# From Triconcepts to Triclusters

Dmitry I. Ignatov, Sergei O. Kuznetsov,
Ruslan A. Magizov, and Leonid E. Zhukov

National Research University Higher School of Economics, Moscow, Russia
dignatov@hse.ru

**Abstract.** A novel approach to triclustering of a three-way binary data is proposed. Tricluster is defined in terms of Triadic Formal Concept Analysis as a dense triset of a binary relation $Y$, describing relationship between objects, attributes and conditions. This definition is a relaxation of a triconcept notion and makes it possible to find all triclusters and triconcepts contained in triclusters of large datasets. This approach generalizes the similar study of concept-based biclustering.

**Keywords:** formal concept analysis, data mining, triclustering, three-way data, folksonomy.

## 1 Introduction

The term biclustering was coined by B.Mirkin in 1996 [15] and the appearance of triclustering and n-clustering was only a matter of time. The similar approach, called direct clustering, was proposed in early 70s by Hartigan [10]. In the Formal Concept Analysis method, introduced in 1982 by R. Wille [17,9], a particular kind of bicluster was used for analysis of binary data, notably formal concept. The Triadic (Formal) Concept Analysis (TCA) was introduced by Lehman and Wille [14] in 1995 as an extension of dyadic Formal Concept Analysis for the case of three-way binary data. The notions of formal concepts and triconcepts describe a useful pattern in binary data that is homogeneous and closed (maximal) in algebraic sense. Due to the rigid structure of formal concepts and computational complexity of processing algorithms (exponential w.r.t. size of input data), some relaxations of the formal concept notion were introduced for dyadic (relevant and dense bisets [3], concept factorization techniques [1], dense biclusters [11]) and triadic cases (triadic concept factors [2]). There are also exist several techniques for reduction of the number of formal concepts to only relevant ones, for example, iceberg lattices and stability indices mining. The need for scalable and efficient triclustering algorithms became clear with the growth of popularity and sizes of social resource tagging systems. The three-way data "user-tag-resource", so called folksonomy, is a core data structure in such systems. One of the known algorithms for mining folksonomies is the TRIAS algorithm [12]. There is also a promising approach to mine n-ary relational data [5,6]; its implementation (DataPeleer) is based on closed sets and outperforms another similar algorithm CubeMiner [13] for mining closed

S.O. Kuznetsov et al. (Eds.): RSFDGrC 2011, LNAI 6743, pp. 257–264, 2011.

trisets. Some researchers go further and actively apply closed trisets for mining complex attribute dependencies in three-way binary data, for example, triadic implications [8].

The paper is organized as follows. In the next section, we introduce main definitions of Triadic Concept Analysis. In the section 3 we define dense triclusters and describe their properties, present the algorithm and evaluate its complexity, discuss some heuristics, and also provide the reader with examples. Section 4 presents the description of the bibsonomy datasets and computer experiments with them. Section 5 concludes the paper.

## 2 Main Definitions

A triadic context $\mathbb{K} = (G, M, B, Y)$ consists of sets $G$ (objects), $M$ (attributes), and $B$ (conditions), and ternary relation $Y \subseteq G \times M \times B$. An incidence $(g, m, b) \in Y$ shows that the object $g$ has the attribute $m$ under condition $b$.

For convenience, a triadic context is denoted by $(X_1, X_2, X_3, Y)$. A triadic context $\mathbb{K} = (X_1, X_2, X_3, Y)$ gives rise to the following diadic contexts $\mathbb{K}^{(1)} = (X_1, X_2 \times X_3, Y^{(1)})$, $\mathbb{K}^{(2)} = (X_2, X_2 \times X_3, Y^{(2)})$, $\mathbb{K}^{(3)} = (X_3, X_2 \times X_3, Y^{(3)})$, where $gY^{(1)}(m, b) :\Leftrightarrow mY^{(1)}(g, b) :\Leftrightarrow bY^{(1)}(g, m) :\Leftrightarrow (g, m, b) \in Y$. The derivation operators (primes or concept-forming operators) induced by $\mathbb{K}^{(i)}$ are denoted by $(.)^{(i)}$. For each induced dyadic context we have two kinds of such derivation operators. That is, for $\{i, j, k\} = \{1, 2, 3\}$ with $j < k$ and for $Z \subseteq X_i$ and $W \subseteq X_j \times X_k$, the $(i)$-derivation operators are defined by:

$$Z \mapsto Z^{(i)} = \{(x_j, x_k) \in X_j \times X_k | x_i, x_j, x_k \text{ are related by Y for all } x_i \in Z\},$$

$$W \mapsto W^{(i)} = \{x_i \in X_i | x_i, x_j, x_k \text{ are related by Y for all } (x_j, x_k) \in W\}.$$

Formally, a triadic concept of a triadic context $\mathbb{K} = (X_1, X_2, X_3, Y)$ is a triple $(A_1, A_2, A_3)$ of $A_1 \subseteq X_1, A_2 \subseteq X_2, A_3 \subseteq X_3$, such that for every $\{i, j, k\} = \{1, 2, 3\}$ with $j < k$ we have $A_i^{(i)} = (A_j \times A_k)$. For a triadic concept $(A_1, A_2, A_3)$, the components $A_1$, $A_2$, and $A_3$ are called the extent, the intent, and the modus of $(A_1, A_2, A_3)$. It is important to note that for interpretation of $\mathbb{K} = (X_1, X_2, X_3, Y)$ as a three-dimensional cross table, according to our definition, under suitable permutations of rows, columns, and layers of the cross table, the triadic concept $(A_1, A_2, A_3)$ is interpreted as a maximal cuboid full of crosses. The set of all triadic concepts of $\mathbb{K} = (X_1, X_2, X_3, Y)$ is called the concept trilattice and is denoted by $\mathfrak{T}(X_1, X_2, X_3, Y)$.

## 3 Mining Dense Triclusters

### 3.1 Prime, Double Prime and Box Operators of 1-Sets

To simplify the notation, we denote by $(.)'$ all prime operators, as it is usually done in FCA. For our purposes consider a triadic context $\mathbb{K} = (G, M, B, Y)$ and

**Table 1.** Concept-forming operators of 1-sets

| Prime operators of 1-sets | Their double prime counterparts |
|---|---|
| $m' = \{ (g,b) \mid (g,m,b) \in Y\}$ | $m'' = \{ \tilde{m} \mid (g,b) \in m' \quad and \quad (g,\tilde{m},b) \in Y\}$ |
| $g' = \{ (m,b) \mid (g,m,b) \in Y\}$ | $g'' = \{ \tilde{g} \mid (m,b) \in g' \quad and \quad (\tilde{g},m,b) \in Y\}$ |
| $b' = \{ (g,m) \mid (g,m,b) \in Y\}$ | $b'' = \{ \widetilde{b} \mid (g,m) \in b' \quad and \quad (g,m,\tilde{b}) \in Y\}$ |

introduce primes, double primes and box operators for particular elements of $G$, $M$, $B$, respectively. In what follows we write $g'$ instead of $\{g\}'$ for 1-set $g \in G$ and similarly for $m \in M$ and $b \in B$: $m'$ and $b'$.

We do not use double primes, but box operators that we introduce below:

$$g^{\square} = \{ g_i \mid (g_i, b_i) \in m' \text{ or } (g_i, m_i) \in b'\}$$
$$m^{\square} = \{ m_i \mid (m_i, b_i) \in g' \text{ or } (g_i, m_i) \in b'\}$$
$$b^{\square} = \{ b_i \mid (g_i, b_i) \in m' \text{ or } (m_i, b_i) \in g'\}$$

Let $\mathbb{K} = (G, M, B, Y)$ be a triadic context. For a triple $(g, m, b) \in Y$, the triple $T = (g^{\square}, m^{\square}, b^{\square})$ is called a tricluster.

The density of a tricluster $(A, B, C)$ of a triadic context $\mathbb{K} = (G, M, B, Y)$ is given by the fraction of all triples of $Y$ in the tricluster, that is $\rho(A, B, C) = |I \cap A \times B \times C|/|A||B||C|$.

The tricluster $T = (A, B, C)$ is called dense if its density is greater than a predefined minimal threshold, i.e. $\rho(T) \geq \rho_{min}$. For a given triadic context $\mathbb{K} = (G, M, B, Y)$ we denote by $\mathbf{T}(G, M, B, Y)$ the set of all its (dense) triclusters.

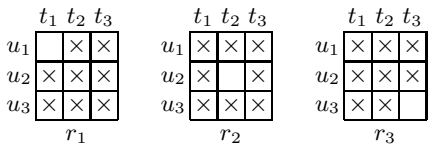
*Property 1.* For every triconcept $(A, B, C)$ of a triadic context $\mathbb{K} = (G, M, B, Y)$ with nonempty sets $A$, $B$, and $C$ we have $\rho(A, B, C) = 1$.

*Property 2.* For every triclucter $(A, B, C)$ of a triadic context $\mathbb{K} = (G, M, B, Y)$ with nonempty sets $A$, $B$, and $C$ we have $0 \leq \rho(A, B, C) \leq 1$.

**Proposition 1.** *Let $\mathbb{K} = (G, M, B, Y)$ be a triadic context and $\rho_{min} = 0$. For every $T_c = (A_c, B_c, C_c) \in \mathfrak{T}(G, M, B, Y)$ there exits a tricluster $T = (A, B, C) \in \mathbf{T}(G, M, B, Y)$ such that $A_c \subseteq A, B_c \subseteq B, C_c \subseteq C$.*

**Example 1.** Consider $3^3 = 27$ formal triconcepts, 24 with $\rho = 1$ and 3 void triconcepts with $\rho = 0$ (there are empty sets of either users, resources or tags). Although this data is small, we have 27 patterns to analyze (maximal number of triconcepts for the context size $3 \times 3 \times 3$ ); this is because the data is the power set triadic context. We can conclude that users $u_1$, $u_2$, and $u_3$ share almost the same sets of tags and resources. So, they are very similar in terms $(tag, resource)$ of shared pairs and it is convenient to reduce the number of patterns describing these data from 27 to 1. The tricluster $T =$

**Table 2.** A small example with Bibsonomy data




$(\{u_1, u_2, u_3\}, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\})$ with $\rho = 0.89$ is exactly such a reduced pattern, but its density is slightly less than 1. Each of triconcepts in $\mathfrak{T} = \{(\emptyset, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\}), (\{u_1\}, \{t_2, t_3\}, \{r_1, r_2, r_3\})...(\{u_1, u_2, u_3\}, \{t_1, t_2\}, \{r_3\})\}$ is contained, w.r.t. componentwise set inclusion, in $T$.

### 3.2 An Algorithm: TRICL

The idea is obvious: For all $(g, m, w) \in I$ with $\rho(g^{\square}, m^{\square}, w^{\square}) \geq \rho_{\min}$ the algorithm stores $T = (g^{\square}, m^{\square}, w^{\square})$ in $\mathbf{T}$. In the pseudo-code of TRICL (Alg. 1) we provide computational details rather than simple algebraic description. It allows us to better evaluate the complexity of the main algorithm's steps and gives ideas on code implementation. The complexity of the first loop (steps 2-10) is $O(|I|)$. The main loop complexity (steps 15-19) is trickier: $O(|Y||\mathbf{T}||\log(|\mathbf{T}|)|G||M||B|)$ or, since we know that $|\mathbf{T}| \leq |Y|$, it is $O(|Y|^2||\log(|Y|)|G||M||B|)$. The factor $|G||M||B|$ appears due to the computation of a tricluster $\rho$ density, this value is indeed hard to compute for large triclusters.

We propose a heuristic to compute $\rho(T)$, based on checking only some amount of randomly selected triples contained in the given tricluster $T$. For a tricluster $T = (A, B, C)$ we perform the density estimation $\hat{\rho}(T) = |P|/|N|$, where $P = \{(g, m, b)|(g, m, b) \in N \cap Y\}$, $N$ is a set of $|N|$ randomly chosen elements of the tricluster. The parameter $|N|$ can be chosen relatively small, say $\frac{1}{10} \cdot |A||B||C|$.

## 4 Real Data and Experiments

In our experiments we analyzed the freely available data from the popular social bookmarking system bibsonomy [4]. We ran the TRICL algorithm on a part of the data consisting of all users, resources, tags and tag assignments to detect communities of users that have similar tagging behavior.

We used only tas file which is actually the list of tuples (tag assignments): who attached which tag to which resource/content.

- 1. user (number, no user names available)
- 2. tag
- 3. content_id (matches bookmark.content_id or bibtex.content_id)
- 4. content_type (1 = bookmark, 2 = bibtex)
- 5. date

For our purposes we need only fields 1, 2, and 3 of the tuple.

**Algorithm 1.** TRICL

**Data**: $K = (G, M, B, Y)$ – formal context, $\rho_{\min}$ – density threshold
**Result**: $\mathbf{T} = \{(A_k, B_k, C_k) | (A_k, B_k, C_k)$ – dense tricluster$\}$
**begin**
  **for** $(g, m, b) \in Y$ **do**
    **if** *g not in PrimesObj* **then**
      $PrimesObj[g] = g'$
    **if** *m not in PrimesAttr* **then**
      $PrimesAttr[m] = m'$
    **if** *b not in PrimesCond* **then**
      $PrimesCond[b] = b'$
    **if** *g not in BoxesObj* **then**
      $BoxesObj[g] = g^{\square}$
    **if** *m not in BoxesAttr* **then**
      $BoxesAttr[m] = m^{\square}$
    **if** *b not in BoxeesCond* **then**
      $BoxesCond[b] = b^{\square}$
  **for** $(g, m, b) \in Y$ **do**
    $Tkey = hash((BoxesObj[g], BoxesAttr[m], BoxesCond[b]))$;
    **if** *Tkey not in* **T** **then**
      **if** $\rho(BoxesObj[g], BoxesAttr[m], BoxesCond[b]) \geq \rho_{\min}$ **then**
        $\mathbf{T}[Tkey] = ((BoxesObj[g], BoxesAttr[m], BoxesCond[b]))$

The resulting folksonomy (bibsonomy) consists of $|U| = 2\,337$ users, $|T| = 67\,464$ tags and $|R| = 28\,920$ resources (bookmarks and bibtex's entries), that are linked by $|Y| = 816,197$ triples. We want to note that we have to deal here with a cuboid consisting of 4 559 624 602 560 cells.

We have also investigated the statistical distribution of the data before using TRICL algorithm. We calculated and plotted histogram for users and the number of (tag, document) assignment pairs, and similar histograms for tags and their number of (user, document) pairs, and for documents and their number of (user, tag) pairs. We found that the data follows Power Law distribution $p(x) = Cx^{-\alpha}$ with $\alpha = 3,6778$ and variance $\sigma = 0,0001$ in the case of documents vs number of (user, tag) assignments. For user and tag data we obtained $\alpha = 2,13$ and $\alpha = 1,8$ respectively. We computed $\alpha$ using ML estimator as described in [16] and verified the results by software mentioned in [7].

This introspection can afford us to use greedy approach to our data if we want to mine large and (relatively) dense triclusters, due to even not so big part of users have the most portion of (tag, user) assignments (similar conclusions for tags and documents distributions).

We measured the run-time performance of our implementation (in Python 2.7.1) on a Pentium Core Duo system with 2 GHz and 2 GB RAM. We used

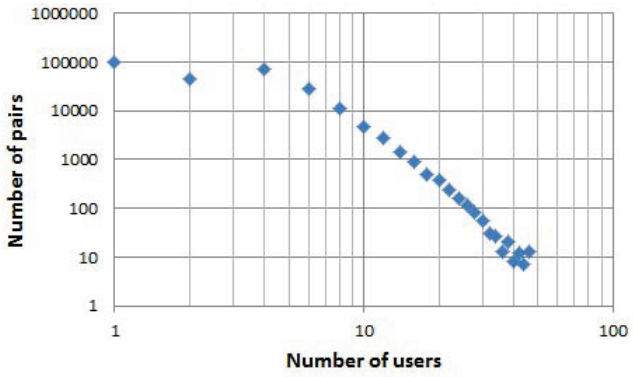



**Fig. 1.** Histogram of numbers of pairs (document, tag) for all triples of bibsonomy data

**Table 3.** Experimental results for $k$ first triples of tas dataset with $\rho_{\min} = 0$

| $k$, number of first triples | $\lvert U\rvert$ | $\lvert T\rvert$ | $\lvert R\rvert$ | $\lvert\mathfrak{T}\rvert$ | $\lvert\mathbf{T}\rvert$ | Trias, s | TriclEx,s | TriclProb,s |
|---|---|---|---|---|---|---|---|---|
| 100 | 1 | 47 | 52 | 57 | 1 | 0.2 | 0.2 | 0.2 |
| 1000 | 1 | 248 | 482 | 368 | 1 | 1 | 1 | 1 |
| 10000 | 1 | 444 | 5193 | 733 | 1 | 2 | 46,7 | 47 |
| 100000 | 59 | 5823 | 28920 | 22804 | 4462 | 3386 | 10311 | 976 |
| 200000 | 340 | 14982 | 61568 | - | 19053 | $>$ 24 h | $>$ 24h | 3417 |

Java implementation of Trias algorithm by R. Jäschke [12] to build all triconcepts of a certain context. The results of the experiments are presented in Table 3. The last two columns shows the mean execution time of Tricl with full and probabilistic density calculation strategies respectively.

**Table 4.** Density of triclusters distribution for 200 000 first triples of tas dataset with $\rho_{\min} = 0$

| low bound of $\rho$ | upper bound of $\rho$ | number of triclusters |
|---|---|---|
| 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 |

In our experiments the $\hat{\rho}$ estimate has only 0.13 mean absolute error for a tricluster size $|N| = 1/10$ , $\rho_{\min} = 0$, and 200 000 first triples of the bibsonomy data. The algorithm becomes drastically faster than Trias and TriclEx in the case of our probabilistic computational strategy.

Density distribution of triclusters for 200 000 first triples of bibsonomy dataset is given in the Table 4.

## 5 Conclusion

We proposed an FCA-based approach to triclustering. We showed that:

- The (dense) triclustering is a good alternative for TCA since the total number of triclusers in real data is significantly less than the number of triconcepts.
- The (dense) triclustering is able to cope with a large number of triconcepts. In the worst cases of tricontexts (or dense cuboids in them) only their main diagonal is empty and considers such cuboids as a whole tricluster. This is very relevant property for mining tricommunities in social bookmarking systems.
- The proposed algorithm has good scalability on real-world data especially when used with greedy covering approach and optimized version of the density calculation procedure.

We will continue our work on triclustering in several directions:

- Investigate mixing of several constraint-based approaches to triclustering (e.g., mining dense triclusters first and then frequent trisets in them).
- Search for better approaches to tricluster's density estimation.
- Develop a unified theoretical framework for triclustering based on closed sets.
- Take into account the nature of real-world data for optimization (data sparsity, distribution of values, etc.).

**Acknowledgements.** This work was partially supported by the Russian Foundation for Basic Research, project No. 08-07-92497-NTSNIL_a. We would like to thank our colleagues Sergei Obiedkov (NRU HSE), Mykola Pechenizsky (TU-Eindhoven) and Alena Pliskina (NRU HSE) for their suggestions and support.

## References

1. Belohlavek, R., Vychodil, V.: Factor analysis of incidence data via novel decomposition of matrices. In: Ferré, S., Rudolph, S. (eds.) ICFCA 2009. LNCS, vol. 5548, pp. 83–97. Springer, Heidelberg (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.) KES 2010. LNCS, vol. 6276, pp. 471–480. Springer, 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: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 144–157. Springer, Heidelberg (2006)
4. http://bibsonomy.org
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, Philadelphia (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.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 186–195. Springer, Heidelberg (2004)
9. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations. Springer, Heidelberg (1999)
10. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
11. Ignatov, D.I., Kaminskaya, A.Y., Kuznetsov, S.O., Magizov, R.A.: A concept-based biclustering algorithm. In: Proceedings of the Eight 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, Los Alamitos (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, New York (2006)
14. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Ellis, G., Levinson, R., Rich, W., Sowa, J. (eds.) ICCS 1995. LNCS, vol. 954, pp. 32–43. Springer, Heidelberg (1995)
15. Mirkin, B.: Mathematical Classification and Clustering. Kluwer, Dordrecht (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, Boston, pp. 445–470 (1982)