Discovering Coverage Patterns for Banner Advertisement Placement

P. Gowtham Srinivas, P. Krishna Reddy, S. Bhargav,
R. Uday Kiran, and D. Satheesh Kumar

International Institute of Information Technology Hyderabad, India
{gowtham.srinivas,bhargav.spg08,uday_rage,satheesh.kumar}@
research.iiit.ac.in,pkreddy@iiit.ac.in

Abstract. We propose a model of coverage patterns and a methodology to extract coverage patterns from transactional databases. We have discussed how the coverage patterns are useful by considering the problem of banner advertisements placement in e-commerce web sites. Normally, advertiser expects that the banner advertisement should be displayed to a certain percentage of web site visitors. On the other hand, to generate more revenue for a given web site, the publisher has to meet the coverage demands of several advertisers by providing appropriate sets of web pages. Given web pages of a web site, a coverage pattern is a set of pages visited by a certain percentage of visitors. The coverage patterns discovered from click-stream data could help the publisher in meeting the demands of several advertisers. The efficiency and advantages of the proposed approach is shown by conducting experiments on real world click-stream data sets.

Keywords: Click stream mining, online advertising, internet monetization, computational advertising, graphical ads delivery.

1 Introduction

We have proposed a model of data mining pattern, called, “coverage patterns” and a methodology to discover coverage patterns from transactional databases. Given a set of data items, a coverage pattern is a set of non-overlapping data items covered by a certain percentage of transactions. An Apriori-like algorithm [1] called *CMine* is proposed for mining coverage patterns.

In the literature, the notion of coverage is being used for solving the set cover problem [2] in set theory and node cover problem [3] in graphs respectively. In [4], the notion of coverage and overlap is used to examine the creation of a tag cloud for exploring and understanding a set of objects. In [5], the notion of coverage and overlap is used to solve the problem of topical query decomposition. In this paper, we have proposed a different kind of knowledge patterns. The proposed patterns can be employed in improving the performance of several applications such as banner advertisements.

The research in this paper is motivated with the problem of banner advertisement placement. The background and problem description is as follows.

Banner advertising is one of the dominant modes of online advertising, in addition to the contextual and sponsored search advertising. A banner advertisement is described as a hypertext link that is associated with a box containing graphics which is redirected to a particular web page when a user clicks on the banner [6]. The following three entities are involved in banner advertising: advertiser, publisher and visitor. An advertiser is interested in endorsing products through banner advertisements. A publisher manages a web site or an advertisement network that sells banner advertisement space. Finally, a visitor visits the web pages of a web site which contains banners.

An advertiser has the goal of spreading his/her advertisement to a certain percentage of people visiting a web site. The goal of the publisher is to make more revenue by efficiently using the advertising space available in the web pages of a web site and meeting the demands of multiple advertisers. For a given web site and period, one can analyse the visitors' behaviour by processing the transactions generated based on click stream dataset and identify the sets of web pages that cover a given percentage of visitors' population. However, the research issue here is to investigate the approaches for discovering the sets of web pages which can cover a given percentage of visitors' population based on transactions extracted from the click stream data.

Most of the research work on online advertisement has been focused on auction models [7], keyword or phrase identification based on user queries [8], contextual advertising [9] and allocation and scheduling of advertisements [10]. To our knowledge, not much amount of research work has been carried out on improving the options offered by the publisher to the advertisers.

The proposed model of coverage patterns could help the advertiser by making his advertisement visible to a certain percentage of web site visitors. With the proposed approach, it is possible to ensure that the publisher can meet the demands of multiple advertisers by considering several groups of potential pages. Through experimental results on the real world datasets we show that the proposed model and algorithm is efficient. It has a potential to improve the performance of banner advertisement placement.

A preliminary approach was presented in [11] to extract coverage patterns for banner advertisement placement. In this paper we have elaborated the model and presented a formal model of coverage patterns. We also proposed an efficient algorithm to extract complete set of coverage patterns and conducted experiments.

The rest of this paper is organized as follows: In section 2, we propose the model and approach to extract coverage patterns. In section 3, we present experimental results. In the last section, we present the conclusion and future work.

2 Model of Coverage Patterns

In this section, we first explain the model of coverage patterns. Next, we discuss the computational issues involved in extracting coverage patterns and explain how the notion of sorted closure property can be exploited for efficient extraction of coverage patterns. Subsequently, we present the algorithm to extract coverage patterns.

2.1 Coverage Patterns

As already mentioned, we identify the issue of banner advertisement placement as one of the potential application of coverage patterns. For a given e-commerce web site, the transactions generated from click stream dataset can be used to identify the sets of web pages that cover a given percentage of visitors' population. Such a knowledge could be used to place the banner advertisements assuming similar visitors' behaviour. The related issues will be investigated as a part of future work.

To present the model of coverage patterns, we consider transactions generated from click stream data of a web site. However, the model can be extended to any transactional data set.

The basic terminology is as follows: Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of identifiers of web pages and D be a set of transactions, where each transaction T is a set of web pages such that $T \subseteq W$. Associated with each transaction is a unique transactional identifier called TID. Let T^{w_i}, $w_i \in W$ be the set of all $TIDs$ in D that contain the web page w_i. A set of web pages $X \subseteq W$ i.e., $X = \{w_p, \cdots, w_q, w_r\}$, $1 \leq p \leq q \leq r \leq n$, is called the pattern. A pattern containing k number of web pages is called a k-pattern. In other words, the length of k-pattern is k.

Example 1 *Consider the transactional database shown in Table 1. It contains 10 transactions. The set of pages, $W = \{a, b, c, d, e, f\}$. The $TIDs$ containing the web page 'a' are 1, 2, 3, 4 and 10. Therefore, $T^a = \{1, 2, 3, 4, 10\}$. The set of web pages 'a' and 'b' i.e., $\{a, b\}$ is a pattern. Since there are two web pages in this pattern it is a 2-pattern.*

Table 1. Transactional database

TID	1	2	3	4	5	6	7	8	9	10
Pages	a, b, c	a, c, e	a, c, e	a, c, d	b, d, f	b, d	b, d	b, e	b, e	a, b

The percentage of transactions in D that contain the web page $w_i \in W$ is known as the "relative frequency of a web page $w_i \in W$" and denoted as $RF(w_i)$.

Definition 1 *(Relative frequency of a web page $w_i \in W$.) Let $|T^{w_i}|$ indicates the total number of transactions that contain w_i. The relative frequency of w_i is denoted as $RF(w_i)$. That is, $RF(w_i) = \frac{|T^{w_i}|}{|D|}$.*

Note that from the advertisement point of view the pages that are visited by more number of users are interesting. We capture this aspect with the notion of frequent page. The frequent web pages are web pages which have relative frequency no less than the user-specified threshold value, called minimum relative frequency.

Definition 2 *(Frequent web page.) A web page $w_i \in W$ is considered frequent if $RF(w_i) \geq minRF$, where $minRF$ is the user-specified minimum relative frequency threshold.*

Example 2 *Continuing with the example, the relative frequency of 'a' i.e., $RF(a) = \frac{|T^a|}{|D|} = \frac{5}{10} = 0.5$. If the user-specified $minRF = 0.5$, then 'a' is called a frequent web page because $RF(a) \geq minRF$.*

Next, we capture the notion that given a set of web pages how many users visit at least one web page in the set. It means that if we place an advertisement on all pages in the set it will guarantee the delivery of advertisement to the users who visit atleast one page. This aspect is captured through the notion of *coverage set*.

Definition 3 *(Coverage set of a pattern $X = \{w_p, \cdots, w_q, w_r\}$, $1 \leq p \leq q \leq r \leq n$.) The set of distinct $TIDs$ containing at least one web page of X is called the coverage set of pattern X and is denoted as $CSet(X)$. Therefore, $CSet(X) = T^{w_p} \cup \cdots \cup T^{w_q} \cup T^{w_r}$.*

A pattern will be interesting if its coverage set contains more than a threshold number of transactions. This aspect is captured through the notion of coverage support.

Definition 4 *(Coverage-support of a pattern X.) The ratio of size of coverage set of X to the transactional database size is called the coverage-support of pattern X and is denoted as $CS(X)$.*

$$CS(X) = \frac{|CSet(X)|}{|D|}. \tag{1}$$

Example 3 *The set of web pages 'a' and 'b' i.e., $\{a, b\}$ is a pattern. The set of tids containing the web page 'a' i.e., $T^a = \{1, 2, 3, 4, 10\}$. Similarly, $T^b = \{1, 5, 6, 7, 8, 9, 10\}$. The coverage set of $\{a, b\}$ i.e., $CSet(\{a, b\}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Therefore, coverage support of $\{a, b\}$ i.e., $CS(\{a, b\}) = \frac{|CSet(\{a,b\})|}{|D|} = \frac{10}{10} = 1$.*

For a pattern X, $CS(X) \in [0, 1]$. If $CS(X) = 0$, no single web page of X has appeared in the entire transactional database. If $CS(X) = 1$, every transaction in T contains at least one web page $w_j \in X$.

It can be noted that once a pattern X has certain coverage-support, then adding other web pages in particular web pages co-occurring with any of the web pages belonging to X to that pattern may not increase the coverage support significantly. From the advertisement point of view, such a pattern can be uninteresting to the advertiser. This is because the same users visit the web pages as there is an overlap of coverage set of X and coverage set of new single web page pattern.

Example 4 *In the transactional database shown in Table 1, $T^{\{a\}} = \{1, 2, 3, 4, 10\}$ and $T^{\{c\}} = \{1, 2, 3, 4\}$. The coverage-support of $\{a, c\}$ i.e., $CS(\{a, c\}) = \frac{5}{10} = 0.5$. If user-specified $minCS = 0.5$, then $\{a, c\}$ is an interesting pattern. However, this pattern is uninteresting as the pattern 'c' has not increased coverage-support of the pattern 'a'.*

To capture this aspect, we introduce the notion *overlap ratio of the pattern*.

Definition 5 *(Overlap ratio of a pattern.) Overlap ratio of a pattern $X = \{w_p, \cdots, w_q, w_r\}$, where $1 \leq p \leq q \leq r \leq n$ and $|T^{w_p}| \geq \cdots \geq |T^{w_q}| \geq |T^{w_r}|$, is the ratio of the number of transactions common in $X - \{w_r\}$ and $\{w_r\}$ to the number of transactions in w_r. It is denoted as $OR(X)$ and is measured as follows.*

$$OR(X) = \frac{|(T^{w_p} \cup \cdots \cup T^{w_q}) \cap (T^{w_r})|}{|T^{w_r}|} \tag{2}$$

For a pattern X, $OR(X) \in [0,1]$. If $OR(X) = 0$, there exists no common transactions between $X - \{w_r\}$ and $\{w_r\}$. If $OR(X) = 1$, w_r has occurred in all the transactions where at least one web page $w_j \in (X - \{w_r\})$ has occurred.

Example 5 *Continuing with Example 3, the* $OR(\{a,b\}) = \frac{|CSet(b) \cap CSet(a)|}{|CSet(a)|} = \frac{2}{5} = 0.4$.

Note that a coverage pattern is interesting if it has high coverage support and low overlap ratio. As a result an advertisement is exposed to more number of users by reducing repetitive display of the advertisement. The definition of coverage pattern is as follows.

Definition 6 *(Coverage pattern X.) A pattern X is said to be a coverage pattern if $CS(X) \geq minCS$, $OR(X) \leq maxOR$ and $RF(w_i) \geq minRF$, $\forall w_i \in X$. The variables, $minCS$ and $maxOR$ represent user-specified minimum coverage support and maximum overlap ratio, respectively. A coverage pattern X having $CS(X) = a\%$ and $OR(X) = b\%$ is expressed as*

$$X \quad [CS = a\%,\ OR = b\%] \tag{3}$$

Example 6 *If $minRF = 0.4$, $minCS = 0.7$ and $maxOR = 0.5$, then the pattern $\{a,b\}$ is a coverage pattern. It is because $RF(a) \geq minRF$, $RF(b) \geq minRF$, $CS(\{a,b\}) \geq minCS$ and $OR(\{a,b\}) \leq maxOR$. This pattern is written as follows:*

$$\{a,b\} \quad [CS = 1\ (= 100\%),\ OR = 0.4\ (= 40\%)]$$

Problem statement: Given a transactional database D, set of web pages W, and user-specified minimum relative frequency ($minRF$), minimum coverage support ($minCS$) and maximum overlap ratio ($maxOR$), discover complete set of coverage patterns such that

i. If X is a coverage 1-pattern (i.e., $k = 1$), then $RF(w_i) \geq minRF$ and $RF(w_i) \geq minCS$, $\forall w_i \in X$.
ii. Otherwise (i.e., when $k > 1$), each coverage pattern X must have $CS(X) \geq minCS$, $OR(X) \leq maxOR$ and $RF(w_i) \geq minRF$, $\forall w_i \in X$.

2.2 Mining Coverage Patterns

A naive approach to find the complete set of coverage patterns for a dataset consisting of n web pages is to generate all possible $(2^n - 1)$ combinatorial patterns (CP) from n web pages. Now, each pattern in CP is added to the coverage pattern set if it satisfies $minCS, minRF$ and $maxOR$ constraints. The problem with this approach is, if n is large the search space will be large leading to high computational cost. The search space can be reduced if the coverage pattern satisfies downward closure property on either coverage support or overlap ratio.

Our analysis on coverage patterns states that the measure *coverage support* does not satisfy *downward closure property*. That is, although a pattern satisfies $minCS$, it is not necessary that all its non-empty subsets will also satisfy $minCS$ value.

Example 7 *Consider the patterns $\{a\}$, $\{e\}$ and $\{a, e\}$. The coverage supports of these patterns are 0.5, 0.4 and 0.7, respectively. If the user-specified $minCS = 0.7$, then the pattern $\{a, e\}$ satisfies $minCS$ value. However, its non-empty subsets do not satisfy $minCS$ value.*

The parameter *overlap ratio* also does not satisfy *downward closure property* if a pattern is considered as an unordered set of web pages. However, this measure satisfies *downward closure property* if a pattern is an ordered set, where web pages are sorted in descending order of their frequencies. This property is known as the *sorted closure property* [12].

Property 1. If $X \subset Y$, then $CSet(X) \subseteq CSet(Y)$.

Property 2. **Sorted closure property**: Let $X = \{w_p, \cdots, w_q, w_r\}$ be a pattern such that $RF(w_p) \geq \cdots \geq RF(w_q) \geq RF(w_r)$ and $1 \leq p \leq q \leq r \leq n$. If $OR(X) \leq maxOR$, all its non-empty subsets containing w_r and having size $k \geq 2$ will also have overlap ratio less than or equal to $maxOR$.

Rationale: Let w_a, w_b and w_c be the web pages having $RF(w_a) \geq RF(w_b) \geq RF(w_c)$. If $OR(w_a \cup w_c) > maxOR$, then $OR(\{w_a \cup w_b\} \cup w_c) > maxOR$ because from $Property$ 1

$$\frac{|CSet(w_a) \cap CSet(w_c)|}{|CSet(w_c)|} \leq \frac{|CSet(\{w_a \cup w_b\}) \cap CSet(w_c)|}{|CSet(w_c)|} \tag{4}$$

Definition 7 *(Non-overlap pattern X.) A pattern X is said to be* non-overlap *if $OR(X) \leq maxOR$ and $RF(w_i) \geq minRF$, $\forall w_i \in X$.*

Every coverage pattern is a non-overlap pattern, however it is not the same vice versa. The *sorted closure property* of non-overlap patterns is used for minimizing the search space while mining complete set of coverage patterns by designing an algorithm similar to the Apriori algorithm [1]. The detailed algorithm for mining the complete coverage patterns is given in next subsection.

2.3 Coverage Pattern Extraction Algorithm

We use the following notations. Let F be a set of frequent items, C_k be a set of candidate k-patterns, L_k be a set of coverage k-patterns and NO_k be a set of non-overlap k-patterns. The proposed algorithm *CMine* employs a *level-wise* search to discover the complete set of coverage patterns. In *level-wise* search, k-patterns are used to explore $(k+1)$-patterns. The proposed CMine algorithm is different from Apriori algorithm [1] used for mining frequent patterns. The main reason is as follows: Frequent patterns satisfy *downward closure property*. Therefore, Apriori algorithm uses frequent k-patterns to explore $(k + 1)$-patterns. CMine cannot explore $(k + 1)$-patterns with coverage k-patterns as coverage patterns no longer satisfy *downward closure property*.

The detailed description of the algorithm is as follows: The algorithm CMine begins with a scan of the database and discovers set of all frequent web pages (denoted as F) and coverage 1-patterns (denoted as L_1). Non-overlap 1-patterns (denoted as NO_1) will be the set of all frequent 1 web pages. Next, web pages in NO_1 are sorted in descending order of their frequencies. This is an exception from Apriori algorithm [1] that has to be carried out in CMine algorithm to efficiently mine coverage patterns. Each web page $w_i \epsilon NO_1$ is of the form $< w_i, T^{w_i} >$ where T^{w_i} denote set of transaction ids which contain the web page w_i. Using NO_1 as a *seed set*, candidate patterns C_2 are generated by combining $NO_1 \bowtie NO_1$. From C_2, the patterns that satisfy $minCS$ and $maxOR$ are generated as coverage 2-patterns, L_2. Simultaneously, all candidate 2-patterns that satisfy $maxOR$ are generated as non-overlap 2-patterns NO_2. Since overlap patterns satisfy *sorted closure property*, C_3 is generated by combining $NO_2 \bowtie NO_2$. From C_3, L_3 and NO_3 are discovered. At each level 'k', a two-step process is followed, consisting of join and prune actions [13].

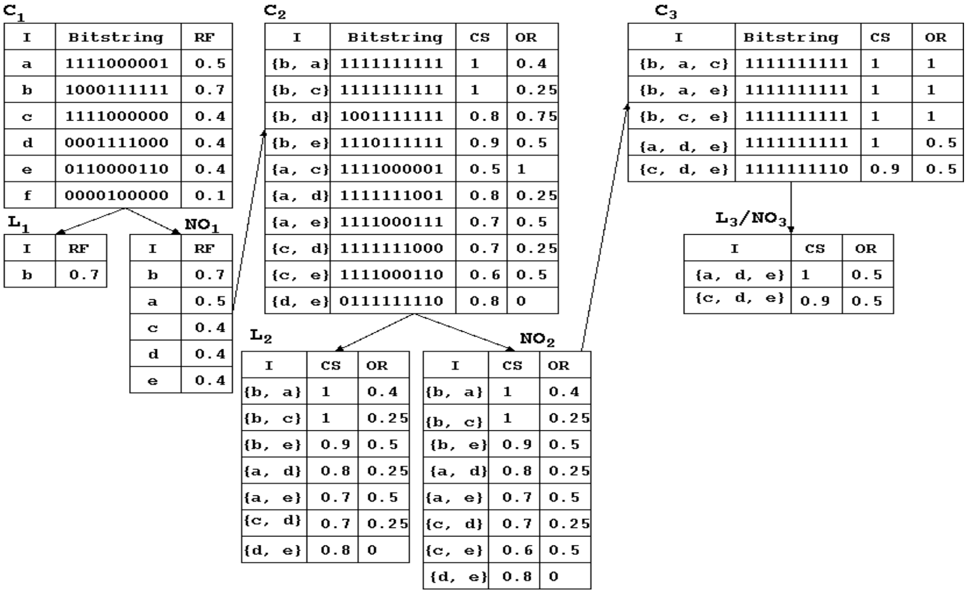
1. The join step: To find L_k, a set of candidate k-web page sets C_k is generated by joining NO_{k-1} with itself. Let l_1 and l_2 be web page sets in NO_{k-1}. Note that the members of NO_{k-1} are join-able if their first $(k-2)$ web pages are in common.
2. The prune step: C_k is a superset of L_k, that is, its members may or may not be coverage patterns, but all of the coverage k-web page sets are included in C_k. The number of k-web page sets in C_k, however, can be huge, and so this could involve heavy computation. To reduce the size of C_k, the Sorted closure property of non-overlap patterns is used as follows. Any $(k-1)$-web page set that is not satisfying the overlap ratio cannot be a subset of a non-overlap k-web page set. Hence, if any $(k-1)$-ordered subset of a candidate k-web page set is not in NO_{k-1}, then the candidate cannot be a non-overlap pattern either and so can be removed from C_k. This pruning step is used to reduce the search space.

The above process is repeated until no new coverage pattern is found or no new candidate pattern can be generated.

The proposed algorithm uses bitwise operations to find the complete set of coverage patterns. So, a single scan of the database (to find the bit strings for all single web page sets) is sufficient for the algorithm to find the complete set of coverage patterns. Generation of bit strings for larger web page sets and computation of CS, OR for a web page set can be carried out by using simple bitwise AND and OR operations which makes the algorithm computationally very fast.

We now explain the working of *CMine* algorithm using the transactional database, T, shown in Table 1. There are 10 transactions in this database, that is, $|T| = 10$. Let the user-specified $minRF$, $minCS$ and $maxOR$ be 0.4, 0.7 and 0.5, respectively. The column titled *Bitstring* represents binary representation of coverage set of pattern i. For example, the bit string corresponding to the pattern "{b,a}" is "1111111111". This implies that every transaction of T contains either b or a or both. For binary representation of TID's, union of coverage sets of two patterns is equal to boolean OR operation of corresponding bit strings. Similarly, intersection of coverage sets of two patterns is equal to boolean AND operation of corresponding bit strings. We use Figure 1 to illustrate the CMine algorithm for finding coverage patterns in T.

i. The algorithm *CMine* scans all the transactions to generate bit string B^{w_i} and relative frequencies (RF) of each web page $w_i \epsilon T$. $RF(w_i) = \frac{|B^{w_i}|}{|T|}$. $|B^{w_i}|$ denotes



C_1

I	Bitstring	RF
a	1111000001	0.5
b	1000111111	0.7
c	1111000000	0.4
d	0001111000	0.4
e	0110000110	0.4
f	0000100000	0.1

L_1

I	RF
b	0.7

NO_1

I	RF
b	0.7
a	0.5
c	0.4
d	0.4
e	0.4

C_2

I	Bitstring	CS	OR
{b, a}	1111111111	1	0.4
{b, c}	1111111111	1	0.25
{b, d}	1001111111	0.8	0.75
{b, e}	1110111111	0.9	0.5
{a, c}	1111000001	0.5	1
{a, d}	1111111001	0.8	0.25
{a, e}	1111000111	0.7	0.5
{c, d}	1111111000	0.7	0.25
{c, e}	1111000110	0.6	0.5
{d, e}	0111111110	0.8	0

L_2

I	CS	OR
{b, a}	1	0.4
{b, c}	1	0.25
{b, e}	0.9	0.5
{a, d}	0.8	0.25
{a, e}	0.7	0.5
{c, d}	0.7	0.25
{d, e}	0.8	0

NO_2

I	CS	OR
{b, a}	1	0.4
{b, c}	1	0.25
{b, e}	0.9	0.5
{a, d}	0.8	0.25
{a, e}	0.7	0.5
{c, d}	0.7	0.25
{c, e}	0.6	0.5
{d, e}	0.8	0

C_3

I	Bitstring	CS	OR
{b, a, c}	1111111111	1	1
{b, a, e}	1111111111	1	1
{b, c, e}	1111111111	1	1
{a, d, e}	1111111111	1	0.5
{c, d, e}	1111111110	0.9	0.5

L_3/NO_3

I	CS	OR
{a, d, e}	1	0.5
{c, d, e}	0.9	0.5

Fig. 1. Working of CMine algorithm. The term 'I' is an acronym for item set.

the number of 1's in the bit string. Each web page, $w_i \in T$ is a member of the set of candidate 1-pattern, C_1.

ii. From C_1, the set of coverage 1-patterns, L_1, are discovered if their frequencies are greater than or equal to $minCS$. Simultaneously, set of non overlap 1-patterns, NO_1, are discovered if candidate 1-patterns have relative support greater than or equal to $minRF$ and finally the web pages in NO_1 are sorted in the decreasing order of their frequencies.

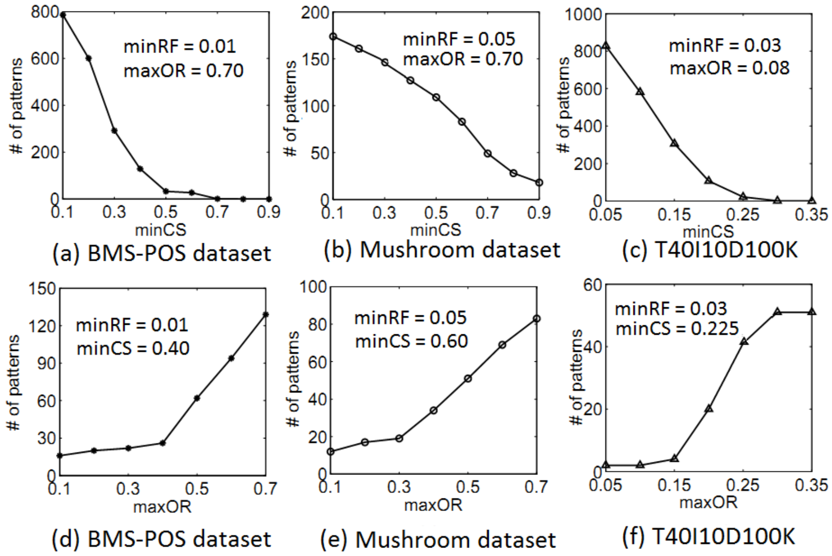
iii. To discover the set of coverage 2-patterns, L_2, the algorithm computes the join of $NO_1 \bowtie NO_1$ to generate a candidate set of 2-patterns, C_2.

iv. Using Equation 1, *coverage support* of each candidate pattern is computed by boolean OR operation. For example, CS(b,a) = $\frac{|B^b \vee B^a|}{|T|}$ = $\frac{|1111111111|}{10}$ =$\frac{10}{10}$ =1.0. Next, *overlap ratio* for each candidate pattern is computed by boolean AND operation. For example, OR(b,a) = $\frac{|B^b \wedge B^a|}{|B^a|}$ = $\frac{|1000000001|}{5}$ =$\frac{2}{5}$ =0.4. The columns titled 'CS' and 'OR' respectively show the *coverage support* and *overlap ratio* for the patterns in C_2.

v. The set of candidate 2-patterns that satisfy $maxOR$ are discovered as non-overlap 2-patterns, denoted as NO_2. Simultaneously, the set of candidate 2-patterns that satisfy both $minCS$ and $maxOR$ are discovered as coverage 2-patterns.

vi. Next, C_3 is generated by $NO_2 \bowtie NO_2$. That is, $C_3 = NO_2 \bowtie NO_2 = \{\{b,a,c\}$, $\{b,a,e\}, \{b,c,e\}, \{a,d,e\}, \{c,d,e\}\}$.

vii. As in step v, we discover non-overlap 3-patterns, NO_3, and coverage 3-patterns, L_3. The algorithm stops as no more candidate 4-patterns can be generated from non-overlap 3-patterns.



(a) BMS-POS dataset (b) Mushroom dataset (c) T40I10D100K

(d) BMS-POS dataset (e) Mushroom dataset (f) T40I10D100K

Fig. 2. Number of patterns generated by CMine algorithm at different $minCS$ and $maxOR$ values for BMS-POS, Mushroom and T40I10D100K datasets.

3 Experimental Results

For experimental purposes we have chosen four real world datasets and one synthetic dataset. The detailed description of the datasets are given below.

i. Kosarak dataset is a sparse dataset with 990,002 number of transactions containing 41,270 distinct items [14].
ii. MSNBC dataset contains data from Internet Information Server (IIS) logs for msnbc.com and news-related portions of msn.com for the entire day of September, 28, 1999 [15]. Requests are at the level of page category. The number of categories are 17 and the number of transactions are 989,818.
iii. Mushroom dataset is a dense dataset containing 8,124 transactions and 119 distinct items [14].
iv. BMS-POS dataset contains click stream data of a dotcom company [14]. The dataset contains 515,597 number of transactions and 1656 distinct items.
v. The synthetic dataset T40I10D100K is generated by the dataset generator [16]. The dataset contains 100,000 transactions and 941 distinct items.

The CMine algorithm was written in Java and run with Windows XP on a 2.66 GHz machine with 2GB memory.

3.1 Coverage pattern generation

The Figure 2(a) shows the number of patterns generated (y-axis) for BMS-POS dataset for different values of *minCS* (x-axis) while *minRF* and *maxOR* are fixed at the values 0.01 and 0.7 respectively. It can be observed from the Figure 2(a) that the number of coverage patterns decrease with the increase in *minCS*, and more importantly, the number of patterns generated are very few when *minCS* is greater than 0.5. In general, for a given *maxOR*, coverage support of coverage patterns increases with the length of the pattern due to the addition of new frequent items. The length of a coverage pattern increases with increasing levels of iteration for generation of candidate itemsets in CMine algorithm. However for higher levels of iteration due to overlap ratio constraint, the number of non-overlap patterns generated is decreased. Therefore, the number of coverage patterns generated having higher coverage support decreases with increasing *minCS*. The relation between the number of coverage patterns generated and *minCS* which was apparent in Figure 2(a) is also observed for Figure 2(b) and 2(c). It can also be observed form Figure 2(c) that no coverage patterns are generated for minCS = 0.35. This implies, that maximum threshold of *minCS* for *minRF* = 0.03, *maxOR* = 0.8 for T40I10D100k dataset is 0.35 since no coverage patterns are generated for minCS greater than 0.35.

The Figure 2(d) shows number of patterns generated (y-axis) for BMS-POS dataset for different values of *maxOR* (x-axis) while *minRF* and *minCS* are fixed at the values 0.01 and 0.40 respectively. It can be observed from Figure 2(d) that the gradient of the curve increases linearly from *maxOR* = 0.1 to 0.4. For maxOR = 0.4, the gradient of the curve changes and again increases linearly from *maxOR* = 0.4 to 0.7. However, the gradient of the curve from *maxOR* = 0.4 to 0.7 is greater than the curve for *maxOR* = 0.1 to 0.4. This implies that the number of patterns generated increases with increasing *maxOR* value. As the *maxOR* value is increased, the number of items of the candidate sets C_i (i =2,3,4...k-1) are increased which will result in increase of number of coverage patterns generated. Similar to the Figure 2(d), the phenomenon of increase in generation of coverage patterns with respect to *maxOR* value is also observed in Figure 2(e) and 2(f). It can be observed form Figure 2(f) that the number of patterns generated become constant for *maxOR* $\geq$ 0.30. This implies that no new nonoverlap patterns are generated for higher levels of iteration for generation of candidate item sets in CMine algorithm such that coverage patterns extracted from non-overlap patterns have coverage support greater than 0.225.

3.2 Scalability Experiment

We used *Kosarak* dataset to conduct scalability experiment. We divided the dataset into five portions of 0.2 million transactions in each part. We investigated the performance of CMine Algorithm after cumulatively adding each portion with previous parts and extracting coverage patterns each time. The values of *minRF*, *minCS* and *maxOR* are fixed at 0.01, 0.1 and 0.5 respectively. The experimental results are shown in Figure 3. It is clear from the Figure 3 that as the database size increases, the execution time also increased linearly.

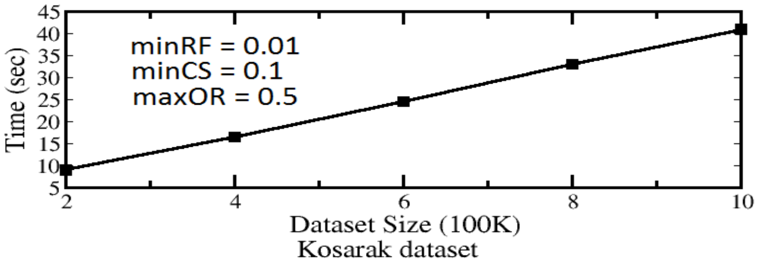


Fig. 3. Scalability of CMine algorithm

3.3 Usefulness of coverage patterns

Table 2. Sample coverage 3-patterns extracted from MSNBC dataset [15]

S.No	Coverage Pattern	CS	S.No	Coverage Pattern	CS
1	$\{local, misc, frontpage\}$	0.42	4	$\{on\text{-}air, news, misc\}$	0.40
2	$\{news, health, frontpage\}$	0.43	5	$\{tech, weather, on\text{-}air\}$	0.41
3	$\{tech, opinion, frontpage\}$	0.41	6	$\{sports, misc, opinion\}$	0.43

Table 2 shows some coverage patterns generated by *Cmine* algorithm for $minCS = 0.4$ and $maxOR = 0.5$ and $minRF = 0.02$ for MSNBC dataset. The names of web page categories involved in MSNBC are “frontpage”, “news”, “tech”, “local”, “opinion”, “on-air”, “misc”, “weather”, “health”, “living”, “business”, “sports”, “summary”, “bbs” (bulletin board service), “travel”, “msn-news”, and “msn-sports”. From Table 2, it can be observed that any of the six coverage patterns ensure about 40 percent coverage. The result indicates how the proposed approach provides flexibility to the publisher to meet the demands of multiple advertisers by considering different sets of web pages.

4 Conclusions and Future work

In this paper we have proposed a new data mining pattern called “coverage pattern” and proposed an efficient methodology to extract the same from transactional databases. We have explained how coverage patterns could be useful by considering the issue of banner advertisement placement. By conducting experiments on different kinds of datasets, we have shown that the proposed model and methodology can effectively discover coverage patterns.

As a part of the future work, we are going to investigate how both frequent and coverage pattern knowledge can be used for efficient banner advertisement placement. In addition we are planning to investigate how the content of the web page and search query can be exploited to explore content specific coverage patterns. We are also exploring how the notion of coverage patterns can be extended to other domains like bio-informatics for extracting potential knowledge patterns.

References

1. R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in *VLDB ’94: Proceedings of the 20th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.
2. V. Chvatal, “A greedy heuristic for the set-covering problem,” *Mathematics of operations research*, pp. 233–235, 1979.
3. M. R. Garey, D. S. Johnson, and L. Stockmeyer, “Some simplified np-complete problems,” in *Proceedings of the sixth annual ACM symposium on Theory of computing*, ser. STOC ’74. ACM, 1974, pp. 47–63.
4. P. Venetis, G. Koutrika, and H. Garcia-Molina, “On the selection of tags for tag clouds,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, ser. WSDM ’11. ACM, 2011, pp. 835–844.
5. F. Bonchi, C. Castillo, D. Donato, and A. Gionis, “Topical query decomposition,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD ’08. ACM, 2008, pp. 52–60.
6. A. Amiri and S. Menon, “Efficient scheduling of internet banner advertisements,” *ACM Trans. Internet Technol.*, vol. 3, no. 4, pp. 334–346, 2003.
7. A. Ghosh, B. I. Rubinstein, S. Vassilvitskii, and M. Zinkevich, “Adaptive bidding for display advertising,” in *WWW ’09: Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 251–260.
8. X. Wu and A. Bolivar, “Keyword extraction for contextual advertisement,” in *WWW ’08: Proceeding of the 17th international conference on World Wide Web*. ACM, 2008, pp. 1195–1196.
9. D. Chakrabarti, D. Agarwal, and V. Josifovski, “Contextual advertising by combining relevance with click feedback,” in *WWW ’08: Proceeding of the 17th international conference on World Wide Web*. ACM, 2008, pp. 417–426.
10. S. Alaei, E. Arcaute, S. Khuller, W. Ma, A. Malekian, and J. Tomlin, “Online allocation of display advertisements subject to advanced sales contracts,” in *ADKDD ’09: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*. ACM, 2009, pp. 69–77.
11. B. Sripada, P. K. Reddy, and R. U. Kiran, “Coverage patterns for efficient banner advertisement placement,” in *WWW (Companion Volume)*, 2011, pp. 131–132.
12. B. Liu, W. Hsu, and Y. Ma, “Mining association rules with multiple minimum supports,” in *KDD ’99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 337–341.
13. J. Han and M. Kamber, *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.
14. Fimi, “Frequent itemset mining implementations repository,” http://fimi.cs.helsinki.fi/ July 2010.
15. A. Frank and A. Asuncion, “UCI machine learning repository,” 2010. [Online]. Available: http://archive.ics.uci.edu/ml
16. R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’93. ACM, 1993, pp. 207–216.