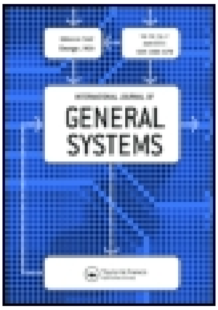
This article was downloaded by: [Higher School of Economics]
On: 11 June 2013, At: 04:49
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of General Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ggen20

Can triconcepts become triclusters?

Dmitry I. Ignatov [a] , Sergei O. Kuznetsov [a] , Jonas Poelmans [a] [b] & Leonid E. Zhukov [a]

[a] School of Applied Mathematics and Information Science & Laboratory for Intelligent Systems and Structural Analysis , National Research University Higher School of Economics , Moscow , Russia

[b] Faculty of Economics and Business , Katholieke Universiteit , Leuven , Belgium

Published online: 28 May 2013.

To cite this article: Dmitry I. Ignatov , Sergei O. Kuznetsov , Jonas Poelmans & Leonid E. Zhukov (2013): Can triconcepts become triclusters?, International Journal of General Systems, DOI:10.1080/03081079.2013.798899

To link to this article: http://dx.doi.org/10.1080/03081079.2013.798899

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

International Journal of General Systems, 2013
http://dx.doi.org/10.1080/03081079.2013.798899

Taylor & Francis
Taylor & Francis Group

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Can triconcepts become triclusters?

Dmitry I. Ignatov[a*], Sergei O. Kuznetsov[a], Jonas Poelmans[ab] and Leonid E. Zhukov[a]

[a] *School of Applied Mathematics and Information Science & Laboratory for Intelligent Systems and Structural Analysis National Research University Higher School of Economics, Moscow, Russia;* [b] *Faculty of Economics and Business Katholieke Universiteit, Leuven, Belgium*

(Received 10 March 2011; final version received 26 February 2013)

Two novel approaches to triclustering of three-way binary data are proposed. Tricluster is defined as a dense subset of a ternary relation Y defined on sets of objects, attributes, and conditions, or, equivalently, as a dense submatrix of the adjacency matrix of the ternary relation Y. This definition is a scalable relaxation of the notion of triconcept in Triadic Concept Analysis, whereas each triconcept of the initial data-set is contained in a certain tricluster. This approach generalizes the one previously introduced for concept-based biclustering. We also propose a hierarchical spectral triclustering algorithm for mining dense submatrices of the adjacency matrix of the initial ternary relation Y. Finally, we describe some applications of the proposed techniques, compare proposed approaches and study their performance in a series of experiments with real data-sets.

Keywords: formal concept analysis; data mining; triclustering; three-way data; folksonomy; spectral triclustering

1. Introduction

Extraction of relevant patterns from two-dimensional (object-attribute) data is one of the most thoroughly studied topics in data mining. Direct clustering was first proposed in Hartigan (1972) and was later extended to biclustering by Mirkin (1996). A bicluster emerges when there is a strong association between a subset of the objects and a subset of the attributes in a data table. A particular kind of bicluster is a formal concept in Formal Concept Analysis (FCA) (Ganter and Wille 1999). A formal concept is a pair of the form (extent, intent), where extent consists of all objects sharing all attributes from the intent and intent consists of all attributes shared by objects from the extent. Formal concepts have the desirable property of being homogeneous and closed in algebraic sense. Agrawal, Imielinski, and Swami (1993) introduced frequent itemset mining of two-dimensional data; many efficient algorithms for computing (closed) itemsests are known, among them CHARM (Zaki and Hsiao 2002), CLOSET (Pei, Han, and Mao 2000), Close (Pasquier et al. 1999). In FCA terms, itemset is a subset of attributes and a closed itemset is intent, a closed subset of attributes. Approximate biclustering was addressed in many settings of soft computation, e.g. in fuzzy-set setting (Belohlávek 2011, 2001), rough-set setting (Ganter and Kuznetsov 2008), and interval setting (Kaytoue et al. 2011, 2013); one can find more examples of biclustering techniques in biological setting in the survey (Madeira and Oliveira 2004).

*Corresponding author. Email: dignatov@hse.ru

© 2013 Taylor & Francis

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Recently, the focus of research shifted from two-dimensional to three-dimensional and n-dimensional data mining. Partially, this evolution is driven by the increase in popularity of social resource sharing systems (e.g. Flickr and Del.icio.us) where users can assign tags to resources. These systems rely on so-called folksonomies (Vander Wal 2007), which are three-way data structures describing interrelations between users, resources, and tags that can be attributed by users to resources. To deal with three-way data within FCA, an extension to Triadic Concept Analysis (TCA) was proposed by Lehmann and Wille (1995) and Wille (1995). In Jäschke et al. (2006) the authors introduced TRIAS algorithm for mining all frequent triconcepts from three-dimensional data and applied it to the popular Bibsonomy (users-tags-papers) data-set in Jäschke et al. (2007). Voutsadakis (2002) extended triadic concept analysis to n-dimensional contexts. Kaytoue et al. (2011) apply the methods of interval pattern structures to the analysis of gene expression data. Kaytoue et al. (2013) analyze these data by representing maximal biclusters of similar values by triconcepts using interordinal scaling. This approach is further extended to n-dimensional data.

Usual difficulties in mining binary data are the lack of fault tolerance, the huge number of patterns leading to large computational complexity, and artifacts in the form of numerous small patterns. In triadic or n-ary contexts, these problems are seriously aggravated. To cope with these issues, several techniques have been introduced for faster selection of interesting patterns. First, these are relaxations of the formal concept for the dyadic case, including relevant and dense bisets (Besson, Robardet, and Boulicaut 2006), concept factorization techniques (Belohlavek and Vychodil 2009), tensor factorization (Miettinen 2011), dense biclusters (Ignatov et al. 2010; Ignatov, Kuznetsov, and Poelmans 2012), and box clustering (Mirkin 2005; Mirkin and Kramarenko 2011). For the triadic case, there is an extended box clustering approach (Mirkin and Kramarenko 2011), triadic concept factorization (Belohlavek and Vychodil 2010). Another approach, called constrain-based mining, also scales up to n-ary relations and is discussed in Cerf et al. (2008, 2009). In this method, the user specifies constraints on the extracted patterns. The Data-Peeler algorithm in Cerf et al. (2008) is able to extract closed patterns in n-ary relations under constraints for $n \geq 3$ and was shown to have better efficiency than that of TRIAS and CubeMiner for 3-ary relations.

There are also pruning techniques based on concept indices. One of the first approaches are based on indices such as sizes of extents (closed sets of objects) and intents (closed sets of attributes, or closed itemsets) (Kuznetsov 1989, 1996) (subsets of concepts satisfying constraints on this kind of indices are also known as iceberg lattices (Stumme et al. 2002)), which, in FCA terms, cut order filters (ideals) of the concept lattice. The index called concept stability (Kuznetsov 1990, 2007) also helps to prune concepts in various applications, some recently proposed efficient indices are independence and concept probability (Klimushkin, Obiedkov, and Roth 2010).

In this paper, we propose a novel triclustering algorithm that extracts dense approximate triclusters from Boolean three-way data. This algorithm has a better theoretical time complexity than existing exact algorithms like TRIAS and is, therefore, better suited for very large data-sets. Moreover, during experimentations with the Bibsonomy data-set, we found that the number of triclusters generated by our algorithm is significantly lower than the number of triconcepts extracted by TRIAS. Manual validation of the extracted tricommunities revealed that a majority of them were meaningful. In this paper, we also adapt spectral clustering, one of the fastest algorithms for graph partitioning (Drineas et al. 1999; Shi and Malik 2000; Ding et al. 2001; Ng, Jordan, and Weiss 2001; Dhillon 2001; Verma and Meila 2003; Kannan, Vempala, and Veta 2004), to the triclustering setting.

The remainder of the paper is organized as follows. In Section 2, we describe some key notions of triadic concept analysis. In Section 3 we define the operators for generating triclusters, introduce some other necessary definitions, and describe our triclustering algorithm TRICL. In

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Section 4, we describe our algorithm SpecTric, an adaptation of the spectral clustering approach to the triadic case. In Section 5, we first discuss experiments with SpecTric and TRICL on a small piece of Bibsonomy data, and then pass to our main big-scale experiments.

In Section 6, we present the results of experiments with the Bibsonomy data and briefly describe the problem setting of the Amsterdam–Amstelland police data. Section 7, concludes our paper and describes some interesting directions for future research.

2. Main definitions

First, we recall some basic notions of FCA (Ganter and Wille 1999). 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 $K = (G, M, I)$, 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*:

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

If we have several contexts, the derivation operators of a context (G, M, I) is denoted by $(.)^I$.

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 $B'' = B$). For $B, D \subseteq M$ the *implication* $B \to D$ holds if $B' \subseteq D'$.

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* $\mathfrak{B}(G, M, I)$.

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

$$\mathbb{K}^{(1)} = (X_1, X_2 \times X_3, Y^{(1)}),$$
$$\mathbb{K}^{(2)} = (X_2, X_1 \times X_3, Y^{(2)}), \tag{2}$$
$$\mathbb{K}^{(3)} = (X_3, X_1 \times X_2, Y^{(3)}),$$

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 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:

$$Z \mapsto Z^{(i)} = \{(x_j, x_k) \in X_j \times X_k \mid x_i, x_j, x_k \text{ are related by Y for all } x_i \in Z\},$$
$$W \mapsto W^{(i)} = \{x_i \in X_i \mid x_i, x_j, x_k \text{ are related by Y for all } (x_j, x_k) \in W\}. \tag{3}$$

Formally, a *triadic concept* (or triconcept) 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 $(A_j \times A_k)^{(i)} = A_i$. For a certain 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 considered as a maximal cuboid (parallelepiped) 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 all dense triclusters

3.1. Prime, double prime, and box operators of 1-sets

To simplify 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 introduce primes, double primes and box operators for particular elements of G, M, B, respectively (see Table 1). 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'.

In what follows, we do not use double primes, because of their rigid structure; they do not tolerate exceptions like missing pairs. To allow for missing pairs, we introduce the following box operators:

$$\begin{aligned} g^{\square} &= \{ g_i \mid (g_i, b_i) \in m' \textit{ or } (g_i, m_i) \in b' \text{ for all } (g_i, m_i, b_i) \in Y\} \\ m^{\square} &= \{ m_i \mid (m_i, b_i) \in g' \textit{ or } (g_i, m_i) \in b' \text{ for all } (g_i, m_i, b_i) \in Y\} \\ b^{\square} &= \{ b_i \mid (g_i, b_i) \in m' \textit{ or } (m_i, b_i) \in g' \text{ for all } (g_i, m_i, b_i) \in Y\}. \end{aligned} \tag{4}$$

3.2. Dense triclusters

Let $\mathbb{K} = (G, M, B, Y)$ be a triadic context. For a certain triple $(g, m, b) \in Y$, the triple $T = (g^{\square}, m^{\square}, b^{\square})$ is called a *tricluster*. As in case of triconcepts, we still call the three components of a tricluster by extent, intent, and modus, respectively.

Table 1. Prime and double prime operators of 1-sets.

Prime operators of 1-sets	Their double prime counterparts
$g' = \{ (m, b) \mid (g, m, b) \in Y\}$	$g'' = \{ \tilde{g} \mid (m, b) \in g' \text{ and } (\tilde{g}, m, b) \in Y \text{ for all } \tilde{g} \in G\}$
$m' = \{ (g, b) \mid (g, m, b) \in Y\}$	$m'' = \{ \tilde{m} \mid (g, b) \in m' \text{ and } (g, \tilde{m}, b) \in Y \text{ for all } \tilde{m} \in M\}$
$b' = \{ (g, m) \mid (g, m, b) \in Y\}$	$b'' = \{ \tilde{b} \mid (g, m) \in b' \text{ and } (g, m, \tilde{b}) \in Y \text{ for all } \tilde{b} \in B\}$

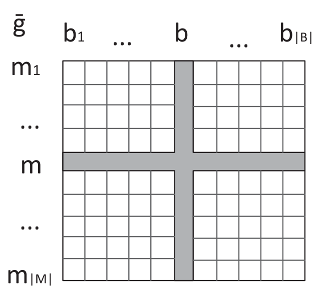


Figure 1. An illustration of the addition of new object $\overline{g}$ to the extent $g^{\square}$ of a certain tricluster.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Table 2. A toy example with Bibsonomy data for users $\{u_1, u_2, u_3\}$, resources $\{r_1, r_2, r_3\}$ and tags $\{t_1, t_2, t_3\}$.

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:

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

Let us explain the structure of the proposed triclusters. Suppose $\mathbb{K} = (G, M, B, I)$ is a triadic context, and triple $(g, m, b) \in I$ is considered. Then, object $\overline{g}$ will be added to $g^{\Box}$ iff $|\{(\overline{g}, m, \widetilde{b})|\widetilde{b} \in B\}| \neq \emptyset \vee |\{(\overline{g}, \widetilde{m}, b)|\widetilde{m} \in M\}| \neq \emptyset$. It is clear that this condition is equivalent to the one in equation (4), and can be illustrated in a simple way (Figure 1): if at least one of the elements from "grey" cells is an element of I, then $\overline{g}$ is added to $g^{\Box}$.

The *density* of a certain 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 $\rho(A, B, C) = \frac{|I \cap A \times B \times C|}{|A||B||C|}$.

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 3.1 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 3.2 For every tricluster (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 3.3 *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 exists a tricluster $T = (A, B, C) \in \mathbf{T}(G, M, B, Y)$ such that $A_c \subseteq A$, $B_c \subseteq B$, $C_c \subseteq C$.*

Proof Let $(g, m, b) \in A_c \times B_c \times C_c$. By the definition of box operators $g^{\Box} = \{ g_i \mid (g_i, b_i) \in m'$ *or* $(g_i, m_i) \in b'$ for all $(g_i, m_i, b_i) \in Y\}$. Since $m \in B_c$, then by the definition of formal triconcept m is related by Y to every $(\tilde{g}, \tilde{b}) \in A_c \times C_c$, therefore $m' \cap A_c \times C_c = A_c \times C_c$. Consequently, for all $g_i \in A_c$ we have $g_i \in g^{\Box}$. For $m^{\Box}$ and $b^{\Box}$ tricluster components the proof is similar. Finally, we have $A_c \subseteq A = g^{\Box}$, $B_c \subseteq B = m^{\Box}$, and $C_c \subseteq C = b^{\Box}$. $\Box$

In other words, proposition 3.3 says that every concept of initial context is included into some tricluster of the same context. Thus, in a certain sense, there is no information loss in triclustering results.

Example 1

In Table 2, we have $3^3 = 27$ formal triconcepts, 24 with $\rho = 1$, and 3 void triconcepts with $\rho = 0$ (they have either emptyset of users or resources or tags). Although the data are small, we have 27 patterns to analyze (maximal number of triconcepts for the context size $3 \times 3 \times 3$); this is due to the data being 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 of $(tag, resource)$ shared pairs and it is convenient to reduce the number of patterns describing this data from 27 to 1. The tricluster $T = (\{u_1, u_2, u_3\}, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\})$ with $\rho = 0.89$ is exactly such a reduced pattern, but its density is slightly less than 1. Each of the triconcepts from $\mathfrak{T} = \{(\emptyset, \{t_1, t_2, t_3\}, \{r_1, r_2, r_3\}), (\{u_1\}, \{t_2, t_3\}, \{r_1, r_2, r_3\}), \ldots (\{u_1, u_2, u_3\}, \{t_1, t_2\}, \{r_3\})\}$ is contained, w.r.t. component-wise set inclusion, in T. In case of large trisubcontext (in initial tricontext) size $n \times n \times n$, where only main diagonal does not contain crosses, we will have tricluster of the same size with $\rho = \frac{n^2-1}{n^3}$.

3.3. *An algorithm: Tricl*

Algorithm 1: TRICL

Data: $K = (G, M, B, Y)$ – formal context, ρ_{min} – density threshold
Result: $\mathbf{T} = \{(A_k, B_k, C_k) | (A_k, B_k, C_k)$ – tricluster$\}$

1 **begin**
2   **for** $(g, m, b) \in Y$ **do**
3   **if** *g not in PrimesObj* **then**
4       $PrimesObj[g] = g'$
5   **if** *m not in PrimesAttr* **then**
6       $PrimesAttr[m] = m'$
7   **if** *b not in PrimesCond* **then**
8       $PrimesCond[b] = b'$
9   **if** *g not in BoxesObj* **then**
10       $BoxesObj[g] = g^{\square}$
11   **if** *m not in BoxesAttr* **then**
12       $BoxesAttr[m] = m^{\square}$
13   **if** *b not in BoxeesCond* **then**
14       $BoxesCond[b] = b^{\square}$
15   **for** $(g, m, b) \in Y$ **do**
16     $Tkey = hash((BoxesObj[g], BoxesAttr[m], BoxesCond[b]))$
17   **if** *Tkey not in* $\mathbf{T}$ **then**
18   **if** $\rho(BoxesObj[g], BoxesAttr[m], BoxesCond[b]) \geq \rho_{min}$ **then**
19         $\mathbf{T}[Tkey] = ((BoxesObj[g], BoxesAttr[m], BoxesCond[b]))$
20

The idea behind TRICL is simple: for all $(g, m, b) \in Y$ with $\rho(g^{\square}, m^{\square}, b^{\square}) \geq \rho_{min}$ the algorithm stores $T = (g^{\square}, m^{\square}, b^{\square})$ in $\mathbf{T}$. In the pseudocode of TRICL (see Algorithm 1), we provide more computational details rather than a simple algebraic description. It allows us to better evaluate the algorithmic complexity of the main algorithm's steps and give some ideas on how to implement the code. The dictionaries $PrimesObj$, $PrimesAttr$, and $PrimesCond$ store pairs $(attribute, condition)$, $(object, condition)$, and $(object, attribute)$ for the respective prime operator $(.)'$. These stored values are then used for computing box operators. In their turn, the results of box operators are placed in $BoxesObj$, $BoxesAttr$, and $BoxesCond$, respectively. The complexity of the first "for" loop (steps 2-10) is $O(|Y|)$. The main loop complexity (steps 15-19) is trickier: $O(|Y||\mathbf{T}||log(|\mathbf{T}|)|G||M||B|)$ or, as we know that $|\mathbf{T}| \leq |Y|$, $O(|Y|^2||log(|Y|)$ $|G||M||B|)$. The factor $|G||M||B|$ arises because of computing the density of a tricluster ρ; this value is indeed difficult to compute for large triclusters.

We propose a heuristic to estimate $\rho(T)$ in Monte-Carlo way, checking only some amount of randomly selected triples contained in the given tricluster T. For a certain tricluster $T = (A, B, C)$, we compute the density estimation $\hat{\rho}(T) = |P|/|N|$, where $P = \{(g, m, b)|(g, m, b) \in N \cap Y\}$, and N is a set of $|N|$ randomly chosen elements of the tricluster. The $|N|$ parameter can be chosen to be relatively small, say $1/10|A||B||C|$. Finally, this heuristic is able to make the density calculation in $|N|/(|A||B||C|)$ times faster. It is also clear that the time complexity of the probabilistic algorithm is in $|N||/(|G||M||B|)$ times better than for the original algorithm with the exhaustive strategy of density calculation. To differentiate between exhaustive TRICL algorithm and its probabilistic version, we call them TriclEx and TriclProb, respectively, in the experiments section.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

4. Spectral triclustering

In this section, we show that a problem of finding (dense) triclusters for a given context can be solved by means of graph-partitioning algorithms. There is a well-known reduction of bipartite graph partitioning to traditional spectral partitioning of a simple graph (see, e.g. (Dhillon 2001)). Some authors also made attempts to extend this approach for the case of tripartite graphs (Gao et al. 2005; Nanopoulos, Gabriel, and Spiliopoulou 2009; Liu, Fang, and Zhang 2010). Furthermore, Gao et al. (2005) worked on a specific case of multiple biclustering in which there is a central type of entity that connects the other types so as to form a star topology of the interrelationships. Doing so, the authors analyzed triadic data which represent the three types of objects. However, they did not use spectral partitioning of this graph directly due to a slightly different problem setting; actually, they analyzed spectral partitioning of two bipartite graphs, which share one partition (the center of the star structure), but not a tripartite graph. Nanopoulos, Gabriel, and Spiliopoulou (2009) describe analyzing folksonomies, which contain (user, tag, resource) triples, however, the researchers also did not perform spectral triclustering to their data directly. They composed a multidigraph which captures multiple similarities between resources (items) and analyzed this graph (namely, corresponding Laplacian tensor) by means of spectral partitioning algorithms. The paper by Liu, Fang, and Zhang (2010) aimed to cope with tag sense ambiguity in folksonomies. Unfortunately, the authors did not keep the tripartite structure of their graphs due to elimination of one valuable relation between users and resources, even though they used spectral clustering of triadic data and called the analyzed graph as "the tripartite hypergraph." Therefore, we cannot consider their algorithm as a relevant candidate for comparison purposes in our paper. The basic notions of spectral clustering and method description can be found in Appendix 1.

4.1. *SpecTric: spectral triclustering algorithm*

Let us formulate the spectral triclustering problem for the triadic case in a similar way as we did for the bipartite graph described above.

We consider the adjacency matrix M for an initial tricontext $\mathbb{K} = (U, T, R, Y \subseteq U \times T \times R)$ (we can always do that since every formal tricontext describes a corresponding tripartite graph), which represents a ternary relation between three sets: users U, tags T, and resources R.

$$\mathbf{M} = \begin{pmatrix} 0 & A_{UT} & A_{UR} \\ A_{UT}^T & 0 & A_{TR} \\ A_{UR}^T & A_{TR}^T & 0 \end{pmatrix} \tag{5}$$

Here, A_{UT} is an adjacency matrix, which shows which tags particular users use. Similarly, A_{UR} is an adjacency matrix of the user resource relation, A_{TR} is an adjacency matrix of the tag resource relation. Hence, a single triple $(user, tag, resource)$ corresponds in the tripartite graph to the triangle, which connects vertices $user$, tag, and $resource$. Let n be a number of users, m be a number of tags, k be a number of resources. Then, M has size $(n + m + k) \times (n + m + k)$. As in the case of bipartite graphs (two-way data), partitioning of a tripartite graph results in eigenvector of Laplacian matrix L, which corresponds to the second largest eigenvalue of the system $Lx = \lambda x$ or for the generalized eigenvector system $Lx = \lambda Wx$.

Similarly, as for a bipartite graph, the Laplacian matrix can be represented as

$$\mathbf{L} = \begin{pmatrix} D_U & -A_{UT} & -A_{UR} \\ A_{UT}^T & D_T & -A_{TR} \\ -A_{UR}^T & -A_{TR}^T & D_R \end{pmatrix}. \tag{6}$$

The corresponding system is written down below

$$\begin{pmatrix} D_U & -A_{UT} & -A_{UR} \\ A_{UT}^T & D_T & -A_{TR} \\ -A_{UR}^T & -A_{TR}^T & D_R \end{pmatrix} \begin{pmatrix} u \\ t \\ r \end{pmatrix} = \lambda \begin{pmatrix} D_U & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_R \end{pmatrix} \begin{pmatrix} u \\ t \\ r \end{pmatrix}. \tag{7}$$

We can easily devise a recursive procedure using the spectral method as a partitioning technique at every level. For every subgraph, we extract the corresponding adjacency matrix from the graph adjacency matrix and then apply spectral bipartition again. Thus, we build a binary tree with leaves containing triclusters (submatrices). Every leaf T_k at level k should ultimately contain at least one user–tag–resource triple. We devise several stopping criteria which are based on size and density constraints for the tree construction:

(1) C_{size} constraint: every leaf $T_k = A_k, B_k, C_k$ has nonempty sets of users A_k, tags B_k, and resources C_k and each set size is greater than 1.

(2) $\rho(T_k) \geq \rho_{min}$.

Some further ideas for the recursive spectral clustering can be found in Cheng et al. (2003).

Algorithm 2: SpecTric

Data: M is a corresponding adjacency matrix of the input formal context $K = (G, M, B, Y)$, ρ_{min} – density threshold

Result: $\mathbf{T} = \{(A_k, B_k, C_k) | (A_k, B_k, C_k)$ – dense tricluster$\}$

begin
- $(T_L, T_R) = SpectralPartitioning(M)$;
- **if** $C_{size}(T_L)$ **then**
  - **if** $\rho(T_L) \geq \rho_{min}$ **then**
    - $\mathbf{T}.Store(T_L)$
    - $SpecTric(TriclusterToMatrix(T_L))$
- **if** $C_{size}(T_R)$ **then**
  - **if** $\rho(T_R) \geq \rho_{min}$ **then**
    - $\mathbf{T}.Store(T_R)$
    - $SpecTric(TriclusterToMatrix(T_R))$

5. Illustration of TRICL, SpecTric, and TRIAS results on a small piece of Bibsonomy-like data

We have three sets U, T, and R: scientists $U = \{$Jäschke, Stumme, Poelmans, Ignatov, Dedene$\}$ use tags $T = \{$Machine Learning, Ontology, Domestic Violence, Formal Concepts, Triclustering$\}$ to mark papers $R = \{$paper$_1$, paper$_2$, paper$_3\}$.

Initial triples are written down below:

$$(Jäschke, Machine\ Learning, paper_1)$$
$$(Jäschke, Formal\ Concepts, paper_2)$$
$$(Stumme, Ontology, paper_2)$$
$$(Poelmans, Domestic\ Violence, paper_3)$$
$$(Ignatov, Machine\ Learning, paper_3)$$
$$(Dedene, Domestic\ Violence, paper_2)$$
$$(Ignatov, Triclustering, paper_3).$$

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

The tripartite graph of the initial context $K = (U, T, R, Y)$ is shown in Figure 2 (To us, scientists being a bit lazy as typical users use the short form of the original words.) Since our users preferred shorter tags as real ones, the full words are included here for clarity.

The corresponding system and its solution for this data are given below.

$$
\begin{array}{c}
J \\ S \\ P \\ I \\ D \\ \hline ML \\ O \\ DV \\ FC \\ T \\ \hline P_1 \\ P_2 \\ P_3
\end{array}
\left(\begin{array}{ccccc|ccccc|ccc}
4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 3 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\
\hline
-1 & 0 & 0 & -1 & 0 & 4 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0-1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 \\
\hline
-1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
-1 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 6 & 0 \\
0 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 5
\end{array}\right)
\begin{pmatrix}
-0.0380 \\ -0.2727 \\ 0.1005 \\ 0.2150 \\ -0.1059 \\ 0.1217 \\ -0.2727 \\ -0.0027 \\ -0.1277 \\ 0.2350 \\ 0.0517 \\ -0.1688 \\ 0.1654
\end{pmatrix}
$$

$$
= 0.1905 \begin{pmatrix} D_U & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_R \end{pmatrix}
\begin{pmatrix}
-0.0380 \\ -0.2727 \\ 0.1005 \\ 0.2150 \\ -0.1059 \\ 0.1217 \\ -0.2727 \\ -0.0027 \\ -0.1277 \\ 0.2350 \\ 0.0517 \\ -0.1688 \\ 0.1654
\end{pmatrix}. \qquad (8)
$$

All the negative components of the solution vector correspond to the components of left tricluster at first splitting on Figure 3, similarly, all the positive components of the solution vector describe the elements of the right tricluster.

A recursive call of the triclustering procedure produces a tree of generated triclusters. We stop triclustering generation if the next step results in a tricluster containing less than either two objects or tags or resources.

As a result, we get three triclusters in Example 2. With respect to the termination criterion, for further interpretation we are interested in those triclusters that contain more than one user, tag or resource for each tricluster component, respectively. In this example, we have only two such triclusters after the first splitting and one after the second splitting (see Figure 3).

To differentiate between triclusters generated by box operators and spectral partitioning hierarchical procedure, we will call former as *conceptual triclusters* and later as *spectral triclusters*.

Let us consider the results of TRICL for the same data.

We consider the triple $(Ignatov, Triclustering, P_3)$ and apply box-operators to its components: $Ignatov^{\square} = \{Ignatov, Poelmans\}$, $T^{\square} = \{MachineLerning, Triclustering,$

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

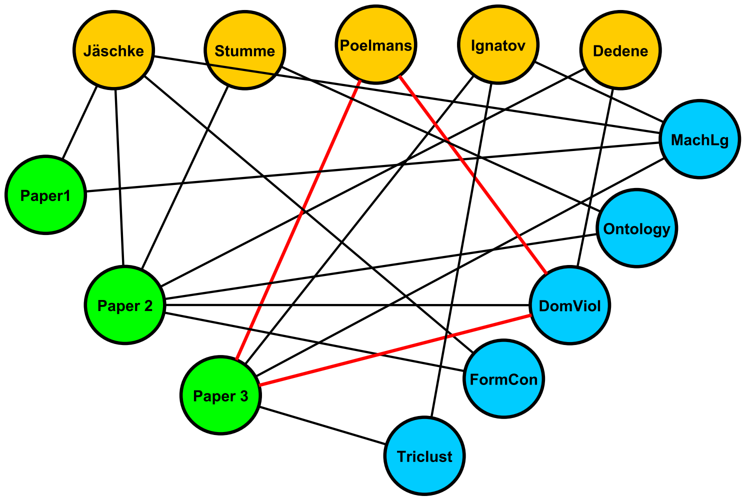


Figure 2. A tripartite graph of the Bibsonomy from Example 2.

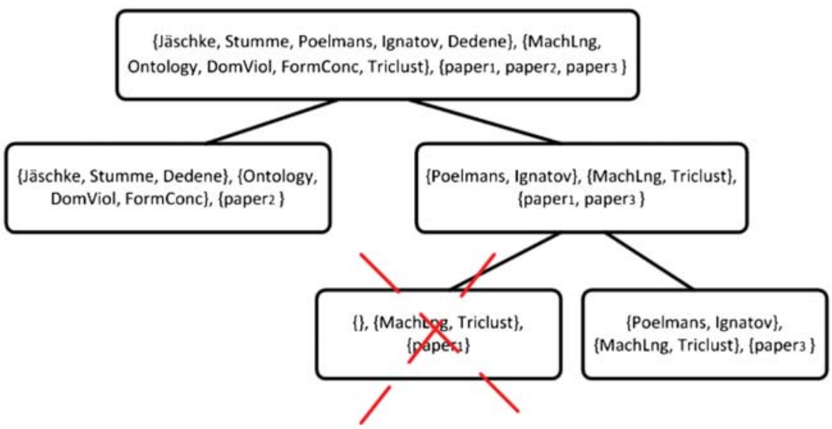


Figure 3. Tree-like partitioning of the Bibsonomy from Example 2.

$DomesticViolence\}$, $P_3^{\Box} = \{P_3\}$. These computation involves the following prime operators results: $Ignatov' = \{(MachineLearning, P_3), (Triclustering, P_3)\}$, $Triclustering' = \{(Ignatov, P_3)\}$, $P_3' = \{(Poelmans, DomesticViolence), (Ignatov, Triclustering), (Ignatov, MachineLearning)\}$.

The first three conceptual triclusters sorted by their density have density no less than 0.25 and are rather well-interpreted scientific communities:

$(\{Poelmans, Ignatov\}, \{MachineLearning, DomesticViolence, Triclustering\}, \{paper3\})$, with $\rho = 0.5$, $(\{Jäschke, Stumme, Dedene\}, \{FormalConcept, Ontology, DomesticViolence\}, \{paper2\})$, with $\rho = 0.3333$,

$(\{Jäschke, Ignatov\}, \{MachineLearning, FormalConcept\}, \{paper1, paper2, paper3\})$, with $\rho = 25$.

The rest four conceptual triclusters are worse interpretable, so $\rho = 0.25$ can be chosen as a suitable minimal threshold:

$(\{Poelmans, Ignatov, Dedene\}, \{MachineLearning, DomesticViolence, Triclustering\}, \{paper2, paper3\})$, with $\rho = 0.2222$,

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

$(\{J\ddot{a}schke, Poelmans, Ignatov\}, \{MachineLearning, DomesticViolence, Triclustering\}, \{paper1, paper3\})$, with $\rho = 0.2222$,

$(\{J\ddot{a}schke, Stumme, Dedene\}, \{MachineLearning, FormalConcept, Ontology, DomesticViolence\}, \{paper1, paper2\})$, with $\rho = 0.1667$,

$(\{J\ddot{a}schke, Stumme, Poelmans, Dedene\}, \{FormalConcept, Ontology, DomesticViolence\}, \{paper2, paper3\})$, with $\rho = 0.1667$.

It is hard to say which results for such a small data are better, but note that both methods capture both different communities $\{J\ddot{a}schke, Stumme, Dedene\}$ and $\{Poelmans, Ignatov\}$ grouped around paper2 and paper3, respectively. The sets of their tags slightly differ for spectral and conceptual triclusters, but anyway they can shed a light on the community interests.

For the same data, we found that the results of TRIAS are unsatisfactory. Among the generated concepts without void extent, intent or modus (6 out of 11), almost all the triconcepts are trivial communities, i.e. they are triples of original relation I:

$\{Dedene\}, \{DomesticViolence\}, \{paper2\}$,
$\{Ignatov\}, \{MachineLearning, Triclustering\}, \{paper3\}$,
$\{Poelmans\}, \{DomesticViolence\}, \{paper3\}$,
$\{Stumme\}, \{Ontology\}, \{paper2\}$,
$\{J\ddot{a}schke\}, \{FormalConcept\}, \{paper2\}$,
$\{J\ddot{a}schke\}, \{MachineLearning\}, \{paper1\}$.

6. Real data and experiments

The performance of our triclustering algorithm was empirically validated on two real world data-sets. In the Bibsonomy case study, communities of researchers were extracted from triples indicating which scientists assigned which tags to scientific papers. We also describe some preliminary results obtained during the case study with the Amsterdam–Amstelland police where we aimed to extract criminal communities from observational police reports.

To make the results of comparison more informative, we introduce some additional measures of triclusters quality.

6.1. Coverage and diversity

In addition to execution time, number of triclusters, and tricluster density, we introduce coverage and diversity.

Coverage is defined simply as a fraction of the triples of the context (alternatively, objects, attributes or conditions) included in at least one of the triclusters of the resulting set.

Diversity is a useful measure for Feature Selection and Ensemble Learning (Tsymbal, Pechenizkiy, and Cunningham 2005). To define *diversity* in triclustering setting, we will use binary function of 2 triclusters $intersect$:

$$intersect(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} 1, & G_{\mathcal{T}_i} \cap G_{\mathcal{T}_j} \neq \emptyset \wedge M_{\mathcal{T}_i} \cap M_{\mathcal{T}_j} \neq \emptyset B_{\mathcal{T}_i} \cap B_{\mathcal{T}_j} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{T}$ is a tricluster set.

It is also possible to define $intersect$ for the sets of objects, attributes, and conditions. For instance, $intersect_G$ is defined as follows:

$$intersect_G(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} 1, & G_{\mathcal{T}_i} \cap G_{\mathcal{T}_j} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

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

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

Once again, it is possible to define diversity for the sets of objects, attributes or conditions:

$$diversity_G(\mathcal{T}) = 1 - \frac{\sum_j \sum_{i<j} intersect_G(\mathcal{T}_i, \mathcal{T}_j)}{\frac{|\mathcal{T}|(|\mathcal{T}|-1)}{2}} \tag{12}$$

We suppose, the higher coverage and diversity, the better results of triclustering algorithm. However, not all triples may be relevant, especially in case of noise or errors in the input data, so, we do not expect 100% coverage for the best algorithms. If the diversity is high and the number of tricluster is relatively small, then we can expect the moderate number of rather well-interpreted triclusters.

Thus, the portrait of the best triclustering results should be described by the following features: the smallest number of dense triclusters with high diversity and coverage values calculated in the shortest time.

6.2. Bibsonomy case study

In our experiments, we have analyzed the popular social bookmarking system Bibsonomy. The data are freely available from http:\Bibsonomy.org for research purposes. For detecting communities of users which have similar tagging behavior, we ran the TRICL algorithm on a part of the data consisting of all users, resources, tags, and tag assignments.

We used only the data-set that contains a 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), and
(5) date.

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

The resulting folksonomy (Bibsonomy) consists of $|U| = 2{,}337$ users, $|T| = 67{,}464$ different tags, and $|R| = 28{,}920$ resources (bookmarks and bibtex entries), which are linked by $|Y| = 816, 197$ triples. We should note that we must deal with a cuboid consisting of 4,559,624,602,560 cells.

Before running the TRICL algorithm, we analyzed the data statistics. In particular, we plotted a histogram for users and their 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.

One can easily see that the data exhibits power law behavior. For example, for Figure 4 plotted in log–log scale we revealed power law ($p(x) = Cx^{-\alpha}$) with $\alpha = 1, 5$ (we followed the method proposed in Newman (2005) to compute this coefficient). The power law distribution of the data points justifies the use of a greedy approach to mining large and (relatively) dense triclusters, since the small part of users have the most portion of (tag, user) assignments (similar conclusions for tags and documents distributions).

We measured the run-time of the implementations (in Python 2.7.1) on a Pentium Core Duo system with 2 GHz and 2 GB RAM. To build all triconcepts of a certain context we used a Java

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

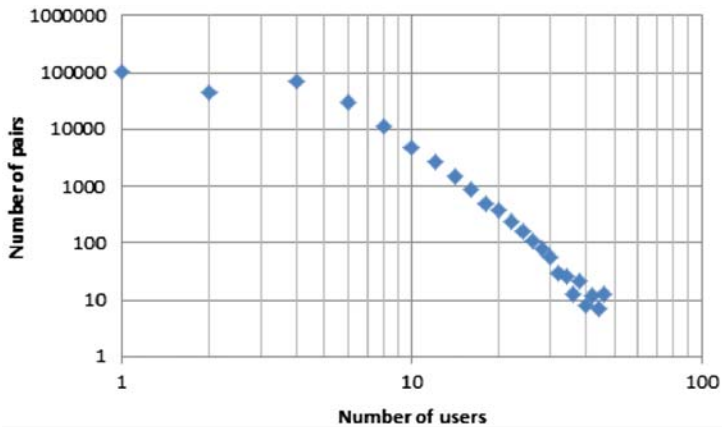


Figure 4. Histogram of numbers of pairs (document, tag) for all triples of the Bibsonomy data.

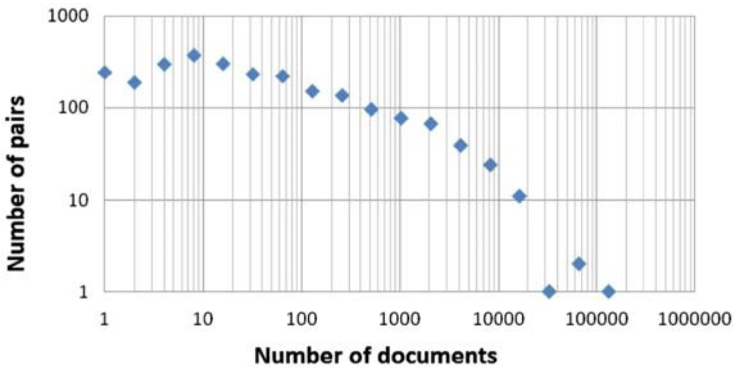


Figure 5. Histogram of numbers of pairs (users, tags) for all triples of the Bibsonomy data.

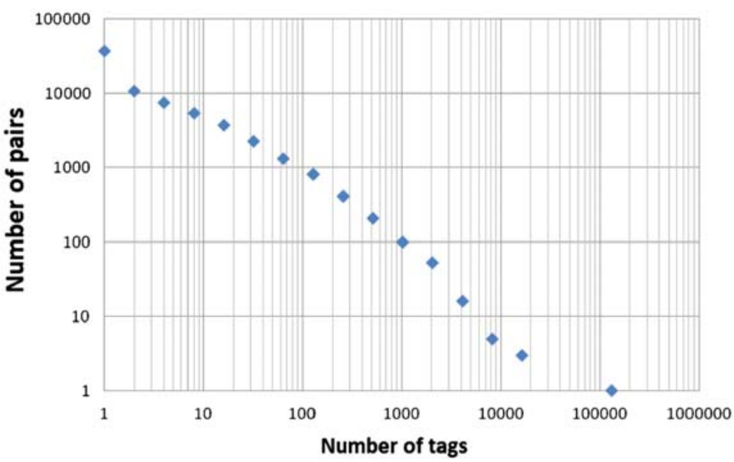


Figure 6. Histogram of numbers of pairs (user, document) for all triples of the Bibsonomy data.

implementation of the TRIAS algorithm by Jäschke et al. (2006). The last two columns in Table 3 mean time of execution of TRICL with full density and probabilistic density calculation strategy, respectively.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Table 3. Experimental results for k first triples of Bibsonomy data set with $\rho_{min} = 0$.

k, number of first triples	$\|U\|$	$\|T\|$	$\|R\|$	$\|\mathfrak{T}\|$	$\|\mathbf{T}_{Tricl}\|$	$\|\mathbf{T}_{SpecTric}\|$	TRIAS, s	TriclEx,s	TriclProb,s	SpecTric,c
100	1	47	52	57	1	1	0.2	0.2	0.2	0.2
1000	1	248	482	368	1	1	1	1	1	1
10,000	1	444	5193	733	1	1	2	46.7	47	2
100,000	59	5823	28,920	22,804	4462	131	3386	10,311	976	6215
200,000	340	14,982	61,568	-	19,053	-	> 24 h	> 24 h	3417	-

Table 4. Density distribution of TRICL-generated triclusters for 200,000 first triples of Bibsonomy data set with $\rho_{min} = 0$.

Low bound of ρ	Upper bound of ρ	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

Table 5. Density distribution of spectral triclusters for 100,000 first triples of Bibsonomy data set with $\rho_{min} = 0$.

Low bound of ρ	Upper bound of ρ	Number of triclusters
0	0.05	117
0.05	0.1	7
0.1	0.2	5
0.2	0.3	0
0.3	0.4	1
0.4	0.5	1
0.5	0.6	0
0.6	0.7	0
0.7	0.8	0
0.8	0.9	0
0.9	1	0

In our experiments, the estimation $\hat{\rho}$ has only 0.13 mean absolute error for $|N| = 1/10$ of a tricluster size, $\rho_{min} = 0$, and 200,000 first triples of the Bibsonomy data. It is clear the probabilistic algorithm TriclProb becomes drastically faster than TRIAS and TriclEx (TRICL with exhaustive density calculation) in the case of our probabilistic computational strategy.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Table 6. Results of the experiments in terms of density, coverage, and diversity.

Algorithm	BibSonomy						
	Number of triclusters	ρ_{av}, %	Cov, %	Div, %	Div_U, %	Div_T, %	Div_R, %
TRICL	398	4.16	100	79.59	67.28	42.83	79.54
SpecTric	2	0.5	100	100	100	100	100
TRIAS	1305	100	100	99.98	91.70	99.78	99.92

Distribution of density of triclusters for 200,000 first triples of Bibsonomy data-set is given in Table 4.

Analyzing the results of triclusters' density distribution in Tables 4 and 5, we can conclude that conceptual triclusters are able to capture the densest triclusters (formal triconcepts) and focus on more dense cuboids of the original data-set. In these experiments, spectral triclustering is not able to find formal triconcepts and results in triclusters of lower density, which are not so ap propriate for further human-expert analysis. However, we should note that our spectral triclustering algorithm is almost two times faster than the direct conceptual triclustering algorithm and results in no more than $2 \times \log(m + n + k)$ spectral triclusters.

In Table 6, we provide the comparison of diversity and coverage for 3000 randomly chosen triples of Bibsonomy data with the following parameters: $|U| = 51$, $|T| = 924$, $|R| = 2844$, and $\rho = 2.2385 \times 10^{-5}$.

TRIAS is the most time consuming algorithm. Though the resulting triclusters (triconcepts) can be easily interpreted, their number and small size make it impossible to understand the general structure of a large context. For example, in Table 3 we can see that TRICL does not consider tags and papers of one user belonging to the same tricluster. On the contrary, TRICL and SpecTric correctly identify only one tricluster in this case. Since all triconcepts have been generated, every triple is covered and coverage is equal to 1. Since the concepts are small, general diversity is rather high. Moreover, the set diversity depends on the size of the corresponding set ($|U|$, $|T|$ or $|R|$): the smaller the set – the greater chance of intersection and the lower the diversity.

SpecTric has displayed rather good computation time. Main part of this time is used for the eigenvalue decomposition of Laplacian matrix. The resulting triclusters can be interpreted, though their average density is too low and size is rather huge. Their small number makes this method good for dividing the context into several nonoverlapping parts. Also, the diversity measure for SpecTric is always equal to 1 because the method generates partitions of the initial context.

TRICL was relatively successful in terms of computation time and number of triclusters even for $\rho = min$. It leads to the high coverage (1 for $\rho_{min} = 0$) and rather low diversities. The diversity measures have acceptable level for users U and resources R, but it is not the case for tags set T.

6.3. Amsterdam–Amstelland Police Triclustering Application

In this case study, we analyzed 22,672 observational police reports filed in the Amsterdam–Amstelland police region in years 2006, 2007, and 2008. The reports contain a textual description of observation made by police officers during a police patrol, a motor vehicle inspection, an intervention, and other similar cases of suspicious situation. We were particularly interested in situations which may indicate that a person is involved in human trafficking. Human trafficking is defined by the United Nations as the recruitment, transportation, harboring, and receipt of

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Table 7. Density distribution of triclusters for the police reports data set with $\rho_{min} = 0$.

Low bound of ρ	Upper bound of ρ	Number of triclusters
0	0.05	4561
0.05	0.1	579
0.1	0.2	333
0.2	0.3	119
0.3	0.4	59
0.4	0.5	30
0.5	0.6	24
0.6	0.7	3
0.7	0.8	13
0.8	0.9	3
0.9	1	146

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

people for the purpose of slavery, forced labor, and servitude (United Nations 2001). Examples of human trafficking-related situations may include a man who walks around in the red light district with a large amount of cash money, or a mail driver who drive the ID papers of the girls in his car, etc. In our data, the 22,672 observational reports are used as objects, the 28,203 persons mentioned in them are used as attributes, and the human trafficking indicators observed for this persons are used as conditions. The result of the triclustering execution on this data are communities of people. These communities may contain criminals who are operating together, but also may reveal which women are potential victims of these communities. Dense communities are persons who are very frequently seen together, e.g. a pimp and his prostitute. Each of these communities is a second component of a certain tricluster. Nonedense communities are persons who are rarely observed to have contact with each other. In our preliminary experiments on specially preselected reports (number of reports – $|G| = 22,672$, number of persons – $|M| = 28,203$, and number of indicators – $|B| = 15$), we have revealed 5859 conceptual triclusters, see their density distribution for $\rho_{min} = 0$ in Table 7. The police experts noticed that for high-density thresholds, our approach results in rather well-interpreted communities. Unfortunately, due to confidentiality reasons, we restrict ourselves to this short description of preliminary experiments in mining criminal communities.

7. Conclusion

FCA-based and spectral partitioning approaches to triclustering were proposed. We showed that:

- (dense) conceptual triclustering is a good alternative for TCA because the total number of triclusters for real data example is drastically less than the number of triconcepts,
- (dense) conceptual triclustering is able to cope with a huge number of triconcepts in the worst cases of tricontexts (or dense cuboids in them) where only their main diagonal is empty and considers such cuboids as a whole tricluster; it is very relevant property for mining tri-communities in social bookmarking systems,
- the proposed algorithm has good scalability on real-world data especially when we apply greedy covering approach and optimized version of the density calculation procedure, and

- spectral hierarchical partitioning of ternary relations can be a good alternative to conceptual triclustering as a tool for fast exploratory (preliminary) analysis.

However, in our study there is no one leading algorithm with reference to all the quality measures. Therefore, we would like better describe typical circumstances in which a particular triclustering method can be applied. There are some methods which seem to be appropriate for deeper evaluation, e.g. TriBox (Mirkin and Kramarenko 2011) and DataPeeler (Cerf et al. 2008), and some possible modification of TRICL, i.e. conceptual triclustering.

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

(1) Improvement and evaluation of the proposed approaches:

- 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,
- developing a unified theoretical framework for triclustering and multimodal clustering based on closed sets,
- developing a multicriteria system of quality measures for triclustering and multimodal clustering (density, diversity, coverage, etc.), and
- taking into account the nature of real-world data for optimization (their sparsity, value distribution, etc.).

(2) New experiments in various application areas where we have made some first steps:

- extracting communities of criminals operating in Amsterdam–Amstelland police region from unstructured observational police reports (Poelmans et al. 2012),
- finding tricommunities in the massive amount of unstructured texts resulting from brainstorm sessions (in collaboration with the Witology company) (Ignatov et al. 2013),
- automatically identifying suitable descriptors for groups on social network sites based on the interests which users indicated on their profile (Gnatyshak et al. 2012), and
- developing Collaborative Filtering algorithms for folksonomy-like data.

There are some prospective ways to adapt the proposed FCA-based approach in rough-set setting using ideas on bireducts from (Slezak and Janusz 2011) and rough clustering from (Lingras and Peters 2011).

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Acknowledgements

The study was implemented in the framework of the Basic Research Program at the National Research University Higher School of Economics in 2012 and in the Laboratory of Intelligent Systems and Structural Analysis. At the early stage, this work was also partially supported by the Russian Foundation for Basic Research, project No. 08-07-92497-NTSNIL_a.

We would like to thank our colleagues Paul Elzinga (Amsterdam–Amstelland Police) who allowed us to experiment with the police data, Ruslan Magizov (NRU HSE) for his programming activity at the early stages of the research, and Zarina Sekinaeva who developed the program for SpecTric algorithm as a part of her master's project. We are specially thankful to Dmitry Gnatyshak for his programming and experimental support and to Dominik Ślęzak for his patience and willingness to constantly stimulate us during the research.

Notes on contributors

Dmitry I. Ignatov works as an Associate Professor for the National Research University Higher School of Economics (Moscow, Russia) at the chair of Artificial Intelligence and Data Analysis. Dr. Dmitry Ignatov graduated in 2004 as a "Specialist in Physics and Mathematics" with distinction at the "Kolomna Teachers' Training Institute" (Kolomna, Russia) and in 2008 as a "Master of Applied Mathematics and Information Sciences" at the "State University Higher School of Economics" (Russia, Moscow). In 2010 he obtained his degree of "Candidate of sciences in Mathematical Modeling, Numerical Methods, and Software Systems" at the "National Research University Higher School of Economics". He did his PhD (Candidate of science in Russian) research in All-Russian Institute for Scientific and Technical Information of Russian Academy of sciences specializing in Theoretical Computer Science. He is an author of more than 35 papers published in peer reviewed conferences, workshops and journals. His main interests include Formal Concept Analysis, Data Mining and Machine Learning, especially biclustering and multimodal clustering. He was a co-organizer or a program co-chair of several international conferences and workshops: ECIR 2013, ICFCA 2012, ICCS 2009 and 2013, RSFDGrC 2011, PReMI 2011, CDUD 2011 and 2012, EEML 2012, SCAKD 2011 etc.

Sergei O. Kuznetsov is the head of Applied Mathematics and Information Science Department at the National Research University Higher School of Economics in Moscow, Russian Federation since 2006. In 1985, he graduated from the Moscow Institute for Physics and Technology, Department of Applied Mathematics and Control. In 1990, he obtained his Candidate of Science Degree (PhD equivalent), dissertation "Improving Performance of Expert Systems". In 2002, he obtained his Doctor of Science Degree (Habilitation) for "Theory of Learning in Concept Lattices". Between 1985 and 2006, he was (Senior) Researcher at the All-Russia Institute for Scientific and Technical Information (VINITI), Moscow. Between 1999 and 2004, he was Humboldt Fellow and invited professor at the Dresden Technical University.

Jonas Poelmans graduated in 2007 as a "Master in Computer Science" with distinction at the "Katholieke Universiteit Leuven". In 2010, he obtained his degree of "Doctor in Applied Economics" at the "Katholieke Universiteit Leuven". He was Conference Chair of the 10th International Conference on Formal Concept Analysis in 2012 and Program Co-chair of International Conference on Conceptual Structures in 2013. He authored more than 45 papers published in international peer-reviewed conferences and journals. He won twice the best paper award at the 'Industrial Conference on Data Mining' and edited several conference and workshop proceeding volumes. His main interests include Formal Concept Analysis and Data Mining.

Leonid E. Zhukov is a Professor for National Research University Higher School of Economics (Moscow, Russia) at the Department of Applied Mathematics and Informatics. He obtained his BS and MS Diploma (summa cum laude) in Theoretical Physics from the University "Moscow Engineering and Physics Institute" in 1993. He also graduated from Department of Computer Science, University of Utah specializing in Computational Engineering and Science Program in 1998. He obtained his PhD degree in Theoretical Physics from Department of Physics, University of Utah in 1998. He worked for different universities and companies: California Institute of Technology, University of Utah, Yahoo! Research Labs at Yahoo! Inc, Jumptap Inc (Boston, MA), Mobile Content Networks (MCN) Inc (Mountain View, CA), Masterhost/SpyLog (Moscow, Russia). He is a co-founder and CTO of Trafica company (Moscow, Russia). He authored many high quality scientific papers in different domains: Biomedical Visualization, Data mining, Information Visualization, Scientific computing, Information Retrieval, Social Network Analysis, etc.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

References

Agrawal, R., T. Imielinski, and A. N. Swami. 1993. "Mining Association Rules between Sets of Items in Large Databases." In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, edited by P. Buneman and S. Jajodia, 207–216. Washington, DC: ACM Press, May 26–28, 1993.

Belohlávek, R. 2001. "Lattices of Fixed Points of Fuzzy Galois Connections." *Mathematical Logic Quarterly* 47 (1): 111–116.

Belohlávek, R. 2011. "What is a Fuzzy Concept Lattice? II." In *RSFDGrC, Vol. 6743 of Lecture Notes in Computer Science*, edited by S. O. Kuznetsov, D. Slezak, D. H. Hepting and B. Mirkin, 19–26. Berlin: Springer.

Belohlavek, R., and V. Vychodil. 2009. *Factor Analysis of Incidence Data via Novel Decomposition of Matrices*. ICFCA '09. Darmstadt: Springer-Verlag, 83–97.

Belohlavek, R., and V. Vychodil. 2010. *Factorizing Three-Way Binary Data with Triadic Formal Concepts*. KES'10. Cardiff: Springer-Verlag, 471–480.

Besson, J., C. Robardet, and J. F. Boulicaut. 2006. "Mining a New Fault-Tolerant Pattern Type as an Alternative to Formal Concept Discovery." In *Conceptual Structures: Inspiration and Application., Vol. 4068 of Lecture Notes in Computer Science*, edited by H. Scharfe, P. Hitzler and P. Ohrstrom, 144–157. Berlin: Springer.

Cerf, L., J. Besson, C. Robardet, and J. F. Boulicaut. 2008. "Data Peeler: Contraint-Based Closed Pattern Mining in n-ary Relations." *SIAM* : 37–48.

Cerf, L., J. Besson, C. Robardet, and J. F. Boulicaut. 2009. "Closed Patterns Meet n-ary Relations." *ACM Transactions on Knowledge Discovery from Data* 3 : 3:1–3:36.

Cheng, D., R. Kannan, S. Vempala, and G. Wang. 2003. *On a Recursive Spectral Algorithm for Clustering from Pairwise Similarities*. Technical report MIT-LCS-TR-906, MIT LCS.

Dhillon, I. S. 2001. *Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning*. KDD '01. San Francisco, CA: ACM, 269–274.

Ding, C. H. Q., X. He, H. Zha, M. Gu, and H. D. Simon. 2001. *A Min-max Cut Algorithm for Graph Partitioning and Data Clustering*. ICDM '01. Washington, DC: IEEE Computer Society, 107–114.

Drineas, P., A. Frieze, R. Kannan, S. Vempala, and V. Vinay. 1999. "Clustering in large graphs and matrices. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms (SODA '99)*, edited by R. E. Tarjan and T. Warnow, 291–299. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Fiedler, M. 1973. "Algebraic Connectivity of Graphs." *Czechosloval Mathematical Journal* 23 (98): 298–305.

Ganter, B., and S. O. Kuznetsov. 2008. "Scale Coarsening as Feature Selection." In *ICFCA, Vol. 4933 of Lecture Notes in Computer Science*, edited by R. Medina and S. A. Obiedkov, 217–228. Berlin: Springer.

Ganter, B., and R. Wille. 1999. *Formal Concept Analysis: Mathematical Foundations*. 1st ed. Secaucus, NJ: Springer-Verlag New York, Inc.

Gao, B., et al. 2005. *Consistent Bipartite Graph Co-partitioning for Star-Structured High-Order Heterogeneous Data Co-clustering*. KDD '05. Chicago, IL: ACM, 41–50.

Gnatyshak, D., D. I. Ignatov, A. Semenov, and J. Poelmans. 2012. "Gaining Insight in Social Networks with Biclustering and Triclustering." In *BIR, Vol. 128 of Lecture Notes in Business Information Processing*, edited by N. Aseeva, E. Babkin and O. Kozyrev, 162–171. Berlin: Springer.

Golub, G., and C. van Loan. 1989. *Matrix Computations*. Baltimore, MD: The John Hopkins University Press.

Hartigan, J. A. 1972. "Direct Clustering of a Data Matrix." *Journal of the American Statistical Association* 67 (337): 123–129.

Ignatov, D., A. Kaminskaya, S. Kuznetsov, and R. Magizov. 2010. *A Concept-Based Biclustering Algorithm* (in Russian). Moscow: MAKS Press, 140–143.

Ignatov, D. I., A. Y. Kaminskaya, A. A. Bezzubtseva, A. V. Konstantinov, and J. Poelmans. 2013. "FCA-Based Models and a Prototype Data Analysis System for Crowdsourcing Platforms." In *ICCS, Vol. 7735 of Lecture Notes in Computer Science*, edited by H. D. Pfeiffer, D. I. Ignatov, J. Poelmans and N. Gadiraju, 173–192. Berlin: Springer.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Ignatov, D. I., S. O. Kuznetsov, and J. Poelmans. 2012. “Concept-Based Biclustering for Internet Advertisement.” In *ICDM Workshops IEEE Computer Society*, edited by J. Vreeken, C. Ling, M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb and X. Wu, 123–130.

Jäschke, R., R. Hotho, A. Schmitz, C. Ganter, and B. Stumme. 2006. *TRIAS-An Algorithm for Mining Iceberg Tri-Lattices*. ICDM ’06. Washington, DC: IEEE Computer Society, 907–911.

Jäschke, R., A. Hotho, C. Schmitz, and G. Stumme. 2007. “Analysis of the Publication Sharing Behaviour in BibSonomy.” In *Proceedings of the 15th International Conference on Conceptual Structures (ICCS 2007), Vol. 4604 of Lecture Notes in Artificial Intelligence*, edited by U. Priss, S. Polovina and R. Hill, 283–295. Berlin: Springer-Verlag.

Kannan, R., S. Vempala, and A. Veta. 2004. “On Clusterings-Good, Bad and Spectral.” *Proceedings 41st Annual Symposium on Foundations of Computer Science* 51 (3): 497–515.

Kaytoue, M., S. Kuznetsov, J. Macko, and A. Napoli. in press. “Biclustering Meets Triadic Concept Analysis.” *Annals of Mathematics and Artificial Intelligence*.

Kaytoue, M., S. O. Kuznetsov, A. Napoli, and S. Duplessis. 2011. “Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis.” *Information Science* 181 (10): 1989–2001.

Klimushkin, M., S. Obiedkov, and C. Roth. 2010. “Approaches to the Selection of Relevant Concepts in the Case of Noisy Data.” In *Formal Concept Analysis., Vol. 5986 of Lecture Notes in Computer Science*, edited by L. Kwuida and B. Sertkaya, 255–266. Berlin: Springer.

Kuznetsov, S. O. 1990. “Stability as an Estimate of the Degree of Substantiation of Hypotheses Derived on the Basis of Operational Similarity.” *Nauchn Tekh Inform Series 2 (Automatic Documentation and Mathematical Linguistics)* 12 : 21–29.

Kuznetsov, S. O. 2007. “On Stability of a Formal Concept.” *Annals of Mathematics and Artificial Intelligence* 49 (1–4): 101–115.

Kuznetsov, S. 1989. “Interpretation on Graphs and Complexity Characteristics of a Search for Specific Patterns.” *Nauchno-Tekhnicheskaya Informatsiya, Seriya 2* 23 (1): 23–27.

Kuznetsov, S. 1996. “Mathematical Aspects of Concept Analysis.” *Journal of Mathematical Science* 80 (2): 1654–1698.

Lehmann, F., and R. Wille. 1995. *A Triadic Approach to Formal Concept Analysis*. London: Springer-Verlag.

Lingras, P., and G. Peters. 2011. “Rough Clustering.” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1 (1): 64–72.

Liu, K., B. Fang, and W. Zhang. 2010. “Unsupervised Tag Sense Disambiguation in Folksonomies.” *JCP* 5 (11): 1715–1722.

Madeira, S. C., and A. L. Oliveira. 2004. “Biclustering Algorithms for Biological Data Analysis: A Survey.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1 (1): 24–45.

Miettinen, P. 2011. “Boolean Tensor Factorization.” In *ICDM 2011, 11th IEEE International Conference on Data Mining, edited by D. Cook, J. Pei, W. Wang, O. Zaïane and X. Wu, 447–456*. Vancouver: CPS.

Mirkin, B. 1996. *Mathematical Classification and Clustering*. Dordrecht: Kluwer Academic Press.

Mirkin, B. 2005. *Clustering For Data Mining: A Data Recovery Approach (Chapman & Hall/Crc Computer Science)*. London: Chapman & Hall/CRC.

Mirkin, B. G., and A. V. Kramarenko. 2011. *Approximate Bicluster and Tricluster Boxes in the Analysis of Binary Data*. RSFDGrC’11. Moscow: Springer-Verlag, 248–256.

Nanopoulos, A., H. H. Gabriel, and M. Spiliopoulou. 2009. “Spectral Clustering in Social-Tagging Systems.” In *WISE, Vol. 5802 of Lecture Notes in Computer Science*, edited by G. Vossen, D. D. E. Long and J. X. Yu, 87–100. Berlin: Springer.

Newman, M. E. J. 2005. “Power Laws, Pareto Distributions and Zipf’s Law.” *Contemporary Physics* 46 (5): 323–351.

Ng, A. Y., M. I. Jordan, and Y. Weiss. 2001. *On Spectral Clustering: Analysis and an Algorithm*. Cambridge, MA: MIT Press, 849–856.

Pasquier, N., Y. Bastide, R. Taouil, and L. Lakhal. 1999. “Efficient Mining of Association Rules Using Closed Itemset Lattices.” *Information Systems* 24 (1): 25–46.

Pei, J., J. Han, and R. Mao. 2000. “CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets.” 21–30.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Poelmans, J., P. Elzinga, D. I. Ignatov, and S. O. Kuznetsov. 2012. "Semi-Automated Knowledge Discovery: Identifying and Profiling Human Trafficking." *International Journal of General Systems* 41 (8): 774–804.

Shi, J., and J. Malik. 2000. "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8): 888–905.

Slezak, D., and A. Janusz. 2011. "Ensembles of Bireducts: Towards Robust Classification and Simple Representation." In *FGIT, Vol. 7105 of Lecture Notes in Computer Science*, edited by T. H. Kim, H. Adeli, D. Slezak, F. E. Sandnes, X. Song, K. I. Chung and K. P. Arnett, 64–77. Berlin: Springer.

Stumme, G., et al. 2002. "Computing Iceberg Concept Lattices with Titanic." *Data & Knowledge Engineering* 42 (2): 189–222.

Tsymbal, A., M. Pechenizkiy, and P. Cunningham. 2005. "Diversity in Search Strategies for Ensemble Feature Selection." *Information Fusion* 6 (1): 83–98.

Vander Wal, T. 2007. "Folksonomy Coinage and Definition." Accessed March 12, 2012. http://vanderwal.net/folksonomy.html

Verma, D., and M. Meila. 2003. *A Comparison of Spectral Clustering Algorithms.* University of Washington Tech Rep UWCSE030501, 1 (03-05-01), 1–18.

Voutsadakis, G. 2002. "Polyadic Concept Analysis." *Order* 19 (3): 295–304.

Wille, R. 1995. "The Basic Theorem of Triadic Concept Analysis." *Order* 12 : 149–158.

Zaki, M. J., and C. J. Hsiao. 2002. "CHARM: An Efficient Algorithm for Closed Itemset Mining." In edited by R. L. Grossman, J. Han, V. Kumar, H. Mannila and R. Motwani. SDM SIAM.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

Appendix 1

Basic spectral partitioning definitions and methodology

Spectral graph partitioning problem can be formulated as a minimization of some objective function through the solution of a corresponding eigenvalue problem. The second smallest eigenvalue provides the lower bound on the value of optimization function and the second eigenvector (Fiedler 1973) is used to construct the partitioning of the graph.

Consider a graph $G = (V, E)$ consisting of a set of vertices $V = \{v_1, v_2 ...\}$ and a set of edges with edge weight e_{ij} connecting those vertices. It can be represented using an adjacency matrix

$$\mathbf{E} = \begin{cases} e_{ij}, & if\ \{v_i, v_j\} \in E \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Since our graphs are undirected, the adjacency matrix is always symmetric.

The vertex set of a graph can be divided into two groups according to some properties, thus inducing a graph partitioning. The degree of dissimilarity between the two subgraphs can be computed as a total weight (sum of all weights) of the edges that have been removed during the *cut* :

$$cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} e_{ij} \tag{14}$$

When using graph partition for clustering, we usually value big clusters more than small ones and would like to prevent cutting off singletons or very small subgraphs. This can be achieved by augmenting the cut value with a normalization procedure. Then, the objective function can be written as

$$Q(V_1, V_2) = \frac{cut(V_1, V_2)}{W(V_1)} + \frac{cut(V_1, V_2)}{W(V_2)} \tag{15}$$

where, W is the sum of the weights of all nodes in the partition

$$W(V) = \sum_{i \in V} w_i \tag{16}$$

Various choices of node weight function W lead to different partition criteria and clustering results. Normalized Cuts criterion was proposed by Shi and Malik (2000) for image segmentation problems and gained further popularity in general clustering algorithms. In Normalized Cuts, weight of every vertex is chosen to be the sum of the weights of incident edges, i.e. $w(i) = \sum_k e_{ik}$.

The graph cut value can be expressed through a partitioning indicator vector x taking integer values of ± 1 depending on the partition side the corresponding node belongs to

$$cut(V_1, V_2) = \frac{1}{4} x^T (\mathbf{D} - \mathbf{E}) x \tag{17}$$

where, $D_{ii} = \sum_j E_{ij}$.

This leads to the following expression for the Normalized Cuts objective function

$$Q = \frac{x^T (\mathbf{D} - \mathbf{E}) x}{x^T \mathbf{D} x} \tag{18}$$

This is a nonlinear integer optimization problem that is NP-hard. Instead of solving the problem exactly, Fiedler (1973) proposed to look for an approximate solution by relaxing the integer constraints and allowing the elements of vector x to take real values.

The minimum of the objective function Q will be reached on an eigenvector (Fiedler vector), corresponding to the second smallest eigenvalue of the problem

$$(\mathbf{D} - \mathbf{E}) x = \lambda \mathbf{D} x \tag{19}$$

The solution vector x can then be used to partition the graph by placing the nodes with greater than zero $x(i)$ values into one partition and those with less than zero values into another. Every partition can then be recursively repartitioned by setting up a new eigenproblem for the corresponding submatrix.

The standard matrix diaganolization methods require $O(n^3)$ operations, where n is the number of nodes in the graph, and are impractical here. We can take advantage of the sparsity of the graph using iterative methods (Lanczoc or Arnoldi algorithms, see for descriptions, e.g., textbook (Golub and van Loan 1989)), especially since only one vector should be computed. The complexity of Lanczos type algorithm is only $O(km)$, where m is the number of edges in the graph and k is the number of iterations required for the convergence (see (Golub and van Loan 1989; Shi and Malik 2000)). In practice, it is usually $k \ll \sqrt{n}$.

Further discussion and details of the spectral graph partitioning method for clustering can be found in Ding et al. (2001), Ng, Jordan, and Weiss (2001), Verma and Meila (2003), Kannan, Vempala, and Veta (2004) and Drineas et al. (1999).

Spectral method for bipartite graphs

Let M be a bipartite graph corresponding to a user-resource matrix A with m users and n resources (in our case $m \ll n$)

$$\mathbf{M} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \tag{20}$$

In this ordering, the first m nodes in the graph are interpreted as users and the last n nodes correspond resources.

Written out component-wise according to definition (see Equation (19)), the eigensystem becomes (Dhillon 2001) :

$$\begin{pmatrix} D_1 & -A \\ -A^T & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{21}$$

Downloaded by [Higher School of Economics] at 04:49 11 June 2013

where $D_{1,ii} = \sum_j A_{ij}$ and $D_{2,jj} = \sum_i A_{ij}$ are diagonal matrices with diagonals equal to corresponding sums of rows and columns of matrix A. The solution vector can be thought of as a sequence of m nodes (subvector x) corresponding to resources followed by n nodes (subvector y) corresponding to users $z = (x, y)^T$.

Downloaded by [Higher School of Economics] at 04:49 11 June 2013