Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 31 (2014) 928 – 938
Procedia Computer Science

2nd International Conference on Information Technology and Quantitative Management, ITQM 2014

Concept Learning from Triadic Data

Roman Zhuk[a,*], Dmitry I. Ignatov[a], Natalia Konstantinova[b]

[a]*National Research University Higher School of Economics, Bolshoy Trehsvyatitelsky 3, Moscow, Russia*
[b]*University of Wolverhampton, MB 108, Stafford Street, Wolverhampton, WV1 1SB, United Kingdom*

Abstract

We propose extensions of the classical JSM-method and the Naïve Bayesian classifier for the case of triadic relational data. We performed a series of experiments on various types of data (both real and synthetic) to estimate quality of classification techniques and compare them with other classification algorithms that generate hypotheses, e.g. ID3 and Random Forest. In addition to classification precision and recall we also evaluated the time performance of the proposed methods.

Selection and peer-review under responsibility of the Organizing Committee of ITQM 2014.
Keywords: Triadic data, Classification, Formal Concept Analysis, JSM-method

1. Introduction

During the last 5-7 years mining of triadic data has attracted attention of scientists working with social Web-services like Bibsonomy [1,2,3,4,5,6,7], which use triadic nature of the data (users, tags, resources). The main results were obtained in the framework of unsupervised learning, namely triadic clustering, whereas, the classification task for triadic data was a missing link. It is worth noting that in 2008 and 2009 Bibsonomy owners organised a series of international competitions on spam detection (classification problem) and recommending tags and resources on triadic data. However, the winners of the competition mainly used the content information. The best results were achieved by employing SVM, however the triadic nature of data was not used. In this paper we try to bridge the gap and conduct missing experiments on the real data.

Thus, we extend conventional JSM-method [8,9,10,10] to the triadic case and propose appropriate modification of the Naïve Bayes classifier. We investigated the method applicability for the Bibsonomy data in the spam detection task, conducted general experiments to analyse methods' behaviour on different types of data sets in terms of accuracy and performance.

The paper will describe several algorithms for a classification task on triadic labeled data and a series of experiments with them on both synthetic and real datasets. The structure of the paper is the following: section 2.1 introduces basic FCA notions, section 2.2 describes an extension of FCA to triadic case, section 3 introduces the task of triadic data

* Corresponding author.
E-mail address: zhukrv@mail.ru,dignatov@hse.ru,n.konstantinova@wlv.ac.uk

Selection and peer-review under responsibility of the Organizing Committee of ITQM 2014.
doi:10.1016/j.procs.2014.05.345

classification and presents main approaches that we designed to this end, section 4 describes datasets and results of the experiments, and, finally, 5 concludes the paper.

2. Basic FCA notions

2.1. FCA for dyadic case

First, we recall some basic notions from the Formal Concept Analysis (FCA)[11]. 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*:

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

If we have several contexts, the derivative operator 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* $\underline{\mathfrak{B}}(G, M, I)$.

2.2. Triadic Formal Concept Analysis

A *triadic context* $\mathbb{K} = (G, M, B, Y)$ consists of sets G (objects), M (attributes), and B (conditions), and ternary relation $Y \subseteq G \times M \times B$[12]. An incidence $(g, m, b) \in Y$ shows that object g has 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 dyadic contexts

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

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

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

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

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

3. Main algorithms for triadic classification

Let a set of objects G be split into three partitions by some target attribute t. The first set includes all the objects that are known to have a target attribute t, the second one consists of those objects that do not have a t, and the third contains objects with unknown status of presence of an t attribute. The first set is called the set of positive examples of objects, or ¡¡+¿¿-class, the second one is a set of negative examples, ¡¡-¿¿-class, the third one is a set of undetermined examples. Therefore a classification task constitutes in defining which of the first two classes undetermined examples belong to.

In addition to a set of objects and a target attribute, we know sets of attributes and conditions for each object (i.e. so called structural attributes and conditions). Then the task can be described in terms of Formal Concept Analysis:

- A positive context $K_+ = (G_+, M, B, I_+)$ describes a positive set of examples
- A negative context $K_- = (G_-, M, B, I_-)$ describes a negative set of examples.
- An undetermined set of examples is described by $K_\tau = (G_\tau, M, B, I_\tau)$ context.

An incidence relation $I_\varepsilon \subseteq G_\varepsilon \times M \times B, \varepsilon \in \{+, -, \tau\}$ determines structural attributes and conditions for each object from the corresponding class. For each context there is its own Galois operator, which we denote as $(.)^+$ $(.)^-$, and $(.)^\tau$.

In figure 1 we provide a basic example of triadic data classification which was inspired by Kaggle competition Dogs vs. Cats.

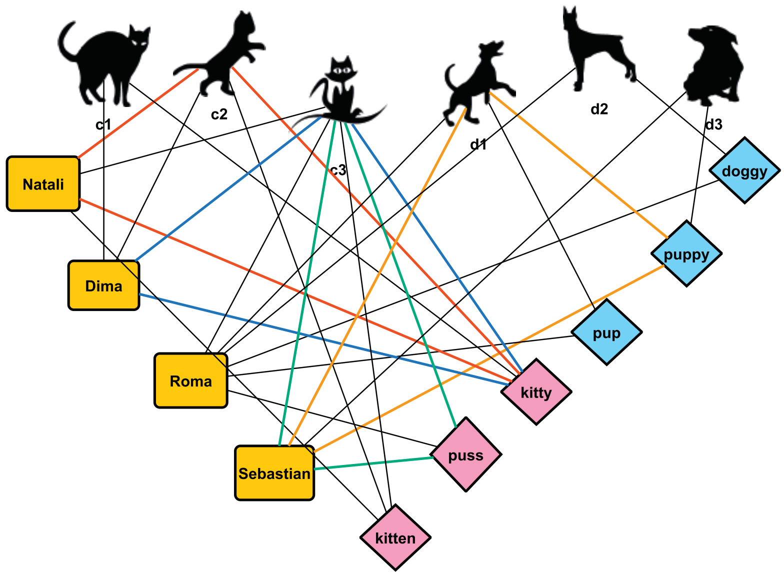

Fig. 1: Dogs vs. Cats triadic classification

ASIRRA (Animal Species Image Recognition for Restricting Access) is a Human Interactive Proof that works by asking users to identify photographs of cats and dogs. Usually, this task is difficult for computers, but people can accomplish it quickly and accurately. In this example users have assigned tags to pictures of dogs and cats. Thus we can employ these tag assignments for cat's and dog's classes done by humans to learn how to classify new undetermined example. We will be predicting whether a picture features a dog or a cat using hypotheses in the form $Set\_of\_people \times Set\_of\_tags$.

This example explains basic idea of triadic classification. We have a set of positive examples, say cats, $G_+ = \{c_1, c_2, c_3\}$, and a set of negative examples, dogs, $G_- = \{d_1, d_2, d_3\}$. A set of people, $M = \{Natali, Dima, Roma, Sebastian\}$ and a set of tags which users assigned to examples, $B = \{kitty, kitten, puss, doggy, pup, puppy\}$. We highlighted some hyperedges on the example graph in figure 1; for example, $Natali$ assigned tag $kitty$ to cat c_2 and $Sebastian$ assigned tag $puppy$ to d_1. Now we have to classify new undetermined examples using hypotheses generated from the sets of people and tags related to positive and negative examples.

3.1. Triadic JSM (weighted)

JSM-method was named in honor of English philosopher John Stuart Mill, who studied schemes of inductive reasoning in the 19th century[13], and was proposed by Viktor K. Finn in late 1970s. This method aims to describe induction in a purely deductive form and give at least partial justification of induction[14,15]. The method was later reformulated in FCA terms and considered as a machine learning technique for learning hypotheses from labeled data (see detailed survey in[16]).

The workload of JSM-method can be split into two phases: learning (training) and classification.

Classification model is based on generic principles of learning by positive and negative examples: for given sets of positive and negative examples we need to find classification hypotheses that cannot cover examples of the contrary class.

So, we have three formal tricontexts: the positive context $K_+ = (G_+, M, B, I_+)$, the negative context $K_- = (G_-, M, B, I_-)$, and the context of undetermined examples $K_\tau = (G_\tau, M, B, I_\tau)$. The learning process is based on K_{--}, K_+ and its results are used for classification of objects from K_τ.

A triple of sets (A_+, D_+, C_+), where $A_+ \subseteq G_+, D_+ \subseteq M, C_+ \subseteq B$, is called a positive formal triconcept if it is the formal triconcept of a context $\mathbb{K}_+$. A set A_+ is called positive formal extent, D_+ is a positive formal intent, C_+ is a positive formal modus. If a positive formal intent D_+ and apositive formal modus C_+ are not contained in any intent and modus of negative examples (i.e. $\forall g_- \in G_-, \{D_+ \times C_+\} \notin \{g_-\}^-$), then the pair (D_+, C_+) is called a positive (+)-hypothesis. A set A_+ is called a formal extent of hypothesis (D_+, C_+). If pair (D_+, C_+) does not fulfill the condition $\forall g_- \in G_-, \{D_+ \times C_+\} \notin \{g_-\}^-$, then this couple is called a positive (+)-falsified hypothesis. Negative hypotheses are defined in a similar way.

Hypotheses are used for classification of undetermined examples.

If an undetermined example contains a positive hypothesis (D_+, C_+) (i.e. $\{D_+ \times C_+\} \subseteq \{g_\tau^\tau\}$), then we call (D_+, C_+) a hypothesis about positive classification of the object g_τ. Similarly the hypothesis about negative classification of object g_τ is defined. The weight of hypothesis $(D_\varepsilon, C_\varepsilon)$ is a number of elements in its extent $(D_\varepsilon \times C_\varepsilon)^\varepsilon$, where $\varepsilon \in \{+, -\}$.

There is a general classification scheme:

1. Find all positive and negative hypotheses
2. For each object g_τ that needs to be classified:

   (a) Calculate a sum of weights for each class of hypotheses that object g_τ satisfies.
   (b) Classify an object as:

   - Positive, if sum of weights of positive hypotheses more than of negative ones
   - Negative, if sum of weights of negative hypotheses more than of positive ones
   - Unclassifiable, if sum of weights of hypotheses from both classes are equal

3.2. Naïve Triadic Bayes

Each example $g \in G_\tau$ is described by a set of attributes and conditions: $\langle m_1 \ldots m_n, b_1 \ldots b_k \rangle$. We have to find the most probable class C, an object with such attributes and conditions belongs to. We assume that elements $m \in M, b \in B$ are independent. Then we have to find C such that:

$$C = \arg\max_{h \in \{+,-\}} p(x = h | m_1 \ldots m_n, b_1 \ldots b_k)$$

According to Bayes theorem:

$$C = \arg\max_{h \in \{+,-\}} \frac{p(m_1 \dots m_n, b_1 \dots b_k | x = h) p(x = h)}{p(m_1 \dots m_n, b_1 \dots b_k)} = \tag{2}$$

$$= \arg\max_{h \in \{+,-\}} p(m_1 \dots m_n, b_1 \dots b_k | x = h) p(x = h) \tag{3}$$

Expand the probability $p(m_1 \dots m_n, b_1 \dots b_k | h)$ as follows:

$$p(m_1 \dots m_n, b_1 \dots b_k | h) = p(m_1|h) p(m_2 \dots m_n, b_1 \dots b_k | h, m_1) = \tag{4}$$

$$= p(m_1|h) p(m_2|h, m_1) p(m_2 \dots m_n, b_1 \dots b_k | h, m_1, m_2) = \dots = \tag{5}$$

$$= p(m_1|h) p(m_2|h, m_1) \dots p(m_n|h, m_1 \dots m_{n-1}) p(b_1 \dots b_k | h, m_1 \dots m_n) = \tag{6}$$

$$= p(m_1|h) \dots p(m_n|h, m_1 \dots m_{n-1}) p(b_1|h, m_1 \dots m_n) \dots p(b_k|h, m_1 \dots m_n, b_1 \dots b_k) = \tag{7}$$

$$= \langle \text{ we apply the fact that } m_1 \dots m_n, b_1 \dots b_k \text{ are independent } \rangle = \tag{8}$$

$$= \prod_{i=1}^{n} P(m_i|h) \prod_{i=1}^{k} P(b_i|h) \tag{9}$$

When counting probabilities as frequencies add smoothing according to Jeffrey-Perks rule:

$$p(f|h) = \frac{n + 1/2}{N + |A|/2},$$

where n is a number of objects in the class having attribute f, N is the total number of objects in the class , A is a set of attributes of the objects.

3.3. *Triadic Close-by-one*

As the main of idea of triadic formal concepts generation we exploit the two-level generation scheme of TRIAS algorithm[6] on associated dyadic contexts.

Let $I = \{(g, (m, b)) | \forall (g, m, b) \in I\}$ be a new incidence relation built on $K = (G, M, B, I)$. We can represent K by a dyadic formal context $K_2 = (G, M \times B, I)$.

If a pair (A_2, Q), where $A_2 \subseteq G, Q \subseteq M \times B$, is a formal triconcept of K_2, then there exist a triple (A_3, D, C) such that $A_3 = A_2, D \times C = Q, D \subseteq M, C \subseteq B$, which is a formal triconcept of K. Thus, by finding concepts of K_2, we generate concepts of K.

In the original TRIAS algorithm the authors use a NextClosure procedure for finding concepts of K_2. In this paper we use ¡¡Close-by-one¿¿ algorithm (CbO)[17] since it maintains a tree structure in the process of concepts generation for more reliable access to the generated concepts. This approach also benefits from its suitability for parallel computations. The original CbO allows to build a tree of canonically generated extents, but we modify it slightly since we do not need to know the order and the relationships between the concepts. Its pseudocode is presented below:

The initial parameters are $A = \emptyset$, $n = 0$, and $currdeep = 0$. W is a set of objects or attributes for a target class. The variable $FConcepts$ stores a set of generated formal concepts. The functions Add and AddRange add to $FConcepts$ one or several new formal concepts respectively. Each new recursive call of CbO is similar to a descent by one level in the tree of canonically generated formal concepts of the original CbO algorithm, therefore the parameter $maxdeep$ can control the descent depth in the tree. It can also be seen that the proposed CbO modification is not able to generate concepts with empty extents or intents, which are evidently useless for classification.

The algorithm ¡¡Close-by-one¿¿ has its dual version with respect to sets of objects and attributes. That is, if we assume $A \subseteq G, W = G$, then formal concepts are generated starting with concepts of minimal extents; similarly, if $A \subseteq M \times B, W = M \times B$, then the generation starts with minimal concept by intent and modus. This peculiarity is a beneficial feature of the algorithm since for JSM-method we enough to have only concepts with maximal (minimal by intent and modus) formal concepts. By setting the target sets $A \subseteq M \times B, W = M \times B$ and the parameter $currdeep = 1$, we obtain the concepts avoiding unnecessary computations.

Algorithm 1 CbO algorithm

Input: A is a set of objects (or attributes),
n is an object (or attribute), $maxdeep$ is a maximal tree depth, $currdeep$ is a current tree level
Output: $FConcepts$

```
if (|A| < |W|)&(currdeep < maxdeep) then
    for all i ∈ range(n, |G|) do
        if min({k|g_k ∈ (A ∪ g_i)'' \ A}) ≥ i then
            FConcepts.Add(A'', A')
            j = min({k|k > i, g_k ∉ (A ∪ g_i)''})
            FConcepts.AddRange(CbO((A ∪ g_i)'', j, currdeep + 1))
        end if
    end for
end if
return FConcepts
```

4. Data sets and experiments

4.1. Synthetic data

4.1.1. Contexts generated with normal distribution

The contexts based on normal distribution are generated according to the following procedure.

Let tricontext K define three-dimensional tensor of size $G \times M \times B$ and k be a number of clusters, then probability of belonging to the cluster K_i is $P(K_i) = 1/k$.

Inside a cluster triple coordinates are defined as follows: $(x, y, z) \sim (N(c_{ix}, \sigma^2_{ix}), N(c_{iy}, \sigma^2_{iy}), N(c_{iz}, \sigma^2_{iz}))$, where parameters of normal distributions are unique and randomly chosen for each $i - th$ cluster. If the density of the context, p, is given, then $|G| * |M| * |B| * p$ triples are generated using the aforementioned law.

4.1.2. Context with cubes and noise

On the main diagonal of tensor $G \times M \times B$, describing the initial triadic context, we have n non-overlapping cuboids of arbitrary sizes. A white noise of density p is also introduced inside the context. Next, the generated context is split into two parts: positive and negative contexts for JSM-method.

4.1.3. Context with random cubes

In a tensor $G \times M \times B$, which defines the triadic context, there are n cuboids of arbitrary sizes and positions. Also a white noise with a rather low density 0.002 is introduced.

4.1.4. Testing whether all possible hypothesis are necessary

Since the task of all concepts' generation for a given context is resource consuming, we use rather small context: in each experiment we use positive and negative context of size $50 \times 50 \times 50$. Three contexts for each class respectively were generated with the following parameters:

1. Random contexts with density 0.15
2. Contexts with 6 Gaussian clusters and 0.2 noise density
3. Context with 6 cubes and 0.1 noise density
4. Context with 8 random cubes and 0.03 noise density

Averaged results for these three types of experiments for JSM-method with weighted votes are given in Table 1. It is clear that, usage of all formal concepts does not increase classification quality. Averaged F-measure for all hypotheses equals 0.752, for maximal hypotheses it is about 0.740, almost identical result. Since there is a large speed up in the computational time for hypotheses of a maximal extent, we use only them later on in the classification framework of JSM-method with weighted voting.

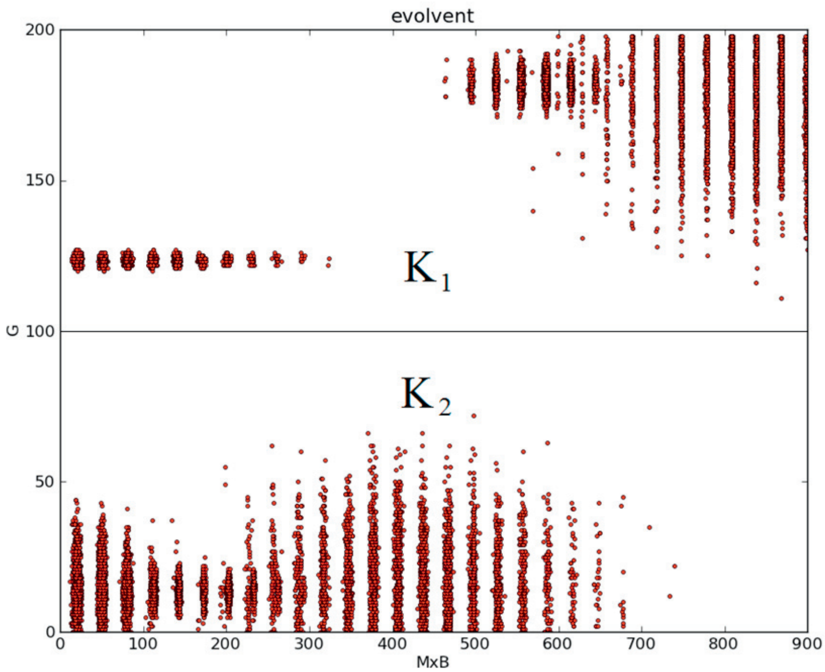

Fig. 2: Plane projection of $(G \times M)$ or 2D-evolvent of two contexts with parameters $|G| = 200$, $|M| = 30$, $|B| = 30, K = 6$, $p = 0.15$

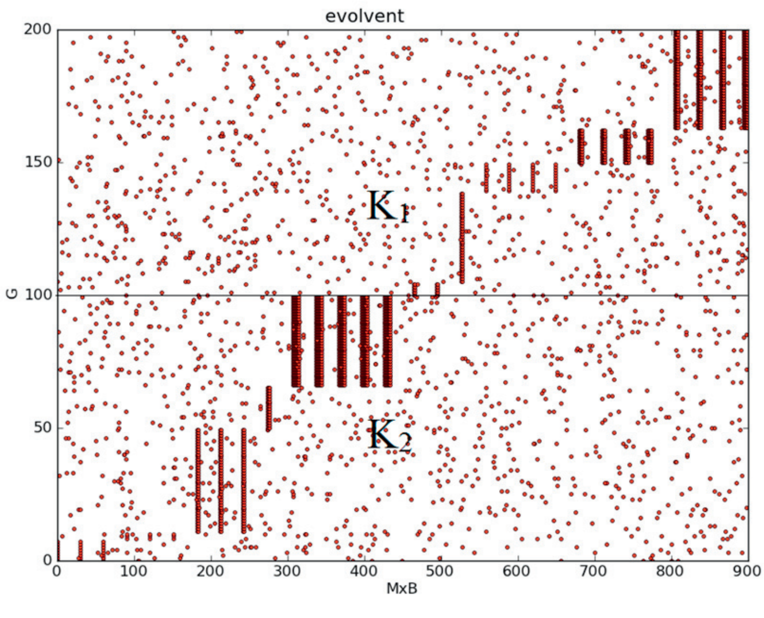

Fig. 3: Plane projection of $(G \times M)$ or 2D-evolvent of two contexts with parameters $|G| = 200$, $|M| = 30$, $|B| = 30$, $N = 10$, $p = 0.005$

4.1.5. Quality assessment

We have built 10 contexts of each type of size $250 \times 100 \times 100$; each of them consists of two equal subcontexts w.r.t. to the number of objects, with the same parameter values as in the previous example.

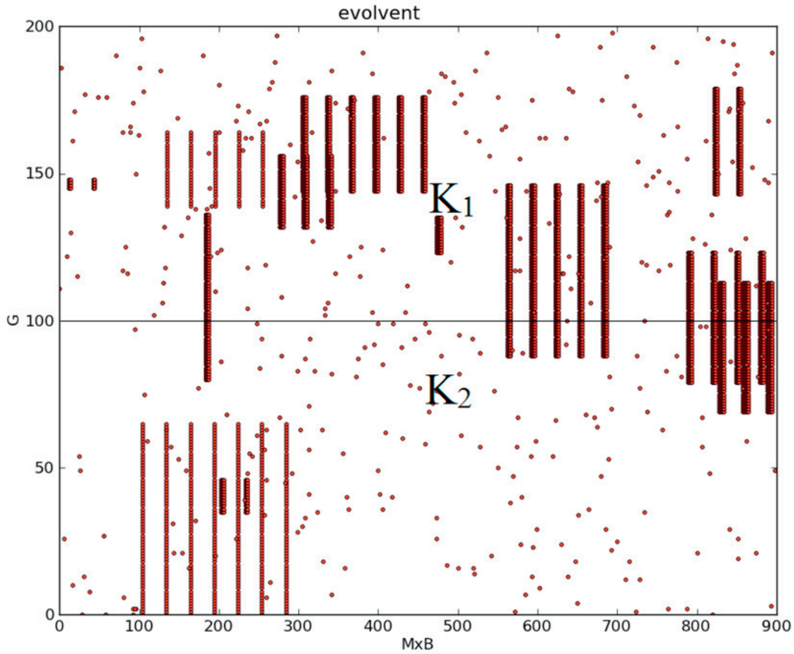

Fig. 4: Plane projection of $(G \times M)$ or 2D-evolvent of two contexts with parameters $|G| = 200$, $|M| = 30$, $|B| = 30$, $N = 8$

Table 1: Classification results of weighted JSM-method on synthetic datasets

Context type	Uniform noise		With cuboids		with rand. cuboids		With clusters	
concepts	all	maximal	all	maximal	all	maximal	all	maximal
True Positive	0,22	0.22	0.5	0.5	0.25	0.20	0.32	0.32
False Positive	0.23	0.27	0	0	0.08	0.08	0	0.03
True Negative	0.26	0.23	0.5	0.5	0.42	0.42	0.45	0.42
False Negative	0.29	0.25	0	0	0.25	0.28	0	0
Unknown	0	0.03	0	0	0	0.02	0.23	0.23

One fifth part of each context was taken as a test, the training was performed on remaining context objects. This rather large number of experiments was performed since we would like to have enough statistics about the behaviour of different methods on different context types. We also include classification methods based on decision trees from machine learning package Weka.

Averaged results for F-measure and a fraction of unclassified objects in 10 experiments for each method and context type are presented in Table 2.

All the methods perfectly classified data with cuboids. The methods show the worst results, as we expected, on uniform contexts. Also rather poor results were demonstrated on random (possibly overlapping) cuboids.

4.2. *Bibsonomy data*

We have also conducted experiments on real data of `bibsonomy.org`, which was provided to us during ECML PKKD Discovery Challenge in 2008 [18]. BibSonomy allows to share reference lists and assign tags to books and papers. In the data set objects consist of bibsonomy users, the set of attributes is a set of tags, and conditions are papers (i.e. id), the target attribute is a label, which indicates whether a given user is a spammer (bot-spammer) or non-spammer (an ordinary human user). Thus, this data gives rise to a triadic context with a target attribute spammer-

Table 2: Classification results of JSM-methods and other machine learning methods on synthetic datasets

Method	Context	F-measure
JSM	Uniform	0.5
	clusters	0.92
	cuboids	1
	rand. cuboids	0.6
Naïve Bayes	Uniform	0.54
	clusters	0.92
	cuboids	1
	rand. cuboids	0.6
Decision trees Id3	Uniform	0.52
	clusters	0.92
	cuboids	1
	rand. cuboids	0.84
Decision trees Random forest	Uniform	0.52
	clusters	0.88
	cuboids	1
	rand. cuboids	0.86

Table 3: Bibsonomy dataset statistics

	$\|U\|$	$\|T\|$	$\|R\|$	Triples count	Density
Non-spammers	2467	268692	69904	816197	$1,761 * 10^{-8}$
Spammers	29248	380434	1626805	13258759	$7,324 * 10^{-10}$

Table 4: Bibsonomy random sample statistics

	$\|U\|$	$\|T\|$	$\|R\|$	Triples count	Density
Non-spammers	565	8720	29372	53129	$3,671 * 10^{-7}$
Spammers	499	17626	22362	161752	$8,224 * 10^{-7}$

non-spammer. There is an additional information for each book: url address, short textual description, label whether a user bookmarked it or not.

For the first experiment with the Bibsonomy data we have used a dataset 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)
5. date

For our purposes we need only fields 1, 2, and 3 of the tuple above. For each record in the datatable we know whether it is a spam record or not.

Considered data is rather large and highly sparse (Table 3), therefore we generate random subsamples from both sets, 500 objects from each set (Table 4). Since the number of objects in the subsamples is drastically less than the number of attributes and conditions, we decided to use all concepts generated by JSM-method; in other words, Close-by-one shows better performance starting with search of concept extents of a smaller size.

Experiment results are given in Table 5.

Table 5: Classification results of triadic versions of JSM and Naïve Bayes on Bibsonomy data

Method	Context	Value	F-measure
JSM	True Positive	0	—
	False Positive	0	
	True Negaitve	0	
	False Negative	0	
	Unknown	213	
Naïve Bayes	True Positive	98	0,9
	False Positive	5	
	True Negaitve	95	
	False Negative	15	
	Unknown	0	

Table 6: Bibsonomy data sample after metainformation fusion

	$\|U\|$	$\|T\|$	$\|R\|$	Triples count	Density
Non-spammers	565	8720	5062	67031	$2,6688 * 10^{-6}$
Spammers	499	17626	15357	323455	$2,394 * 10^{-6}$

Table 7: Classification results of triadic versions of JSM and Naïve Bayes on Bibsonomy data after metainformation fusion

Method	Context	Value	F-measure
JSM	True Positive	24	0.54
	False Positive	24	
	True Negative	92	
	False Negative	40	
	Unknown	55	
Naïve Bayes	True Positive	38	0.5
	False Positive	0	
	True Negative	100	
	False Negative	75	
	Unknown	0	

All JSM-based methods were not able to classify the objects. The explanation lies in the data peculiarities: pairs (tag, book_id) of each (spam) user are unique. Surprisingly good results were shown by Naïve Bayes classifier. It can be explained by the assumption made: all attributes (tags) are independent from conditions (content_id), which does not take into account triadic data nature. Tag sets of each class are almost unique, this implies comparatively good performance of Naïve Bayes.

Since the results of the first experiment were rather disappointing, we were seeking different ways to cope with the unique content id's. We made an assumption that even though paper ids are different, but there are, almost for sure, papers with the same content. Since the database contains some additional metainformation, we came up with an idea to use it in the classification. Each unique condition (paper id) was associated with two new conditions: url-address and bookmark/reference label, i.e. each triple (user, tag, book_id) generates two new triples (user, tag, url) and (user, tag, type). By doing so we have a new context with the following parameters (see Table 6). The increase of the size of the set B is explained by the fact that many papers feature the same url-address. As a result we have a context with the parameters described in the Table 6.

Experimental evaluation (Table 7) shows that the methods demonstrated their average values of precision, which is quite acceptable. It also worth noting that even though Naïve Bayes demonstrated the lowest F-measure, it did not leave objects unclassified. According to the results we make a conclusion that the idea of using metainformation was fruitful: the methods showed their average performance in the object classification task.

5. Conclusion

We have considered several methods of triadic data classification in this paper. We proposed a modification of Naïve Bayes classifier for the case of triadic contexts as well as two JSM method modifications. We conducted a series of experiments on the data of various types to investigate the quality of classification for a particular data type.

JSM-method with votes showed relatively good results, whereas the original JSM-method, even having high F-measure values, left a large fraction of examples unclassified.

Due to the peculiarities of the real Bibsonomy data (description of each spammer is unique in terms of tags and resources), all the classification methods showed unsatisfactory results. It was partially overcome by using meta-information as additional formal conditions.

In the future studies on the topic we plan to consider more flexible classification techniques based on OAC-triclusters[2].

Acknowledgments

The study was conducted in the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) in 2013 and 2014 and in the Laboratory of Intelligent Systems and Structural Analysis at HSE as a part of the project "Mathematical Models, Algorithms, and Software Tools for Intelligent Analysis of Structural and Textual Data". Second author was also supported by Russian Foundation for Basic Research (grant #13-07-00504).

References

1. D. Gnatyshak, D. I. Ignatov, S. O. Kuznetsov, From Triadic FCA to Triclustering: Experimental Comparison of Some Triclustering Algorithms, in: M. Ojeda-Aciego, J. Outrata (Eds.), CLA, Vol. 1062 of CEUR Workshop Proceedings, CEUR-WS.org, 2013, pp. 249–260.
2. D. I. Ignatov, S. O. Kuznetsov, J. Poelmans, L. E. Zhukov, Can triconcepts become triclusters?, International Journal of General Systems 42 (6) (2013) 572–593.
3. D. I. Ignatov, S. O. Kuznetsov, R. A. Magizov, L. E. Zhukov, From Triconcepts to Triclusters, in: S. O. Kuznetsov, D. Slezak, D. H. Hepting, B. Mirkin (Eds.), RSFDGrC, Vol. 6743 of Lecture Notes in Computer Science, Springer, 2011, pp. 257–264.
4. D. Benz, A. Hotho, R. Jäschke, B. Krause, F. Mitzlaff, C. Schmitz, G. Stumme, The social bookmark and publication management system bibsonomy – A platform for evaluating and demonstrating Web 2.0 research, VLDB J. 19 (6) (2010) 849–875.
5. R. Missaoui, L. Kwuida, Mining Triadic Association Rules from Ternary Relations, in: P. Valtchev, R. Jäschke (Eds.), ICFCA, Vol. 6628 of Lecture Notes in Computer Science, Springer, 2011, pp. 204–218.
6. R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, G. Stumme, TRIAS–An Algorithm for Mining Iceberg Tri-Lattices, in: Proceedings of the Sixth International Conference on Data Mining, ICDM '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 907–911.
7. R. Belohlávek, C. Glodeanu, V. Vychodil, Optimal Factorization of Three-Way Binary Data Using Triadic Concepts, Order 30 (2) (2013) 437–454.
8. O. Anshakov, The JSM method: A set-theoretical explanation, Automatic Documentation and Mathematical Linguistics 46 (5) (2012) 202–220.
9. B. Ganter, S. O. Kuznetsov, Hypotheses and Version Spaces, in: A. de Moor, W. Lex, B. Ganter (Eds.), ICCS, Vol. 2746 of Lecture Notes in Computer Science, Springer, 2003, pp. 83–95.
10. S. O. Kuznetsov, Mathematical aspects of concept analysis, Journal of Mathematical Science Vol. 80 (Issue 2) (1996) pp. 1654–1698.
11. B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, 1st Edition, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.
12. F. Lehmann, R. Wille, A Triadic Approach to Formal Concept Analysis, in: Proceedings of the Third International Conference on Conceptual Structures: Applications, Implementation and Theory, Springer-Verlag, London, UK, 1995, pp. 32–43.
13. J. Mill, A System of Logic. Ratiocinative and Inductive, London, 1843.
14. V. Finn, On Machine-oriented Formalization of Plausible Reasoning in F.Bacon-J.S.Mill Style., Semiotika i Informatika 20 (1983) 35101, (in Russian).
15. V. Finn, Plausible Reasoning in Systems of JSM Type, Itogi Nauki i Tekhniki. Seriya Informatika 15 (1991) 54101, (in Russian).
16. S. O. Kuznetsov, Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research, in: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis, Vol. 3626 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 196–225. doi:10.1007/11528784_11.
17. S. O. Kuznetsov, S. A. Obiedkov, Comparing performance of algorithms for generating concept lattices, J. Exp. Theor. Artif. Intell. 14 (2-3) (2002) 189–216.
18. A. Hotho, D. Benz, R. Jäschke, B. Krause (Eds.), ECML PKDD Discovery Challenge 2008 (RSDC'08), Workshop at 18th Europ. Conf. on Machine Learning (ECML'08) / 11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'08), 2008.