Разработка данных систем совместного пользования ресурсами: от трипонятий к трикластерам*

Игнатов Д. И., Кузнецов С. О., Пульманс Й.
dignatov@hse.ru
г. Москва, НИУ Высшая школа экономики

В работе предложен новый подход к трикластеризации трехмерных бинарных данных. Трикластер определен в терминах триадического анализа формальных понятий (Triadic Formal Concept Analysis) как плотное тримножество тернарного отношения Y между объектами, признаками и условиями. Такое определение является ослаблением определения трипонятия и дает возможность найти все трикластеры и трипонятия, содержащиеся в трикластерах больших наборов данных. Данный подход обобщает аналогичные исследования, проведенные нами для случая бикластеризации, основанной на формальных понятиях.

Термин «бикластер» был введен Б. Г. Миркиным в 1996 [15] и появление трикластеризации и n-кластеризации оставалось только делом времени. Сходный подход, называемый прямой кластеризацией (direct clustering), был предложен в начале 1970-х в работе Дж. Хертигана [10]. В анализе формальных понятий (АФП), предложенном в 1982 г. Р. Вилле [9, 17], используется частный случай бикластера, а именно формальное понятие. Триадический анализ формлаьных понятий (ТАФП) был разработан Леманом и Вилле [14] в 1995 г. как расширение АФП для случая трехмерных бинарных данных. Термины «формальные понятия» и «трипонятия» описывают полезные паттерны в бинарных данных, которые однородны и замкнуты (максимальны) в алгебраическом смысле. По причине жесткой структуры формальных понятий и вычислительной сложности порождающих их алгоритмов (экспоненциальной от размера входа) были предложены различные ослабления определения формального понятия для диадического случая (релевантные и плотные бимножества [3], методы факторизации на понятиях [1], плотные бикластеры [11]) и для триадического случая (триадическая факторизация на понятиях [2]). Существует несколько подходов для поиска только релевантных понятий, например «решетки-айсберги» и индексы устойчивости. Необходимость масштабируемых и эффективных алгоритмов трикластеразции очевидна в связи с возрастающей популярностью и размерами систем совместного пользования ресурсами (social resource tagging systems). Трехмерные данные вида «user-tag-resource», так называемые фолксономии, являются ключевой структурой данных в таких системах. TRIAS — один из хорошо известных алгоритмов для разработки данных фолксономий [12]. Есть также многообещающий подход для разработки данных n-арных отношений [5, 6]; его реализация (DataPeleer) основана на замкнутых множествах и превосходит аналогичные алгоритмы, такие как CubeMiner [13] для разработки данных замкнутых тримножеств. Некоторые исследователи пошли дальше и активно применяют замкнутые тримножества для разработки данных сложных признаковых зависимостей в трехмерных данных, например триадических импликаций [8].

Работа выполнена при финансовой поддержке РФФИ, проект №08-07-92497-НЦНИЛ_а.

Основные определения

Триадический контекст $\mathbb{K} = (G, M, B, Y)$ состоит из множеств G (*объектов*), M (*признаков*), и B (*условий*), а также тернарного отношения $Y \subseteq G \times M \times B$. Запись $(g, m, b) \in Y$ показывает, что объект g обладает признаком m при условии b.

Для удобства записи триадический контекст обозначают (X_1, X_2, X_3, Y). Триадический контекст $\mathbb{K} = (X_1, X_2, X_3, Y)$ порождает следующие диадические контексты: $\mathbb{K}^{(1)} = (X_1, X_2 \times X_3, Y^{(1)})$, $\mathbb{K}^{(2)} = (X_2, X_2 \times X_3, Y^{(2)})$, $\mathbb{K}^{(3)} = (X_3, X_2 \times X_3, Y^{(3)})$, где $gY^{(1)}(m, b) :\Leftrightarrow mY^{(1)}(g, b) :\Leftrightarrow :\Leftrightarrow bY^{(1)}(g, m) :\Leftrightarrow (g, m, b) \in Y$. *Операторы штрих* (операторы, формирующие понятия), порождаемые контекстом $\mathbb{K}^{(i)}$, обозначаются $(.)^{(i)}$. Для каждого порожденного диадического контекста мы имеем два типа таких операторов, т. е. для $\{i, j, k\} = \{1, 2, 3\}$ с $j < k$, $Z \subseteq X_i$ и $W \subseteq X_j \times X_k$ (i)-е операторы штрих определяются следующим образом: $Z \mapsto Z^{(i)} = \{(x_j, x_k) \in X_j \times X_k | x_i, x_j, x_k$ связаны Yдля всех $x_i \in Z\}$, $W \mapsto W^{(i)} = \{x_i \in X_i | x_i, x_j, x_k$ связаны Yдля всех $(x_j, x_k) \in W\}$. Формально *триадическое понятие* триадического контекста $\mathbb{K} = (X_1, X_2, X_3, Y)$ является тройкой вида (A_1, A_2, A_3), где $A_1 \subseteq X_1, A_2 \subseteq X_2, A_3 \subseteq X_3$ и для для любых $\{i, j, k\} = \{1, 2, 3\}$ при условии $j < k$ мы имеем $A_i^{(i)} = (A_j \times A_k)$. Для трипонятия (A_1, A_2, A_3) его компоненты A_1, A_2 и A_3 называются *объемом*, *содержанием* и *модусом* соответственно. Заметим, что интерпратация $\mathbb{K} = (X_1, X_2, X_3, Y)$ как трехмерной таблицы инциденций согласно нашему определению при подходящей перестановке строк, столбцов и слоев этой таблицы влечет интерпретацию триадического понятия (A_1, A_2, A_3) как максимального трехмерного параллелепипеда, заполненного крестиками. Множество всех триадических понятий контекста $\mathbb{K} =$

$= (X_1, X_2, X_3, Y)$ называется *трирешеткой понятий* и обозначается $\mathfrak{T}(X_1, X_2, X_3, Y)$.

Поиск плотных трикластеров

Операторы штрих и бокс-операторы одноэлементных множеств. Для упрощения обозначений мы используем $(.)'$ для всех операторов штрих, как это обычно и делается в АФП. Для наших целей мы рассмотрим триадический контекст $\mathbb{K} = (G, M, B, Y)$ и введем штрихи и *бокс операторы* для конкретных элементов множеств G, M, B соответственно. В дальнейшем мы будем писать g' вместо $\{g\}'$ для 1-множества $g \in G$ и аналогично для $m \in M$ и $b \in B$: m' and b'.

Операторы, формирующие понятия для 1-множеств:

$$\begin{aligned} m' &= \{\ (g,b)\ |(g,m,b) \in Y\}; \\ g' &= \{\ (m,b)\ |(g,m,b) \in Y\}; \\ b' &= \{\ (g,m)\ |(g,m,b) \in Y\}. \end{aligned}$$

Мы не используем операторы двойного штриха, а только введеные нами бокс-операторы:

$$\begin{aligned} g^{\square} &= \{\ g_i \mid (g_i, b_i) \in m' \text{ or } \ (g_i, m_i) \in b'\}; \\ m^{\square} &= \{\ m_i \mid (m_i, b_i) \in g' \text{ or } \ (g_i, m_i) \in b'\}; \\ b^{\square} &= \{\ b_i \mid (g_i, b_i) \in m' \text{ or } \ (m_i, b_i) \in g'\}. \end{aligned}$$

Пусть $\mathbb{K} = (G, M, B, Y)$ — триадический контекст. Для тройки $(g, m, b) \in Y$ назовем *трикластером* $T = (g^{\square}, m^{\square}, b^{\square})$.

Плотность трикластера (A, B, C) триадического контекста $\mathbb{K} = (G, M, B, Y)$ определяется как доля всех троек Y в трикластере, т. е. $\rho(A, B, C) = |I \cap A \times B \times C|/|A||B||C|$.

Трикластер $T = (A, B, C)$ называется *плотным*, если его плотность больше некоторого минимального порога, т. е. $\rho(T) \geqslant \rho_{\min}$. Для данного триконтекста $\mathbb{K} = (G, M, B, Y)$ мы обозначим $\mathbf{T}(G, M, B, Y)$ множество всех его (плотных) трикластеров.

Утверждение 1. *Для любого трипонятия (A, B, C) триадического контекста $\mathbb{K} = (G, M, B, Y)$ и непустых множеств A, B и C мы имеем $\rho(A, B, C) = 1$.*

Утверждение 2. *Для любого трикластера (A, B, C) триадического контекста $\mathbb{K} = (G, M, B, Y)$ и непустных множеств A, B и C мы имеем $0 \leqslant \rho(A, B, C) \leqslant 1$.*

Теорема 1. *Пусть $\mathbb{K} = (G, M, B, Y)$ будет триадическим контекстом и $\rho_{\min} = 0$. Для любого $T_c = (A_c, B_c, C_c) \in \mathfrak{T}(G, M, B, Y)$ существует трикластер $T = (A, B, C) \in \mathbf{T}(G, M, B, Y)$ такой, что $A_c \subseteq A, B_c \subseteq B, C_c \subseteq C$.*

Таблица 1. Фрагмент данных Bibsonomy

	t_1	t_2	t_3
u_1		×	×
u_2	×	×	×
u_3	×	×	×

r_1

	t_1	t_2	t_3
u_1	×	×	×
u_2	×		×
u_3	×	×	×

r_2

	t_1	t_2	t_3
u_1	×	×	×
u_2	×	×	×
u_3	×	×	

r_3

Пример 1. Для примера табл. 67 получим $3^3 = 27$ формальных понятий, 24 с $\rho = 1$ и 3 трипонятия с $\rho = 0$ (они имеют либо пустое множество пользователей, либо ресурсов или тегов). Хотя данные небольшие, мы получили 27 паттернов для анализа (максимальное количество формальных понятий для контекста размера $3 \times 3 \times 3$) — результат того, что мы имеем дело со степенным триадическим контекстом. Однако мы можем заключить, что пользователи u_1, u_2 и u_3 использовали почти одинаковые множества ресурсов и их пометки. Таким образом, они очень сходны в терминах (tag, resource) общих пар и целесообразно уменьшить число паттернов, описывающих данные, с 27 до 1. Трикластер $T = (\{u_1, u_2, u_3\}, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\})$ с $\rho = 0{,}89$ есть в точности искомый паттерн, а его плотность несколько меньше 1. Каждое из трипонятий триконтекста в $\mathfrak{T} = \{(\varnothing, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\}), (\{u_1\}, \{t_2, t_3\}, \{r_1, r_2, r_3\}), \ \ldots, \ (\{u_1, u_2, u_3\}, \{t_1, t_2\}, \{r_3\})\}$ содержится (в смысле покомпонентного вложения множеств) в T.

Мы предложили эвристику для вычисления $\rho(T)$, основанную на проверке только небольшого количества случайно выбранных троек, содержащихся в заданном трикластере T. Для трикластера $T = (A, B, C)$ мы провели оценку плотности $\hat{\rho}(T) = |P|/|N|$, где $P = \{(g, m, b) | (g, m, b) \in N \cap Y\}$, N — множество размера $|N|$ случайно выбранных элементов трикластера. Параметр $|N|$ может быть выбран относительно небольшим, скажем $0{,}1|A||B||C|$.

Реальные данные и эксперименты

В наших экспериментах мы анализировали свободно доступные данные популярной системы социальных закладок Bibsonomy [4]. Мы запускали алгоритм TRICL на части данных, состоящих из всех пользователей, ресурсов и присвоенных тегов для выявления сообщества пользователей, имеющих сходное поведение при тегировании.

Результирующая фолксономия (бибсономия) состит из $|U| = 2\ 337$ пользователей, $|T| = 67\ 464$ тегов и $|R| = 28\ 920$ ресурсов (закладок или bibtex описаний), которые связаны $|Y| = 816\ 197$ тройками. Отметим, что мы имеем дело с параллелепипедом, состоящим из 4 559 624 602 560 ячеек.

Алгоритм 1. Алгоритм поиска трикластеров TRICL

Вход: $K = (G, M, B, Y)$ – триконтекст;
$\rho_{\min}$ – порог плотности;
Выход: $\mathbf{T} = \{(A_k, B_k, C_k) | (A_k, B_k, C_k)$ – плотный трикластер$\}$.

1: **для всех** $(g, m, b) \in Y$
2: **если** g not in PrimesObj **то**
3: PrimesObj$[g] = g'$;
4: **если** m not in PrimesAttr **то**
5: PrimesAttr$[m] = m'$;
6: **если** b not in PrimesCond **то**
7: PrimesCond$[b] = b'$;
8: **если** g not in BoxesObj **то**
9: BoxesObj$[g] = g^{\square}$;
10: **если** m not in BoxesAttr **то**
11: BoxesAttr$[m] = m^{\square}$;
12: **если** b not in BoxesCond **то**
13: BoxesCond$[b] = b^{\square}$;
14: **для всех** $(g, m, b) \in Y$
15: $T = (\text{BoxesObj}[g], \text{BoxesAttr}[m], \text{BoxesCond}[b])$
16: Tkey $= \text{hash}(T)$
17: **если** Tkey not in **T то**
18: **если** $\rho(T) \geqslant \rho_{\min}$ **то**
19: **T**[Tkey] $= (T)$;

Мы исследовали статистическое распределение перед применением алгоритма TRICL. Мы вычислили и построили гистограммы для пользователей и числа пар (tag, document) (рис. 1), аналогичные гистограммы для тегов и числа пар (user, document) и для документов и их пар (user, tag). Мы обнаружили, что данные следуют степенному закону распределения $p(x) = Cx^{-\alpha}$ с $\alpha = 3,6778$ и дисперсией $\sigma = 0,0001$ в случае документов и количества пар (user, tag). Для пользователй и тегов мы получили $\alpha = 2{,}13$ и $\alpha = 1{,}8$ соответственно. Мы вычислили α, используя оценку максимального правдоподобия, как описано в [16], и проверили результаты с помощью программного обеспечения, описанного в [7].

Это наблюдение позволит нам использовать жадную стратегию поиска, если мы хотим искать большие и относительно плотные трикластеры, по той причине, что только небольшая часть пользователей совершает присваивание (tag, user) (аналогичные выводы для распределения тегов документов).

Мы измерили производительность нашей реализации (Python 2.7.1) на системе Pentium Core Duo с тактовой частотой процессора 2 ГГц и 2 Гб ОЗУ. Мы использовали реализацию алгоритма TRIAS на Java из работы [12] для построения всех трипонятий заданного контекста. Результаты экспериментов представлены в табл. 2. Два последних

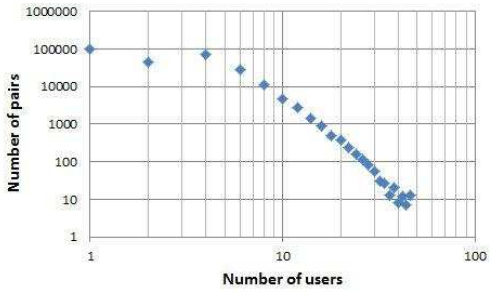

Рис. 1. Гистограмма количества пар (document, tag) для 800 000 записей Bibsonomy

Таблица 2. Экспериментальные результаты для k первых троек набора данных tas с $\rho_{\min} = 0$

k	$\lvert U\rvert$	$\lvert T\rvert$	$\lvert R\rvert$	$\lvert\mathfrak{T}\rvert$	$\lvert\mathbf{T}\rvert$
100	1	47	52	57	1
1000	1	248	482	368	1
10000	1	444	5193	733	1
100000	59	5823	28920	22804	4462
200000	340	14982	61568	—	19053

Trias, с	TriclEx, с	TriclProb, с
0,2	0,2	0,2
1	1	1
2	46,7	47
3386	10311	976
> 24 ч	> 24 ч	3417

Таблица 3. Распределение плотности трикластеров для 200 000 первых троек набора данных tas с $\rho_{\min} = 0$

Нижняя граница ρ	Верхняя граница ρ	Число трикластеров
0	0,05	18617
0,05	0,1	195
0,1	0,2	112
0,2	0,3	40
0,3	0,4	20
0,4	0,5	10
0,5	0,6	8
0,6	0,7	1
0,7	0,8	1
0,8	0,9	0
0,9	1	49

столбца показывают среднее время работы Tricl с полной (TriclEx) и вероятностной (TriclProb) стратегией вычислений.

В наших экспериментах оценка $\hat{\rho}$ имела абсолютную ошибку 0б13 для параметров $|N| = 1/10$, $\rho_{\min} = 0$ и 200 000 троек данных Бибсономии. Алгоритм работает значительно быстрее, чем Trias и TriclExб в случае нашей вероятностной стратегии.

В табл. 3 представлено распределение плотности трикластеров для 200 000 первых троек набора данных bibsonomy дано.

Заключение

Мы предложили АФП-подход к трикластеризации и показали, что:

— (плотная) трикластеризация — неплохая альтернатива для ТАФП, т. к. общее число трикластеров для некоторого набора данных значительно меньше числа формальных понятий;
— (плотная) трикластеризация способна справиться с большим количеством трипонятий в худшем случае триконтекстов (или плотных кубоидов в них), когда их главная диагональ пуста, и рассматривает такие кубоиды как целые трикластеры. Это очень релевантное свойство для исследования трисообществ в системах социальных закладок;
— предложенный алгоритм имеет хорошую масштабируемость на реальных данных, особенно когда используется подход жадного покрытия и оптимизированная версия процедуры вычисления плотности.

Мы продолжаем нашу работу над трикластеризацией в следующих направлениях:

— исследование различных, основанных на ограничениях подходах к трикластеризации (например, анализ данных плотных трикластеров и затем частых множеств тримножеств в них);
— поиск лучших стратегий оценки плотности трикластеров;
— разработка обобщенной теоретической формализации трикластеризации на основе замкнутых множеств;
— учет природы реальных данных с целью оптимизации алгоритмов (разреженность данных, распределение значений и т. д.).

Литература

[1] *Belohlavek, R., Vychodil, V.* Factor analysis of incidence data via novel decomposition of matrices // In: Ferre, S., Rudolph, S. (eds.) ICFCA. Lecture Notes in Computer Science, vol. 5548, pp. 83–97. Springer (2009).

[2] *Belohlavek, R., Vychodil, V.* Factorizing three-way binary data with triadic formal concepts // In: Setchi, R., Jordanov, I., Howlett, R., Jain, L. (eds.) Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 6276, pp. 471–480. Springer Berlin / Heidelberg (2010).

[3] *Besson, J., Robardet, C., Boulicaut, J.F.* Mining a new fault-tolerant pattern type as an alternative to formal concept discovery // In: Scharfe, H., Hitzler, P., Ohrstrom, P. (eds.) Conceptual Structures: Inspiration and Application, Lecture Notes in Computer Science, vol. 4068, pp. 144–157. Springer Berlin / Heidelberg (2006).

[4] `bibsonomy.org` — Сервис библиографических закладок.

[5] *Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.* Data peeler: Contraint-based closed pattern mining in n-ary relations // In: SDM. pp. 37–48. SIAM (2008).

[6] *Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.* Closed patterns meet -ary relations TKDD 3(1) (2009).

[7] *Clauset, A., Shalizi, C.R., Newman, M.E.J.* Power-law distributions in empirical data // SIAM Review 51(4), 661–703 (2009).

[8] *Ganter, B., Obiedkov, S.* Implications in triadic formal contexts // In: Wolff, K., Pfeiffer, H., Delugach, H. (eds.) Conceptual Structures at Work, Lecture Notes in Computer Science, vol. 3127, pp. 237–237. Springer Berlin / Heidelberg (2004).

[9] *Ganter, B., Wille, R.* Formal concept analysis: Mathematical foundations // Springer, Berlin-Heidelberg (1999).

[10] *Hartigan, J.A.* Direct clustering of a data matrix. Journal of the American Statistical Association // 67(337), 123–129 (March 1972).

[11] *Ignatov, D.I., Kaminskaya, A.Y., Kuznetsov, S.O., Magizov, R.A.* A concept-based biclustering algorithm // In: Proceedings of the Eighth International conference on Intelligent Information Processing (IIP-8), pp. 140–143. MAKS Press (2010), in Russian.

[12] *Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.* Trias — an algorithm for mining iceberg tri-lattices // In: ICDM, pp. 907–911. IEEE Computer Society (2006).

[13] *Ji, L., Tan, K.L., Tung, A.K.H.* Mining frequent closed cubes in 3D datasets // In: Dayal, U., Whang, K.Y., Lomet, D.B., Alonso, G., Lohman, G.M., Kersten, M.L., Cha, S.K., Kim, Y.K. (eds.) VLDB, pp. 811–822. ACM (2006).

[14] *Lehmann, F., Wille, R.* A triadic approach to formal concept analysis // In: Ellis, G., Levinson, R., Rich, W., Sowa, J. (eds.) Conceptual Structures: Applications, Implementation and Theory, Lecture Notes in Computer Science, vol. 954, pp. 32–43. Springer Berlin / Heidelberg (1995).

[15] *Mirkin, B.* Mathematical Classification and Clustering. — Kluwer, 1996.

[16] *Newman, M.E.J.* Power laws, pareto distributions and Zipf's law // Contemporary physics 46(5), 323–351 (2005).

[17] *Wille, R.* Restructuring lattice theory: An approach based on hierarchies of concepts // In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Boston (1982).