From Triadic FCA to Triclustering: Experimental Comparison of Some Triclustering Algorithms

Dmitry V. Gnatyshak, Dmitry I. Ignatov, and Sergei O. Kuznetsov

National Research University Higher School of Economics
dignatov@hse.ru
http://www.hse.ru

Abstract. In this paper we show the results of the experimental comparison of five triclustering algorithms on real-world and synthetic data wrt. resource efficiency and 4 quality measures. One of the algorithms, the OAC-triclustering based on prime operators, is presented first time in this paper. Interpretation of results for real-world datasets is provided.

Keywords: formal concept analysis, triclustering, triadic data, data mining

1 Introduction

Recently analysis of triadic data attracts more and more attention in Data Mining community [1,2,3,4,5]. One particular example is mining of so-called folksonomies, structures composed by three sets: users, objects and tags. One of the first attempts of such an analysis[1] was done within the framework of Formal Concept Analysis (FCA) [6], namely in Triadic Formal Concept Analysis (TCA) [7]. Triclustering methods, which are objects of our study, allow one to simultaneously discover three-component homogeneous groups in the considered three-sets. For example, these triclusters can be used for community detection [8] and recommender algorithms [9]. As a consequence, we would like to investigate peculiarities and reveal advantages and drawbacks of various triclustering approaches to choose the optimal one depending on the task.

In this paper we compare the following triclustering methods: object-attribute-condition triclustering (2 modifications including new one) [10,11], TriBox [2], spectral triclustering [11] and Trias algorithm [1]. Even though there are some recent efficient [12] and even more general algorithms than Trias [13], it is well-known to the FCA community and sometimes outperforms its competitors [12]. Moreover, the aim of the paper is not in comparison of different algorithms for triadic formal concepts generation, but it is rather in comparison of the original patterns with their various approximations (triclusters) in terms of the introduced quality measures. We also suggest investigating different matrix decomposition approaches, e.g. Boolean ones [3,4], in a further study; note that

Boolean matrix factorization can be considered as an approach to the reduction of the number of the resulting triconcepts.

The rest of the paper is organised as follows. In section 2 we give main definitions and describe the triclustering methods selected for the comparison. Section 3 describes all the experiments and their results on real and synthetic data along with specially introduced quality measures. Section 4 concludes the paper and indicates some further research direction.

2 Triclustering models and methods

Object-attribute-condition triclustering (OAC-triclustering) is based on Formal Concept Analysis [6] and its triadic extension, Triadic Formal Concept Analysis (TCA) [7]. In this paper we consider 2 types of OAC-triclustering: box operator based (box OAC-triclustering) [10], and prime operator based (prime OAC-triclustering). The latter is introduced in this paper. The set of triclusters is generated one by one and complete enumeration strategy is used.

First, we recall some basic notions of Formal Concept Analysis (FCA) [6]. Let G and M be sets, called the set of objects and attributes, respectively, and let I be a relation $I \subseteq G \times M$: for $g \in G$, $m \in M$, gIm holds iff the object g has the attribute m. The triple $\mathbb{K} = (G, M, I)$ is called a *(formal) context*. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then the *Galois connection* is given by the following *derivation operators*:

$$\begin{aligned} A' &= \{m \in M \mid gIm \text{ for all } g \in A\}, \\ B' &= \{g \in G \mid gIm \text{ for all } m \in B\}. \end{aligned} \tag{1}$$

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *(formal) concept (of the context K)* with *extent* A and *intent* B (in this case we have also $A'' = A$ and $B'' = B$).

The concepts, ordered by $(A_1, B_1) \geq (A_2, B_2) \iff A_1 \supseteq A_2$ form a complete lattice, called *the concept lattice* $\underline{\mathfrak{B}}(G, M, I)$.

2.1 Object-attribute-condition triclustering

Here let us define the box operators and describe **box OAC-triclustering**. We use a slightly different introduction of the main TCA notions because of their further technical usage.

Let $\mathbb{K} = (G, M, B, I)$ be a triadic context, where G, M, and B are sets, and I is a ternary relation: $I \subseteq G \times M \times B$. In addition to set of objects, G, and set of attributes, M, we have B, a set of conditions.

Derivation (prime) operators for a triple $(\widetilde{g}, \widetilde{m}, \widetilde{b}) \in I$ from triadic context $\mathbb{K}$ can be defined as follows:

$$\widetilde{g}' := \{\, (m, b) \mid (\widetilde{g}, m, b) \in I\} \tag{2}$$

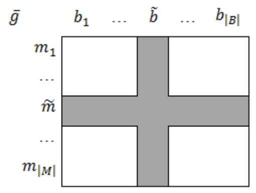

Fig. 1. $\overline{g}$ addition condition

$$(\widetilde{g},\widetilde{m})' := \{\, b \mid (\widetilde{g},\widetilde{m},b) \in I\} \tag{3}$$

$\widetilde{m}'$, $\widetilde{b}'$, $(\widetilde{g},\widetilde{b})'$, $(\widetilde{m},\widetilde{b})'$ prime operators can be defined the same way.

Now for a triple $(\widetilde{g},\widetilde{m},\widetilde{b}) \in I$ let us define box operator $\widetilde{g}^{\square}$ ($\widetilde{m}^{\square}$ and $\widetilde{b}^{\square}$ are introduced in the same way):

$$\widetilde{g}^{\square} := \{\, g \mid \exists m (g,m) \in \widetilde{b}' \vee \exists b (g,b) \in \widetilde{m}'\} \tag{4}$$

Definition 1. *Suppose* $\mathbb{K} = (G,M,B,I)$ *is a triadic context. For a triple* $(g,m,b) \in I$ *a triple* $T = (g^{\square}, m^{\square}, b^{\square})$ *is called a* box operator based OAC-tricluster*. Traditionally, its components are respectively called* extent, intent, and modus.

The density of a tricluster $T = (X,Y,Z)$ is defined as the fraction of all triples of I in $X \times Y \times Z$:

$$\rho(T) := \frac{|I \cap (X \times Y \times Z)|}{|X||Y||Z|} \tag{5}$$

Definition 2. *The tricluster* T *is called* dense *iff its density is not less than some predefined threshold, i.e.* $\rho(T) \geq \rho_{min}$.

Let us elaborate on the structure of box operator based triclusters. Suppose $\mathbb{K} = (G,M,B,I)$ is a triadic context, and the triple $(\widetilde{g},\widetilde{m},\widetilde{b}) \in I$ is considered. Then object $\overline{g}$ will be added to $\widetilde{g}^{\square}$ iff $\{(\overline{g},\widetilde{m},b) \mid b \in B \wedge (\overline{g},\widetilde{m},b) \in I\} \neq \emptyset \vee \{(\overline{g},m,\widetilde{b}) \mid m \in M \wedge (\overline{g},m,\widetilde{b}) \in I\} \neq \emptyset$. It is clear that this condition is equivalent to the one in eq. (4), and can be easily illustrated (Fig. 1): if at list one of the elements from "grey" cells is an element of I, then $\overline{g}$ is added to $\widetilde{g}^{\square}$.

The idea of box OAC-triclustering is to enumerate all triples of the ternary relation I for a context $\mathbb{K}$ generating a box operator based tricluster for each. If generated tricluster T was not added to the set of all triclusters $\mathcal{T}$ on previous steps, then T is added to $\mathcal{T}$. It is possible to implement hash-fuctions for triclusters in order to significantly optimize computational time by simplifying the comparison of triclusters. Also a minimal density threshold can be used.

Prime OAC-triclustering extends the biclustering method from [14] to the triadic case. It uses prime operators (eq. 3) to generate triclusters.

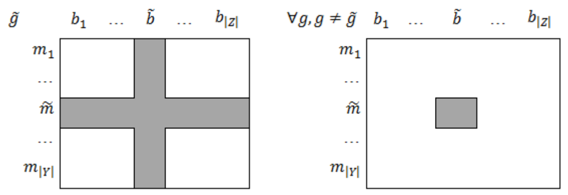
$\tilde{g}$
b_1 … $\tilde{b}$ … $b_{|Z|}$
m_1 … $\tilde{m}$ … $m_{|Y|}$
$\forall g, g \neq \tilde{g}$
b_1 … $\tilde{b}$ … $b_{|Z|}$
m_1 … $\tilde{m}$ … $m_{|Y|}$

Fig. 2. Prime operator based tricluster structure

Definition 3. *Suppose* $\mathbb{K} = (G, M, B, I)$ *is a triadic context. For a triple* $(g, m, b) \in I$ *a triple* $T = ((m, b)', (g, b)', (g, m)')$ *is called a* prime operator based OAC-tricluster*. Its components are called respectively* extent, intent, and modus*.*

Prime based OAC-triclusters are more dense than box operator based ones. Their structure is illustrated on Fig. 2: every element corresponding to the "grey" cell is an element of I. Thus, prime operator based OAC-triclusters in a three-dimensional matrix form contain a cross-like structure of ones.

It may also be useful to implement hash functions for triclusters.

Algorithm 1 Algorithm for prime OAC-triclustering.

```
Input: $\mathbb{K} = (G, M, B, I)$ — tricontext;
    $\rho_{min}$ — density threshold
Output: $\mathcal{T} = \{T = (X, Y, Z)\}$
 1: $\mathcal{T} := \emptyset$
 2: for all $(g, m)$: $g \in G, m \in M$ do
 3:     $PrimesObjAttr[g, m] = (g, m)'$
 4: end for
 5: for all $(g, b)$: $g \in G, b \in B$ do
 6:     $PrimesObjCond[g, b] = (g, b)'$
 7: end for
 8: for all $(m, b)$: $m \in M, b \in B$ do
 9:     $PrimesAttrCond[m, b] = (m, b)'$
10: end for
11: for all $(g, m, b) \in I$ do
12:     $T = (PrimesAttrCond[m, b], PrimesObjCond[g, b], PrimesObjAttr[g, m])$
13:     $Tkey = hash(T)$
14:     if $Tkey \notin \mathcal{T}.keys \wedge \rho(T) \geq \rho_{min}$ then
15:         $\mathcal{T}[Tkey] := T$
16:     end if
17: end for
```

2.2 TriBox method

The TriBox method [2] implements an optimization approach in tricluster generation. Suppose $\mathbb{K} = (G, M, B, I)$ is a triadic context. The idea is to select some triple of I, take it for the initial tricluster, and then to modify its extent, intent, and modus so that they covered a significant part of the context while maintaining high density. Thus, TriBox aims at finding a set of triclusters $\mathcal{T} = \{T = (X, Y, Z)\}$ that maximize the criterion 6. The resulting triclusters compose locally optimal solution for the trade-off problem between the density and the volume of various possible triclusters.

$$f(T) = \rho(T)^2 |X||Y||Z| \tag{6}$$

Once again, hash-functions for triclusters may be used for optimizations.

2.3 Spectral triclustering method

Spectral triclustering method [11] is based on the graph partition problem. The idea is to represent the given triadic context as a tripartite graph and then recursively divide it. To find an optimal partitioning spectral clustering uses the second minimal eigenvector of the Laplacian matrix.

Let us elaborate on this technique. Suppose $\mathbb{K} = (G, M, B, I)$ is a triadic context First we need to transform $\mathbb{K}$ into tripartite graph $\Gamma = \langle V, E \rangle$. Since I is a *ternary* relation it is only possible to represent $\mathbb{K}$ as a tripartite hypergraph without the loss of information. The following transformation technique is considered: $V := G \sqcup M \sqcup B$, for each triple $(g, m, b) \in I$ edges (g, m), (g, b) and (m, b) are added to E to form an undirected non-weighted graph. As the result some additional triples will be added to I after inverse transformation. Still it is clear that these triples will be added only in „dense“ areas of I thus possibly filling missing values and “optimizing” tricontext for methods aiming at finding formal concepts. Thus this technique is acceptable for solving the problem.

After the transformation Laplacian matrix is built for Γ:

$$L_{ij} = \begin{cases} degree(v_i), & \text{if } i = j \\ -1, & \text{if } i \neq j \text{ and } \exists \text{ edge } (v_i, v_j) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where v_i is the i^{th} vertex of V.

The second minimal eigenvector of L is an optimal solution of the continuous variant of the optimal partition problem for Γ (finding the minimal set $\tilde{E} \subseteq E$ so that the graph $\tilde{\Gamma} = (V, E \setminus \tilde{E})$ is not connected). The sign of each component of this vector indicates one of the 2 new connected components. For convenience we find the optimal partition vector as the approximation of the obtained vector by setting its components to ± 1 values.

In order to avoid partitioning of dangling vertices or small subgraphs the generalized eigenvalue problem must be considered ([11], Appendix I):

$$Lv = \lambda D v \tag{8}$$

where D is a diagonal matrix containing vertices' degrees on the main diagonal.

Also, some minimum size constraint can be used to avoid too deep partitioning. Since spectral triclustering is not able to generate the same tricluster more than once, it is not necessary to use hash functions to speed up the calculations.

2.4 Trias method

Formally, TRIAS [1] is a method for finding triadic formal concepts that are closed 3-sets. Since triadic formal concepts can be interpreted as absolutely dense triclusters, this method was added to the study.

TRIAS is based on the NEXTCLOSURE algorithm that enumerates all formal concepts of the dyadic context in lexicographical order. In TRIAS this approach is extended to the triadic case and minimal support constraints are added (triclusters with too small extent, intent or modus are skipped).

As well as spectral triclustering, TRIAS is not able to generate the same tricluster more than once.

3 Experiments

3.1 Noise-tolerance.

In order to test noise-tolerance of the algorithms 26 triadic contexts have been generated. The initial context contains 30 objects, 30 attributes, 30 conditions, and 3 non-overlapping $10 \times 10 \times 10$ cuboids of ones on the main diagonal in its three-dimensional matrix form. Then this context has been sequentially noised by the inversions with the probability of an inversion of a triple varying from 0.1 to 0.5 with 0.1 interval (the latter context can be called uniform context, because probability of $(g, m, b) \in I$ is equal for every triple). There have been 5 such series of context. Table 1 contains the average number of triples and total density for these sets of contexts.

Table 1. Noised contexts.

Context	# triples	Density
$p = 0$	3000	0.1111
$p = 0.1$	5069.6	0.1873
$p = 0.2$	7169.4	0.2645
$p = 0.3$	9290.2	0.344$\underline{0}$
$p = 0.4$	11412.8	0.4222
$p = 0.5$	13533.4	0.5032

The noise tolerance of an algorithm has been defined as the ability to build triclusters similar to initial cuboids. We used the Jaccard similarity coefficient to find the most similar tricluster t for the given cuboid c and their similarity. Total similarity has been defined as follows (C is a number of cuboids):

$$sim(c) = \frac{1}{C} \sum_{c=c_1}^{c_C} \max_{t=t_1,\ldots,t_T} \frac{|G_c \cap G_t|}{|G_c \cup G_t|} \frac{|M_c \cap M_t|}{|M_c \cup M_t|} \frac{|B_c \cap B_t|}{|B_c \cup B_t|} \tag{9}$$

Following size measure for spectral triclustering has been chosen:

$$Size(X, Y, Z) = \frac{|X| + |Y| + |Z|}{|G| + |M| + |B|} \tag{10}$$

s_{min} has been set to 0.34 to stop at the triclusters of correct size.

All of the methods have been implemented by authors and incorporated in a single triclustering toolbox in order to make the comparison more accurate. The toolbox has been implemented in C# using MS Visual Studio 2010/2012. All the experiments have been performed on Windows 7 SP1 x64 system equipped with an Intel Core i7-2600 @ 3.40GHz processor and 8 GB of RAM. AlgLib[1] library was used for eigenvalue decomposition.

The results of the experiments are represented on Figure 3. It is clear that every method has managed to successfully find initial cuboids, but the results quickly deteriorate for most of methods with the growth of inversion probability. TriBox has shown the best results as it tries to optimize the density-volume trade-off (which most probably is the best for the areas of the former cuboids for small error probability). Though prime OAC-triclustering has been also rather noise-tolerant, it generated significantly more triclusters (most likely the high number of triclusters is the reason for these results). All the other methods have been unable to provide significant results for noisy contexts. Moreover, as it was expected, no adequate triclusters were generated by any of the methods for the contexts with inversion probability 0.5.

3.2 Time, quantity, coverage, density and diversity.

The experiments on the computation time, triclusters count, coverage, density, and diversity were conducted on the following contexts (Table 2):

1. Uniform context $(\forall (g, m, b)\, P((g, m, b) \in I) = 0.1)$
2. Top 250 movies context from `www.imdb.com`, objects — movie titles, attributes — tags, conditions — genres
3. Random sample of 3000 of the first 100000 triples of the `www.bibsonomy.org` dataset, objects — users, attributes — tags, conditions – bookmark names

Parallel versions of OAC-triclustering algorithms and TriBox have also been implemented via parallelization of their outer loops and the computation time for them has been compared. Coverage and diversity measures were introduced as additional quality measures. *Coverage* is defined simply as a fraction of the triples of the context (alternatively, objects, attributes or conditions) included

[1] `http://www.alglib.net/`

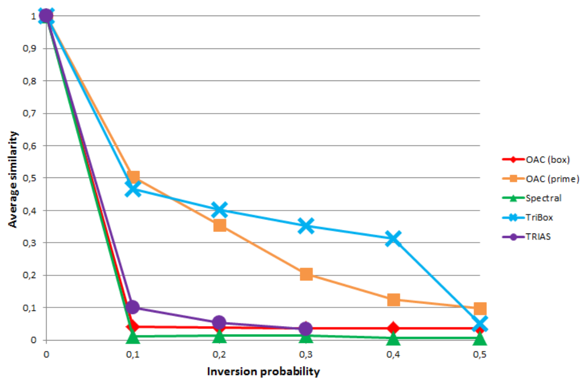

Fig. 3. Similarity for the noise-tolerance experiments

Table 2. Contexts for the experiments with 5 chosen evaluation measures.

Context	$\|G\|$	$\|M\|$	$\|B\|$	# triples	Density
Uniform	30	30	30	2660	0.0985
IMDB	250	795	22	3818	0.00087
BibSonomy	51	924	2844	3000	0.000022

in at least one of the triclusters of the resulting set. To define diversity we will use a binary function of 2 triclusters $intersect(\mathcal{T}_i, \mathcal{T}_j)$ that equals to 1 if both triclusters $\mathcal{T}_i$ and $\mathcal{T}_j$ have nonempty intersection of the sets of contained triples, and 0 otherwise.

It is also possible to define *intersect* for the sets of objects, attributes and conditions. For instance, $intersect_G(\mathcal{T}_i, \mathcal{T}_j)$ is equal to 1 if triclusters $\mathcal{T}_i$ and $\mathcal{T}_j$ have nonempty intersection of their extents, and 0 otherwise.

Now we can define diversity of the tricluster set $\mathcal{T}$:

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

The diversity for the sets of objects, attributes or conditions is similarly defined.

Table 3 contains the results for the experiments. The following values for parameters were selected:

1. OAC-triclustering: $\rho_{min} = 0$
2. SpecTric: $s_{min} = 0$
3. Trias: $\tau_G = \tau_M = \tau_B = 0$

Table 3. Results of the experiments on the computation time (t), tricusters count (n), density (ρ), coverage (Cov), and diversity (Div).

Algorithm	t, ms	t_{par}, ms	n	ρ_{av}, %	Cov, %	Div, %	Div_G, %	Div_M, %	Div_B, %
	Uniform random context								
OAC ($\Box$)	407	**196**	73	9.88	**100.00**	0.00	0.00	0.00	0.00
OAC ($\prime$)	**312**	877	2659	32.23	**100.00**	92.51	60.07	59.80	59.45
SpecTric	277	-	**5**	8.74	8.84	**100.00**	**100.00**	**100.00**	**100.00**
TriBox	6218	1722	1011	74.00	96.02	97.42	66.25	79.53	84.80
Trias	29367	-	38356	**100.00**	**100.00**	99.99	99.93	4.07	3.51
	IMDB								
OAC ($\Box$)	2314	**1573**	1500	1.84	**100.00**	15.65	9.67	0.70	7.87
OAC ($\prime$)	**547**	2376	1274	53.85	**100.00**	96.55	94.56	92.14	28.52
SpecTric	98799	-	**21**	17.07	20.88	**100.00**	**100.00**	**100.00**	**100.00**
TriBox	197136	55079	328	91.65	98.90	98.89	98.46	95.21	30.94
Trias	102554	-	1956	**100.00**	**100.00**	99.89	99.69	52.52	26.18
	BibSonomy								
OAC ($\Box$)	19297	**6803**	398	4.16	**100.00**	79.59	67.28	42.83	79.54
OAC ($\prime$)	**13556**	9400	1289	94.66	**100.00**	99.74	88.58	99.51	99.53
SpecTric	5906563	-	**2**	50	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
TriBox	$>$ 24 hrs								
Trias	110554	-	1305	**100.00**	**100.00**	99.98	91.70	99.78	99.92

3.3 Results

Trias is one of the most time consuming algorithms compared in the paper along with TriBox and SpecTric for large contexts. Though the resulting triclusters (triconcepts) can easily be interpreted their number and small size make it impossible to understand the general structure of the context. Since all triconcepts have been generated and thus every triple has been covered, the coverage is equal to 1. Since the concepts are small general diversity is rather high. Still, the set diversity depends on the size of the corresponding set: the smaller the set — the greater the chance of intersection and the lower the diversity.

Examples of the triconcepts for the IMDB context:

1. {The Princess Bride (1987), Pirates of the Caribbean: The Curse of the Black Pearl (2003)}, {Pirate}, {Fantasy, Adventure}
2. {Platoon (1986), Letters from Iwo Jima (2006)}, {Battle}, {Drama, War}
3. {V for Vendetta (2005)}, {Fascist, Terrorist, Government, Secret Police , Fight}, {Action, Sci-Fi, Thriller}

SpecTric has displayed rather good computation time only for small contexts. Most of this time is used for the eigenvalue decomposition of Laplacian matrix. Thus we intend to test some alternative linear algebra libraries in the toolbox and compare the results as well. The resulting triclusters can be reasonably interpreted, though their average density is low. Their small number makes this method good for dividing the context into several non-overlapping parts. Also the diversities for SpecTric are always equal to 1 because the method generates partitions of the initial context. High diversity though leads to low coverage.

Examples of the triclusters for the IMDB context:

1. $\rho = 23.08\%$, {Alien (1979), The Shining (1980), The Thing (1982), The Exorcist (1973)}, {Spaceship, Egg, Parasite, Creature, Caretaker, Colorado, Actress, Blood, Helicopter, Scientist, Priest, Washington D.C., Faith}, {Horror}
2. $\rho = 2.09\%$, {The Shawshank Redemption (1994), The Godfather (1972), The Godfather: Part II (1974), . . . , Bonnie and Clyde (1967), Arsenic and Old Lace (1944)}, {Prison, Cuba, Business, 1920s, . . . , Texas, Cellar}, {Crime, Thriller }

TriBox in this study generates the best triclusters. The only drawback of this method is high computation time, though the use of the parallel version of TriBox can significantly lower it for multi-core processors. Average density of the resulting triclusters is rather high, they have good interpretability. Coverage and diversities are also high in most of the cases. The only exception is set diversity in the situation when some of the sets are small, just as for Trias.

Examples of the triclusters for the IMDB context:

1. 100%, {Million Dollar Baby (2004), Rocky (1976), Raging Bull (1980)}, {Boxer, Boxing}, {Drama, Sport}
2. 83.33%, {The Sixth Sense (1999), The Exorcist (1973), The Silence of the Lambs (1991)}, {Psychiatrist}, {Drama, Thriller}
3. 33.33%, {Platoon (1986), All Quiet on the Western Front (1930), Glory (1989), Apocalypse Now (1979), Lawrence of Arabia (1962), Saving Private Ryan (1998), Paths of Glory (1957), Full Metal Jacket (1987)}, {Army, General, Jungle, Vietnam, Soldier, Recruit}, {Drama, Action, War}

Box OAC-triclustering has been not that successful. Despite being rather fast (only OAC-triclustering based on prime operators and SpecTric for small contexts are faster) and having good parallel version the resulting triclusters are quite large, have relatively low density and many intersections. It leads to the high coverage (1 for $rho_{min} = 0$) and rather low diversities. Also these triclusters are difficult to interpret (unlike SpecTric's triclusters that also have large size and low density). In many cases extent sizes are small. Examples are given below:

1. 0.9%, {The Shawshank Redemption (1994), The Godfather (1972), Ladri di biciclette (1948), Unforgiven (1992), Batman Begins (2005), Die Hard (1988), . . . , The Green Mile (1999), Sin City (2005), The Sting (1973)}, {Prison, Murder, Cuba, FBI, Serial Killer, Agent, Psychiatrist,. . . , Window, Suspect, Organized Crime , Revenge, Explosion, Assassin, Widow}, {Crime, Drama, Sci-Fi, Fantasy, Thriller, Mystery}

2. 1.07%, {The Great Escape (1963), Star Wars: Episode VI - Return of the Jedi (1983), Jaws (1975), Batman Begins (2005), Blade Runner (1982), Die Hard (1988),..., Metropolis (1927), Sin City (2005), Rebecca (1940)}, {Prison, Murder, Cuba, FBI, Serial Killer, Agent, Psychiatrist,..., Shower, Alimony, Phoenix Arizona, Assassin, Widow}, {Drama, Thriller, War}

Prime OAC-triclustering showed rather good results. It is one of the fastest algorithms (though some additional optimizations specified for unparallel version made parallelization inefficient for small contexts). The number of triclusters is high, but they are easily interpreted. Once again for $\rho_{min} = 0$ coverage is equal to 1, but remains high for different ρ_{min}. At the same time diversities are also rather high. Examples of the triclusters for the IMDB context are given below:

1. 36%, {The Shawshank Redemption (1994), Cool Hand Luke (1967), American History X (1998), A Clockwork Orange (1971), The Green Mile (1999)}, {Prison, Murder, Friend, Shawshank, Banker}, {Crime, Drama}
2. 56, 67%, {The Godfather: Part II (1974), The Usual Suspects (1995)}, {Cuba, New York, Business, 1920s, 1950s}, {Crime, Drama, Thriller}
3. 60%, {Toy Story (1995), Toy Story 2 (1999)}, {Jealousy, Toy, Spaceman, Little Boy, Fight}, {Fantasy, Comedy, Animation, Family, Adventure}

4 Conclusion

We compared several different triclustering approaches and showed that there is no algorithm-winner with respect to the considered criteria. Also, we can conclude that either (dense) prime OAC-triclustering or TriBox is a good alternative for TCA approach because the total number of triclusers for real data example is drastically less than the number of triconcepts. The proposed algorithm has a good scalability on real-world data and shows acceptable noise-tolerance level. Probably investigated spectral hierarchical partitioning of ternary relations can be a good alternative of concept-based triclustering as a tool for fast exploratory (preliminary) analysis.

Our further work on triclustering will go in the following directions:

– developing a unified theoretical framework for n-clustering,
– mixing several constraint-based approaches to triclustering (e.g., mining dense triclusters first and then frequent tri-sets in them),
– finding better approaches for estimating tricluster's density,
– taking into account the nature of real-world data for optimization (their sparsity, value distribution, etc.).
– investigation of the existing approaches and developing their extensions to triadic numerical data for comparison purposes,
– applying triclustering in recommender systems and social network analysis.

Acknowledgements. We would like to thank our colleagues Boris Mirkin, Jonas Poelmans, Leonid Zhukov, Andrey Kramarenko, Svetlana Vasyukova for helpful comments, and Ruslan Magizov (box OAC-triclustering) and Zarina Sekinaeva (SpecTric) for their programming support. This research was carried out in 2013 within a project of the Basic Research Program of the National Research University Higher School of Economics (Moscow) and affiliated Laboratory of Intelligent Systems and Structural Analysis.

References

1. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias–an algorithm for mining iceberg tri-lattices. In: Proceedings of the Sixth International Conference on Data Mining. ICDM '06, Washington, DC, USA, IEEE Computer Society (2006) 907–911
2. Mirkin, B., Kramarenko, A.V.: Approximate bicluster and tricluster boxes in the analysis of binary data. [15] 248–256
3. Miettinen, P.: Boolean tensor factorization. In Cook, D., Pei, J., Wang, W., Zaïane, O., Wu, X., eds.: ICDM 2011, 11th IEEE International Conference on Data Mining, Vancouver, Canada, IEEE Computer Society, CPS (2011) 447–456
4. Belohlávek, R., Glodeanu, C., Vychodil, V.: Optimal factorization of three-way binary data using triadic concepts. Order **30**(2) (2013) 437–454
5. Kaytoue, M., Kuznetsov, S., Macko, J., Napoli, A.: Biclustering Meets Triadic Concept Analysis. Ann. of Math. and Art. Intell. (2013) (to appear).
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
7. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Proceedings of the Third International Conference on Conceptual Structures: Applications, Implementation and Theory, London, UK, Springer-Verlag (1995) 32–43
8. Gnatyshak, D., Ignatov, D.I., Semenov, A., Poelmans, J.: Gaining insight in social networks with biclustering and triclustering. In: BIR. Volume 128 of Lecture Notes in Business Information Processing., Springer (2012) 162–171
9. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: Musicbox: Personalized music recommendation based on cubic analysis of social tags. IEEE Transactions on Audio, Speech & Language Processing **18**(2) (2010) 407–412
10. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E.: From triconcepts to triclusters. [15] 257–264
11. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J., Zhukov, L.E.: Can triconcepts become triclusters? International Journal of General Systems **42**(6) (2013) 572–593
12. Trabelsi, C., Jelassi, N., Ben Yahia, S.: Scalable mining of frequent tri-concepts from folksonomies. In Tan, P.N., Chawla, S., Ho, C., Bailey, J., eds.: Advances in Knowledge Discovery and Data Mining. Volume 7302 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 231–242
13. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed patterns meet n-ary relations. ACM Trans. Knowl. Discov. Data **3** (March 2009) 3:1–3:36
14. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-based biclustering for internet advertisement. In: ICDM Workshops, IEEE Computer Society (2012) 123–130
15. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing - 13th International Conference, 2011. Proceedings. In: RSFDGrC. Volume 6743 of Lecture Notes in Computer Science., Springer (2011)