TRIAS — An Algorithm for Mining Iceberg Tri-Lattices

Robert Jäschke[1,2], Andreas Hotho[1], Christoph Schmitz[1], Bernhard Ganter[3], Gerd Stumme[1,2]

[1] Knowledge & Data Engineering Group, University of Kassel, Germany,
http://www.kde.cs.uni-kassel.de

[2] Research Center L3S, Hannover, Germany, http://www.l3s.de

[3] Institute for Algebra, Dresden University of Technology, Germany,
http://www.math.tu-dresden.de/∼ganter/

Abstract

In this paper, we present the foundations for mining frequent tri-concepts, which extend the notion of closed itemsets to three-dimensional data to allow for mining folksonomies. We provide a formal definition of the problem, and present an efficient algorithm for its solution as well as experimental results on a large real-world example.

1 Introduction

With the observation that *Formal Concept Analysis (FCA)* [13, 5], a mathematical theory for deriving concept hierarchies from data, provides an elegant framework for reducing the effort of mining association rules [9, 14, 12], a new research area emerged around 1999 which became known as *closed itemset mining* in the data mining community and as *iceberg concept lattices* in FCA.

Independent of this development, Formal Concept Analysis has been extended about ten years ago to deal with three-dimensional data [8]. This line of *Triadic Concept Analysis* did not receive a broad attention up to now. With the rise of *folksonomies* as core data structure of *social resource sharing systems*, however, the interest in Triadic Concept Analysis increased again.

With this paper, we initiate the confluence of both lines of research, Triadic Concept Analysis and closed itemset mining. In particular, we give a formal definition of the *problem of mining all frequent tri-concepts* (the three-dimensional version of mining all frequent closed itemsets), and present our algorithm TRIAS for mining all frequent tri-concepts of a given dataset.

The paper is organized as follows. In the next section, we introduce folksonomies and social resource sharing systems and motivate our conceptual clustering approach for this kind of data. In Section 3, we discuss Triadic Formal Concept Analysis and provide in Section 4 a formal definition of the problem of mining frequent triadic formal concepts; in Section 5, we introduce our TRIAS algorithm. Section 6 comes up with a real-world application, the folksonomy of the bookmark sharing system del.icio.us, and concludes with experiments on the effectiveness and efficiency of the algorithm.

2 Motivation: Conceptual Clustering of Folksonomies

Social resource sharing systems on the web, such as the shared photo gallery Flickr[1], the bookmarking system del.icio.us,[2] or our own system BibSonomy[3] (see Figure 1) have acquired large numbers of users within less than two years. The reason for their immediate success is the fact that no specific skills are needed for participating, and that these tools yield immediate benefit for each individual user (e.g. organizing ones bookmarks in a browser-independent, persistent fashion), without too much overhead. The core data structure of a social resource sharing system is a *folksonomy*. It consists of assignments of arbitrary catchwords – called ’tags’ – to resources by users. Thus it consists of three sets U, T, R of users, tags, and resources, resp., and a ternary relation Y between them. Folksonomies have thus one dimension more than typical basket analysis datasets (which consist of the two dimensions ‘items’ and ‘transactions’). A formal model for Folksonomies as well as an overview of related work can be found in [6].

Our algorithm solves the problem of frequent closed itemset mining for this kind of data. It will return a tri-ordered set of (frequent) triples, where each triple (A, B, C) consists of a set A of users, a set B of tags, and a set C of resources. These triples – called (frequent) *tri-concepts* in the

[1] http://flickr.com [2] http://del.icio.us [3] http://www.bibsonomy.org

sequel — have the property that each user in A has tagged each resource in C with all tags from B, and that none of these sets can be extended without shrinking one of the other two dimensions. They are thus the three-dimensional version of closed itemsets.

We can additionally impose minimum support constraints on each of the three dimensions 'users', 'tags', and 'resources'. In the dyadic case, they equal the minimum support and minimal length thresholds from association rule mining. By setting higher values, we can focus on the largest conceptual components of the folksonomy before having a more detailed look with lower thresholds.

3 Triadic Concept Analysis

Inspired by the pragmatic philosophy of Charles S. Peirce with its three universal categories [10], Rudolf Wille and Fritz Lehmann extended Formal Concept Analysis in 1995 with a third category [8]. They defined a *triadic formal context* as a quadruple $\mathbb{K} := (G, M, B, Y)$ where G, M, and B are sets, and Y is a ternary relation between G, M, and B, i. e., $Y \subseteq G \times M \times B$. The elements of G, M, and B are called *objects*, *attributes*, and *conditions*, resp, and $(g, m, b) \in Y$ is read "object g has attribute m under condition b". A *triadic concept* of $\mathbb{K}$ is a triple (A_1, A_2, A_3) with $A_1 \subseteq G$, $A_2 \subseteq M$, $A_3 \subseteq B$, and $A_1 \times A_2 \times A_3 \subseteq Y$ such that none of its three components can be enlarged without violating this condition. This is the natural extension of the definition of a formal concept[4] to the triadic case.

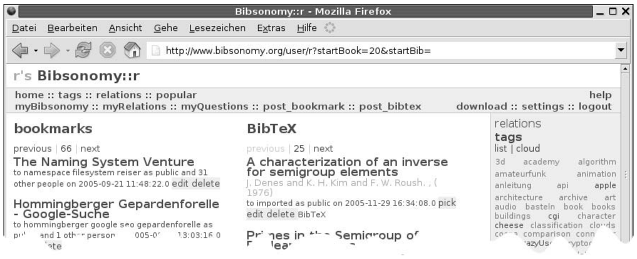
Lehmann and Wille present in [8] an extension of the theory of ordered sets and (concept) lattices to the triadic case, and discuss structural properties. This approach initiated research on the theory of *concept trilattices*.[5] With the rise of social resource sharing systems on the web, triadic data became recently interesting for many researchers. In particular, one needs knowledge discovery and information retrieval methods that are able to handle very large datasets.

In [11], we discussed how to compute association rules from a triadic context based on projections. A first step towards truly 'triadic association rules' has been done in [4].

4 The Problem of Mining all Frequent Tri-Concepts

We will now formalize the problem of mining all frequent tri-concepts. We start with an adaptation of the notion of 'frequent itemsets' to the triadic case.

Definition 1. *Let* $\mathbb{F} := (U, T, R, Y)$ *be a folksonomy/triadic context. A* tri-set *of* $\mathbb{F}$ *is a triple* (A, B, C) *with* $A \subseteq U$, $B \subseteq T$, $C \subseteq R$ *such that* $A \times B \times C \subseteq Y$.


Bibsonomy::r - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://www.bibsonomy.org/user/r?startBook=20&startBib=

r's Bibsonomy::r

home :: tags :: relations :: popular help
myBibsonomy :: myRelations :: myQuestions :: post_bookmark :: post_bibtex download :: settings :: logout

bookmarks

previous | 66 | next

The Naming System Venture
to namespace filesystem reiser as public and 31 other people on 2005-09-21 11:48:22.0 edit delete

Hommingberger Gepardenforelle - Google-Suche
to hommingberger google seo gepardenforelle as pu[illegible] and 1 oth[illegible] person [illegible] 005-0[illegible] 13:03:16.0 [illegible]lete

BibTeX

previous | 25 | next

A characterization of an inverse for semigroup elements
J. Denes and K. H. Kim and F. W. Roush. , (1976)
to imported as public on 2005-11-29 16:34:08.0 pick edit delete BibTeX

Pri[illegible]es in [illegible] Semigroup of [illegible]

relations
tags
list | cloud
3d academy algorithm amateurfunk animation anleitung api apple architecture archive art audio basteln book books buildings cgi character cheese classification clouds co[illegible] comparison conn[illegible] [illegible]azyUs[illegible] [illegible]yptor

Figure 1. BibSonomy displays bookmarks and bibliographic references simultaneously.

As folksonomies have three dimensions which are completely symmetric, one can establish minimum support thresholds on all of them. The general problem of mining frequent tri-sets is then the following:

Problem 1 (Mining all frequent tri-sets). *Let* $\mathbb{F} := (U, T, R, Y)$ *be a folksonomy/triadic context, and let u-minsup, t-minsup, r-minsup* $\in [0, 1]$. *The task of mining all frequent tri-sets consists in determining all tri-sets* (A, B, C) *of* $\mathbb{F}$ *with* $\frac{|A|}{|U|} \geq$ *u-minsup,* $\frac{|B|}{|T|} \geq$ *t-minsup, and* $\frac{|C|}{|R|} \geq$ *r-minsup.*

This is actually a harder problem than the direct adaptation of frequency to one more dimension: In classical frequent itemset mining, one has a constraint – the frequency – only on one dimension (the number of transactions). Thus the equivalent triadic version of the problem would need two minimum support thresholds only (say u-minsupp and t-minsupp). However, this seems not natural as it breaks the symmetry of the problem. Hence we decided to go for the harder problem directly (which equals in the dyadic case the addition of a minimal length constraint on the itemsets). The lighter version with only two constraints is then just a special case (e. g., by letting r-minsupp:= 0).

As in the classical case, our thresholds are antimonotonic constraints: If (A_1, B_1, C_1) with A_1 being maximal for $A_1 \times B_1 \times C_1 \subseteq Y$ is not u-frequent then all (A_2, B_2, C_2) with $B_1 \subseteq B_2$ and $C_1 \subseteq C_2$ are not u-frequent either. The same holds symmetrically for the other two directions.

With the step from two to three dimensions, however, the direct symmetry between monotonicity and antimonotonicity (which results in the dyadic case from the dual order isomorphism between the set of concept extents and the set of concept intents) breaks. All we have in the triadic case is the following lemma which results (via the three quasi-orders defined in Section 3) from the triadic Galois connection [1] induced by a triadic context.

Lemma 1 (cf. [8]). *Let* (A_1, B_1, C_1) *and* (A_2, B_2, C_2) *be tri-sets with* A_i *being maximal for* $A_i \times B_i \times C_i \subseteq Y$*, for*

[4] In terms of association rules: a closed itemset together with all its related transactions, see [13, 5] for details.

[5] See `http://www.bibsonomy.org/tag/triadic+fca`

$i = 1, 2$.[6] *If $B_1 \subseteq B_2$ and $C_1 \subseteq C_2$ then $A_2 \subseteq A_1$. The same holds symmetrically for the other two directions.*

As the set of all frequent tri-sets is highly redundant, we will in particular consider a specific condensed representation, i. e., a subset which contains the same information, namely the set of all frequent tri-concepts.

Problem 2 (Mining all frequent tri-concepts). *Let $\mathbb{F} := (U, T, R, Y)$ be a folksonomy/triadic context, and let u-minsup, t-minsup, r-minsup $\in [0, 1]$. The task of mining all frequent tri-concepts consists in determining all tri-concepts (A, B, C) of $\mathbb{F}$ with $\frac{|A|}{|U|} \geq$ u-minsup, $\frac{|B|}{|T|} \geq$ t-minsup, and $\frac{|C|}{|R|} \geq$ r-minsup.*

Sometimes it is more convenient to use absolute rather than relative thresholds. For this case we let $\tau_u := |U| \cdot$ u-minsupp, $\tau_t := |T| \cdot$ t-minsupp, and $\tau_r := |R| \cdot$ r-minsupp.

Once Problem 2 is solved, we obtain the answer to Problem 1 in a straightforward enumeration as $\{(A, B, C) \mid \exists$ frequent tri-concept $(\hat{A}, \hat{B}, \hat{C})\colon A \subseteq \hat{A}, B \subseteq \hat{B}, C \subseteq \hat{C}, |A| \geq \tau_u, |B| \geq \tau_t, |C| \geq \tau_r\}$.

5 The TRIAS Algorithm for Mining all Frequent Tri-Concepts

Our algorithm for mining all frequent tri-concepts of a folksonomy $\mathbb{F} := (U, T, R, Y)$ is listed as Algorithm 5.1. A prior version was used for analysing psychological studies [7]. That application varied from TRIAS as it aimed at an iterative pruning of the data set. Furthermore, it did not take into account any frequency constraints.

We let $\tilde{Y} := \{(u, (t, r)) \mid (u, t, r) \in Y\}$, and we identify U, T, and R with natural numbers, i. e. $U = \{1, \ldots, |U|\}$ (and symmetrically for T, R). In both its outer and its inner loop, TRIAS calls the pairs of subroutines *FirstFrequentConcept*$((G, M, I), \tau)$ and *NextFrequentConcept*$((A, B), (G, M, I), \tau)$. These two routines provide an enumeration of all frequent dyadic concepts (A, B) of the formal (dyadic) context (G, M, I). The context is passed over as input parameter. *FirstFrequentConcept* returns in (A, B) the first concept of the enumeration. *NextFrequentConcept* takes the current concept (A, B) and modifies it to the next concept of the enumeration. This way, we compute all frequent maximal cuboids in the relation Y by consecutively computing maximal rectangles in the binary relations $\tilde{Y}$ and I, resp, whereas the condition in line 10 of Algorithm 5.1 checks if the rectangle layers form a maximal cuboid. Note that $A \subseteq (B \times C)^{\tilde{Y}}$ trivially holds, because of $A = I^{\tilde{Y}}$ and $(B \times C) \subseteq I$. Hence only "$\supseteq$" has to be checked.

```
TRIAS$(U, T, R, Y, \tau_u, \tau_t, \tau_r)$
    $\tilde{Y} := \{(u, (t, r)) \mid (u, t, r) \in Y\}$
    $(A, I) :=$ FirstFrequentConcept$((U, T \times R, \tilde{Y}), \tau_u)$
    repeat
        if $|I| \geq \tau_t \cdot \tau_r$ then begin
            $(B, C) :=$ FirstFrequentConcept$((T, R, I), \tau_t)$
            repeat
                if $|C| \geq \tau_r$ then
                    if $A = (B \times C)^{\tilde{Y}}$ then output$(A, B, C)$
            until not NextFrequentConcept$((B, C), (T, R, I), \tau_t)$
        endif
    until not NextFrequentConcept$((A, I), (U, T \times R, \tilde{Y}), \tau_u)$
```

Algorithm 5.1: The TRIAS algorithm

```
FirstFrequentConcept$(\mathbb{K}, \tau)$
    $A := \emptyset'$
    $B := A'$
    if $|A| < \tau$ then
        NextFrequentConcept$((A, B), \mathbb{K}, \tau)$
    endif
    return $(A, B)$
```

Algorithm 5.2: The *FirstFreqentConcept* function

```
NextFreqentConcept$((A, B), (G, M, I), \tau)$
    while defined$(i)$ begin
        $A := (B \oplus i)'$
        if $|A| \geq \tau$ then
            $D := A'$
            if $B <_i D$ then
                $B := D$
                return true
            endif
        endif
        $i :=$ max$(M \setminus B \cap \{1, \ldots, i - 1\}$
    end
    return false
```

Algorithm 5.3: The *NextFreqentConcept* function

For computing all (frequent) maximal rectangles in a binary relation, one can resort to any algorithm for computing (iceberg) concept lattices. The enumeration can be done in any convenient way. For the inner and the outer loop, one could use different algorithms for that task.

In our implementation we equipped the NEXTCLOSURE algorithm [3, 5] of the fourth author with frequency pruning for implementing the *FirstFrequentConcept* and *NextFrequentConcept* routines (see Algorithms 5.2 and 5.3, resp.) for both the outer and the inner loop. This algorithm has the advantage that it needs almost only the space for the data in main memory.

[6] This holds in particular if the tri-sets are tri-concepts.

NextClosure computes concepts in lectic order. This means that, for a given concept (A, B), NextClosure computes the concept (C, D) whose intent D is the next set after B in the so-called lectic order. The lectic order on sets is a total order and is equivalent to the lexicographic order of bit vectors representing those sets.

To find the next concept we define for $B \subseteq M$ and $i \in M$: $B \oplus i := (B \cap \{1, \ldots, i-1\}) \cup \{i\}$.

By applying the closure operator $X \mapsto X''$ to $B \oplus i$ the algorithm computes for a given B the set $D := (B \oplus i)''$. This is the lectically next intent, if $B <_i D$ holds, where $B <_i D$ means, that i is the smallest element in which B and D differ, and $i \in D$.

The method *NextFrequentConcept* adopts this idea and additionally checks if the computed extent $A := (B \oplus i)'$ fullfills the minimal support criterion before computing the intent $D := A'$. This is done in line 5 of Algorithm 5.3 by considering the extent A only if it is large enough.

6 Evaluation

In order to evaluate our approach, we have analyzed the popular social bookmarking system del.icio.us. For detecting communities of users which have the same tagging behaviour, we ran the Trias algorithm on a snapshot consisting of all users, resources, tags and tag assignments we could download (cf. [6]) that were entered to the system on or before June 15, 2004. The resulting folksonomy consists of $|U| = 3,301$ users, $|T| = 30,416$ different tags, $|R| = 22,036$ resources (URLs), which are linked by $|Y| = 616,819$ triples.

As a first step, we ran Trias on the dataset without restricting the minimum supports (i. e., $\tau_u := \tau_t := \tau_r := 0$). The resulting concept tri-lattice consists of $246,167$ tri-concepts.

We then investigated the concepts which contain more than one user, tag and resource, i. e., with $\tau_u := \tau_t := \tau_r := 2$. There were $1,062$ such tri-concepts. Figure 2 shows two examples. They may be exploited further for extracting relations between tags or for recommending a user to get in touch with the other one, as they both use the same terminology for the same URLs and are thus likely to be on a similar line of thought.

As in the dyadic case, the size of the result may grow exponentially in the worst case. Biedermann has shown in [2] that the concept tri-lattice of the triadic context of size $n \times n \times n$ where only the main diagonal is empty has size 3^n. In typical applications, however, one is far from this theoretical boundary. Therefore we focus on empirical evaluations on a large scale real-world dataset.

A	fischer gnat
B	css design web
C	http://www.quirksmode.org/ http://webhost.bridgew.edu/etribou/layouts/ http://www.picment.com/articles/css/funwithforms/ http://www.alistapart.com/articles/sprites/
A	bibi poppy
B	women cinema film
C	http://www.reelwomen.org/ http://www.people.virginia.edu/~pm9k/libsci/womFilm.html http://www.lib.berkeley.edu/MRC/womenbib.html http://www.beaconcinema.com/womfest/ http://www.widc.org/ http://www.wftv.org.uk/home.asp http://www.feminist.com/resources/artspeech/media/femfilm.htm http://www.duke.edu/web/film/pioneers/ http://www.womenfilmnet.org/index.htm#top http://208.55.250.228/

Figure 2. Examples of frequent tri-concepts of del.icio.us

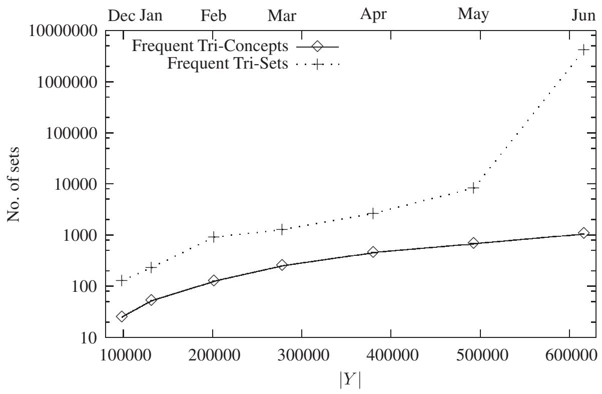


Figure 3. Number of frequent tri-sets vs. number of frequent tri-concepts

For measuring the runtime and the number of frequent concepts we have evaluated the performance of Trias on the del.icio.us dataset described before. From the base set we created monthly snapshots as follows. $\mathbb{F}_0$ contains all tag assignments performed on or before Dec 15, 2003, together with the involved users, tags, and resources; $\mathbb{F}_1$ all tag assignments performed on or before Jan 15, 2004, together with the involved users, tags, and resources; and so on until $\mathbb{F}_6$ which contains all tag assignments performed on or before June 15, 2004, together with the involved tags, users, and resources. This represents seven monotonously growing contexts describing the folksonomy at different points in time. For mining frequent tri-sets and frequent tri-concepts we used minimum support values of $\tau_u := \tau_t := \tau_r := 2$ and measured the run-time of the implementations on a dual-core Opteron system with 2 GHz and 8 GB RAM.

Figure 3 shows the number of frequent tri-concepts versus the number of frequent tri-sets on the logarithmically scaled y-axis, whereas the x-axis depicts the number of

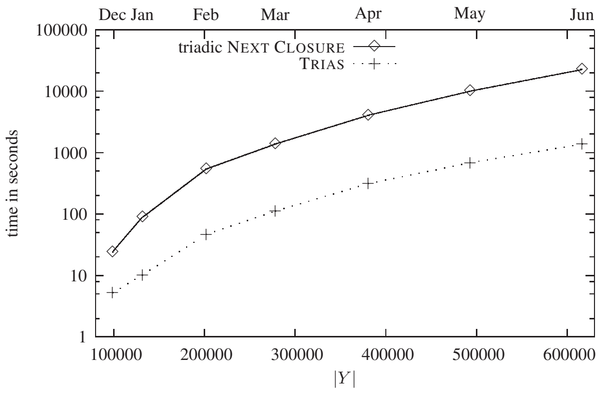


Figure 4. Runtime of triadic Next Closure and Trias algorithm on del.icio.us datasets

triples in Y – which grows from 98,870 triples in Dec 2003 to 616,819 in June 2004. What can be seen is the massive increase of frequent tri-sets in June 2004 while the number of frequent tri-concepts grows at a different level. This can be explained when looking further on the size of the tri-concepts found which also grows from month to month, since more and more users appear and start agreeing on a common vocabulary. Especially such large concepts as shown in Figure 2 do not appear until June 2004 but they are responsible for the steep increase of frequent tri-sets. Overall one can observe that the number of frequent tri-sets of every snapshot is always at least one magnitude of size larger than the number of frequent tri-concepts. Consequently, computing frequent tri-sets is much more demanding than computing frequent tri-concepts – without providing any additional information.

A comparison of the speed improvement gained from not computing all tri-concepts with an algorithm like Next Closure and afterwards pruning the non-frequent concepts but using the Trias algorithm for directly mining frequent tri-concepts is shown in Figure 4. The logarithmically scaled y-axis depicts the runtime of the algorithms in seconds while the x-axis shows again the size of the Y relation. One can see that computing all tri-concepts is more than one magnitude more expensive than mining only the frequent tri-concepts one is interested in.

With the results seen we can conclude that the Trias algorithm provides an efficient method to mine frequent tri-concepts in large scale conceptual structures.

Acknowledgement. Part of this research was funded by the EU in the Nepomuk project (FP6-027705).

References

[1] K. Biedermann. Triadic Galois connections. In K. Denecke and O. Lüders, editors, *General algebra and applications in discrete mathematics*, pages 23–33, Aachen, 1997. Shaker Verlag.

[2] K. Biedermann. Powerset trilattices. In M. Mugnier and M. Chein, editors, *Conceptual Structures: Theory, Tools and Applications*, volume 1453 of *Lecture Notes in Computer Science*, pages 209–221. Springer, 1998.

[3] B. Ganter. Algorithmen zur formalen Begriffsanalyse. In B. Ganter, R. Wille, and K. E. Wolff, editors, *Beiträge zur Begriffsanalyse*, pages 241–254. B.I.–Wissenschaftsverlag, Mannheim, 1987.

[4] B. Ganter and S. A. Obiedkov. Implications in triadic contexts. In *Conceptual Structures at Work: 12th International Conference on Conceptual Structures*, volume 3127 of *Lecture Notes in Computer Science*, pages 186–195. Springer, 2004.

[5] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.

[6] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[7] S. Krolak-Schwerdt, P. Orlik, and B. Ganter. TRIPAT: a model for analyzing three-mode binary data. In H. H. Bock, W. Lenski, and M. M. Richter, editors, *Studies in Classification, Data Analysis, and Knowledge Organization*, volume 4 of *Information systems and data analysis*, pages 298–307. Springer, Berlin, 1994.

[8] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual structures: applications, implementation and theory*, volume 954 of *Lecture Notes in Artificial Intelligence*, pages 32–43. Springer Verlag, 1995.

[9] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Closed set based discovery of small covers for association rules. In *Actes des 15èmes journées Bases de Données Avancées (BDA'99)*, pages 361–381, Octobre 1999.

[10] C. S. Peirce. *Collected Papers*. Harvard Universit Press, Cambridge, 1931–1935.

[11] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna, editors, *Data Science and Classification: Proc. of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin, Heidelberg, 2006. Springer.

[12] G. Stumme. Conceptual knowledge discovery with frequent concept lattices. FB4-Preprint 2043, TU Darmstadt, 1999.

[13] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.

[14] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed association rule mining. technical report 99–10. Technical report, Computer Science Dept., Rensselaer Polytechnic, October 1999.