19

Finding Conceptual Document Clusters Based on Top-*N* Formal Concept Search: Pruning Mechanism and Empirical Effectiveness

Yoshiaki OKUBO and Makoto HARAGUCHI
IST, Hokkaido University
JAPAN

1. Introduction

We often rely on the *World Wide Web* as useful and rich resources of information and knowledge. A large number of web pages (documents) are on the Internet and we can easily browse and enjoy them anytime. It is, however, not so easy to efficiently find useful pages because of the hugeness of the Web space. For example, *Google*, a popular information retrieval (*IR*) engine, often gets a number of web pages with the order of hundred thousands for a given keyword set.

In general, an information retrieval system shows us an ordered list of web pages, where the ordering is determined by its own ranking mechanism. Then, only some of the higher-ranked pages in the list are actually browsed and the others are discarded as less important ones, because the list contains a large number of pages. Thus, web pages with lower ranks are usually invisible for us even if they are similar to higher-ranked ones. In this sense, we might be missing many useful pages or documents. Therefore, if we can make such hidden significant pages visible, our chance to obtain valuable information and knowledge on the Web can be enhanced. Extracting *clusters* each of which consists of similar (web) documents would be a promising approach to realizing it.

Although several clustering methods for web documents have already been investigated (e. g. (Vakali et al., 2004)), most of them adopt traditional *hierarchical* or *partitional* approaches. That is, the whole set of documents is divided into k clusters, where the number of clusters, k, is given as a parameter. As is well-known, however, providing an adequate value for k is quite difficult. This fact has motivated us to investigate a new clustering method, a *pinpoint extraction of Top-N nice clusters* (Haraguchi & Okubo, 2010; 2006b; Okubo et al., 2005; Okubo & Haraguchi, 2003).

As has been pointed out (e. g. (Hotho et al., 2003)), a meaningful cluster should have a clear explanation of what the conceptual meaning of the cluster is. Agreeing with that, we have made an informal constraint on clusters to be extracted (Haraguchi & Okubo, 2007; 2006a):

> *The notion of relevance or interestingness depends only on a conceptual class of documents, not dependent on particular instances of documents. Then the clusters we have to find must be concepts of documents that can be definable by means of feature terms.*

This kind of clusters has been originally formalized in (Haraguchi & Okubo, 2007; 2006a) with the notion of *Formal Concept Analysis* (Ganter & Wille, 1999). A *formal concept* (*FC* in short) in our case is defined as a pair of *closed sets* of documents X and feature terms Y, where the former is called the *extent* and the latter the *intent* of the concept. Such a concept means each document in X shares all the feature terms in Y and such a document never exists any more. Thus, a set of documents as the extent of an *FC* corresponds to a conceptual cluster of documents which is definable by the feature terms shared with the documents. We call this kind of cluster an *FC-cluster*.
In general, we can extract a huge number of *FC*-clusters for a given document set. In order to obtain meaningful *FC*s, we try to extract only Top-N *FC*s in the sense that their intents retain a certain degree of quality (constraint on intents by a lower threshold δ) and their extents are evaluated as in the top N (preference on extents). In (Haraguchi & Okubo, 2007; 2006a), it has been formalized as *Top-N δ-Valid FC Problem*.
In this chapter, we precisely present an algorithm for efficiently extracting Top-N *FC*s. It is based on a *maximum clique algorithm* (Balas & Yu, 1986) and can be viewed as an improved version of our previous algorithm (Haraguchi & Okubo, 2007; 2006a). The point is to *safely* and *completely* exclude duplications of extents already obtained. We show useful theoretical properties for the task with their proofs. Several pruning rules based on these properties are incorporated in our improved algorithm. The *safeness* and *completeness* of the pruning rules are also discussed with theoretical proofs. Our experimental results show that it can extract Top-N *FC*s with practical times for a real dataset. Moreover, since the exact correspondence between the notion of *closed itemsets* (Pasquier et al., 1999) and *FC*s can be observed, our improved algorithm is compared with several excellent closed itemset miners in computation times. The experimental results also show that our algorithm outperforms them in certain difficult cases.

This chapter is an extended version of the literature (Okubo & Haraguchi, 2006). We especially focuses on the algorithmic viewpoint of our method. The remainder of this chapter is organized as follows: In the next section, we introduce a basic terminology used throughout this chapter. Section 3 describes the notion of formal concepts and discusses the correspondence between *FC*s and closed itemsets. Our conceptual document clusters are introduced in Section 4. Section 5 formalizes our problem for finding Top-N δ-Valid *FC*s. Our algorithm for the problem is discussed in Section 6. Some pruning rules are presented in details and a pseudo-code of the algorithm is also given. In Section 7, we conduct our experimentation with a real dataset of web pages. Computational performance of our algorithm is compared with several efficient closed itemset miners. In the final section, we conclude this chapter with a summary and important future directions.

2. Preliminaries

In this chapter, we are concerned with a *simple weighted undirected graph*. A graph is denoted by $G = (V, E, w)$, where V is a set of *vertices*, $E \subseteq V \times V$ a set of *undirected edges* [1] and $w : V \to \mathbb{R}^+$ a (positive real-valued) *weight function* for vertices.
For any vertices $v, v' \in V$, if $(v, v') \in E$, v is said to be *adjacent* to v' and vice versa. For a vertex $v \in V$, the set of vertices adjacent to v is denoted by $N_G(v)$, where $|N_G(v)|$ is called the *degree* of v. If it is clear from the context, it is simply denoted by $N(v)$.

[1] That is, any edge $(v, v') \in E$ is identified with (v', v).

Object ID	Features
1	a b c d e f
2	b c e
3	b e
4	a b d e f
5	b d
6	a d f
7	c d f

Fig. 1. Formal Context

For any pair of vertices v and v' in V $(v \neq v')$, if $(v, v') \in E$, then G is said to be *complete*.
For a subset V' of V, a graph $G(V')$ defined by $G(V') = (V', E \cap V' \times V', w)$ is called a *subgraph* of G and is said to be *induced by* V'. If the subgraph is complete, then it is called a *clique* in G. A clique is simply referred to as the set of vertices by which it is induced. Note here that any subset of a clique is also a clique.
For a clique Q, its *size* is defined by $|Q|$. Since each vertex in G is assigned a weight, it would be reasonable to evaluate cliques by providing an adequate evaluation function based on vertex weights.
For cliques Q and Q', if $Q \subset Q'$, then Q' is called an *expansion* of Q. Moreover, if there exists no clique Q'' such that $Q \subset Q'' \subset Q'$, Q' is called an *immediate expansion* of Q. If a clique Q has no immediate expansion, that is, any proper superset of Q is not a clique, then Q is said to be *maximal*. A maximal clique whose size is largest among all maximal ones is especially called a *maximum clique*. In general, a maximum clique is not uniquely found in G.

3. Formal concept analysis

Formal Concept Analysis (FCA) (Ganter & Wille, 1999) is a theory of data analysis which identifies *conceptual structures among objects* (*individuals*). We first introduce some terminologies for FCA.

3.1 Formal concepts

Let $\mathcal{O}$ be a set of *objects* (*individuals*) and $\mathcal{F}$ a set of *features* (*attributes*). Assume we have a binary relation $R \subseteq \mathcal{O} \times \mathcal{F}$, where a tuple $(x, y) \in R$ means that the object x is associated with the feature y (that is, x has y). A triple of $\mathcal{O}$, $\mathcal{F}$ and R, $< \mathcal{O}, \mathcal{F}, R >$, is called a *formal context*.
A formal context is often represented as a table. Figure 1 shows an example of a formal context in a table format. For example, the object `2` is associated with the features `b`, `c` and `e`.
Under a formal context $< \mathcal{O}, \mathcal{F}, R >$, for each object $x \in \mathcal{O}$, the set of features with which x is associated is denoted by $\mathcal{F}(x)$, that is, $\mathcal{F}(x) = \{y \mid (x, y) \in R\}$.
Given a formal context $< \mathcal{O}, \mathcal{F}, R >$, for a set of objects $X \subseteq \mathcal{O}$ and a set of features $Y \subseteq \mathcal{F}$, we define two mappings $\varphi : 2^{\mathcal{O}} \to 2^{\mathcal{F}}$ and $\psi : 2^{\mathcal{F}} \to 2^{\mathcal{O}}$, respectively, as follows.

$$\varphi(X) = \{y \in \mathcal{F} \mid \forall x \in X, (x, y) \in R\} = \bigcap_{x \in X} \mathcal{F}(x) \quad \text{and}$$
$$\psi(Y) = \{x \in \mathcal{O} \mid Y \subseteq \mathcal{F}(x)\}.$$

The former computes the set of features shared by every object in X. The latter, on the other hand, returns the set of objects each of which is associated with all of the features in Y.

Based on the mappings, a *formal concept* (*FC* in short) under the formal context is defined as a pair of an object set $X \subseteq \mathcal{O}$ and a feature set $Y \subseteq \mathcal{F}$, (X, Y), such that $\varphi(X) = Y$ and $\psi(Y) = X$. Especially, X and Y are called the *extent* and *intent* of the concept, respectively. From the definition, it is obvious that $\psi(\varphi(X)) = X$ and $\varphi(\psi(Y)) = Y$. That is, a formal concept is defined as a pair of *closed* sets of objects and features under the composite mappings. We often denote the composite mappings $\psi \circ \varphi$ and $\varphi \circ \psi$ by E and I, respectively.
For a set of objects $X \subseteq \mathcal{O}$, $\varphi(\psi(\varphi(X))) = I(\varphi(X)) = \varphi(X)$ holds. In other words, $\varphi(X)$ is always closed under I. Therefore, $\varphi(X)$ can be the intent of an *FC*. More precisely speaking, for any set of objects X, we can uniquely obtain an *FC*, $(\psi(\varphi(X)), \varphi(X)) = (E(X), \varphi(X))$. Similarly, for a set of features $Y \subseteq \mathcal{F}$, we can always consider its corresponding *FC*, $(\psi(Y), \varphi(\psi(Y))) = (\psi(Y), I(Y))$.
From the definition of the mappings, we can observe the following theoretical properties.

Observation 1.
Let X and X' be subsets of $\mathcal{O}$ such that $X \subseteq X'$. Then $\varphi(X) \supseteq \varphi(X')$ and $E(X) \subseteq E(X')$. Dually, for subsets of $\mathcal{F}$, Y and Y', such that $Y \subseteq Y'$, $\psi(Y) \supseteq \psi(Y')$ and $I(Y) \subseteq I(Y')$ hold. ■

3.2 Formal concept lattice

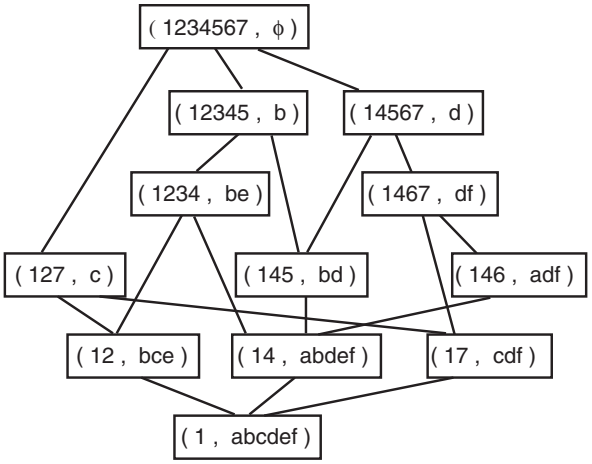
Given a formal context $< \mathcal{O}, \mathcal{F}, R >$, let $\mathcal{FC}$ be the set of *FC*s under the context. We introduce here an ordering on $\mathcal{FC}$.

Definition 1. (Partial Ordering on $\mathcal{FC}$)
Let FC_i and FC_j be a pair of formal concepts in $\mathcal{FC}$ such that $FC_i = (X_i, Y_i)$ and $FC_j = (X_j, Y_j)$. Then FC_i *precedes* FC_j, denoted by $FC_i \prec_{fc} FC_j$, iff $X_i \subset X_j$ and $Y_i \supset Y_j$. ■

For a pair of formal concepts FC_i and FC_j such that $FC_i \prec_{fc} FC_j$, we often say that FC_i is more *specific* than FC_j and FC_j is more *general* than FC_i.
The partially ordered set $(\mathcal{FC}, \prec_{fc})$ forms a *lattice*, called a *formal concept lattice*. For the formal context in Figure 1, we have the formal concept lattice shown in Figure 2. In the figure, `xyz` is an abbreviation of a set $\{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$. The most general concept is put at the top and the most specific one at the bottom. For any pair of FC_i and FC_j, if FC_i immediately precedes FC_j, they are connected by an edge.

3.3 Exact correspondence between formal concepts and closed itemsets

In the field of *Data Mining* (Han & Kamber, 2006), many researchers have investigated *Frequent Itemset Mining* (Agrawal & Srikant, 1994; Han et al., 2007; Pasquier et al., 1999). Particularly, the notion of *closed itemsets* is well-known as a useful lossless condensed representation of itemsets (Pasquier et al., 1999).
Let $\mathcal{I}$ be a set of *items*. A *transaction* T is a subset of $\mathcal{I}$ and is assigned a unique identifier, denoted by $id(T)$. A *transaction database* $\mathcal{TD}$ is given as a (multiple) set of transactions. Let $\mathcal{ID}$ be the set of identifiers of transactions in $\mathcal{TD}$, that is, $\mathcal{ID} = \{id(T) \mid T \in \mathcal{TD}\}$. We can obtain a binary relation $R \subseteq \mathcal{ID} \times \mathcal{I}$ which is defined as $R = \{(id(T), x) \mid T \in \mathcal{TD} \ \wedge \ x \in T\}$. Regarding each transaction in $\mathcal{TD}$ as an object and each item in $\mathcal{I}$ as a feature, we can represent the transaction database as a formal context $< \mathcal{ID}, \mathcal{I}, R >$.
A set of items in $\mathcal{I}$, I, is called an itemset. For a transaction database $\mathcal{TD}$, the *support* (or *frequency*) of I, denoted by $sup(I)$, is defined as $sup(I) = |\{T \in \mathcal{TD} \mid I \subseteq T\}|$. An itemset I is said to be *closed* if there exists no itemset I' such that $I \subset I'$ and $sup(I) = sup(I')$. Since the support of an itemset is defined as the number of transactions containing the itemset, a closed itemset I exactly corresponds to the intent of a formal concept. For the closed itemset I,



Object ID	Features
1	a b c d e f
2	b c e
3	b e
4	a b d e f
5	b d
6	a d f
7	c d f

Fig. 2. Formal Concept Lattice

furthermore, the set of identifiers of the transactions containing I, denoted by ID_I, is uniquely identified. Particularly, ID_I is closed in the sense that any other transaction never contains I, that is, ID_I also corresponds to the extent of an FC. Therefore, we can always observe the exact correspondence between a closed itemset I and the formal concept (ID_I, I).
From the correspondence, any efficient closed itemset miner would be helpful for finding FCs. For example, `AFOPT` (Liu et al., 2003), `DCI-Closed` (Lucchese et al., 2004) and `LCM` (Uno et al., 2004) are well-known as such efficient systems (algorithms).

4. Conceptual document clusters based on formal concepts

Let $\mathcal{D}$ be a set of *documents* and $\mathcal{T}$ a set of *feature terms*. We assume that each *document* in $\mathcal{D}$ is represented as a set of feature terms in $\mathcal{T}$ appearing in the document. Then, we have a binary relation $R \subseteq \mathcal{D} \times \mathcal{T}$, where a tuple $(d, t) \in R$ means that the term t appears in the document d. Under the assumption, the triple $\mathcal{C} =< \mathcal{D}, \mathcal{T}, R >$ is regarded as a formal context, where for each formal concept (D, T) under $\mathcal{C}$, D is considered as a *document cluster*. It should be emphasized here that we can clearly explain why the documents in D are grouped together. Each document in D shares the set of feature terms T and any other document never contains T. In this sense, D can be viewed as a conceptual document cluster which is definable by the feature terms shared with the documents. Thus, by restricting our document clusters to extents of formal concepts under $\mathcal{C}$, we can explicitly provide conceptual meanings based on their intents.

5. Top-N δ-valid formal concept problem

For a given formal context $\mathcal{C} =< \mathcal{O}, \mathcal{F}, R >$, the number of FCs under $\mathcal{C}$ often becomes large. It is actually impossible for users to check and analyze all of them. Therefore, selectively extracting FCs with a certain degree of quality would be a practical and reasonable approach. Needless to say, quality of an FC is affected by both its extent and intent. For example, a concept with a larger intent might have convincing evidence (similarity) for the grouping of objects (extent). Its extent, however, tends to be smaller. We often consider that concepts

with too small extents would not be so meaningful because they might be too specific or exceptional. Conversely, although a concept with a smaller intent will have a larger extent, evidence for the grouping seems to be weak. In other words, the extent would consist of less similar objects. Thus, in order to obtain meaningful *FC*s, it is required to control their quality from the viewpoint of extents and intents. For such a requirement, we formalize our *FC*s to be found as follows:

Constraint on Intents: In order for *FC*s to retain a certain degree of similarity, a constraint on intents is imposed. Concretely speaking, a threshold for evaluation value of intents, δ, is provided. For an *FC*, if its intent value is greater than or equal to δ, then the *FC* is considered to have sufficient quality of intent. Such an *FC* is said to be *δ-valid*.

Preference in Extents: *FC*s with higher evaluation values of extents are preferred among the δ-valid *FC*s. Particularly, given an integer N, the *FC*s with Top-N extent values are extracted.

In order to evaluate extents and intents of *FC*s, we assume that each object $x \in \mathcal{O}$ and feature $y \in \mathcal{F}$ are assigned their positive real-valued weights, referred to as $w_{\mathcal{O}}(x)$ and $w_{\mathcal{F}}(y)$. Then, evaluation functions for extents and intents, $eval_E$ and $eval_I$, are defined with $w_{\mathcal{O}}$ and $w_{\mathcal{F}}$, respectively.
Although we can define various evaluation functions, *increasing monotone* functions are strongly preferred from the computational point of view, where a function f is said to be increasing monotone (under set-inclusion) iff $S \subseteq S'$ implies $f(S) \leq f(S')$ for any pair of sets, S and S'. In what follows, we assume our evaluation functions to be increasing monotone. The reason why such a function is preferable will become clear in the next section.
We can now formalize our problem of finding *Top-N δ-valid formal concepts* as follows:

Definition 2. (Top-N δ-Valid Formal Concept Problem)
Let $\mathcal{C} =< \mathcal{O}, \mathcal{F}, R >$ be a formal context, δ a threshold for admissible evaluation value of intent and N an integer for Top-N. The problem of finding *Top-N δ-valid formal concepts for* $\mathcal{C}$ is to extract the set of formal concepts $\{(X, Y)\}$ such that $eval_I(Y) \geq \delta$ (as constraint) and $eval_E(X)$ is in the top N among such ones (as preference). ■

From the viewpoint of problem specification, our Top-N δ-valid *FC* problem is closely related to *Top-N Frequent Closed Itemset Mining* (Wang et al., 2005) and *Constraint-Based Concept Mining* (Besson et al., 2005).
Given a pair of parameters, N and *minlen*, Top-N Frequent Closed Itemset Mining (Wang et al., 2005) is to find closed itemsets of length at least *minlen* whose frequencies (supports) are in the top N. Since length (size) of itemsets is a measure which evaluates itemsets, the parameter *minlen* controls the quality of closed itemsets (that is, intents) to be extracted, as is similar to our problem. Furthermore, frequencies of closed itemsets are equivalent to sizes of their corresponding extents. In case we simply evaluate each intent and extent by their sizes, therefore, our Top-N *FC* problem is identical with the Top-N frequent closed itemset mining. In the framework, however, length and frequency are only measures by which we can evaluate itemsets. On the other hand, in our framework, we can consider various evaluation measures as well as length and frequency. In this sense, our problem can be regarded as a generalization of Top-N closed itemset mining.
Given a pair of parameters, *minsup* and *minlen*, Constraint-Based Concept Mining (Besson et al., 2005) is a task of finding all closed itemsets I such that $sup(I) \geq minsup$ and $|I| \geq minlen$. That is, this mining task is to compute all frequent closed itemsets with enough length.

Since the frequency of closed itemsets is equivalent to the size of extents, *minsup implicitly* gives some integer N such that the N-th frequency of closed itemsets with enough length is equal to *minsup*. Therefore, if extents are evaluated by their sizes, providing *minsup* in essence corresponds to providing some N for Top-N in our problem. However, the authors consider that providing N is more simple and intuitive than providing *minsup*.

6. Finding top-N δ-valid formal concepts with clique search

In this section, we precisely discuss our algorithm for finding Top-N δ-valid FCs.
Top-N δ-valid FCs can be extracted by finding certain *cliques* in an weighted undirected graph. Before going into details, we present a basic strategy of our search algorithm.

6.1 Basic search strategy

Let $\mathcal{C} =< \mathcal{O}, \mathcal{F}, R >$ be a formal context. For each FC under $\mathcal{C}$, there always exists a set of objects $X \subseteq \mathcal{O}$ such that $E(X) = \psi(\varphi(X))$ and $\varphi(X)$ correspond to the extent and the intent of the FC, respectively. Therefore, by applying the mappings φ and ψ to each set of objects $X \subseteq \mathcal{O}$, we can obtain all of the FCs under $\mathcal{C}$.
Let us consider a *total ordering* on $\mathcal{O} = \{x_1, \ldots, x_{|\mathcal{O}|}\}$, $\prec$, simply defined as $x_i \prec x_j$ iff $i < j$. It is assumed that for each subset $X \subseteq \mathcal{O}$, the elements in X is ordered based on $\prec$.
For a subset of $\mathcal{O}$, $X_i = \{x_{i_1}, \ldots, x_{i_n}\}$, the first element x_{i_1} is denoted by $head(X_i)$ and the last one, x_{i_n}, by $tail(X_i)$. Furthermore, the set of first k elements, $\{x_{i_1}, \ldots, x_{i_k}\}$, is called the *k-prefix* of X_i and is referred to as $prefix(X_i, k)$, where $0 \leq k \leq n$ and $prefix(X_i, 0)$ is defined as ϕ.
We introduce here a *partial ordering* on $2^{\mathcal{O}}$, $\prec_s$, as follows.

Definition 3. (Partial Ordering on $2^{\mathcal{O}}$)
Let X_i and X_j be subsets of $\mathcal{O}$ such that $X_i \neq X_j$. Then X_i *precedes* X_j or X_j *succeeds* X_i, denoted by $X_i \prec_s X_j$, iff X_i is a prefix of X_j, that is, $X_i = prefix(X_j, |X_i|)$. If X_j is a successor of X_i, then X_j is called a *descendant* of X_i. Particularly, X_j is called a *child* of X_i if X_j is an immediate successor of X_i. ■

It can be easily observed that the partially ordered set $(2^{\mathcal{O}}, \prec_s)$ forms a *tree* with the root node ϕ which is well-known as a *set enumeration tree* (Rymon, 1992). Figure 3 shows an example of a set enumeration tree for $\mathcal{O} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$.
For each (non-leaf) subset $X \subseteq \mathcal{O}$ in the tree, its child is simply obtained as $X \cup \{x\}$, where $tail(X) \prec x$. Based on the fact, therefore, any subset of $\mathcal{O}$ can be generated systematically *without any duplications*, starting with the empty set.
In the tree, a simple theoretical property can be observed.

Observation 2.
Let X_i and X_j be subsets of $\mathcal{O}$ such that $X_i \prec_s X_j$. Then, $eval_I(\varphi(X_i)) \geq eval_I(\varphi(X_j))$.

Proof:
It is immediately proved from Observation 1. From $X_i \subseteq X_j$, $\varphi(X_i) \supseteq \varphi(X_j)$ holds. Since $eval_I$ is assumed to be increasing monotone, we have $eval_I(\varphi(X_i)) \geq eval_I(\varphi(X_j))$. ■

As a direct consequence, a simple pruning rule will be available in our search.

Pruning 1.
For a set of objects $X \subseteq \mathcal{O}$, if $eval_I(\varphi(X)) < \delta$ holds, then there is no need to examine any descendant of X. ■

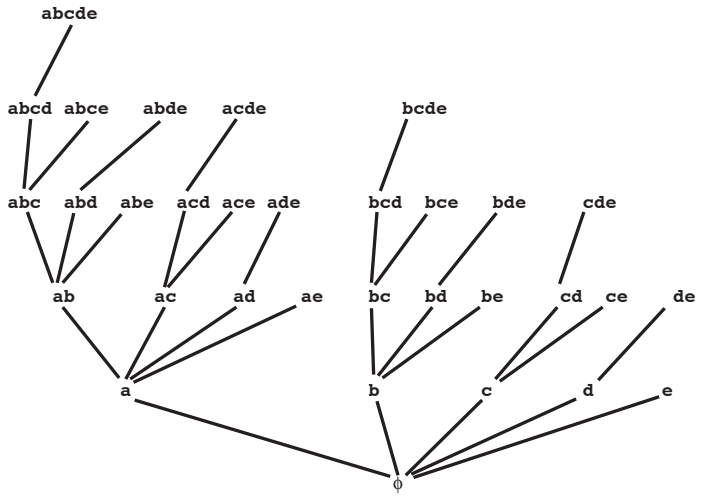


Fig. 3. Set Enumeration Tree

Proof:
It can be easily verified. Let X' be a descendant of X. Since $X \subset X'$, $eval_I(\varphi(X)) \geq eval_I(\varphi(X'))$ holds. From the assumption in the rule, we have $eval_I(\varphi(X')) \leq eval_I(\varphi(X)) < \delta$. This means that no δ-valid FC can be obtained from X'. Thus, we can safely prune any descendant of X. ■

From the above discussion, our search for finding Top-N δ-valid FCs can be performed in *depth-first manner* with the simple pruning. During our search, we maintain a list which stores Top-N δ-valid FCs already found. That is, the list keeps *tentative* Top-N FCs. For a set of objects $X \subseteq \mathcal{O}$, we check whether $eval_I(\varphi(X)) \geq \delta$ holds or not. If it holds, then $(E(X), \varphi(X))$ is a δ-valid FC and the tentative Top-N list is adequately updated for the FC. Then a child of X is generated and the same procedure is recursively performed for the child. If $eval_I(\varphi(X)) < \delta$, we can immediately backtrack without examining any descendant of X. Starting with the initial X of the empty set, the procedure is iterated in depth-first manner until no X remains to be examined. A pseudo-code of our basic algorithm is presented in Figure 4.

6.2 Finding formal concepts by clique search

Although a pruning rule is available, the simple basic algorithm just discussed above is required further improvement for efficient computation. In order to improve it, we try to find our Top-N FCs by *clique search* with *depth-first branch-and-bound* strategy (Balas & Yu, 1986).

6.2.1 Graph construction

Given a formal context $\mathcal{C} =< \mathcal{O}, \mathcal{F}, R >$ and a threshold δ, an weighted undirected graph $G = (\mathcal{O}, E, w_{\mathcal{O}})$, is first constructed, where the set of edges, E, is defined as

$$E = \{(x_i, x_j) \mid x_i, x_j \in \mathcal{O}(i \neq j) \wedge eval_I(\mathcal{F}(x_i) \cap \mathcal{F}(x_j)) \geq \delta\}.$$

That is, if a pair of objects share a set of features whose evaluation value is greater than or equal to δ, then they are connected by an edge.

Input :
$< \mathcal{O}, \mathcal{F}, R >$: a formal context where
$\mathcal{O}$: a set of objtects, $\mathcal{F}$: a set of features and R : a binary relation on $\mathcal{O}$ and $\mathcal{F}$
δ : a threshold for intent value
N : an integer for Top-N
$eval_E$: an evaluation function for extents defined with an weight function $w_{\mathcal{O}}$ for $\mathcal{O}$
$eval_I$: an evaluation function for intents defined with an weight function $w_{\mathcal{F}}$ for $\mathcal{F}$
Output :
$\mathcal{FC}$: the set of δ-valid formal concepts whose extent values are in the top N

---

procedure main() :
  $\mathcal{FC} \leftarrow \phi$; /* Grobal variable */
  $min = 0.0$; /* Grobal variable */
  for each $x \in \mathcal{O}$ in predefined order **do**
    begin
      if $eval_I(\mathcal{F}(x)) \geq \delta$ **then** /* Pruning 1 */
        TopNFCFindBasic($\{x\}, \mathcal{F}(x)$) ;
      endif
    end
  return $\mathcal{FC}$;

---

procedure TopNFCFindBasic(X, I) :
  TopNListUpdata($(E(X), I)$) ;
  for each $x \in \mathcal{O}$ such that $tail(X) \prec x$ in predefined order **do**
    begin
      $NewX \leftarrow X \cup \{x\}$;
      $NewI \leftarrow I \cap \mathcal{F}(x)$;
        if $eval_I(NewI) \geq \delta$ **then** /* Pruning 1 */
          TopNFCFindBasic($NewX$, $NewI$) ;
        endif
    end

---

procedure TopNListUpdate(FC) :
  $\mathcal{FC} \leftarrow \mathcal{FC} \cup \{FC\}$;
  if $\mathcal{FC}$ tentatively contains N-th ones **then**
    $min \leftarrow N$-th extent value ;
    Remove M-th ones from $\mathcal{FC}$ such that $N < M$;
  endif

Fig. 4. Basic Algorithm for Finding Top-N δ-Valid Formal Concepts

6.2.2 Clique enumeration tree

Since each clique Q in G is a subset of $\mathcal{O}$, $(E(Q), \varphi(Q))$ always becomes a formal concept. It should be noted here that from the definition of G, for each δ-valid FC, there always exists a clique Q in G such that $E(Q)$ and $\varphi(Q)$ are the extent and intent of the FC, respectively. Therefore, subsets to be examined in our basic algorithm can be restricted to only cliques in G. Based on δ, thus, we can *statically* excludes many useless subsets from which we can never obtain δ-valid FCs.

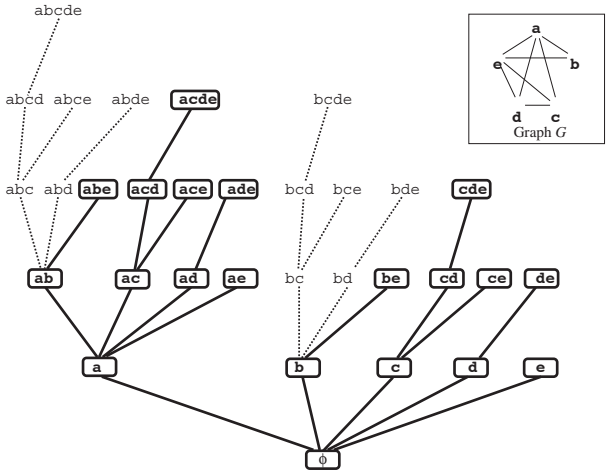


Fig. 5. Clique Enumeration Tree for Graph G

Since any subset of a clique is also a clique, the ordering $\prec_s$ is still valid for cliques and the cliques in G also form a tree, called a *clique enumeration tree*. An example of a clique enumeration tree for a graph G is shown in Figure 5, where any clique in G is boxed.
Needless to say, Pruning 1 is still available in the clique enumeration tree. That is, the constraint based on δ can work not only statically in the graph construction, but also *dynamically* in the search process.
A child of a clique Q is generated by adding a certain vertex (object) to Q. Such an object to be added is precisely defined with a notion of *extensible candidates*

Definition 4. (Extensible Candidates for Clique)
Let $G = (V, E, w)$ be a graph and Q a clique in G. A vertex $v \in V$ adjacent to any vertex in Q is called an *extensible candidate* for Q. The set of extensible candidates is denoted by $cand(Q)$, that is,

$$cand(Q) = \{v \in V \mid \forall u \in Q \ \ (v, u) \in E\} = \bigcap_{v \in Q} N_G(v). \qquad \blacksquare$$

Since it is obvious from the definition that for any extensible candidate $v \in cand(Q)$, $Q \cup \{v\}$ always becomes a clique, we can easily generate a child of Q by adding $v \in cand(Q)$ such that $tail(Q) \prec v$. Thus, we can explore a clique enumeration tree in depth-first manner.

6.3 Pruning redundant cliques

As has just been discussed, Pruning 1 excludes useless cliques from which no δ-valid FCs can be obtained. In addition to such useless ones, our clique enumeration tree in general contains redundant cliques whose corresponding FCs are identical. Therefore, it would be desirable for efficient computation to prune such redundant cliques as well. We can observe the following three simple properties of FCs which will be useful for realizing it.

Observation 3.
Let (X, Y) be a formal concept. Then, there always exists a clique Q in G such that $E(Q) = X$ and $head(Q) = head(X)$. $\blacksquare$

Proof:
It is trivial. Since X is a clique in G, X itself can be such a clique Q. Then, it is obvious that $E(Q) = E(X) = X$ and $head(Q) = head(X)$. ■

Observation 4.
Let Q be a clique in G. For any element $\alpha \in E(Q) \backslash Q$, $E(Q \cup \{\alpha\}) = E(Q)$ holds. That is, their corresponding FCs are identical. ■

Proof:
It can be easily proved from Observation 1. Let α be an element such that $\alpha \in E(Q) \backslash Q$. From a property of the mapping φ, $\varphi(Q \cup \{\alpha\}) = \varphi(Q) \cap \varphi(\{\alpha\})$. Moreover, since $\alpha \in E(Q)$, $\varphi(Q) \subseteq \varphi(\{\alpha\})$. Therefore, we have $\varphi(Q \cup \{\alpha\}) = \varphi(Q)$ and $\psi(\varphi(Q \cup \{\alpha\})) = \psi(\varphi(Q))$. ■

Observation 5.
Let Q be a clique in G and $Q \cup \{\alpha\}$ a child of Q. For any element $\beta \in E(Q \cup \{\alpha\}) \backslash E(Q)$ such that $\beta \prec \alpha$, there exists Q' such that $E(Q') = E(Q \cup \{\alpha\})$ and Q' is examined prior to $Q \cup \{\alpha\}$ in our depth-first search. ■

Proof:
Let β be an element such that $\beta \in E(Q \cup \{\alpha\}) \backslash E(Q)$ and $\beta \prec \alpha$. Since $\beta \notin E(Q)$ and $Q \subseteq E(Q)$, we have $\beta \notin Q$ and then $\beta \notin Q \cup \{\alpha\}$. Therefore, $\beta \in E(Q \cup \{\alpha\}) \backslash Q \cup \{\alpha\}$ holds. From Observation 4, $E(Q \cup \{\alpha\} \cup \{\beta\}) = E(Q \cup \{\alpha\})$ holds. It should be noted here that since $\beta \prec \alpha$, $Q \cup \{\alpha\} \cup \{\beta\}$ is processed prior to $Q \cup \{\alpha\}$ in our depth-first search. Thus, the observation can be verified. ■

Each of the above observations provides us a pruning rule to exclude redundant cliques.

Pruning 2.
Let Q be a clique in G. If $head(Q) \neq head(E(Q))$ holds, then Q and its descendants do not have to be examined. ■

Pruning 3.
Let Q be a clique in G. For any element $\alpha \in E(Q) \backslash Q$ such that $tail(Q) \prec \alpha$, $Q \cup \{\alpha\}$ and its descendants do not have to be examined. ■

Pruning 4.
Let Q be a clique in G and $Q \cup \{\alpha\}$ a child of Q. If there exists an element $\beta \in E(Q \cup \{\alpha\}) \backslash E(Q)$ such that $\beta \prec \alpha$, then $Q \cup \{\alpha\}$ and its descendants do not have to be examined. ■

A remarkable point we should emphasize is that the pruning rules are safe. In other words, for each formal concept (X, Y), there always exists a clique Q such that $E(Q) = X$ and Q is never pruned with the pruning rules.

Theorem 1.
Pruning 2, 3 and 4 are safe. ■

Proof:
For a formal concept $FC = (X, Y)$, let α_0 be the head element of X, that is, $\alpha_0 = head(X)$. Moreover, assume $P = X \backslash E(\{\alpha_0\}) = \{\alpha_1, \ldots, \alpha_k\}$.

Since $X = E(\{\alpha_0\}) \cup P$,

$$\begin{aligned}\varphi(X) &= \varphi(E(\{\alpha_0\}) \cup P) \\ &= \varphi(E(\{\alpha_0\})) \cap \varphi(P) \\ &= \varphi(\{\alpha_0\}) \cap \varphi(P) \\ &= \varphi(\{\alpha_0\} \cup P).\end{aligned}$$

Therefore, we immediately have

$$X = E(X) = \psi(\varphi(X)) = \psi(\varphi(\{\alpha_0\} \cup P)) = E(\{\alpha_0\} \cup P).$$

For each i ($1 \leq i \leq k$), we consider

$$\begin{aligned}D_i &= E(\{\alpha_0\} \cup \{\alpha_1\} \cup \cdots \cup \{\alpha_i\}) \setminus \{\alpha_0\} \cup \{\alpha_1\} \cup \cdots \cup \{\alpha_i\} \\ &= E(\{\alpha_0\} \cup prefix(P,i)) \setminus \{\alpha_0\} \cup prefix(P,i).\end{aligned}$$

In case of $i = 0$, D_0 is defined as $D_0 = E(\{\alpha_0\}) \setminus \{\alpha_0\}$. Observation 4 implies that, for each i ($1 \leq i \leq k$), if $\alpha_i \in D_{i-1}$, then

$$E(\{\alpha_0\} \cup prefix(P,i)) = E(\{\alpha_0\} \cup prefix(P,i-1)).$$

On the other hand, if $\alpha_i \notin D_{i-1}$,

$$E(\{\alpha_0\} \cup prefix(P,i)) \supset E(\{\alpha_0\} \cup prefix(P,i-1)).$$

Here, let us consider a subset of P, R, defined as

$$R = \{\alpha_i \in P \mid \alpha_i \notin D_{i-1}\}.$$

Assuming $R = \{\alpha'_1, \ldots, \alpha'_\ell\}$ and $\alpha'_0 = \alpha_0$, P can be represented as a union of $\ell + 1$ (ordered) subsets, $P = P_0 \cup \cdots \cup P_\ell$, where P_i is defined as

$$P_i = \begin{cases} \{\alpha \in P \mid \alpha'_i \preceq \alpha \prec \alpha'_{i+1}\} & \text{if } 0 \leq i < \ell, \\ \{\alpha \in P \mid \alpha'_i \preceq \alpha\} & \text{if } i = \ell. \end{cases}$$

That is, each $\alpha \in P$ belongs to a certain P_i. If $\alpha \in P$ is contained in P_i, then the index i is referred to as $f(\alpha)$.
We first verify that for each $\alpha_i \in P$ ($1 \leq i \leq k$),

$$E(\{\alpha_0\} \cup prefix(P,i)) = E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i))). \tag{1}$$

The statement can be proved with mathematical induction on the index i of α_i.
In case of $i = 1$, α_1 belongs to P_0 or P_1, that is, $f(\alpha_1)$ is 0 or 1, respectively. For the former, α_1 is in $D_0 = E(\{\alpha_0\}) \setminus \{\alpha_0\}$. Hence, $E(\{\alpha_0\} \cup \{\alpha_1\}) = E(\{\alpha_0\} \cup prefix(P,1)) = E(\{\alpha_0\}) = E(\{\alpha_0\} \cup prefix(R,0))$ holds. For the latter, since $prefix(P,1) = prefix(R,1) = \{\alpha_1\}$, the statement is obviously true.
For the induction step, let us assume that for some i, $E(\{\alpha_0\} \cup prefix(P,i)) = E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i)))$ holds. In case of $\alpha_{i+1} \in D_i$, α_{i+1} belongs to $P_{f(\alpha_i)}$, that is, $f(\alpha_{i+1}) =$

$f(\alpha_i)$. In addition, $E(\{\alpha_0\} \cup prefix(P,i) \cup \{\alpha_{i+1}\}) = E(\{\alpha_0\} \cup prefix(P,i))$ holds. From the assumption, therefore, we have

$$\begin{aligned} E(\{\alpha_0\} \cup prefix(P,i) \cup \{\alpha_{i+1}\}) &= E(\{\alpha_0\} \cup prefix(P,i+1)) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i))) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_{i+1}))), \end{aligned}$$

showing the statement is true.
On the other hand, in case of $\alpha_{i+1} \notin D_i$, α_{i+1} belongs to $P_{f(\alpha_i)+1}$ and is particularly identical with $\alpha'_{f(\alpha_i)+1}$. Since $f(\alpha_{i+1}) = f(\alpha_i) + 1$, we also have $\alpha_{i+1} = \alpha'_{f(\alpha_{i+1})}$. From the assumption, therefore, it can be verified that the statement is true as follows.

$$\begin{aligned} E(\{\alpha_0\} \cup prefix(P,i) \cup \{\alpha_{i+1}\}) &= E(\{\alpha_0\} \cup prefix(P,i+1)) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i)) \cup \{\alpha_{i+1}\}) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i)) \cup \{\alpha'_{f(\alpha_i)+1}\}) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_i)+1)) \\ &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_{i+1}))). \end{aligned}$$

As the result, we can conclude the statement is true for any $\alpha_i \in P$ $(1 \leq i \leq k)$.
We can now obtain

$$E(\{\alpha_0\} \cup prefix(P,k)) = E(\{\alpha'_0\} \cup prefix(R, f(\alpha_k))).$$

It is noted here that since $prefix(P,k) = P$ and $prefix(R, f(\alpha_k)) = prefix(R,\ell) = R$, $E(\{\alpha_0\} \cup P) = X = E(\{\alpha'_0\} \cup R)$ holds. This implies that examining $\{\alpha'_0\} \cup R$ is sufficient to obtain the extent X. In order to visit $\{\alpha'_0\} \cup R$ in our depth-first search, we have to take into account the ordered sequence of cliques,

$$\{\alpha'_0\} \cup prefix(R,0) \prec_s \{\alpha'_0\} \cup prefix(R,1) \prec_s \cdots \prec_s \{\alpha'_0\} \cup prefix(R,\ell).$$

Therefore, it should be verified that our pruning rules never exclude any $\{\alpha'_0\} \cup prefix(R,i)$ in the sequence.
For any i $(0 \leq i \leq \ell)$, since $\{\alpha'_0\} \cup prefix(R,i) \subseteq E(\{\alpha'_0\} \cup prefix(R,i)) \subseteq X$ and $\alpha'_0 = \alpha_0 = head(X)$, $head(\{\alpha'_0\} \cup prefix(R,i)) = head(E(\{\alpha'_0\} \cup prefix(R,i)))$ always holds. Hence, Pruning 2 can never prevent us from examining $\{\alpha'_0\} \cup prefix(R,i)$.
In order to show Pruning 3 is safe, we have to verify that for any $\alpha'_i \in R$,

$$\alpha'_i \notin E(\{\alpha'_0\} \cup prefix(R,i-1)) \backslash \{\alpha'_0\} \cup prefix(R,i-1).$$

Let r be the original index of α'_i in P, that is, $\alpha'_i = \alpha_r$ $(\in P)$. Since $\alpha'_i \in R$, $\alpha_r \notin D_{r-1} = E(\{\alpha_0\} \cup prefix(P,r-1)) \backslash \{\alpha_0\} \cup prefix(P,r-1)$. It is, particularly, clear that $\alpha_r \notin E(\{\alpha_0\} \cup prefix(P,r-1))$ because $\alpha_r \notin \{\alpha_0\} \cup prefix(P,r-1)$. From the statement (1), it is noted here that $E(\{\alpha_0\} \cup prefix(P,r-1)) = E(\{\alpha'_0\} \cup prefix(R, f(\alpha_{r-1})))$. Moreover, since $\alpha'_i = \alpha_r$, $f(\alpha_{r-1})$ should be $i-1$. Therefore, $E(\{\alpha_0\} \cup prefix(P,r-1)) = E(\{\alpha'_0\} \cup prefix(R,i-1))$ holds. Since $\alpha'_i \notin E(\{\alpha'_0\} \cup prefix(R,i-1))$, it is clear that $\alpha'_i \notin E(\{\alpha'_0\} \cup prefix(R,i-1)) \backslash \{\alpha'_0\} \cup prefix(R,i-1)$. This means that for each i $(1 \leq i \leq \ell)$, $\{\alpha'_0\} \cup prefix(R,i)$ can never be a target of Pruning 3.

Safeness of Pruning 4 can be confirmed by showing the following statement is true for any i $(1 \leq i \leq \ell)$:

$$\forall \beta \in E(\{\alpha'_0\} \cup prefix(R,i)) \backslash E(\{\alpha'_0\} \cup prefix(R,i-1)),\ \ \alpha'_i \preceq \beta.$$

From Observation 1, for any $X' \subseteq X$, $E(X') \subseteq E(X)$ holds. Moreover, since X is the extent of FC, $E(X) = X$. It implies that for any $X' \subseteq X$, $E(X') \subseteq X$. From $E(\{\alpha_0\}) \cup P = X$ and for any i $(0 \leq i < k)$, $P = prefix(P,i) \cup \{\alpha_{i+1}\} \cup \cdots \cup \{\alpha_k\}$, therefore, we have

$$\begin{aligned} D_i &= E(\{\alpha_0\} \cup prefix(P,i)) \setminus \{\alpha_0\} \cup prefix(P,i) \\ &\subseteq E(\{\alpha_0\}) \cup \{\alpha_{i+1}\} \cup \cdots \cup \{\alpha_k\}. \end{aligned}$$

In other words, for any $\beta \in D_i$, $\beta \in E(\{\alpha_0\})$ or $\alpha_i \prec \beta$. It is, therefore, easy to see that for any $\beta \in E(\{\alpha_0\} \cup prefix(P,i)) \setminus \{\alpha_0\} \cup prefix(P,i-1)$, $\beta \in E(\{\alpha_0\})$ or $\alpha_i \preceq \beta$. From $\{\alpha_0\} \cup prefix(P,i-1) \subseteq E(\{\alpha_0\} \cup prefix(P,i-1))$ and $E(\{\alpha_0\}) \subseteq E(\{\alpha_0\} \cup prefix(P,i-1))$, it is immediately derived that for each $\beta \in E(\{\alpha_0\} \cup prefix(P,i)) \setminus E(\{\alpha_0\} \cup prefix(P,i-1))$, $\alpha_i \preceq \beta$ holds.
Following the argument just before, we assume here that for $\alpha'_i \in R$, its original index in P is r. From the statement (1),

$$\begin{aligned} E(\{\alpha_0\} \cup prefix(P,r)) &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_r))) \\ &= E(\{\alpha'_0\} \cup prefix(R,i)). \end{aligned}$$

Moreover,

$$\begin{aligned} E(\{\alpha_0\} \cup prefix(P,r-1)) &= E(\{\alpha'_0\} \cup prefix(R, f(\alpha_{r-1}))) \\ &= E(\{\alpha'_0\} \cup prefix(R,i-1)) \end{aligned}$$

because $f(\alpha_{r-1}) = i-1$. That is,

$$\begin{aligned} &E(\{\alpha_0\} \cup prefix(P,r)) \setminus E(\{\alpha_0\} \cup prefix(P,r-1)) \\ &\quad = E(\{\alpha'_0\} \cup prefix(R,i)) \setminus E(\{\alpha'_0\} \cup prefix(R,i-1)). \end{aligned}$$

Therefore, we can see that for any $\beta \in E(\{\alpha'_0\} \cup prefix(R,i)) \setminus E(\{\alpha'_0\} \cup prefix(R,i-1))$, $\alpha'_i \preceq \beta$ holds.
As the result, in our depth-first search with Pruning 2, 3 and 4, we can surely visit the clique $\{\alpha'_0\} \cup R$ from which the extent X can be obtained. Thus, our prunings are safe. ■

With the help of the prunings, thus, we can safely exclude only cliques from which we certainly obtain duplicate FCs. One might be interested here in whether our removal of duplications is complete or not. We can affirmatively answer to the question. That is, the prunings can *completely* eliminate all of the duplicate FCs.

Theorem 2.
Let $\mathcal{Q}$ be the set of cliques examined in our search with Pruning 2, 3 and 4. Then, for any pair of (different) cliques in $\mathcal{Q}$, their corresponding FCs are not identical. ■

Proof:
We prove the theorem by showing their extents are not identical.
Let Q_i and Q_j be a pair of cliques in $\mathcal{Q}$, where $i \neq j$. There are two cases to be considered.

Case 1 : Q_i and Q_j are comparable under the ordering $\prec_s$.

Case 2 : Q_i and Q_j are not comparable under $\prec_s$.

In each case, we try to verify that $E(Q_i) \neq E(Q_j)$ holds.

Case 1 : Without loss of generality, we assume $Q_i \prec_s Q_j$. There exists a child of Q_i, $Q_i \cup \{\alpha\}$, such that $Q_i \prec_s Q_i \cup \{\alpha\} \preceq_s Q_j$. Since Pruning 3 could not exclude $Q_i \cup \{\alpha\}$, $tail(Q_i) \prec \alpha$ and $\alpha \notin E(Q_i) \backslash Q_i$ hold. Then, we have $\alpha \notin E(Q_i)$.
On the other hand, since $Q_i \cup \{\alpha\} \preceq_s Q_j$, $Q_i \cup \{\alpha\} \subseteq E(Q_i \cup \{\alpha\}) \subseteq E(Q_j)$ holds. This means $\alpha \in E(Q_j)$. Therefore, we obtain $E(Q_i) \neq E(Q_j)$.

Case 2 : Since Q_i and Q_j are not comparable under $\prec_s$, they have common ancestors. Let Q be the maximum (youngest) ancestor among them. That is, Q is equivalent to the maximum common prefix of Q_i and Q_j.
In case of $Q = \phi$, from the definition of the ordering $\prec_s$, $head(Q_i)$ is not equivalent to $head(Q_j)$. Moreover, according to Pruning 2, $head(E(Q_i))$ and $head(E(Q_j))$ should be $head(Q_i)$ and $head(Q_j)$, respectively. Hence, $E(Q_i) \neq E(Q_j)$ holds.
In order to prove $E(Q_i) \neq E(Q_j)$ in case of $Q \neq \phi$, we assume without loss of generality that in our depth-first search, Q_i is examined prior to Q_j. Since $Q \prec_s Q_i$, there exists a child of Q, $Q \cup \{\alpha_i\}$, such that $Q \prec_s Q \cup \{\alpha_i\} \preceq_s Q_i$ Similarly, we can consider another child, $Q \cup \{\alpha_j\}$, such that $Q \prec_s Q \cup \{\alpha_j\} \preceq_s Q_j$. Note here that from the assumption, $tail(Q) \prec \alpha_i \prec \alpha_j$.
Since Pruning 4 could not exclude $Q \cup \{\alpha_i\}$, for each $\beta \in E(Q \cup \{\alpha_i\}) \backslash E(Q)$, $\alpha_i \preceq \beta$ holds. Similarly, for each $\beta \in E(Q \cup \{\alpha_j\}) \backslash E(Q)$, $\alpha_j \preceq \beta$ holds. Furthermore, α_i and α_j are not contained in $E(Q)$ according to Pruning 3. Hence, we have $\alpha_i \in E(Q \cup \{\alpha_i\})$ and $\alpha_i \notin E(Q \cup \{\alpha_j\})$. It should be noted here that by Pruning 4, for any clique Q' such that $Q \cup \{\alpha_j\} \preceq_s Q'$, $E(Q')$ can never contain α_i. That is, $\alpha_i \notin E(Q_j)$. On the other hand, since $Q \cup \{\alpha_i\} \preceq_s Q_i$, it is obvious that $\alpha_i \in E(Q_i)$. Therefore, we can conclude $E(Q_i) \neq E(Q_j)$. ■

The above prunings are basically based on theoretical properties of FCs. In addition to them, we can also enjoy a simple branch-and-bound pruning based on a theoretical property of cliques. It is adopted as a basic pruning mechanism in several efficient algorithms for finding the maximum clique in a given graph (Balas & Yu, 1986; Fahle, 2002; Tomita & Kameda, 2007). This kind of pruning is based on the following simple property.

Observation 6.
For cliques Q and Q' in G such that $Q \subseteq Q'$, $Q \cup cand(Q) \supseteq Q' \cup cand(Q')$. ■

Proof:
It can be easily proved. Since $Q \subseteq Q'$ and each $v \in cand(Q')$ is adjacent to all vertices in Q', v is adjacent to any vertex in Q, that is, $v \in cand(Q)$. For each vertex $v \in Q' \backslash Q$, $Q \subseteq Q' \backslash \{v\}$. Since such a vertex v is adjacent to all vertices in $Q' \backslash \{v\}$, v is always adjacent to any vertex in Q, that is, $v \in cand(Q)$. Thus, we have $Q \cup cand(Q) \supseteq Q' \cup cand(Q')$. ■

Let us assume that we have a list of tentative Top-N δ-valid FCs already found in our search. Based on the above property, we can obtain a simple but quite effective pruning rule.

Pruning 5.
Let min be the minimum extent value in the tentative list. For a clique Q in G, if $eval_E(Q \cup cand(Q)) < min$, then no descendant of Q has to be examined.

Proof:
For a clique Q, let Q' be a descendant of Q, that is, $Q \subset Q'$. Since $E(Q')$ is also a clique containing Q', $Q \subset Q' \subseteq E(Q')$. There always exists a maximal clique in G, Q_{max}, such that $Q \subset Q' \subseteq E(Q') \subseteq Q_{max}$. Note here that the extensible candidates of Q_{max} becomes ϕ because Q_{max} is maximal. From Observation 6, therefore, $Q_{max} \subseteq Q \cup cand(Q)$ holds. Then, we also have $E(Q') \subseteq Q \cup cand(Q)$. It gives an inequality $eval_E(E(Q')) \leq eval_E(Q \cup cand(Q))$. From the assumption in the pruning rule, we know $eval_E(E(Q')) < min$. This means that the FC obtained from Q', $(E(Q'), \varphi(Q'))$, can never be in Top-N because its extent value is less than the tentative N-th value of extents. Therefore, we can safely prune any descendant of Q as useless ones. ■

In the pruning rule, thus, $eval_E(Q \cup cand(Q))$ can work as an *upper bound* of evaluation values we can observe by expanding Q.

Remark: If our evaluation function $eval_E$ is simply defined as size, the upper bound can be represented as

$$eval_E(Q \cup cand(Q)) = |Q \cup cand(Q)| = |Q| + |cand(Q)|.$$

In this case, the upper bound can be further improved.

Let Q_{max} be a maximum clique which is an expansion of Q. From the definition of $cand(Q)$, $Q_{max} \subseteq Q \cup cand(Q)$ holds. Since $cand(Q)$ is in general not a clique, we can consider a maximum clique Q'_{max} in $G(cand(Q))$, the subgraph of G induced by $cand(Q)$, such that $Q_{max} = Q \cup Q'_{max}$. Therefore, if we can compute any upper bound K for the size of a maximum clique in $G(cand(Q))$, we have

$$|Q_{max}| = |Q \cup Q'_{max}| = |Q| + |Q'_{max}| \leq |Q| + K \leq |Q| + |cand(Q)|.$$

That is, $|Q| + K$ can work as a better (tighter) upper bound of evaluation values we can obtain by expanding Q.

Upper bounds for the size of a maximum clique have been widely utilized in efficient branch-and-bound algorithms for finding a maximum clique (Fahle, 2002; Tomita & Kameda, 2007). The literature (Fahle, 2002) has summarized several classes of upper bounds. According to the argument in (Fahle, 2002), the (*vertex*) *chromatic number* χ can be tightest among well-known upper bounds. However, identifying χ is an NP-complete problem. Therefore, *approximations* of χ are computed in algorithms previously proposed. For more detailed discussion, see the literature (Fahle, 2002; Tomita & Kameda, 2007).

6.4 Algorithm for finding top-N δ-valid formal concepts

With the help of the pruning rules, we can design a depth-first branch-and-bound algorithm for finding Top-N δ-valid FCs. The pruning rules are adequately incorporated into the basic algorithm previously presented. A pseudo-code of our algorithm is shown in Figure 6.

7. Experimental results

We present in this section our experimental results. An example of document cluster actually extracted is presented. Since this chapter mainly focuses on the algorithmic viewpoint, we also discuss the empirical computational performance of our algorithm.

Input :
$< \mathcal{O}, \mathcal{F}, R >$: a formal context where
$\mathcal{O}$: a set of objects, $\mathcal{F}$: a set of features, and R : a binary relation on $\mathcal{O}$ and $\mathcal{F}$
δ : a threshold for intent value
N : an integer for Top-N
$eval_E$: an evaluation function for extents defined with an weight function $w_{\mathcal{O}}$ for $\mathcal{O}$
$eval_I$: an evaluation function for intents defined with an weight function $w_{\mathcal{F}}$ for $\mathcal{F}$
Output :
$\mathcal{FC}$: the set of δ-valid formal concepts whose extent values are in the top N

---

procedure main() :
  $\mathcal{FC} \leftarrow \phi$; /* Global variable */
  $min = 0.0$; /* Global variable */
  $G \leftarrow (\mathcal{O}, E, w_{\mathcal{O}})$ where
    $E = \{(x_i, x_j) \mid x_i, x_j \in \mathcal{O}(i \neq j) \wedge eval_I(\mathcal{F}(x_i) \cap \mathcal{F}(x_j)) \geq \delta\}$; /* Global variable */
  for each $x \in \mathcal{O}$ in predefined order **do**
    begin
      if $eval_I(\mathcal{F}(x)) \geq \delta$ **then** /* Pruning 1 */
        TopNFCFind($\{x\}, \mathcal{F}(x), \phi, N_G(x)$) ;
    end
  return $\mathcal{FC}$;

---

procedure TopNFCFind(X, I, $Prev$, $Cand$) :
  if $head(X) \neq head(E(X))$ or /* Pruning 2 */
      $\exists x \in E(X) \backslash Prev$ such that $x \prec tail(X)$ **then** /* Pruning 4 */
    return ;
  else
    TopNListUpdata($(E(X), I)$) ;
  endif
  for each $x \in Cand \backslash E(X)$ such that $tail(X) \prec x$ in predefined order **do** /* Based on Pruning 3 */
    begin
      $NewX \leftarrow X \cup \{x\}$;
      $NewI \leftarrow I \cap \mathcal{F}(x)$;
      $NewCand \leftarrow Cand \cap N_G(x)$;
      if $eval_I(NewI) < \delta$ **then** /* Pruning 1 */
        continue ;
      endif
      if $\mathcal{FC}$ tentatively contains N-th ones and
          $eval_E(NewX \cup NewCand) < min$ **then** /* Pruning 5 */
        continue ;
      else
        TopNFCFind($NewX$, $NewI$, $E(X)$, $NewCand$) ;
      endif
    end

---

procedure TopNListUpdate(FC) :
  $\mathcal{FC} \leftarrow \mathcal{FC} \cup \{FC\}$;
  if $\mathcal{FC}$ tentatively contains N-th ones **then**
    $min \leftarrow N$-th extent value ;
    Remove M-th ones from $\mathcal{FC}$ such that $N < M$;
  endif

Fig. 6. Algorithm for Finding Top-N δ-Valid Formal Concepts

Extent :

```
http : //news.bbc.co.uk/sport3/worldcup2002/hi/
          matches_wallchart/south_korea_v_poland/default.stm          [10]
http : //news.bbc.co.uk/sport3/worldcup2002/hi/
          matches_wallchart/tunisia_v_japan/default.stm               [265]
                              ⋮
http : //news.bbc.co.uk/sport3/worldcup2002/hi/
          team_pages/france/newsid_2097000/2097020.stm                [328]
http : //news.bbc.co.uk/sport3/worldcup2002/hi/
          matches_wallchart/england_v_brazil/
          newsid_2049000/2049924.stm                                  [562]
```

Intent:

`Russia`, `Belgium`, `go`, `bbc`, ... etc.

Fig. 7. Example of 500.0-Valid *FC* for *WorldCup*

Our algorithm has been implemented in language *C* and compiled by *gcc* with the optimization of *O*3 on *FreeBSD*. For comparison in the viewpoint of computational efficiency, two closed itemset miners, `AFOPT` (Liu et al., 2003) and `LCM` (Uno et al., 2004), have been also compiled. All of the systems have been run on a PC with Xeon 2.4 GHz CPU and 1GB main memory.

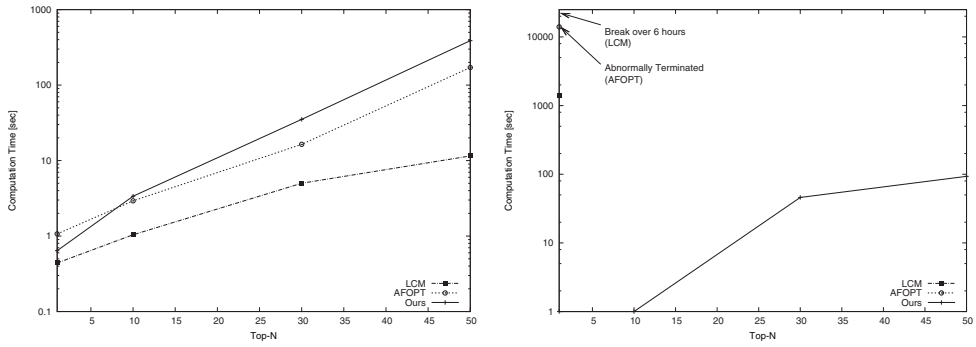
7.1 Dataset

For the experimentation, we have prepared a dataset, *WorldCup*, which is a collection of web pages. They have been retrieved with Google SOAP Search API [2] under the key words, "World Cup" and { "Germany", "France", "Brazil", "Korea", "Japan", "ticket" }. The number of documents (pages) is 5,971. Each document corresponds to a snippet retrieved by Google API. After *Stemming Process* with *Porter Stemmer* (Porter, 1980), we have discarded any words whose document frequencies are out of the range of $[50, 700]$. The remaining words are regarded as feature terms and the number of them is 2,824.
Each document has been *linearly* assigned a weight according to its rank. Moreover, each feature term has been given the IDF value as its weight. Each extent and intent have been evaluated by the sum of individual weights.

7.2 Extracted document clusters

Figure 7 shows an example cluster extracted from $WorldCup$, under $N = 50$ and $\delta = 500.0$. It is 33-th one. In the figure, each URL is accompanied with its rank on the right. A point worthy to remark is that the cluster consists of web pages with their ranks within a wide range. Since we usually browse web pages only with relatively higher ranks, lower-ranked pages are almost discarded in many cases, regardless of their contents. However, if such a lowly ranked page is concerned with some feature terms shared with several higher-ranked ones, we can expect that it is probably valuable. Thus, our cluster can make such hidden useful pages visible. A similar effect has been also observed in (Haraguchi & Okubo, 2010; 2006b; Okubo et al., 2005).

[2] It is no longer available.



(a) $\delta = 80$ (b) $\delta = 120$

Fig. 8. Computation Time under Various Top-N for *WorldCup*

7.3 Computational performance

As has been discussed, any closed itemset miner would be helpful for finding Top-N δ-valid FCs. We compare computational performance of our algorithm with some of efficient closed itemset miners. Such a system usually enumerates all frequent closed itemsets under a given minimum support threshold (*minsup*). Therefore, we can naively obtain Top-N FCs by first enumerating frequent closed itemsets including Top-N FCs and then choosing the targets from them. Since the former task is the dominant part of the whole process, we particularly report on computation times for the task.

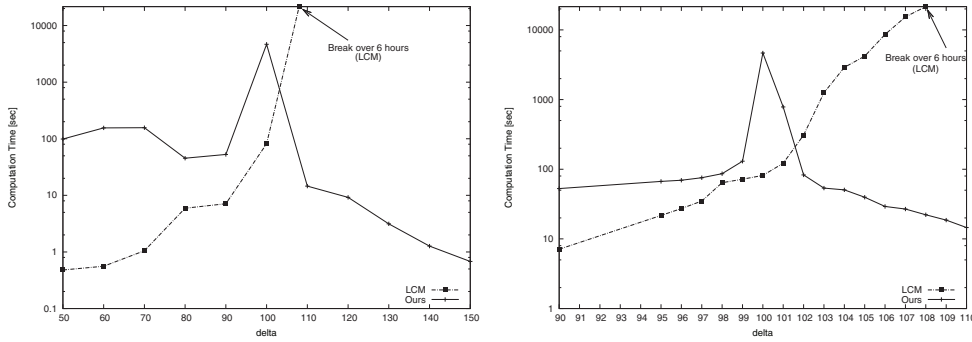
It is noted here that weights of objects (transactions) and features (items) are out of considerations in these miners. Therefore, each object and feature are assigned uniform weights, 1.0. Then each extent and intent are evaluated by the weight sum, that is, by their sizes.

For the comparison, `LCM` (Uno et al., 2004) and `AFOPT` (Liu et al., 2003) have been selected as very fast frequent closed itemset miners. Their implementations are available at *Frequent Itemset Mining Implementations Repository* [3]. In order to find Top-N δ-valid FCs by the systems, we have to provide some *minsup* under which all of the Top-N FCs can be enumerated as frequent ones. However, we have no idea about an adequate *minsup* in advance. If a given *minsup* is too high, some δ-valid FCs will be lost because a higher *minsup* forces us to extract only smaller closed itemsets equivalent to intents with lower evaluation values. Conversely, if a *minsup* is too low, we will obtain a large number of closed itemsets, even though we can find all of the targets from them. The most adequate value of *minsup* is given as the extent size of the N-th δ-valid FCs. Under the optimal *minsup*, the number of closed itemsets to be enumerated can be minimized. In our experimentations, the closed itemset miners are executed under the optimal *minsup* in each problem setting. It should be emphasized here that it is certainly advantageous to the miners.

For the dataset *WorldCup*, we try to find Top-N δ-valid FCs under $\delta = 80$ and $\delta = 120$. For each δ setting, we observe computation times changing the parameter N. After the execution of our system for each problem setting, `LCM` and `AFOPT` have been given the N-th extent size as the optimal *minsup*. The results are shown in Figure 8.

From the figure, we guess that in case of lower δ, the closed itemset miners can quickly enumerate candidates including the targets. `LCM` is especially a system worthy to remark.

[3] `http://fimi.cs.helsinki.fi/`



(a) (b) Around $\delta = 100$

Fig. 9. Computation Time under Various δ for *WorldCup* ($N = 30$)

As the result, they will find final targets faster than our system does. However, the authors emphasize here that although they have been ideally given the best *minsup* for each problem, we never know such best values of *minsup* in actual situations.

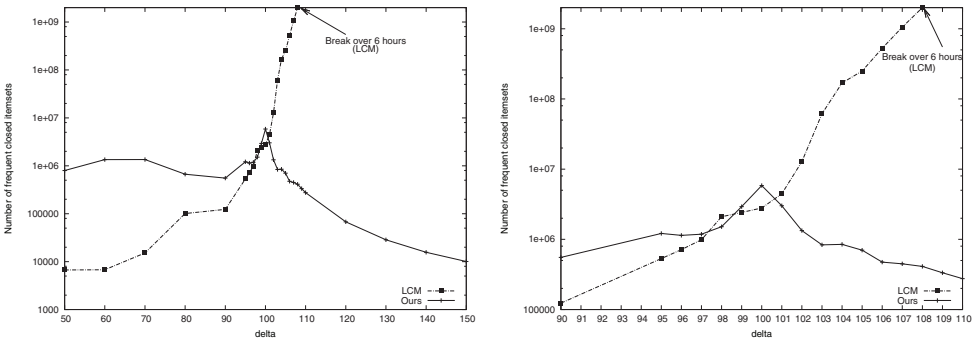
On the other hand, in case of higher δ, our system outperforms the other systems. These miners have failed to enumerate the closed itemsets within 6 hours. As δ becomes higher, the miners are given a lower *minsup*. It is well known that in such a *minsup*-based system, a lower *minsup* brings just a small reduction of the search space. In other words, under such a lower *minsup*, the number of frequent (closed) itemsets becomes quite huge. The figure seems to reflect this fact.

In order to analyze such a performance behavior more precisely, we observe computation times by `LCM` and our system changing the parameter δ under $N = 30$. The results are shown in Figure 9 (a). From the figure, as we have guessed just above, in lower range of δ, `LCM` can enumerate the closed itemsets including Top-N FCs much faster than our system. On the other hand, in higher range of δ, our system outperforms `LCM`. In particular, since the performance curves cross in a range around $\delta = 100$, detailed curves around the range are shown in Figure 9 (b).

As is shown in Figure 9 (b), the performance curves cross between $\delta = 101$ and $\delta = 102$. Roughly speaking, the computational performance of `LCM` is primary affected the number of frequent closed itemsets, and that of our system the number of FCs we examined in our search. Therefore, we also observe these numbers in order to precisely compare the performance in more detail.

The results are presented in Figure 10 (a) and the detailed curves around $\delta = 100$ are also shown in Figure 10 (b). The curves cross between $\delta = 100$ and $\delta = 101$. In a strict sense, this observation is slightly different from the case of computation time. This observation stems from the difference between the processing cost for each closed itemset and that for each FC. The latter cost (that is, ours) is slightly higher than the former. Since `LCM` can enumerate a more number of frequent closed itemsets, we have had such slight different observations. Therefore, the authors consider that they are consistent.

We observe an exponential growth in the number of closed itemsets `LCM` has actually enumerated, as is shown in Figure 10. The fact can be clearly explained as follows. The parameter δ corresponds to some threshold for the minimum size of itemsets we can accept. Furthermore, frequencies (supports) of itemsets are monotonically decreasing, as itemsets



(a) (b) Around $\delta = 100$

Fig. 10. Number of frequent closed itemsets and examined FCs for $WorldCup$ ($N = 30$)

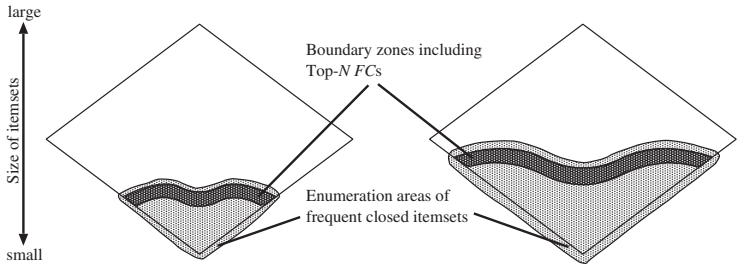


Fig. 11. Boundary Zones including Top-N δ-valid FCs and enumeration areas of frequent closed itemsets in itemset-lattices

become larger. Our task of finding Top-N FCs is equivalent to extract closed itemsets with enough sizes and Top-N frequencies. Intuitively speaking, therefore, our targets of Top-N δ-valid FCs can be found in some boundary zone determined by δ, as is illustrated in Figure 11. If frequent closed itemset miners like `LCM` is used for finding Top-N FCs, they have to enumerate the closed itemsets in the lower part of the itemset lattice including the boundary zone. In case of lower δ, since such a boundary zone lies lower in the lattice, the number of closed itemsets will not be so large (Figure 11 (a)). Therefore, they can enumerate all of them very quickly as we have observed above. On the other hand, as δ gets higher, a boundary zone in the lattice rises and the number of closed itemsets to be enumerated will drastically increase (Figure 11 (b)).

From these figures, in higher range of δ, any algorithms for finding frequent closed itemsets are no longer practical for our Top-N δ-valid FC problem, even though they are quite helpful in lower range of δ. The authors, however, expectantly assert that in order to find FCs which are actually interesting for us, we need to provide a relatively higher value for δ. Under a lower δ, we will obtain FCs with larger extents (clusters). Such large clusters seem to be ordinary and not so interesting for us. The authors expect that clusters which are not too large would provide us valuable information in practical sense. In order to find such interesting clusters,

we are required to give a relatively higher value of δ. We can, therefore, consider our algorithm to be suitable and effective for this purpose.

8. Concluding remarks

In this chapter, we discussed a method for conceptual clustering of documents. Since our document clusters are extracted as extents of formal concepts, each document cluster can be provided a clear conceptual meaning in terms of feature terms. Our task was formalized as the Top-N δ-valid FC problem. We designed an efficient depth-first branch-and-bound algorithm which is an improved version of our previous algorithm with some new pruning rules. The safeness and completeness of the prunings were verified with theoretical proofs. Our experimental results showed that our document clusters can be extracted with reasonable computation time. Furthermore, we verified that our algorithm outperforms several efficient closed itemset miners in certain problem settings.

Document clusters we can actually extract are much affected by terms we provide as features. In case of web pages, it might depend on the result of converting HTML sources into texts. We need to adequately remove useless terms such as advertisements. Those points should be further investigated as important future work.

Quality of our clusters will be also related to how we assign a weight to each feature term and document. We need to analyze relationship between weight assignment and quality of clusters in more details.

It is important to investigate the scalability of our algorithm. By conducting further experimentations, we need to observe its computational performance for datasets with the order of hundreds thousands.

Needless to say, our method is not only for datasets of documents (web pages). We can apply the method to any dataset in which each object to be clustered can be represented as a set of features, like *relational data*. Applying the method to other practical domain will be interesting work.

9. References

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules, *Proceedings of the 20th International Conference on Very Large Databases - VLDB'94*, pp.487 – 499, ISBN 1-55860-153-8, Santiago de Chile, September 1994, Morgan Kaufmann Publishers, CA.

Balas, E. & Yu, C. S. (1986). Finding a Maximum Clique in an Arbitrary Graph, *SIAM Journal on Computing*, Vol. 15, No. 4, pp. 1054 – 1068, ISSN 0097-5397.

Besson, J.; Robardet, C. & Boulicaut, J. (2005). Constraint-Based Concept Mining and Its Application to Microarray Data Analysis, *Intelligent Data Analysis*, Vol. 9, No. 1, pp. 59 – 82, ISSN 1088-467X.

Fahle, T. (2002). Simple and Fast: Improving a Branch and Bound Algorithm for Maximum Clique, *Proceedings of the 10th European Symposium on Algorithms - ESA'02*, LNCS 2461, pp. 485 – 498, ISBN 3-540-44180-8, Rome, September 2002, Springer, Berlin.

Ganter, B & Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations*, Springer, ISBN 978-3540627715, Berlin.

Han, J.; Cheng, H.; Xin, D. & Yan, X. (2007). Frequent Pattern Mining - Current Status and Future Directions, *Data Mining and Knowledge Discovery*, Vol. 15, No. 1, pp. 55 – 86, ISSN 1384-5810.

Han, J. & Kamber, M. (2006). *Data Mining - Concepts and Techniques (Second Edition)*, Morgan Kaufmann Publishers, ISBN 1-55860-901-6, CA.

Haraguchi, M. & Okubo, Y. (2010). Pinpoint Clustering of Web Pages and Mining Implicit Crossover Concepts, In: *Web Intelligence and Intelligent Agents*, Usmani, Z. (Ed.), pp. 391 – 410, INTECH, ISBN 978-953-7619-85-5, Rijeka.
(Online version : http://sciyo.com/articles/show/title/pinpoint-clustering-of-web-pages-and-mining-implicit-crossover-concepts)

Haraguchi, M. & Okubo, Y. (2007). An Extended Branch-and-Bound Search Algorithm for Finding Top-*N* Formal Concepts of Documents, In: *New Frontiers in Artificial Intelligence, JSAI 2006 Conference and Workshops, Tokyo, Japan, June 5-9, 2006, Revised Selected Papers* , Washio, T., Satoh, K., Takeda, H. and Inokuchi, A. (Eds.), LNCS 4384, pp. 276 – 288, Springer, ISBN 3-540-69901-5, Berlin.

Haraguchi, M. & Okubo, Y. (2006a). An Extended Branch-and-Bound Search Algorithm for Finding Top-*N* Formal Concepts of Documents, *Proceedings of the 4th Workshop on Learning with Logics and Logics for Learning - LLLL'06*, pp. 41 – 47, Tokyo, June 2006, JSAI, Tokyo.

Haraguchi, M. & Okubo, Y. (2006b). A Method for Pinpoint Clustering of Web Pages with Pseudo-Clique Search, In: *Federation over the Web, International Workshop, Dagstuhl Castle, Germany, May 1 - 6, 2005, Revised Selected Papers*, Jantke, K. P., Lunzer, A., Spyratos, N. and Tanaka, Y. (Eds.), LNAI 3847, pp. 59 – 78, Springer, ISBN 3-540-31018-5, Berlin.

Hotho, A.; Staab, S. & Stumme, G. (2003). Explaining Text Clustering Results Using Semantic Structures, *Proceedings of the 7th European Conference on Principles of Data Mining and Knowledge Discovery - PKDD'03*, LNCS 2838, pp. 217 – 228, ISBN 3-540-20085-1, Cavtat-Dubrovnik, September 2003, Springer, Berlin.

Liu, G.; Lu, H.; Yu, J. X.; Wei, W. & Xiao, X. (2003). AFOPT: An Efficient Implementation of Pattern Growth Approach, *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations - FIMI'03*, http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-90/, ISSN 1613-0073.

Lucchese, C.; Orlando, S. & Perego, R. (2004). DCI-Closed: A Fast and Memory Efficient Algorithm to Mine Frequent Closed Itemsets, *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations - FIMI'04*, http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-126/, ISSN 1613-0073.

Okubo, Y. & Haraguchi, M. (2006). Finding Conceptual Document Clusters with Improved Top-N Formal Concept Search, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence - WI'06*, pp. 347 – 351, ISBN 0-7695-2747-7, Hong Kong, December 2006, IEEE Computer Society, CA.

Okubo, Y.; Haraguchi, M. & Shi, B. (2005). Finding Significant Web Pages with Lower Ranks by Pseudo-Clique Search, *Proceedings of the 8th International Conference on Discovery Science - DS'05*, LNAI 3735, pp. 346 – 353, ISBN 3-540-29230-6, Singapore, October 2005, Springer, Berlin.

Okubo, Y. & Haraguchi, M. (2003). Creating Abstract Concepts for Classification by Finding Top-*N* Maximal Weighted Cliques, *Proceedings of the 6th International Conference on Discovery Science - DS'03*, LNAI 2843, pp. 418 – 425, ISBN 3-540-20293-5, Sapporo, October 2003, Springer, Berlin.

Pasquier, N.; Bastide, Y.; Taouil, R. & Lakhal, L. (1999). Efficient Mining of Association Rules Using Closed Itemset Lattices, *Information Systems*, Vol. 24, No. 1, pp. 25 – 46, ISSN 0306-4379.

Porter, M. F. (1980). An Algorithm for Suffix Stripping, *Program: Electronic Library and Information Systems*, Vol. 14, No. 3, pp. 130 – 137, ISSN 0033-0337.

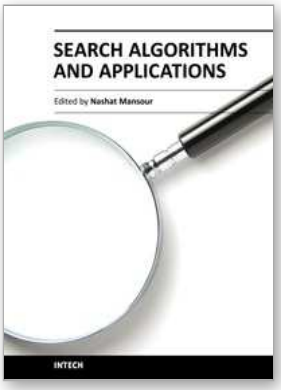
Rymon, R. (1992). Search through Systematic Set Enumeration, *Proceedings of International Conference on Principles of Knowledge Representation Reasoning - KR'92*, pp. 539 – 550, ISBN 1-55860-262-3, Cambridge, October 1992, Morgan Kaufmann Publishers, CA.

Tomita, E. & Kameda, T. (2007). An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique with Computational Experiments, *Journal of Global Optimization*, Vol. 37, No. 1, pp. 95 – 111, ISSN 0925-5001.

Uno, T.; Kiyomi, M. & Arimura, H. (2004). LCM ver. 2: Efficient Mining Algorithm for Frequent/Closed/Maximal Itemsets, *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations - FIMI'04*, `http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-126/`, ISSN 1613-0073.

Vakali, A.; Pokorný, J. & Dalamagas, T. (2004). An Overview of Web Data Clustering Practices, *Current Trends in Database Technology - EDBT 2004 Workshops, EDBT 2004 Workshops PhD, DataX, PIM, P2P&DB, and ClustWeb, Heraklion, Crete, Greece, March 14-18, 2004, Revised Selected Papers*, Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y. and Vakali, A. (Eds.), LNCS 3268, pp. 597 – 606, Springer, ISBN 3-540-23305-9, Berlin.

Wang, J.; Han, J.; Lu, Y. & Tzvetkov, P. (2005). TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 5, pp. 652 – 664, ISSN 1041-4347.



Search Algorithms and Applications

Edited by Prof. Nashat Mansour

ISBN 978-953-307-156-5
Hard cover, 494 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Search algorithms aim to find solutions or objects with specified properties and constraints in a large solution search space or among a collection of objects. A solution can be a set of value assignments to variables that will satisfy the constraints or a sub-structure of a given discrete structure. In addition, there are search algorithms, mostly probabilistic, that are designed for the prospective quantum computer. This book demonstrates the wide applicability of search algorithms for the purpose of developing useful and practical solutions to problems that arise in a variety of problem domains. Although it is targeted to a wide group of readers: researchers, graduate students, and practitioners, it does not offer an exhaustive coverage of search algorithms and applications. The chapters are organized into three parts: Population-based and quantum search algorithms, Search algorithms for image and video processing, and Search algorithms for engineering applications.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821