Экспериментальное сравнение некоторых алгоритмов трикластеризации

Гнатышак Д. В., Игнатов Д. И., Жуков Л. Е., Кузнецов С. О., Миркин Б. Г.
dignatov@hse.ru
НИУ ВШЭ, Москва, Россия

В статье приводится экспериментальное сравнение пяти алгоритмов трикластеризации на реальных и синтетических данных по ресурсной эффективности, 4 мерам качества.

Experimental comparison of some triclustering algorithms

Gnatyshak D. V., Ignatov D. I., Zhukov L. E., Kuznetsov S. O., Mirkin B. G.
NRU HSE, Moscow, Russia

In this paper we show the results of the experimental comparison of five triclustering algorithms on real-world and synthetic data by resource efficiency and 4 quality measures. We also discuss the results' interpretation.

В настоящее время всё большее внимание в анализе данных уделяется триадическим данным, частным примером которых служат фолксономии, структуры из трёх множеств: пользователи, объекты и теги. Методы трикластеризации позволяют одновременно выделять 3-х компонентные группы в 3-х рассматриваемых множествах. После этого полученные трикластеры могут быть использованы для поиска сообществ, построения рекомендательных систем и т. д. Как следствие, имеется необходимость выявить преимущества и недостатки различных методов трикластеризации для выбора наиболее оптимальных из них в зависимости от задачи. В работе проводится сравнение следующих методов: трикластеризация объект-признак-условие (2 вида) [2], метод TriBox [5], метод спектральной трикластеризации [8] и метод TRIAS [3].

Модели и методы трикластеризации

Трикластеризация объект-признак-условие (OAC-трикластеризация) опирается непосредственно на аппарат анализа формальных понятий. В данной работе исследуются две её разновидности: OAC-трикластеризация, основанная на бокс-операторах [2], и OAC-трикластеризация, основанная на штрих-операторах, предложенная в данной работе. Трикластеры порождаются по одному (стратегия полного перебора).

Для начала определим алгоритм трикластеризаци объект-признак-условие на основе бокс-операторов. Пусть дан триадический контекст $\mathbb{K} = (G, M, B, I)$, где G, M, B — множества, а $I \subseteq G \times M \times B$ — тернарное отношение. Для фиксированной тройки $(\tilde{g}, \tilde{m}, \tilde{b}) \in I$ определим бокс-операторы:

$$\tilde{g}^{\square} := \{ g \,|\, (g, m) \in \tilde{b}' \vee (g, b) \in \tilde{m}' \} \quad (1)$$

$$\tilde{m}^{\square} := \{ m \,|\, (g, m) \in \tilde{b}' \vee (m, b) \in \tilde{g}' \} \quad (2)$$

$$\tilde{b}^{\square} := \{ b \,|\, (g, b) \in \tilde{m}' \vee (m, b) \in \tilde{g}' \} \quad (3)$$

где $(\cdot)'$ — штрих-оператор для триадического случая [4].

Назовём *OAC-трикластером на основе бокс-операторов*, построенным на тройке $(g, m, b) \in I$, тройку множеств $T = (g^{\square}, m^{\square}, b^{\square})$. Его компоненты будем называть, по аналогии с трипонятием, *объёмом*, *содержанием* и *модусом*.

Также необходимо определить плотность трикластера: $\rho(X, Y, Z) = \frac{|I \cap (X \times Y \times Z)|}{|X||Y||Z|}$.

Метод последовательно перебирает все тройки контекста и для каждой из них получает трикластер, который добавляется в общее множество трикластеров. Для отслеживания повторного порождения трикластера рекомендуется использование хеш-значений, что даёт значительный выигрыш по времени. Возможно также задание порога на минимальную плотность трикластера.

Теперь перейдём к трикластеризации объект-признак-условие на основе штрих-операторов. Данный метод использует несколько иную схему построения трикластеров, являющуюся, по сути, расширением на триадический случай метода, описанного в [7]. Пусть имеется триадический контекст $\mathbb{K} = (G, M, B, I)$. Назовём *OAC-трикластером, основанным на штрих-операторах*, построенным на тройке $(g, m, b) \in I$, тройку множеств $T = ((m, b)', (g, b)', (g, m)')$, где

$$(g, m)' = \{ b \,|\, (g, m, b) \in I \} \quad (4)$$

$$(g, b)' = \{ m \,|\, (g, m, b) \in I \} \quad (5)$$

$$(m, b)' = \{ g \,|\, (g, m, b) \in I \} \quad (6)$$

Псевдокод алгоритма данного вида трикластеризации приведён ниже.

Метод TriBox [5] использует оптимизационный подход для построения трикластеров, вносящих наибольший вклад в контекст. Алгоритм перебирает все тройки и пытается дополнять их таким образом, чтобы покрыть как можно большую часть контекста, сохранив при этом высокую плотность. Следовательно, он использует жадный подход и порождает трикластеры по одному.

Алгоритм 1. Алгоритм OAC-трикластеризации, основанной на штрих-операторах.

Вход: $\mathbb{K} = (G, M, B, I)$ — триконтекст;
ρ_{min} — порог плотности
Выход: $TSet = \{(X, Y, Z)\}$
1: **для всех** (g, m): $g \in G, m \in M$
2: $PrOA[g, m] = (g, m)'$
3: **для всех** (g, b): $g \in G, b \in B$
4: $PrOC[g, b] = (g, b)'$
5: **для всех** (m, b): $m \in M, b \in B$
6: $PrAC[m, b] = (m, b)'$
7: **для всех** $(g, m, b) \in I$
8: $T = (PrAC[m, b], PrOC[g, b], PrOA[g, m])$
9: $Tkey = hash(T)$
10: **если** $Tkey \notin Tset.keys \wedge \rho(T) \geqslant \rho_{min}$ **то**
11: $Tset[Tkey] = T$

В процессе перебора троек метод определяет для каждого элемента множеств объектов, признаков и условий насколько полезно будет добавить или удалить данный элемент из трикластера. Это продолжается до тех пор, пока добавление/удаление элемента не будет давать отрицательное значение меры полезности.

Метод спектральной трикластеризации, рассмотренный в [8], использует методы линейной алгебры для последовательного разбиения триадического контекста на несколько подмножеств троек. Итоговые подконтексты выдаются в качестве трикластеров. Таким образом, данный метод является дивизивным по стратегии поиска и порождает множество трикластеров одновременно.

В первую очередь для поиска трикластеров методом спектральной трикластеризации необходимо представить триконтекст в виде трёхдольного графа. Затем для него строится матрица Лапласа:

$$L_{ij} = \begin{cases} deg(v_i), & \text{если } i = j \\ -1, & \text{если } i \neq j \text{ и } \exists \text{ ребро } (v_i, v_j) \\ 0, & \text{иначе} \end{cases} \quad (7)$$

Второй наименьший собственный вектор v_2 этой матрицы является оптимальным решением непрерывного аналога задачи наилучшего разбиения данного графа. Остаётся лишь округлить компоненты собственного вектора до ± 1, чтобы получить вектор наилучшего разбиения. Также, для того, чтобы избежать отсечения висячих вершин, рекомендуется использовать собственный вектор для обобщённой задачи: $Lv = \lambda Dv$, где D — матрица со степенями вершин на главной диагонали [6]. Для данного метода возможно использование различных мер размера трикластеров $S(X, Y, Z)$ для определения условия остановки разбиений текущей ветви.

Таблица 1. Зашумлённые контексты.

Контекст	Число троек	Плотность
$p = 0$	3000	0.1111
$p = 0{,}1$	5069,6	0,1873
$p = 0{,}2$	7169,4	0,2645
$p = 0{,}3$	9290,2	0,3440
$p = 0{,}4$	11412,8	0,4222
$p = 0{,}5$	13533,4	0,5032

Алгоритм TRIAS, описанный в [3] является методом поиска триадических формальных понятий. Впрочем, формальные понятия можно рассматривать как абсолютно плотные трикластеры.

TRIAS основан на алгоритме NextClosure, который находит все формальные понятия диадического контекста, перебирая их в лексикографическом порядке. Он расширяет данный алгоритм на триадический случай, а также добавляет условия минимальной поддержки, т.е. формальные понятия со слишком малым объёмом, содержанием и/или модусом будут отбрасываться.

Машинные эксперименты

Исследование шумоустойчивости. Для проверки алгоритмов на шумоустойчивость было создано 6 контекстов. Они представляют собой зашумлённые контексты 30×30×30, у которых изначально на главной диагонали были построены кубоиды из троек 10×10×10. Зашумление создавалось с помощью инверсии, т.е. если в исходном контексте тройка присутствовала, то с заданной вероятностью p в зашумлённом контексте её не было, и наоборот. Для исходного контекста было создано 5 наборов из 5 зашумлённых контекстов с вероятностями ошибки от 0.1 до 0.5 с шагом 0.1 (последний можно назвать равномерным контекстом). Таблица 1 содержит среднее число троек и плотность для данных наборов контекстов.

Шумоустойчивость — его способность построить трикластеры максимально похожие на изначальные кубоиды. Максимально похожий трикластер t для данного кубоида c и его значение сходства определялись с помощью меры сходства, а общая мера качества находилась следующим образом (C — количество кубоидов):

$$sim(c) = \frac{1}{C} \sum_{c=c_1}^{c_C} \max_{t=t_1,\ldots,t_T} \frac{|G_c \cap G_t|}{|G_c \cup G_t|} \times \frac{|M_c \cap M_t|}{|M_c \cup M_t|} \frac{|B_c \cap B_t|}{|B_c \cup B_t|} \quad (8)$$

Для данных экспериментов в качестве меры S для спектральной трикластеризации для исходного контекста $\mathbb{K} = (G, M, B, I)$ использовалась следую-

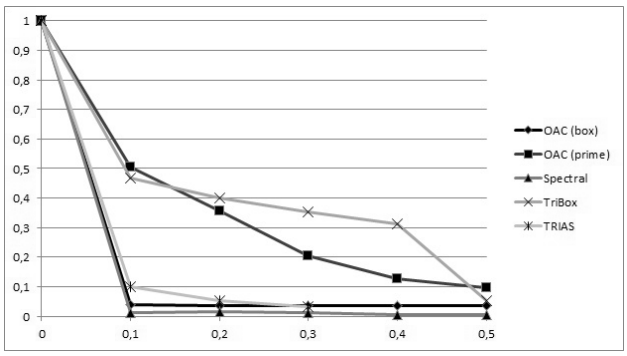

Рис. 1. Сходство для экспериментов на зашумлённых данных в зависимости от вероятности ошибки

щая:

$$S(X,Y,Z) = \frac{|X|+|Y|+|Z|}{|G|+|M|+|B|} \quad (9)$$

s_{min} было выбрано равным 0,34, чтобы алгоритм не пытался дробить дальше найденные трикластеры нужного размера.

Все тесты проводились на компьютере с процессором Intel Core i7-2600 3,4 ГГц и 8 ГБ оперативной памяти. Для их проведения была создана специальная платформа, в которой были реализованы все вышеперечисленные методы. Также для некоторых методов были реализованы эффективные параллельные версии алгоритмов.

На рисунке 1 приведены результаты экспериментов. Как видно, в незашумлённых контекстах все алгоритмы успешно справляются с обнаружением исходных кубоидов. С ростом погрешности результаты ухудшаются. Стоит отметить высокую шумоустойчивость метода TriBox. Также неплохую шумоустойчивость показал метод OAC-трикластеризации, основанный на штрих-операторах, однако он выдавал значительно больше трикластеров. Остальные же методы показали плохие результаты. И для контекста с вероятностью шума 0,5, как и следовало ожидать, никаких правдоподобных трикластеров не получено.

Время, количество, плотность и разнообразие. Данные эксперименты проводились на следующих контекстах (таблица 2):

1) равномерный контекст с вероятностью наличия тройки 0,1;
2) Top 250 лучших фильмов с ресурса `www.imdb.com`, объекты — названия фильмов, признаки — теги, условия — жанры;
3) случайная выборка 3000 троек из первых ста тысяч троек ресурса www.bibsonomy.org, объекты — пользователи, признаки — теги, условия — названия книг;

В качестве дополнительных мер качества, помимо времени, количества (N) и плотности (ρ) использовались покрытие и разнообразие для множеств всех троек (D), объектов (D_G), признаков (D_M) и условий (D_B). Введем функцию *intersect* от двух трикластеров:

Таблица 2. Контексты для экспериментов на время, количество, плотность и разнообразие.

Контекст	$\lvert G\rvert$	$\lvert M\rvert$	$\lvert B\rvert$	# троек	Плотность
Равномерный	30	30	30	2660	0,0985
IMDB	250	795	22	3818	0,00087
BibSonomy	51	924	2844	3000	2,2385e-005

$$intersect(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} 1, & G_{\mathcal{T}_i} \cap G_{\mathcal{T}_j} \neq \varnothing \wedge \\ & \wedge M_{\mathcal{T}_i} \cap M_{\mathcal{T}_j} \neq \varnothing \wedge \\ & \wedge B_{\mathcal{T}_i} \cap B_{\mathcal{T}_j} \neq \varnothing \\ 0, & \text{иначе} \end{cases} \quad (10)$$

А также $intersect_X$ для отдельных множеств объектов, признаков или условий.

Тогда мера разнообразия множества трикластеров $\mathcal{T}$ будет равна:

$$diversity(\mathcal{T}) = 1 - \frac{\sum_j \sum_{i<j} intersect(\mathcal{T}_i, \mathcal{T}_j)}{\frac{|\mathcal{T}|(|\mathcal{T}|-1)}{2}} \quad (11)$$

Основные результаты по сравнению алгоритмов представлены в таблице 3 (значения плотности, покрытия и разнообразия приведены в процентах). В приведённых результатах все методы работали с нулевыми значениями параметров. Также проводилось сравнение для прочих значений параметров, где также исследовалось значение меры покрытия полученными трикластерами множеств объектов, признаков и условий (в силу ограничений объёма данной статьи таблицы опущены).

Выводы

Как показали результаты, метод TRIAS работает дольше всех остальных наряду с методом TriBox и методом спектральной трикластеризации для больших контекстов. Он также выдаёт больше всего трипонятий малого размера, которые, несмотря на хорошую интерпретируемость, не могут чётко представлять структуру контекста. Покрытие в случае нулевого значения параметра, как и следовало ожидать, равно 1. Для больших значений минимальной поддержки его значение резко падает. Разнообразие, опять же, выше для нулевых порогов поддержки, но с их увеличением и, как следствие, уменьшением количества трикластеров, разнообразие также уменьшается.

Спектральная кластеризации показывает крайне хорошие результаты по времени только на небольших контекстах. Как было установлено, основная часть времени уходит на нахождение собственных

Таблица 3. Результаты экспериментов на время, количество, плотность, покрытие и разнообразие.

Метод	T, мс	N	ρ	C	D	D_G	D_M	D_B
	Равномерный случайный контекст							
OAC (□)	407	73	9,88	100	0	0	0	0
OAC (/)	312	2659	32,23	100	92,51	60,07	59,80	59,45
Спектр.	277	5	8,74	8,84	100	100	100	100
TriBox	6218	1011	74,00	96,02	97,42	66,25	79,53	84,80
TRIAS	29367	38356	100	100	99,99	99,93	4,07	3,51
	IMDB							
OAC (□)	2314	1500	1,84	100	15,65	9,67	0,70	7,87
OAC (/)	547	1274	53,85	100	96,55	94,56	92,14	28,52
Спектр.	98799	21	17,07	20,88	100	100	100	100
TriBox	197136	328	91,65	98,9	98,89	98,46	95,21	30,94
TRIAS	102554	1956	100	100	99,89	99,69	52,52	26,18
	BibSonomy							
OAC (□)	19297	398	4,16	100	79,59	67,28	42,83	79,54
OAC (/)	13556	1289	9466	100	99,74	88,58	99,51	99,53
Спектр.	5906563	2	50	100	100	100	100	100
TriBox	Время > 24 часов							
TRIAS	110554	1305	100	100	99,98	91,70	99,78	99,92

векторов первой матрицы Лапласа. Интерпретируемость метода достаточно хорошая, хотя средняя плотность полученных трикластеров при этом низка. Количество полученных трикластеров также невелико, что позволяет рекомендовать этот метод для случая, когда контекст необходимо разбить на небольшое количество подконтекстов. Также стоит отметить, что в силу дивизивной природы метода, разнообразие полученных трикластеров, как общее, так и по всем множествам, всегда равно 1, что влечет низкое значение на покрытия.

Метод TriBox даёт самые качественные трикластеры, но требует значительных вычислительных затрат. Полученные трикластеры обладают достаточно высокой плотностью и хорошо интерпретируются. Покрытие полученного множества и разнообразие в большинстве случаев достаточно высокие. Исключением является случай, когда одно из множеств значительно меньше других (например $|B| \ll |G|$): в этом случае разнообразие по данному множеству естественным образом будет низким. Использование параллельной версии алгоритма сокращает время работы примерно в 3-4 раза.

Метод трикластеризации объект-признак-условие, основанный на бокс-операторах показал не самые лучшие результаты. Хоть алгоритм по времени опережают лишь OAC-трикластеризация, основанная на штрих-операторах, и спектральная кластеризация для небольших контекстов, полученные трикластеры достаточно большие, обладают большими пересечениями, часто вкладываются друг в друга и имеют относительно низкую плотность. Это приводит к очень высокому покрытию контекста, но ценой этого является низкое разнообразие, порой даже очень близкое к нулю. Также, из-за большого размера трикластеров, результаты достаточно плохо интерпретируются. Зачастую, основу содержания и модуса составляют признаки и условия, отвечающие всего нескольким объектам. Параллельная версия алгоритма даёт примерно сокращение времени работы от 1,5 до 4 раз.

Наконец, метод OAC-трикластеризации, основанный на штрих-операторах, показал себя достаточно хорошо. По времени работы он уступает лишь спектральной трикластеризации для небольших контекстов. Несмотря на большое количество трикластеров, все они неплохо интерпретируются, лишь немного уступая по качеству методу TriBox. Исходя из природы метода, при нулевом пороге плотности покрытие равняется единице, но и для других порогов оно остаётся достаточно высоким. Разнообразие полученных трикластеров на маленьких порогах плотности невысокое, но оно значительно повышается для больших порогов, не достигая, впрочем, 1. Наконец, стоит отметить, что в связи с некоторой программной оптимизацией непараллельной версии алгоритма и его малой временной сложностью ($O(|G||M||B|)$), не очень эффективна его параллельная версия, выигрывающая по времени только для больших контекстов.

Литература

[1] *Ganter B., Wille R.* Formal concept analysis: Mathematical Foundations — Berlin-Heidelberg: Springer, 1999.

[2] *Ignatov D. I., Kuznetsov S. O., Magizov R. A., Zhukov L. E.* From Triconcepts to Triclusters // 13-th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC-2011), 2011. — Pp. 257-264.

[3] *Jaschke R., Hotho A., Schmitz C., Ganter B., Stumme G.* TRIAS - An Algorithm for Mining Iceberg Tri-Lattices // ICDM, 2006. — Pp.907-911.

[4] *Lehmann F., Wille R.* A triadic approach to formal concept analysis // Conceptual Structures: Applications, Implementation and Theory. Springer, 1995. — Pp.32-43.

[5] *Mirkin B., Kramarenko A.* Approximate Bicluster and Tricluster Boxes in the Analysis of Binary Data // 13-th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC-2011), 2011. — Pp. 248-256.

[6] *Zhukov L.,* Spectral Clustering of Large Advertiser Datasets // Technical report Overture R&D, 2004.

[7] *Игнатов Д. И., Каминская А. Ю., Кузнецов С., О., Магизов Р. А.* Метод бикластеризации на основе объектных и признаковых замыканий // Международная конференция «Интеллектуализация обработки информации» (ИОИ-8), 2010. — С.140–143.

[8] *Секинаева З., Игнатов Д.* Метод спектральной трикластеризации для систем совместного пользования ресурсами // Доклады всероссийской научно-практической конференции «Анализ Изображений, Сетей и Текстов» (АИСТ 2012), 2012. — С.246-254.