

## 2. ЛЕКЦИЯ - Традиционные подходы

Прежде чем перейти концепции обучения представлению графов и глубокого обучения на графах, необходимо дать некоторые методологические предпосылки. Таким образом, необходимо рассмотреть использование машинного обучения на графиках до появления современных подходов к глубокому обучению.

### 2.1. Характеристика графов и методы ядра

Рассмотрим традиционные подходы к классификации с использованием графических данных следуют стандартной парадигме машинного обучения, которая была популярна до появления глубокого обучения.

#### *Характеристики и функции на уровне узла*

В принципе, свойства и характеристики данных могут быть использованы в качестве признаков в модели классификации узлов (например, в качестве входных данных для модели логистической регрессии).

*Степень узла.* Наиболее очевидной и простой характеристикой узла для изучения является степень, которая обычно обозначается  $d_u$  для узла  $u \in V$  и просто подсчитывает количество ребер, инцидентных узлу (2.1):

$$d_u = \sum_{v \in V} A(u, v) \quad (2.1)$$

Необходимо обратить внимание, что в случае ориентированных и взвешенных графов можно различать разные понятия степени, например, соответствие исходящим ребрам или входящим ребрам путем суммирования по строкам или столбцам в уравнении (2.1). В целом, степень узла (рис.2) является важной характеристикой, которую необходимо учитывать, и часто является одним из наиболее информативных признаков в традиционных моделях машинного обучения, применяемых к задаче уровня узла [5].

*Центральность узла.* Степень узла просто измеряет, сколько соседей имеет узел, но этого не обязательно достаточно для измерения важности узла в графе. Во многих случаях, можно извлечь дополнительную информацию из более мощных показателей важности узла. Для этого можно рассмотреть меру центральности узлов, которая формирует полезные функции для самых разных задач классификации узлов.

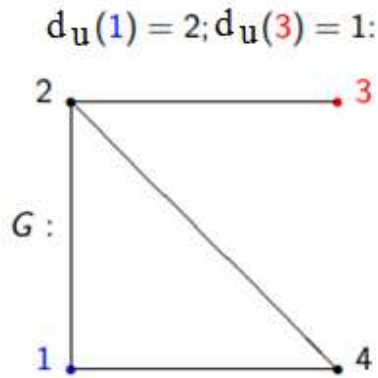


Рис. 2.2. Пример расчета степени узла

Одной из популярных и важных мер центральности является так называемая центральность собственного вектора, которая также, как и степень узла учитывает, насколько важны соседи узла. В частности, определить центральность собственного вектора узла  $e_u$  через рекуррентное соотношение, в котором центральность узла пропорциональна средней центральности его соседей (2.2):

$$e_u = \frac{1}{\lambda} \sum_{u \in V} A(u, v) e_v \quad \forall u \in V \quad (2.2)$$

где  $\lambda$  постоянная (собственное значение). Переписав это уравнение в векторной записи с  $e$  в качестве вектора центральностей узлов, можно увидеть, что эта рекуррентность определяет стандартное уравнение собственного вектора для матрицы смежности (2.3):

$$\lambda e = Ae \quad (2.3)$$

Другими словами, мера центральности, удовлетворяющая повторяемости в уравнении 2.2, соответствует собственному вектору матрицы смежности. Предполагая, что требуются положительные значения центральности, можно применить теорему Перрона-Фробениуса для дальнейшего определения того, что вектор значений центральности  $e$  задается собственным вектором, соответствующим наибольшему собственному значению  $A$ .

Один из взглядов на центральность собственного вектора состоит в том, что он ранжирует вероятность посещения узла при случайном блуждании бесконечной длины на графе. Эту точку зрения можно проиллюстрировать, рассмотрев использование степенной итерации для получения значений центральности собственного вектора. То есть, поскольку собственный вектор  $A$  с наибольшим зна-

чением, можно вычислить  $e$ , используя степенную итерацию через (2.4) [7]:

$$e^{(t+1)} = Ae^{(t)} \quad (2.4)$$

Таким образом, повторяя этот процесс бесконечно, мы получаем оценку, пропорциональную количеству посещений узла на путях бесконечной длины. Как правило процесс умножения вектора центральности собственного вектора на матрицу смежности повторяется до тех пор, пока значения вектора центральности собственного вектора для узлов на графике не достигнут равновесия или не перестанут показывать заметные изменения.

Есть, конечно, и другие меры центральности, которые можно использовать. К ним относятся центральность между узлами, которая измеряет, как часто узел лежит на кратчайшем пути между двумя другими узлами, а также центральность по близости, которая измеряет среднюю длину кратчайшего пути между узлом и всеми другими узлами и т.д.

*Коэффициент кластеризации.* Важность степени и центральности узла, несомненно. Однако интересно узнать полезность других узлов при схожести степеней и центральности узла собственного вектора. Такое важное структурное отличие можно измерить, используя вариации коэффициента кластеризации, который измеряет долю замкнутых треугольников в локальной окрестности узла. Популярный локальный вариант коэффициента кластеризации для неориентированных графов вычисляется следующим образом (2.5) [7]:

$$c_u = \frac{|(v_j, v_k) \in E: v_j, v_k \in N(u)|}{(d_u/2)} \quad (2.5)$$

Числитель в этом уравнении подсчитывает количество ребер между соседями узла  $u$  (где используется  $N(u) = \{v \in V: (u, v) \in E\}$  для обозначения окрестности узла). Знаменатель вычисляет, сколько ребер может находиться в окрестности  $u$ . Тогда если у вершины  $v_i$  есть  $k_i$  соседи,  $d_u$  = ребра могут существовать между вершинами в окрестности  $u$ .

Рассмотрим пример коэффициента локальной кластеризации на неориентированном графе (рис. 2.3).

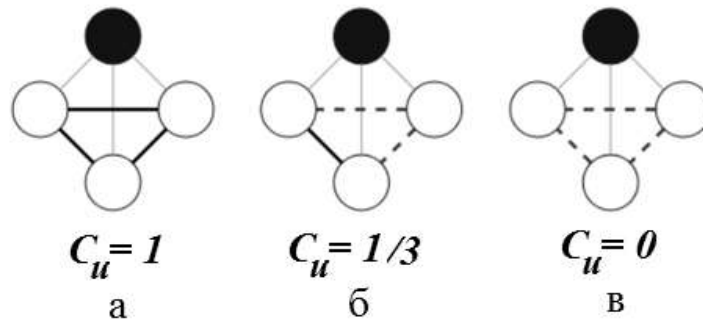


Рис. 2.3. Пример локального коэффициента кластеризации на неориентированном графе.

На рис. 2.3 помеченный узел (черный) имеет трех соседей, между которыми может быть не более 3 соединений. На рис. 2.3а реализовано все три возможных соединения (толстые черные сегменты), что дает локальный коэффициент кластеризации, равный  $c_{u_a} = \text{число связей между соседями данного узла} - 3 / \text{возможное число связей между соседями} - 3 = 1$ . На рис. 2.3б реализовано только одно соединение (толстая черная линия), а 2 соединения отсутствуют (пунктирные линии). Тогда локальный кластерный коэффициент  $c_{u_b} = \text{число связей между соседями данного узла} - 1 / \text{возможное число связей между соседями} - 3 = 1/3$ . На рис. 2.3в нет ни одно из возможных соединений между соседями черного узла. Это даст значение локального коэффициента кластеризации равным 0.

Теперь реализуем расчет локального коэффициента кластеризации на неориентированном графе для нашего примера (рис. 2.4).

$c_{u_1} = \text{число связей между соседями данного узла} - 1 / \text{возможное число связей между соседями} - 1 = 1$ .

$c_{u_2} = \text{число связей между соседями данного узла} - 0 / \text{возможное число связей между соседями} - 1 = 0$ .

$c_{u_3} = \text{число связей между соседями данного узла} - 1 / \text{возможное число связей между соседями} - 1 = 1$ .

$c_{u_4} = \text{число связей между соседями данного узла} - 1 / \text{возможное число связей между соседями} - \left( \frac{k_i(k_i-1)}{2} = \frac{3 \times 2}{2} \right) - 3 = 1/3$ .

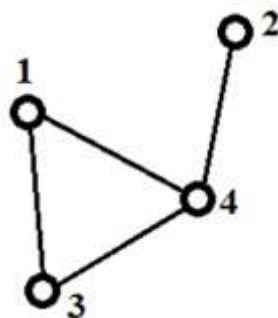


Рис. 2.4. Пример неориентированного графа

Данные значения сопоставимы с центральностью узла, т.е. узлы 1 и 3 имеют самые большие значения как коэффициент кластеризации, так и центральность узла.

Существует альтернативные способы рассмотрения коэффициента кластеризации. Например, глобальный коэффициент кластеризации основанный на триплетах узлов. Триплет — это три узла, которые соединены двумя (открытый триплет) или тремя (замкнутый триплет) неориентированными связями. Поэтому треугольный граф включает три замкнутых триплета, по одному центрированному на каждом из узлов (это означает, что три триплета в треугольнике происходят из перекрывающихся выборок узлов).

Таким образом, глобальный коэффициент кластеризации — это отношение числа замкнутых триплетов (или трёх треугольников) над общим количеством триплетов (как открытых, так и закрытых).

### **Функции уровня и ядро графа**

Рассмотрим ряд определений для методов ядра графа, которые представляют собой подходы к разработке функций и используются в моделях машинного обучения:

1. Ядро графа — это понятие, описывающее поведение графа в отношении гомоморфизмов графа, которые представляют отображение между двумя графами, не нарушающее структуру. Граф является ядром, если каждый гомоморфизм является изоморфизмом, то есть это биекция вершин графа. Ядро содержит в себе много информации о спрямляющем пространстве, и позволяет производить в нем различные операции, не зная самого пространства.

2. Гомоморфизм графов — это отображение между двумя графами, не нарушающее структуру. Это отображение между набором вершин двух графов, которое отображает смежные вершины в смежные. Гомоморфизм графов — это подобие одного графа другому, но не наоборот (рис. 2.5).

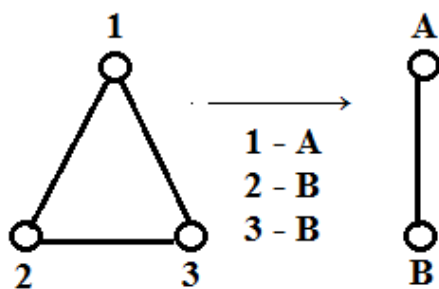


Рис. 2.5. Пример гомоморфизма графа

Гомоморфизмы обобщают различные понятия раскрасок графов, как част-

ный случай разметки графа и позволяет выражение важных классов задач удовлетворения ограничений. Частными случаями гомоморфизма являются изоморфизм.

3. Изоморфизм графа  $G = \{V_G, E_G\}$  и  $H = \{V_H, E_H\}$  – есть биекция между множествами вершин графов  $f: V_G \rightarrow V_H$  такая, что любые две вершины  $u$  и  $v$  графа  $G$  смежны тогда и только тогда, когда вершины  $f(u)$  и  $f(v)$  смежны в графе  $H$ . При этом графы понимаются неориентированными и не имеющими весов вершин и ребер (рис. 2.6).

Граф G	Граф H	Изоморфизм между графами G и H	Подстановка изоморфизма f
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$	$\begin{pmatrix} a & b & c & d & g & h & i & j \\ 1 & 6 & 8 & 3 & 5 & 2 & 4 & 7 \end{pmatrix}$

Рис. 2.6. Пример изоморфизма графа

Таким образом, изоморфизм, есть совпадение двух графов, т.е. подобие в обе стороны.

Факт, что гомоморфизмы могут быть использованы, приводит к мощным алгебраическим структурам – предпорядку на графах. Вычислительная сложность поиска гомоморфизма между заданными графами в общем случае запредельная, но известно много частных случаев, когда задача выполнима за полиномиальное время. Границы между решаемыми и непреодолимыми случаями находятся в области активных исследований. В этом заключается проблема изоморфизма

*Агрегирование статистики на уровне узла.* Самый простой подход к определению функции на уровне графа — просто агрегировать статистику на уровне узла. Например, можно вычислить гистограммы или другую сводную статистику на основе степеней, центральности и коэффициентов кластеризации узлов в графе. Эта агрегированная информация может быть использована в качестве представления на уровне графа. Недостатком этого подхода является то, что он полностью основан на информации на уровне локальных узлов и может упустить важные глобальные свойства в графе.

*Ядро Вайсфейлера-Лемана.* Один из способов улучшить базовый подход с набором узлов – использовать стратегию итеративной агрегации окрестностей. Идея этих подходов состоит в том, чтобы извлечь функции уровня узла, которые содержат больше информации, чем просто их локальной информации. Таким образом, агрегируются функции маркировки в представление на уровне графа. Про-

цесс маркировки останавливается, если получена совершенная раскраска, т. е. никакой цвет не разделяется на группы.

Возможно, наиболее важной и известной из этих стратегий является алгоритм и ядро Вейсфейлера-Лемана. Основная идея алгоритма заключается в следующем [10]:

1. Сначала каждому узлу присваиваем начальную метку  $l^{(0)}(v)$ . В большинстве графов эта метка представляет собой просто степень, т. е.  $l^{(0)}(v) = d_v \quad \forall v \in V$ .

2. Затем итеративно присваивается новая метка каждому узлу, хешируя (преобразование в битовую строку фиксированной длины) мультимножества текущих меток в окрестности узла (2.6):

$$l^{(i)}(v) = \text{HASH} \left( \left\{ \left\{ l^{(i-1)}(v) \quad \forall v \in N(v) \right\} \right\} \right) \quad (2.6)$$

где двойные фигурные скобки используются для обозначения мультинабора, а функция *HASH* (хеш-функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины) сопоставляет каждый уникальный мультинабор с уникальной новой меткой.

3. После выполнения  $K$  итераций перемаркировки (т. е. шаг 2) теперь у нас есть метка  $l^{(K)}(v)$  для каждого узла, которая резюмирует структуру его  $K$ -окрестности. Затем можно вычислить сводную статистику по этим меткам в качестве представления функции для графика. Другими словами, ядро Вейсфейлера-Лемана вычисляется путем измерения разницы между результирующими наборами меток для двух графов.

Ядро Вейсфейлера-Лемана обладает важными теоретическими свойствами. Например, один из популярных способов аппроксимации изоморфизма графов состоит в том, чтобы проверить, имеют ли два графа один и тот же набор меток после  $K$  раундов алгоритма. и известно, что этот подход решает проблему изоморфизма для широкого набора графов.

*Методы на основе графлетов и путей.* При обсуждении функций на уровне узлов, есть еще одна часто применяемая и мощная стратегия, которая состоит в том, чтобы просто подсчитывать появление различных небольших структур подграфов, в этом контексте обычно называемых графлетами. Графлеты – это классы изоморфизма индуцированных подграфов в графе. Формально ядро графлета включает в себя перечисление всех возможных структур графа определенного размера и подсчет того, сколько раз они встречаются в полном графе (на рис. 2.7 показаны различные графлеты размера 3). Проблема с этим подходом состоит в

том, что подсчет этих графлетов представляет собой комбинаторно сложную задачу, хотя было предложено множество приближений.

Альтернативой перечислению всех возможных графлетов является использование методов на основе пути. В этих подходах вместо перечисления графлетов просто исследуются различные виды путей, встречающихся в графе. Например, ядро случайного блуждания, включает в себя выполнение случайных блужданий по графу с последующим подсчетом появления последовательностей различных степеней, в то время как ядро кратчайших путей включает аналогичную идею, но использует только кратчайшие пути между узлами (быстрее, чем случайные блуждания).

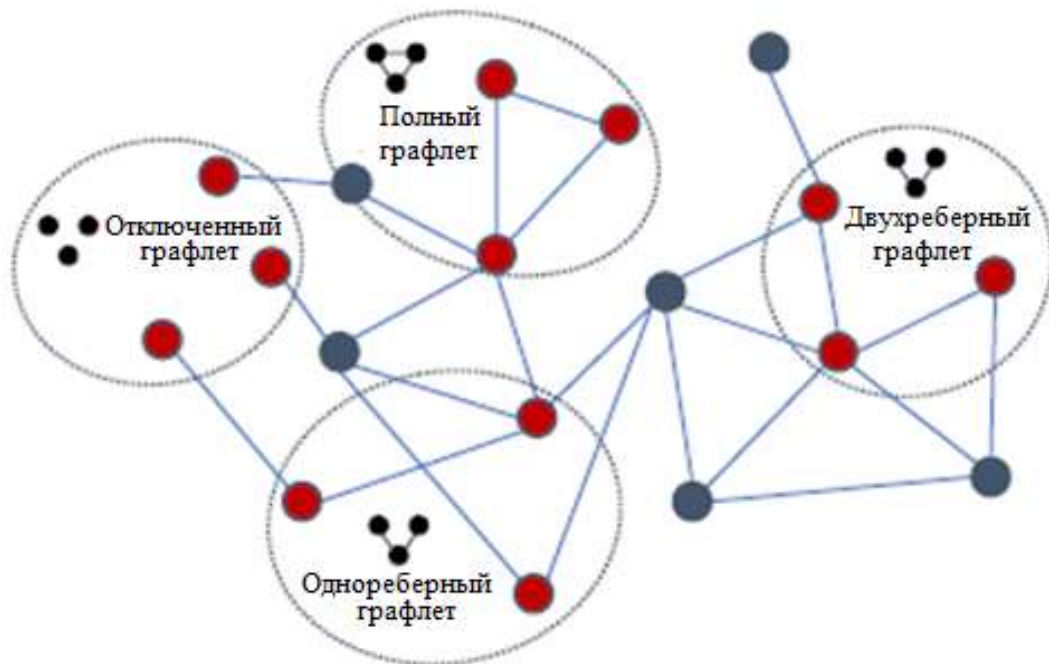


Рис. 2.7. Четыре различных графлета размера 3, которые могут встречаться в простом графе.

При этом случайное блуждание, есть стохастический процесс, который описывает путь, состоящий из последовательности случайных шагов в каком-либо пространстве.

## 2.2. Обнаружение метрики сходства соседства

Рассмотренные различные подходы к извлечению признаков или статистики об отдельных узлах и целых графах полезны для многих задач классификации. Однако они ограничены тем, что не определяют количественную оценку отношений между узлами. Например, эти методы не очень полезны для задачи прогнозирования отношений, где цель – предсказать существование ребра между двумя узлами (рис. 2.8).



На рис. 2.8 пунктирные ребра на обучающем графике удаляются при обучении модели или вычислении статистики сходства. Модель оценивается на основе ее способности предсказывать наличие этих тестовых ребер (то есть невидимых).

Рассмотрим различные статистические меры сходства соседства между парами узлов, которые количественно определяют степень, в которой пара узлов связана. Примером является, простейшая мера сходства соседей просто подсчитывает количество соседей, которые разделяют два узла (2.7):

$$S[u, v] = |N(u) \cap N(v)| \quad (2.7)$$

где используется  $S[u, v]$  для обозначения значения, определяющего отношение между узлами  $u$  и  $v$ . Тогда  $S \in \mathbb{R}^{|V| \times |V|}$  обозначает матрицу подобия, суммирующую всю статистику попарных узлов.

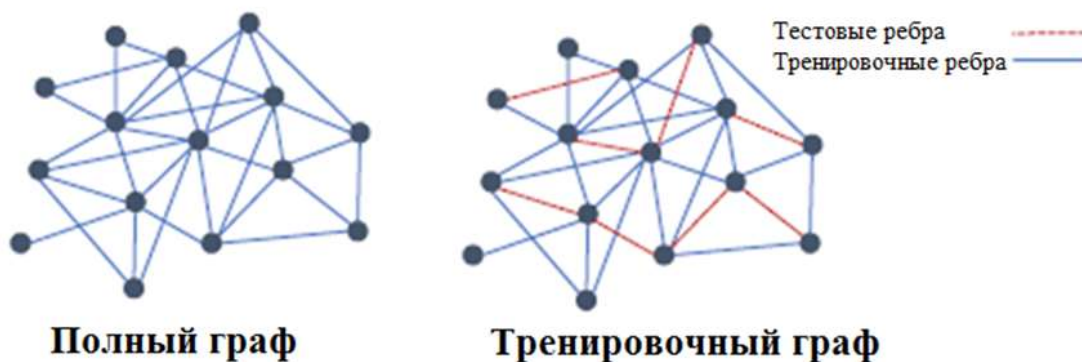


Рис. 2.8. Иллюстрация полного графика и графика с подвыборкой, используемых для обучения.

Хотя здесь не за действительно «машинное обучение», однако этот метод по-прежнему являются очень полезными и мощными базовым средством для прогнозирования отношений. Общая стратегия статистики сходства соседей  $S[u, v]$ , состоит в том, чтобы предположить, что вероятность ребра  $(u, v)$  просто пропорциональна  $S[u, v]$  (2.8):

$$P(S[u, v] = 1) \propto S[u, v] \quad (2.8)$$

Таким образом, чтобы подойти к задаче прогнозирования отношений с использованием меры сходства соседей, нужно установить порог, для определения, когда следует прогнозировать существование ребра. Необходимо обратить внимание, что в настройке прогнозирования отношений обычно предполагается, что

известно только подмножество истинных ребер  $E_{train} \subset E$ .

### ***Локальные меры сходства.***

Локальные статистические данные о сходстве – это мера количества общих соседей, которые разделяют два узла, т. е.  $|N(u) \cap N(v)|$ . Например, индекс Серенсена определяет матрицу  $S_{Sorenson} \in \mathbb{R}^{|V| \times |V|}$  соседства узел-узел, перекрывающуюся с элементами, заданными формулой (2.9).

$$S_{Sorenson}[u, v] = \frac{2|N(u) \cap N(v)|}{d_u + d_v} \quad (2.9)$$

который нормализует количество общих соседей по сумме степеней узлов. Обычно очень важна какая-то нормализация; в противном случае мера сходства будет сильно смещена в сторону прогнозирования ребер для узлов с большими степенями. Другие подобные подходы включают индекс Солтона (2.10), который нормируется произведением степеней  $u$  и  $v$ , т.е.

$$S_{Salton}[u, v] = \frac{2|N(u) \cap N(v)|}{\sqrt{d_u d_v}} \quad (2.10)$$

а также индекс Жаккара (2.11):

$$S_{Jaccard}[u, v] = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \quad (11)$$

В общем, эти меры направлены на количественную оценку сходства между соседями узлов при минимизации любых смещений степеней узлов. Можно найти в литературе много других вариаций этого подхода.

Есть также меры, которые выходят за рамки простого подсчета количества общих соседей и пытаются каким-то образом учитывать важность общих соседей. Индекс распределения ресурсов (Resource Allocation –  $RA$ ) подсчитывает обратные степени общих соседей (2.12),

$$S_{RA}[v_1, v_2] = \sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{d_u} \quad (2.12)$$

в то время как индекс Адама-Адара (Adamic-Adar –  $AA$ ) выполняет аналогичные вычисления с использованием обратного логарифма степеней (2.13) [12]:

$$S_{RA}[v_1, v_2] = \sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{\log(d_u)} \quad (2.13)$$

где  $d(u)$  – множество узлов, смежных с  $u$ . Определение основано на концепции, согласно которой общие элементы с очень большими окрестностями менее значимы при прогнозировании соединения между двумя узлами по сравнению с элементами, разделяемыми небольшим количеством узлов.

Таким образом, обе эти меры придают больший вес общим соседям с низкой степенью, при этом интуитивно понятно, что общий сосед с низкой степенью более информативен, чем общий сосед с высокой степенью.

### ***Меры глобального сходства.***

Меры локального сходства являются чрезвычайно эффективной эвристикой для прогнозирования связей и часто обеспечивают конкурентоспособную производительность даже по сравнению с передовыми подходами к глубокому обучению. Однако локальные подходы ограничены тем, что они рассматривают только окрестности локальных узлов. Например, два узла могут не иметь локального сходства в своих окрестностях, но при этом быть членами одного и того же сообщества на графе. Статистика глобального сходства пытается учитывать такие отношения.

*Индекс Каца.* Индекс Каца является самой основной глобальной статистикой сходства. Чтобы вычислить индекс Каца, необходимо подсчитать количество путей между парой узлов (2.14):

$$S_{Katz}[u, v] = \sum_{i=1}^{\infty} \beta^i A^i[u, v] \quad (2.14)$$

где  $\beta$  – определяемый пользователем параметр, определяющий, какой вес придается коротким путям по сравнению с длинными. Небольшое значение  $\beta < 1$  снизит важность длинных путей.

Индекс Каца возможно рассматривать как центральность, которая вычисляет относительное влияние узла в сети путём измерения числа ближайших соседей (узлы первой степени), а также всех других узлов в сети, которые соединяются через этих ближайших соседей.

Любому пути или связи между парой узлов назначается вес, определённый значением  $\beta$  и расстоянием между узлами как  $\beta^i$ . Соответственно, вес соединений с удалёнными соседями уменьшаются на множитель  $\beta$ .

Рассмотрим пример как измеряется центральность «Ивана» в социальной сети, если  $\beta = 0,5$  (рис.2.9).

Видно, у «Ивана» существует связь с его непосредственными соседями «Леной» и «Борисом». Тогда можно назначить вес каждой связи, который будет равен  $0,5^1 = 0,5$ . Так как «Коля» связан с «Иваном» косвенно через «Бориса», вес, назначенный этому соединению (состоящему из двух связей), будет равен  $0,5^2 = 0,25$ . Тогда вес, назначенный связи между «Анной» и «Иваном» через «Катю» и «Лену», будет равен  $0,5^3 = 0,125$ . Соответственно вес, назначенный связи между «Анной» и «Иваном» через «Диму», «Колю» и «Бориса», будет равен  $0,5^4 = 0,0625$ . Остальные веса рассчитываются аналогично.

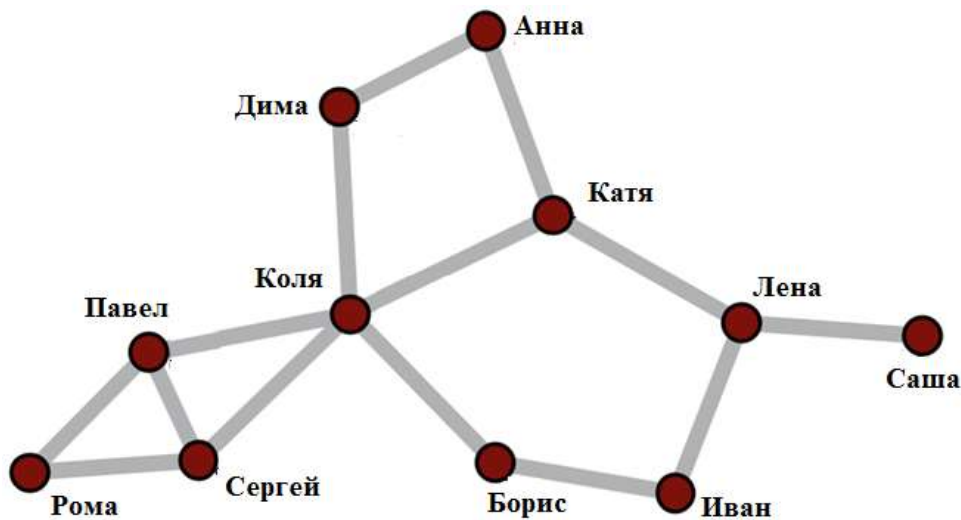


Рис. 2.9. Фрагмент социальной сети.

Необходимо отметить факт, что элемент в позиции  $(u, v)$  матрицы  $A^i$  отражает общее число соединений степени  $i$  между узлами  $u$  и  $v$ . Тогда значение множителя затухания  $\beta$  следует выбрать так, чтобы оно было меньше, чем обратное значение абсолютного значения наибольшего собственного значения матрицы  $A$ . В этом случае следующее выражение может быть использовано для вычисления индекса Каца (2.15):

$$S_{Katz} = (I - \beta A)^{-1} - I \quad (2.15)$$

где  $S_{Katz} \in \mathbb{R}^{|V| \times |V|}$  – полная матрица значений сходства между узлами;  $I$  – вектор размера  $n$  ( $n$  равно числу узлов), состоящий из единиц.

Индекс Каца используют для вычисления центральности в ориентированных сетях, в оценке относительного статуса или влияния объектов в социальной сети. В нейронауках обнаружено, что центральность по Кацу коррелирует с относительной частотой возбуждения нейронов в нейронной сети.

*Сходство Лейхта, Холма и Ньюмана.* Один из самых значимых недостатков индекса Каца заключается в том, что он сильно смещен по степени узла. Уравнение (2.14) обычно дает более высокие общие оценки подобия при рассмотрении узлов высокой степени по сравнению с узлами низкой степени, поскольку узлы высокой степени обычно участвуют в большем количестве путей. Чтобы сгладить этот недостаток, предлагают улучшенную метрику, рассматривая соотношение между фактическим количеством наблюдаемых путей и количеством ожидаемых путей между двумя узлами (2.16):

$$\frac{A^i}{\mathbb{E}[A^i]} \quad (2.16)$$

т. е. количество путей между двумя узлами нормализовано на основе ожиданий того, сколько путей ожидается в рамках случайной модели.

Вычисление математического ожидания  $\mathbb{E}[A^i]$ , основывается на так называемой конфигурационной модели, которая представляет собой метод генерации случайных сетей из заданной последовательности степеней. Таким образом, изображается случайный граф с тем же набором степеней, что и наш данный граф. При этом предположении можно аналитически вычислить, что (2.17) [13]:

$$\mathbb{E}[A[u, v]] = \frac{d_u d_v}{2m} \quad (2.17)$$

где используется  $m = |E|$  для обозначения общего количества ребер в графе.

Уравнение (2.17) утверждает, что в модели со случайной конфигурацией вероятность ребра просто пропорциональна произведению двух степеней узлов. Это можно увидеть, заметив, что из  $u$  выходит  $d_u$  ребер, и каждое из этих ребер имеет шанс  $dv/2m$  закончиться в  $v$ . Для  $\mathbb{E}[A^2[u, v]]$  можно аналогичным образом вычислить (2.18):

Это следует из того факта, что путь длины 2 может проходить через любую промежуточную вершину  $u$ , и тогда вероятность такого пути пропорциональна вероятности того, что ребро, выходящее из  $v_1$ , попадет в  $u$  — заданное  $\frac{d_{v_1} d_u}{2m}$  — умноженной на вероятность того, что ребро выходящее из  $u$  попадает в  $v_2$  — заданное  $\frac{d_{v_2}(d_u - 1)}{2m}$  (где вычитается единица, так как уже израсходовано одно из ребер  $u$  для входящего ребра из  $v_1$ ).

$$\mathbb{E}[A^2[v_1, v_2]] = \frac{d_{v_1} d_{v_2}}{(2m)^2} \sum_{u \in V} (d_u - 1) d_u \quad (2.18)$$

К сожалению, аналитическое вычисление количества ожидаемых узловых путей в модели со случайной конфигурацией становится трудновыполнимым, когда длина пути выходит за пределы значения три. Чтобы получить математическое ожидание  $\mathbb{E}[A^i]$  для более длинных путей (т. е.  $i > 2$ ), полагаются на тот факт, что наибольшее собственное значение можно использовать для аппроксимации роста числа путей. В частности, если определить  $p_i \in \mathbb{R}^{|V|}$  как вектор, подсчитывающий количество путей длины  $i$  между узлом  $u$  и всеми остальными узлами, то для больших  $i$  (2.19)

$$A p_i = \lambda_1 p_{i-1} \quad (2.19)$$

поскольку  $p_i$  в конечном итоге сойдется к доминирующему собственному вектору графа. Это означает, что количество путей между двумя узлами увеличивается в  $\lambda_1$  раз на каждой итерации, где  $\lambda_1$  является наибольшим собственным значением  $A$ . На основе этого приближения для больших  $i$ , а также точного решения для  $i = 1$  получаем (2.20) :

$$\mathbb{E}[A[u, v]] = \frac{d_u d_v \lambda_1^{i-1}}{2m} \quad (2.20)$$

Наконец, собрав все вместе, можно получить нормализованную версию индекса Каца (2.21), которую называв индексом LNH (по инициалам авторов, предложивших алгоритм Leicht, Newman, Holme (Лейхта, Ньюмана, Холма)):

$$S_{\text{LNH}}[u, v] = I[u, v] + \frac{2m}{d_u d_v} \sum_{i=0}^{\infty} \beta^i \lambda_1^{1-i} A^i[u, v] \quad (2.21)$$

где  $I$  – это единичная матрица  $|V| \times |V|$ , последовательно проиндексированная как  $A$ .

В отличие от индекса Каца, индекс LNH учитывает ожидаемое количество путей между узлами и дает высокую меру сходства только в том случае, если два узла встречаются на большем количестве путей, чем ожидается. Решение матричного ряда (после игнорирования диагональных членов) можно записать как (2.22):

$$S_{\text{LNH}} = 2 \propto m \lambda_1 D^{-1} (I - \frac{\beta}{\lambda_1} A)^{-1} D^{-1} \quad (2.22)$$

где  $D$  — матрица со степенями узлов на диагонали.

*Методы случайного блуждания.* Другой набор глобальных мер подобия рассматривает случайные блуждания, а не точное количество путей по графу. Можно привести пример известного алгоритма, персонализированного PageRank, где определяется стохастическая матрица  $P = AD^{-1}$  и вычислить (2.23):

$$g_u = cPg_u + (1 - c)e_u \quad (2.23)$$

В этом уравнении  $e_u$  является однократным индикаторным вектором для узла  $u$ , а  $g_u[v]$  дает стационарную вероятность того, что случайное блуждание, начавшееся в узле  $u$ , посетит узел  $v$ . Таким образом, определяется вероятность того, что случайное блуждание возобновится в узле  $u$  в каждый момент временного шага. Без этой вероятности перезапуска вероятности случайных блужданий просто сходились бы к нормализованному варианту центральности собственного вектора. Вероятность перезапуска определяет меру важности, специфичную для узла  $u$ , поскольку случайные блуждания постоянно «телепортируются» обратно в этот узел. Решение этой повторяемости дается формулой (2.24).

$$g_u = (1 - c)(I - cP)^{-1}e_u \quad (2.24)$$

Тогда можно определить меру сходства случайных блужданий между узлами как (2.25)

$$S_{RW}[u, v] = g_u[v] + g_v[u] \quad (2.25)$$

т. е. сходство между парой узлов пропорционально тому, насколько вероятно, будет достигнут каждый узел в результате случайного блуждания, начиная с другого узла.

Также можно после окончания процесса блуждания, можно выделить вершины, которые посещались наибольшее количество раз.

### **2.3. Графовые лапласианы и спектральные методы**

Рассмотрев традиционные подходы к классификации данных графа, а также традиционные подходы к предсказанию отношений, мы теперь обратимся к проблеме обучения кластеризации узлов в графе. Эта тема также мотивирует задачу изучения низких размерных вложений узлов.

### Графы лапласианы.

Матрицы смежности могут представлять графы без потери информации. Однако существуют альтернативные матричные представления графиков, обладающие полезными математическими свойствами. Эти матричные представления называются лапласианами и образованы различными преобразованиями матрицы смежности.

*Ненормализованный лапласиан.* Матрица Лапласа – это ненормированный лапласиан, определяемый следующим образом (2.26):

$$L = D - A \quad (2.26)$$

где  $A$  – матрица смежности, а  $D$  – матрица степеней.

Если выбран простой граф  $G$ , то  $A$  (матрица смежности) содержит только единицы или 0, а все его диагональные элементы равны 0. При этом в матрице степеней  $D$  диагональ заполнена значениями степенями вершин. Рассмотрим простой пример помеченного неориентированного графа и его матрицы Лапласа (рис. 2.10) [15].

Лапласиан матрица простого графа обладает рядом важных свойств:

1. Лапласиан симметричен ( $L^T = L$ ) и положительно полуопределен ( $x^T L x \geq 0, \forall x \in \mathbb{R}^{|V|}$ ). Это видно из того факта, что лапласиан является симметричным и диагонально доминирующим.

2. Выполняется следующее векторное тождество  $\forall x \in \mathbb{R}^{|V|}$  (2.27, 2.28)

$$x^T L x = \frac{1}{2} \sum_{u \in V} \sum_{v \in V} A[u, v] (X[u] - X[v])^2 \quad (27)$$

$$= \sum_{(u,v) \in E} (X[u] - X[v])^2 \quad (28)$$

3. Лапласиан  $L$  имеет  $|V|$  неотрицательных собственных значений:  $0 = \lambda_{|V|} \leq \lambda_{|V|-1} \leq \dots \leq \lambda_1$ .

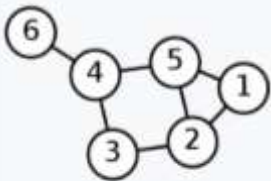
Помеченный граф	Матрица степеней	Матрица смежности	Матрица Лапласа
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

Рис. 2.10. Пример расчета матрицы Лапласа



*Лапласиан и компоненты связности.* Лапласиан суммирует многие важные свойства графа. Например, у нас есть следующая теорема:

**Теорема.** Геометрическая кратность нулевого собственного значения лапласиана  $L$  соответствует числу компонент связности в графе.

**Доказательство.** Это можно увидеть, что для любого собственного вектора  $e$  собственного значения 0 имеем, согласно (2.29) [15]

$$e^T L e = 0 \quad (2.29)$$

определения уравнения собственного значения для собственного вектора. И результат в уравнении (2.29) подразумевает, что (2.30)

$$\sum_{(u,v) \in E} (e[u] - e[v])^2 = 0 \quad (2.30)$$

Приведенное равенство (30) означает, что  $e[u] = e[v], \forall (u, v) \in E$ , что, в свою очередь, означает, что  $e[u]$  — одна и та же константа для всех узлов  $u$ , находящихся в одной компоненте связности. Таким образом, если граф полносвязный, то собственный вектор для собственного значения 0 будет постоянным вектором, состоящим из единиц для всех узлов графа, и это будет единственный собственный вектор, так как в этом случае существует только одно единственное решение к уравнению (2.29).

И наоборот, если граф состоит из нескольких компонентов связности, то уравнение 29 выполняется независимо для каждого блока лапласиана, соответствующего каждому компоненту связности. То есть, если граф состоит из  $K$  компонентов связности, то существует такой порядок узлов в графе, что матрица Лапласа может быть записана как (2.31)

$$L = \begin{bmatrix} L_1 & & \\ & L_2 & \\ & & \ddots & L_K \end{bmatrix} \quad (2.31)$$

где каждый из  $L_K$  блоков в этой матрице является действительным графом Лапласиана полностью связного подграфа исходного графа.

Поскольку они являются действительными лапласианами полносвязных графов, для каждого из блоков  $L_K$  будем иметь, что выполняется уравнение (2.29) и что каждый из этих подлапласианов имеет собственное значение 0 с кратностью

1 и собственный вектор из всех единиц (определенный только над узлами в этом компоненте). Более того, поскольку  $L$  — блочно-диагональная матрица, ее спектр задается объединением спектров всех  $L_K$  блоков, т. е. собственные значения  $L$  — это объединение собственных значений  $L_K$ -матриц, а собственные векторы  $L$  являются объединением собственных векторов всех матриц  $L_K$  со значениями 0, заполненными на позициях других блоков. Таким образом, каждый блок вносит один собственный вектор для собственного значения 0, и этот собственный вектор является вектором-индикатором для узлов в этой компоненте связности.

*Нормализованные лапласианы.* В дополнение к ненормализованному лапласиану есть также два популярных нормализованных варианта лапласиана. Симметричный нормированный лапласиан определяется как (2.32)

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \quad (2.32)$$

в то время как лапласиан случайного блуждания определяется как (2.33) [15]

$$L_{RW} = D^{-1} L \quad (2.33)$$

Обе эти матрицы имеют сходные свойства с лапласианом, но их алгебраические свойства отличаются малыми константами из-за нормализации. Например, теорема о компонентах связности верна точно для  $L_{RW}$ . Для  $L_{sym}$  теорема верна, но с собственными векторами для собственного значения 0, масштабированными на  $D^{\frac{1}{2}}$ . Эти разные варианты лапласиана могут быть полезны для разных задач анализа и обучения.

### ***Разрезы графа и кластеризация***

В теореме о компонентах связности графа видно, что собственные векторы, соответствующие нулевому собственному значению лапласиана, могут использоваться для назначения узлов кластерам в зависимости от того, к какому компоненту связности они принадлежат. Однако этот подход позволяет только кластеризовать узлы, которые уже находятся в отключенных компонентах, что тривиально. Рассмотрим, что лапласиан можно использовать для получения оптимальной кластеризации узлов в полносвязном графе.

*Графические разрезы.* Чтобы реализовать лапласовский подход к спектральной кластеризации, необходимо сначала определить, что подразумевается под оптимальным кластером. Для этого определим понятие разреза на графе. Пусть  $A \subset V$  обозначает подмножество узлов графа, а  $\bar{A}$  обозначает дополнение

этого множества, т. е.  $A \cup \bar{A} = V, A \cap \bar{A} = \emptyset$ . Учитывая разбиение графа на  $K$  непесекающихся подмножеств  $A_1, \dots, A_K$  определяется значение сечения как (2.34) [7]

$$cut(A_1, \dots, A_K) = \frac{1}{2} \sum_{k=1}^K |(u, v) \in E : u \in A_k, v \in \bar{A}_k| \quad (2.34)$$

Другими словами, разрез – это просто подсчет того, сколько ребер пересекает границу между разделами узлов. Теперь одним из вариантов определения оптимальной кластеризации узлов  $K$  кластеров будет выбор раздела, минимизирующего значение отсечения. Существуют эффективные алгоритмы для решения этой задачи, но известная проблема этого подхода заключается в том, что он имеет тенденцию просто создавать кластеры, состоящие из одного узла/

Таким образом, вместо того, чтобы просто свести к минимуму разрез, обычно стремятся минимизировать разрез, одновременно добиваясь, чтобы все разделы были достаточно большими. Одним из самых популярных способов является минимизация коэффициента разреза (*RatioCut*) (2.35) :

$$RatioCut(A_1, \dots, A_K) = \frac{1}{2} \sum_{k=1}^K \frac{|(u,v) \in E : u \in A_k, v \in \bar{A}_k|}{|A_k|} \quad (2.35)$$

что показывает способность выбора малых размеров кластера. Другое популярное решение — минимизировать нормализованный разрез (*Normalized Cut (NCut)*) (2.36) [7]:

$$NCut(A_1, \dots, A_K) = \frac{1}{2} \sum_{k=1}^K \frac{|(u,v) \in E : u \in A_k, v \in \bar{A}_k|}{vol(A_k)} \quad (2.36)$$

где  $vol(A) = \sum_{u \in A} d_u$ . Решение NCut обеспечивает, чтобы все кластеры имели одинаковое количество ребер, связанных с их узлами.

*Аппроксимация коэффициента разреза лапласовским спектром.* Рассмотрим подход к поиску назначения кластера, который минимизирует RatioCut (коэффициента разреза), используя лапласовский спектр. (Аналогичный подход можно использовать и для минимизации значения нормализованного разреза NCut.) Для простоты возьмем случай, когда  $K = 2$  и разделим наши узлы на два кластера. Наша цель состоит в том, чтобы решить следующую оптимизационную задачу (2.37)

$$\min_{A \in V} \text{RatioCut}(A, \bar{A}) \quad (2.37)$$

Чтобы переписать эту задачу в более удобной форме, определим следующий вектор  $a \in \mathbb{R}^{|V|}$  (2.38) [15]:

$$a[u] = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}} & \text{if } u \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}} & \text{if } u \in \bar{A} \end{cases} \quad (38)$$

Комбинируя этот вектор со свойствами лапласиана графа, можем видеть, что (2.39 – 2.44)

$$a^T L a = \sum_{(u,v) \in E} (a[u] - a[v])^2 \quad (39)$$

$$= \sum_{(u,v) \in E: u \in A, v \in \bar{A}} (a[u] - a[v])^2 \quad (40)$$

$$= \sum_{(u,v) \in E: u \in A, v \in \bar{A}} \left( \sqrt{\frac{|\bar{A}|}{|A|}} - \left( -\sqrt{\frac{|A|}{|\bar{A}|}} \right) \right)^2 \quad (41)$$

$$= \text{cut}(A, \bar{A}) \left( \frac{|A|}{|\bar{A}|} + \frac{|\bar{A}|}{|A|} + 2 \right) \quad (42)$$

$$= \text{cut}(A, \bar{A}) \left( \frac{|A| + |\bar{A}|}{|\bar{A}|} + \frac{|A| + |\bar{A}|}{|A|} \right) \quad (43)$$

$$= |V| \text{RatioCut}(A, \bar{A}) \quad (44)$$

Таким образом, видно, что вектор  $a$  позволяет записать *Ratio Cut* в терминах лапласиана (с точностью до постоянного множителя). Кроме того,  $a$  обладает двумя другими важными свойствами:

Свойство 1 (2.45)

$$\sum_{u \in V} a[u] = 0, \text{ что эквивалентно } a \perp \mathbf{1} \quad (2.45)$$

Свойство 2 (2.46)

$$\|a\|^2 = |V| \quad (2.46)$$

где  $\mathbf{1}$  — вектор всех единиц.

Собирая все это вместе, можно переписать задачу минимизации *RatioCut* в уравнении (2.37) как (2.47)

$$\min_{A \in V} a^T L a \quad (2.47)$$

Однако, это NP-трудная задача, так как ограничение, определяемое как в уравнении 38, требует, оптимизации дискретного набора. Очевидное ослабление состоит в том, чтобы удалить это условие дискретности и упростить минимизацию для векторов с действительным знаком (2.48) [16]:

$$\min_{a \in \mathbb{R}^{|V|}} a^T L a \quad (2.48)$$

По теореме Рэлея-Ритца решение этой задачи оптимизации дается вторым наименьшим по величине собственным вектором  $L$  (поскольку наименьший собственный вектор равен  $\mathbf{1}$ ).

Таким образом, можно аппроксимировать минимизацию *RatioCut*, установив  $a$  вторым наименьшим собственным вектором лапласиана. Чтобы превратить этот вектор с действительным знаком в набор дискретных назначений кластеров, можно просто назначать узлы кластерам на основе знака  $a[u]$ , т. е. (2.49) [16].

$$\begin{cases} u \in A & \text{if } a[u] \geq 0 \\ u \in \bar{A} & \text{if } a[u] < 0 \end{cases} \quad (2.49)$$

Таким образом, второй наименьший собственный вектор лапласиана является непрерывным приближением к дискретному вектору, который дает оптимальное назначение кластера (относительно *RatioCut*). Аналогичный результат можно показать для аппроксимации значения *NCut*, но он основан на втором наименьшем собственном векторе нормализованного лапласиана  $L_{RW}$ .

### **Обобщенная спектральная кластеризация**

Таким образом, спектр лапласиана позволил нам найти осмысленное разбиение графа на два кластера. В частности, , второй наименьший собственный вектор можно использовать для разделения узлов на разные кластеры. Эту общую идею можно распространить на произвольное число  $K$  кластеров, исследуя  $K$  наименьших собственных векторов лапласиана. Шаги этого общего подхода следующие:

1. Найти  $K$  наименьших собственных векторов  $L$  (исключая наименьшие):  $e_{|V|-1}, e_{|V|-2}, \dots, e_{|V|-K}$ .

2. Сформировать матрицу  $U \in \mathbb{R}^{|V| \times (K-1)}$  из собственных векторов из шага 1 в качестве столбцов.

3. Представить каждый узел соответствующей ему строкой в матрице  $U$ , т.е.  $z_u = U[u] \forall u \in V$ .

4. Запустить кластеризацию  $K$ -средних на вложениях  $z_u \forall u \in V$ .

Этот подход может быть адаптирован для использования нормализованного лапласиана, и результат аппроксимации для  $K = 2$ , а также может быть обобщен на этот случай  $K > 2$ .

Общий принцип спектральной кластеризации является мощным. Можно представить узлы в графе, используя спектр лапласиана графа, и это представление может быть мотивировано как принципиальное приближение к оптимальной кластеризации графа. Также существуют тесные теоретические связи между спектральной кластеризацией и случайными блужданиями на графах, а также в области обработки графовых сигналов.