральной в том или ином смысле точки, характеризующей кластер в целом.

Метод Уорда основан на объединении не максимально близких кластеров, а тех, слияние которых дает наименьший прирост внутрикластерной дисперсии.

4.0.2. Свойство монотонности.

Определение 6. Пусть R_t — расстояние между ближайшими кластерами, выбранными на t-м шаге для слияния. Тогда говорят, что функция R обладает свойством *монотонности*, если при каждом слиянии расстояние между объединяемыми кластерами только увеличивается, то есть $R_2 < R_3 < ... < R_l$.

Свойство монотонности позволяет визуализировать процесс кластеризации с помощью *дендрограммы*. По вертикальной оси откладываются объекты, по горизонтальной — расстояния R_t.

Теорема 4.0.1 (Миллиган, 1979)**.** *Если выполняются условия:*

1. $\alpha_U \geq 0, \quad \alpha_V \geq 0;$
2. $\alpha_U + \alpha_V + \beta \geq 1;$
3. $\min\{\alpha_U, \alpha_V\} + \gamma \geq 0,$

то кластеризация является монотонной.

4.0.3. Свойства растяжения и сжатия

Расстояния обладают свойством растяжения, если по мере роста кластера расстояния от него до других кластеров увеличиваются, как будто пространство вокруг кластера растягивается. Свойство растяжения способствует более чёткому отделению кластеров. С другой стороны, при слишком сильном растяжении возможно найти кластеры там, где их изначально не было.

Расстояния обладают свойством сжатия, если по мере роста кластера расстояния от него до других кластеров уменьшается, и кажется, что пространство вокруг кластера сжимается.

Отношение $\frac{R_t}{\rho(\mu_U, \mu_V)}$ определяет свойство растяжения и сжатия, где $R_t = R(U, V)$ — расстояние между ближайшими кластерами, объединяемыми на t-м шаге, μ_U и μ_V — центры этих кластеров.

Если $\frac{R_t}{\rho(\mu_U,\mu_V)} > 1 \quad \forall t$, то расстояние R является растягивающим.
Если $\frac{R_t}{\rho(\mu_U,\mu_V)} < 1 \quad \forall t$, то сжимающим.

Есть расстояния, которые не являются ни сжимающими, ни растягивающими, они сохраняют метрику пространства.

На практике часто применяются *гибкое расстояние*, которое представляет собой компромисс между методами ближнего соседа, дальнего соседа и среднего расстояния. Оно определяется одним параметром β: $\alpha_U = \alpha_V = \frac{1-\beta}{2}, \gamma = 0, \beta < 1$. Гибкое расстояние является сжимающим при $\beta > 0$ и растягивающим при $\beta < 0$. Стандартная рекомендация: $\beta = -0,25$[10].

4.0.4. Свойства редуктивности

Зачастую алгоритм кластеризации применяется к выборкам длиной в несколько сотен объектов и более. Таким образом, необходима оптимизация работы алгоритма, во избежание попарных сравнений индивидов.

Если задать параметр δ, ограничивающий окрестность индивида, то сравниваться будут наиболее близкие пары $(U, V) : \{R(U, V) \leq \delta\}$. Параметр δ увеличивается на каждой итерации объединения кластеров.

Определение 7. Расстояние R называется редуктивным, если для $\forall \delta > 0$ и $\forall U, V :$ $R(U, V) \leq \delta$ объединение $\delta-$окрестностей U и V содержит в себе $\delta-$окрестность кластера $W = U \cup V$:

$$\{S \mid R(U \cup V, S) < \delta, R(U, V) \leq \delta\} \subseteq \{S \mid R(S, U) < \delta \text{ или } R(S, V) < \delta\}.$$

Теорема 4.0.2 (Диде и Моро, 1984)**.** *Если выполняются условия:*

1. $\alpha_U \geq 0, \alpha_V \geq 0$;
2. $\alpha_U + \alpha_V + \min\{\beta, 0\} \geq 1$;
3. $\min\{\alpha_U, \alpha_V\} + \gamma \geq 1$,

то расстояние R является редуктивным.

Сравнение условий теорем 4.0.1 и 4.0.2 показывает, что всякое редуктивное расстояние является монотонным, следовательно позволяет визуализировать кластеризацию в виде дендограммы.

Глава 5

Алгоритм кластеризации Маркова

5.0.1. Описание метода

Алгоритм клатеризации Маркова (Markov Cluster Algorithm или MCL) — быстрый и масштабируемый алгоритм кластеризации без учителя, основанный на моделировании случайных блужданий в графах.

Алгоритм моделирует поток, используя два простых матричных оператора. Первый оператор — expansion, который совпадает с нормальным матричным умножением. Второй, inflation, — возведение матрицы в степень через произведение Адамара с последующим диагональным масштабированием. Expansion соответствует вычислению случайных блужданий более высокой длины, то есть блужданий со многими шагами. Длинные пути являются более распространенными в пределах кластеров, нежели чем между ними. Inflation повышает вероятности внутрикластерных блужданий и уменьшает межкластерные. Действие операторов более подробно описано в разделе 5.0.4.

Предполагается отсутствие априорных знаний о структуре кластера.

5.0.2. Стохастические матрицы и случайные блуждания

Стохастическая матрица — это неотрицательная матрица, чьи строки или столбцы каждые дают в сумме единицу.

Согласно [11] введем обозначения.

Пусть M — матрица из $\mathbb{R}^{n\times n}$, α и β — различные последовательности индексов в диапазоне $\{1,\dots,n\}$. Дополнение к α имеет вид α^c. Обозначим за $M[\alpha|\beta]$ подматрицу, состоящую из строк под номерами α и столбцов с индексами β.

Определение 8. Матрица M называется неприводимой, если для $\forall\alpha : 1 \le \alpha \le n-1$ подматрица $M[\alpha|\alpha^c]$ имеет хотя бы один ненулевой элемент.

Введенные выше понятия так же могут быть сформулированы в терминах графов.

Проассоциируем с матрицей M граф $G = (V, w)$, где $V = \{v_1, .., v_n\}$ — конечный набор элементов, $w : V \times V \to \mathbb{R}_{\ge 0}$.

Положим $w(v_p, v_q) = 1 \Leftrightarrow M_{pq} \neq 0$. Тогда существование таких последовательностей α, что элементы $M[\alpha|\alpha^c] = 0$ означает, что нет дуг в G, соединяющих узлы под-

графа, определенного на α, с узлами подграфа, определенного на α^c. Таким образом, матрица M является приводимой, если ассоциированный с ней граф G можно разбить на два непересекающихся подграфа, которые не имеют дуг, соединяющих их.

Случайное блуждание — это конечная цепь Маркова, которая является обратимой во времени. Дело в том, что нет большой разницы между теорией случайных блужданий на графах и теории конечных цепей Маркова. Каждую цепь Маркова можно рассматривать как случайное блуждание на ориентированном графе со взвешенными дугами.

Определение 9. Пусть G — граф с n узлами, $M = \mathcal{M}_G$ — ассоциированная с ним матрица. Стохастическая матрица, ассоциированная с графом G, обозначается как $\mathcal{T}_G$ и является нормализованной формой матрицы M.

Пусть d — диагональная матрица, на диагонали которой стоят "веса" столбцов матрцы M. То есть $d_{kk} = \sum\limits_i M_{ik}$, где $d_{ij} = 0$, $i \neq j$. Тогда $\mathcal{T}_G$ определяется как

$$\mathcal{T}_G = \mathcal{M}_G d^{-1}.$$

Пусть G — граф, ассоциированный со стохастической по столбцам матрицей $T = \mathcal{T}_G$. Рассмотрим столбец q, здесь T_{pq} означает "меру притяжения вершины q к вершине p", это имеет смысл только в контексте столбца q.

Если для каждого из столбцов матрицы все отличные от нуля значения равномерно распределены, то этот факт можно интерпретировать как "каждый узел одинаково притягивается ко всем своим соседям", или "каждый узел движется к каждому из своих соседей с равной вероятностью".

Замечание 2. Пусть M — стохастическая по столбцам матрица, тогда $M^m = (M_{ij}^m)$ — m-я степень матрицы, так же стохастическая по столбцам матрица.

Определение 10. Конечная стохастическая матрица $M = (M_{ij})$, $i, j = 1, \ldots, n$ называется регулярной, если $\exists m \in \mathbb{N} : M_{ij}^m > 0$, где M_{ij}^m — элементы m-й степени матрицы M.

Теорема 5.0.1 ([10])**.** *Если M — регулярная стохастическая матрица, то найдется вектор $\Pi = (\Pi_1, \ldots, \Pi_n)$: $M^n \to \mathbb{I}^{\mathrm{T}}\Pi$, где $\mathbb{I} = (1, \ldots, 1)$ — вектор размерности $n \times 1$.*

5.0.3. Основная парадигма Markov Cluster Algorithm

Возникает вопрос, как определить понятие “кластер” в рамках кластеризации графами? Существует несколько альтернативных формулировок. Пусть G — граф, обладающий кластерной структурой, тогда

- Количество длинных путей в G велико для пар вершин, лежащих в одном кластере, и мало для пар вершин, принадлежащих к разным кластерам.
- Случайное блуждание в G, которое посещает кластер, скорее всего, не покидает его.
- Связи между различными кластерами, вероятно, будут заключены в кратчайших путях между парами вершин G.

Дадим формальное определение кластера.

Определение 11. Пусть M — неотрицательная идемпотентная матрица, такая что $\forall j$ $\exists i : M[i, j] \neq 0$. Пусть G — ассоциированный с M граф, обозначим отношение ассоциации через $\to$. Через V_x обозначим набор аттракторов в G, тогда $E = \{E_1, \ldots, E_d\}$ — множество классов эквивалентности относительно $\to$ на V_x. Определим отношение v на $E \times V$ как

$$v(E, \alpha) = \begin{cases} 1, & \text{если } \exists \beta \in E : \alpha \to \beta, \\ 0, & \text{иначе.} \end{cases}$$

Кластеризация с перекрытием определяется как $C\mathcal{L}_M = \{C_1, \ldots, C_d\}$, которая определена на V, имеет d элементов и ассоциирована с M. Тогда i-й кластер C_i, $i = 1, \ldots, d$ определяется соотношением $C_i = \{v \in V | v(E_i, v) = 1\}$.

Заметим, что множество кластеров — в точности множество слабо связанных объектов в графе G. Принадлежность $E_i \in C_i$ означает то, что каждый кластер имеет по меньшей мере один элемент, являющийся уникальным для этого кластера.

Таким образом, основной парадигмой MCA можно считать следующее утверждение: любое случайное блуждание в графе G скорее всего не покинет кластер, в то время как многие другие будут посещать его.

В основе алгоритма лежит идея моделирования потока в пределах графа, с целью усиления связей, где они являются сильными и ослабления в противном случае. Это становится возможным благодаря операторам expansion и inflation.

5.0.4. Операторы expansion и inflation

Алгоритм MCL , называемых expansion и inflation.

Пусть $M \in \mathbb{R}^{k \times l}$ — стохастическая по столбцам матрица.

Expansion

Оператор expansion совпадает с обычным возведением матрицы в степень e. Стандартным параметром является $e = 2$.

Возведение в степень повышает вероятности внутрикластерных блужданий, то есть вероятности перехода внутри кластера будут выше, чем между ними.

Inflation

Положим $r \geq 0, r \in \mathbb{R}$. Обозначим inflation оператор как $\Gamma_r : \mathbb{R}^{k \times l} \to \mathbb{R}^{k \times l}$ который определяется следующим образом:

$$(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^{k} (M_{iq})^r.$$

Опять же стандартным значением параметра является $r = 2$. Параметр лежит в диапазоне $r \geq 0$, так как при таких значениях r данные имеют разумное толкование. Значения $0 \leq r \leq 1$ увеличивают однородность вектора вероятностей, то есть “каждый узел движется к каждому из своих соседей с равными вероятностями”. В то время как значения $1 < r < \infty$ увеличивают неоднородность.

5.0.5. Описание алгоритма MCL

Алгоритм работы MCL состоит из чередования применения операторов expansion и inflation в цикле. На k-й итерации цикла вычисляются две матрицы, обозначенные как T_{2k} и T_{2k+1}. Матрица T_{2k} вычисляется возведением в степень e_k матрицы T_{2k-1}, вычисленной на предыдущем этапе. Матрица T_{2k+1} вычисляется применением оператора Γ_{r_k} к T_{2k}.

Таким образом, алгоритм имеет следующую структуру:

$MCL(G, \Delta, e_{(i)}, r_{(i)})$ {

$G = G + \Delta$

$T_1 = \mathcal{T}_G$

for $k = 1, \ldots, \infty$ {

$T_{2k} = Exp_{e_k}(T_{2k-1})$

$T_{2k+1} = \Gamma_{r_k}(T_{2k})$

if (T_{2k+1} *идемпотентна*) **break**

}

Интерпретируем T_{2k+1} *согласно определению* 11.

}

Заключение

Была изучена литература, освещающая актуальность и проблематику анализа больших данных. В работе дан обзор основых характеристик больших данных и методов их кластеризации. Представлена иерархическая кластеризация со всем многообразием метрик. Более детально были изучены алгоритмы MCL и SBM.

Собрана и структурирована теоретическая основа алгоритма SBM, который представлял наибольший интерес благодаря заявленной эффективности алгоритма и отсутствию четкого и формального описания. Реализована работа алгоритма стохастической блочной модели и симметричной стохастической блочной модели. Также реализована проверка разделимсти графа. В том числе была решена проблема наглядной визуализации полученной кластерной структуры графа. Реализация представлена в математической среде R.

Также представлены примеры успешного восстановления кластерной структуры графа при различных объёмах выборки, количествах кластеров и параметров вероятностей связи. Приведен пример графа, который не удовлетворяет заявленным условиям разделимости, кластерную структуру которого восстановить, как и ожидалось, не удалось.

Алгоритм SBM находится на пике развития и имеет множество открытых проблем, решение которых является перспективой работы. Также общей задачей для любых алгоритмов является минимизация вычислительной сложности без потери точности. Стоит отметить, что даже в период моего изучения алгоритма, среди заинтересованных людей были представлены различные варианты реализации и модификации алгоритма, что опять же говорит о востребованности и актуальности метода.

Список литературы

1. Emmanuel Abbe. Community detection and the stochastic block model. — 2016.
2. Emmanuel Abbe, Colin Sandon. Recovering communities in the general stochastic block model without knowing the parameters. — 2015.
3. Emmanuel Abbe, Afonso S. Bandeira, Georgina Hall. Exact Recovery in the Stochastic Block Model. — 2014.
4. Emmanuel Abbe, Colin Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. — 2015.
5. Emmanuel Abbe. Community detection and the stochastic block model: recent developments. — 2016.
6. Viktor Mayer-Schonberger, Kenneth Cukier. BIG DATA A Revolution That Will Transform How We Live, Work, and Think. — Eamon Dolan/Mariner Books, 2013.
7. Beyer M. A., Laney D. The importance of «big data»: A definition. — 2012. — URL: https://www.gartner.com/doc/2057415.
8. Райзин Д. В. Классификация и кластеризация. — М.:Мир, 1980. — 390 с.
9. Воронцов К. В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций. — МГУ, 2007.
10. Lance G. N., Willams W. T. A General Theory of Classificatory Sorting Strategies 1. Hierarchical Systems. — 1967. — 373–380 p.
11. Stijn van Dongen. Graph clusteering by flow simulation : Ph.D. thesis. — 2000.
12. Jure Lescovec, Anand Rajaraman, Jeffrey D. Ullman. Mining of Massive Datasets. — 2014.

Приложение А

Реализация в R SBM и SSBM

Генерируем граф, состоящий из $n * k$ узлов, где n — общее количество вершин графа, k — предполагаемое количество кластеров.

```
## SBM ##
npc <- 12
k <- 2
n <- npc * k
```

Моделирование матрицы W, элементами которой являются вероятности связи вершин графа.

```
Z <- diag(k) % x % matrix(1,npc,1)
W <- matrix(runif(k*k),k,k)
```

Моделирование матрицы смежности M, которая является матричным представлением графа.

```
M <- 1*(matrix(runif(n*n),n,n)<Z%*%W%*%t(Z))
```

Используя пакет R, разобьем на кластеры смоделированный нами граф, представленный в виде матрицы M.

```
library(blockmodels)
model <- BM_bernoulli("SBM",M )
model$estimate()
which.max(model$ICL) # итоговое количество кластеров
```

Аналогичное моделирование графа и соответствующей матрицы смежности M. Используем метод SSBM, следовательно промоделируем симметричную матрицу W, т.к. в данной модели нет разницы между кластерами, значит и вероятности связи между кластерами совпадают. Далее разбиваем на кластеры смоделированный граф.

```
## SBM symmetric ##
npc <- 30
k <- 2
n <- npc * k
```

```
Z <- diag(k)%x%matrix(1,npc,1)
W <- matrix(runif(k*k),k,k)
W[lower.tri(W)]<-t(W)[lower.tri(W)]
M <- 1*(matrix(runif(n*n),n,n)<Z%*%P%*%t(Z))
M[lower.tri(M)]<-t(M)[lower.tri(M)]


model <- BM_bernoulli("SBM_sym",M )
model$estimate()
which.max(model$ICL)
```

Далее представлена визуализация графа по матрице смежности.

```
library("igraph")
library("RColorBrewer")


graph <- graph_from_adjacency_matrix(M, mode = "undirected",
                                     diag = FALSE)
plot(graph, vertex.size = 15, vertex.color = "plum2",edge.width = 4)
```

Согласно полученной разметки графа sbmclust ниже представлена программная визуализация представления кластерной структуры.

```
sbmclust <- apply(model$memberships[[5]]$Z, 1, which.max)
graph1 <- graph_from_adjacency_matrix(M, mode = "undirected",
                                      diag = FALSE)
V(graph1)$color <- ifelse(sbmclust == 1, "lightblue3", "deeppink3")
graph1 <- set_graph_attr(graph1, "layout", layout_with_kk(graph1))
plot(graph1, vertex.size = 15, edge.width = 4)
```

Приложение Б

Реализация в R SBM и SSBM примеров для $n = 1000, k = 5$

```
 npc <- 200
k <- 2
p <- c(0.2, 0.2, 0.2, 0.2, 0.2)
n <- npc * k

Z <- diag(k)%x%matrix(1,npc,1)

W <- diag(1/50, 5,5)
W[upper.tri(W)] <- 1/1000
W[lower.tri(W)] <- 1/1000

M<-1*(matrix(runif(n*n),n,n)<Z%*%W%*%t(Z))
M[lower.tri(M)]<-t(M)[lower.tri(M)]

model <- BM_bernoulli("SBM_sym",M )
model$estimate()
which.max(model$ICL)

model$memberships[[k]]$Z

sbm_clust <- apply(model$memberships[[k]]$Z, 1, which.max)

graph <- graph_from_adjacency_matrix(
                    M[-c(129,522,921,542,610,859,253,698),
                      -c(129,522,921,542,610,859,253,698)],
                      mode = "undirected", diag = FALSE)
plot(graph, vertex.size = 3, vertex.color = "plum2",
```

```
                                edge.width = 1, vertex.label = NA)

graph1 <- graph_from_adjacency_matrix(
                    M[-c(129,522,921,542,610,859,253,698),
                      -c(129,522,921,542,610,859,253,698)],
                      mode = "undirected", diag = FALSE)
V(graph1)$color <- ifelse(sbm_clust == 1,
                    "gold", ifelse(sbm_clust == 2, "skyblue1",
                    ifelse(sbm_clust == 3,"deeppink3",
                    ifelse(sbm_clust == 4, "springgreen",
                    "slateblue4"))))
graph1 <- set_graph_attr(graph1, "layout", layout_with_kk(graph1))
plot(graph1, vertex.size = 3, edge.width = 1, vertex.label = NA)
```

Приложение B

Реализация в R SBM и SSBM примеров для n = 12, k = 2

```
npc <- 6
k <- 2
p <- c(0.5, 0.5)
n <- npc * k # nodes
Z <- diag(k)%x%matrix(1,npc,1)

W <- diag(8/12, 2,2)
W[upper.tri(W)] <- 2/12
W[lower.tri(W)] <- 2/12

M<-1*(matrix(runif(n*n),n,n)<Z%*%W%*%t(Z))
M[lower.tri(M)]<-t(M)[lower.tri(M)]

model <- BM_bernoulli("SBM_sym",M )
model$estimate()
which.max(model$ICL)

sbm_clust <- apply(model$memberships[[2]]$Z, 1, which.max)
model$memberships[[2]]$Z

graph <- graph_from_adjacency_matrix(M,
                    mode = "undirected", diag = FALSE)

plot(graph, vertex.size = 15,
             vertex.color = "plum2",edge.width = 4)

graph1 <- graph_from_adjacency_matrix(M,
```

```
                                    mode = "undirected", diag = FALSE)
V(graph1)$color <- ifelse(sbm_clust == 1,
                                  "lightblue3", "deeppink3")
graph1 <- set_graph_attr(graph1, "layout", layout_with_kk(graph1))
plot(graph1, vertex.size = 15, edge.width = 4)
```