Mining Top-k Frequent/Regular Patterns Based on User-Given Trade-Off between Frequency and Regularity

Komate Amphawan[1,⋆] and Philippe Lenca[2]

[1] CIL, Faculty of Informatics, Burapha University, Chonburi 20131, Thailand
komate@gmail.com
[2] Institut Telecom, Telecom Bretagne, UMR CNRS 6285 Lab-STICC, France
philippe.lenca@telecom-bretagne.eu

Abstract. Frequent-Regular pattern mining has been introduced to extract interesting patterns based on their occurrence behavior. This approach considers the terms of frequency and regularity to determine significant of patterns under user-given support and regularity thresholds. However, it is well-known that setting of thresholds to discover the most interesting results is a very difficult task and it is more reasonable to avoid specifying the suitable thresholds by letting users assign only simple parameters. In this paper, we introduce an alternative approach, called *Top-k frequent/regular pattern mining based on weights of interests*, which allows users to assign two simple parameters: (i) a weight of interest on frequency/regularity and (ii) a number of desired patterns. To mine patterns, we propose an efficient single-pass algorithm, *TFRP-Mine*, to quickly mine patterns with frequent/regular appearance. Experimental results show that our approach can effectively and efficiently discover the valuable patterns that meet the users' interest.

Keywords: Association rule, Frequent patterns, Top-k frequent patterns, Frequent-regular itemset, Top-k frequent regular patterns.

1 Introduction

The frequent pattern mining problem, introduced in [1], plays a major role in many data mining tasks that aim to find interesting patterns from databases. Many efficient algorithms have been proposed for a large category of patterns. Excellent surveys can be found in [2, 3]. Roughly speaking, a pattern X is called frequent if its support is no less than a given minimal absolute support threshold where the support of X is the number of its occurrences in the database (support is also often considered from relative frequency point of view with a minimal frequency threshold). Obviously, frequent pattern mining gives an important role to occurrence frequency.

⋆ Corresponding author.

B. Papasratorn et al. (Eds.): IAIT 2013, CCIS 409, pp. 1–12, 2013.

However, the occurrence behavior of patterns (*i.e.* whether a pattern occurs regularly, irregularly, or mostly in a specific time interval) may also be an important criteria in various applications (*e.g.* retail marketing [4], stock marketing [5], elderly daily habits' monitoring [6], etc.). The frequent-regular pattern mining problem thus also consider the maximum interval at which a pattern occurs (or disappears) in the database. To mine frequent-regular patterns, the users have to assign two parameters: (*i*) support and (*ii*) regularity thresholds. However, it is well-known that it is not an obvious task. If the support threshold is too large, then there may be only a small number of results or even no result. In that case, the user may have to fix a smaller threshold and do the mining again. If the threshold is too small, then there may be too many results for the users. Thus, asking the number of desired outputs is considered easier and mining top-k patterns has become a very popular task (see for example [7–13]).

The problem of top-k frequent-regular patterns mining was introduced in [14] and enhanced in [15, 16]. It aims at mining the k patterns with highest support that occur regularly (the users has to specify a regularity threshold and the number k of desired results). Moreover, frequent-regular patterns mining was extended in several manners (e.g. on incremental transactional databases [17], on data stream [18], for both frequent and rare items [19], with maximum items' support constraints [20]).

However, all approaches mentioned above need to specify an appropriate value of regularity threshold to discover results. In the same way as the support threshold, if the regularity threshold is set too small, there may be only few regular patterns or even no patterns. On the other hand, there may be too many patterns with a large regularity threshold. We thus propose a trade-off between support and regularity. Indeed, the user may also be interested by a mix between frequency and regularity. The trade-off can simply be tuned with a weight parameter. According to his preferences the user may then choose to focus more or less on frequency and/or regularity.

The rest of this paper is organized as follows. Sections 2 introduce the top-k frequent-regular patterns mining problem. Section 3 extend the problem to top-k frequent-regular pattern under user-given weights of interest. The proposed *TFRP-Mine* algorithm is described in details in Section 4. Section 5 reports the experimental study. Finally, we conclude the paper in Section 6.

2 Notations on Top-k Frequent-Regular Patterns Mining

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items. A set $X \subseteq I$ is called a pattern (itemset), or a k-pattern when it contains k items. A transaction database $TDB = \{t_1, t_2, \ldots, t_m\}$ is a set of m transactions which each transaction $t_q = (tid, Y)$ is a tuple containing: (*i*) a unique transaction identifier, $tid = q$, and (*ii*) Y is a set of items. If $X \subseteq Y$, then X occurs in t_q, denoted as t_q^X. We define $T^X = \{t_p^X, \ldots, t_q^X\}$ $(1 \leq p \leq q \leq m)$, as the set, called *tidset*, of all *ordered tids* in which X appears. The support (frequency) of a pattern X, denoted as $s^X = |T^X|$, is also defined as the number of *tids* (transactions) that contain X.

To measure the regularity of occurrence of X (as described in [5]), let t_p^X and t_q^X be any two consecutive *tids* in T^X. The regularity between the two consecutive *tids*, *i.e.* the number of *tids* not containing X between t_p^X and t_q^X, can be expressed as $rtt_q^X = t_q^X - t_p^X$. In addition, for the first and the last occurrences of X, their regularities are also regarded as (i) the first regularity is the number of *tids* not containing X before its first appearance, $fr^X = t_1^X$, and (ii) the last regularity is the number of *tids* not containing X from the last occurrence of X to the end of database, $lr^X = |TDB| - t_{|T^X|}^X$. Based on the regularity values mentioned above, we can define the total regularity of X as follows:

Definition 1. *The regularity of X is the maximum number of tids (transactions) that X disappears from database which can be defined as,*

$$r^X = max(fr^X, rtt_2^X, rtt_3^X, \ldots, rtt_{|T^X|}^X, lr^X)$$

From the regularity of X, we can guarantee that X will appear at least once in every set of r^X consecutive transactions. The problem of frequent-regular pattern mining is thus to discover patterns that frequently and regularly appear in a database. However, frequent-regular patterns mining still suffer from setting of appropriate thresholds [14]. Hence, the *Top-k frequent-regular pattern mining* [14] was proposed to avoid the difficulties of setting an appropriate support threshold by allowing users to control the number of patterns to be mined. It can be defined as follows:

Definition 2. *A pattern X is called a top-k frequent-regular pattern if (i) its regularity is no greater than a user-given regularity threshold σ_r and (ii) there exist no more than $k-1$ patterns whose support is higher than that of X.*

The users have thus to specify only the number k of desired patterns and a regularity threshold σ_r to obtain the k regular patterns with highest support. However, it can be still difficult to specify a proper regularity threshold.

3 Defining of the User-Given Weights of Interest on Support/Regularity

We here introduce an alternative approach, called *Top-k frequent/regular pattern mining based on user-given weight of interest*, to discover patterns by alleviating difficulties of setting the appropriate thresholds. This approach requires users to specify two simple parameters: (i) the number of desired patterns (k) and (ii) a weight indicated users' interest on frequency/support(λ) or regularity(ϕ) (the users can identify only one or both weights in range $[0, 1]$; the summation of the two weights is equal to 1). This lets the users to express their interest on the support and/or regularity measures. We thus define the following notations/definitions for our approach.

Definition 3. *Let $\lambda \in [0,1]$ be the weight of support given by a user; the weight of regularity is then $\phi = 1 - \lambda$. On the other hands, if the user gives weight of regularity, $\phi \in [0,1]$, the weight of support is $\lambda = 1 - \phi$.*

With the user-given weights of interest on support/regularity values, we then define the *support-regularity value* of the patterns to measure their interestingness based on weights of interest under the support and regularity constraints.

Definition 4. *Let the weights of interest be λ and ϕ. The value of support-regularity of a pattern X is defined as $sr^X = (\lambda)(s^X) - (\phi)(r^X)$.*

The *support-regularity* of any pattern is in $[-m, m]$ where m is the number of transactions. Patterns with high support and low regularity will have high *support-regularity*. Otherwise, the value of *support-regularity* depends on user-given weights. If user gives high weight of support λ (weight of regularity ϕ will be low), the patterns with high support and high regularity will have the *support-regularity* greater than that of patterns with low support and low regularity. Lastly, patterns with low support and high regularity will absolutely have low *support-regularity value*. We can thus define the the top-k frequent-regular pattern mining problem under user-given weights of interest as follow.

Definition 5. *A pattern X is called a top-k frequent/regular pattern based on under-given trade-off between frequency and regularity if there exist no more than $k-1$ patterns whose support-regularity values are greater than that of X.*

The top-k frequent/regular patterns mining problem is to discover the k patterns whith highest support-regularity values from transactional databases under two user-given parameters: (i) weight of interest on support λ or regularity ϕ, and (ii) the number of expected output k.

4 Mining Top-k Frequent/Regular Patterns Mining under User-Given Weights of Interest

We here introduce an efficient algorithm, *TFRP-Mine*, for mining top-k frequent/regular patterns based on user-given weights of interest. It consists of two steps: (i) *TFRP-Initialization (TFRP-I)* captures the database content with only one database scan (patterns (items or itemsets), their support, regularity, support-regularity) into a top-k list, and (ii) *TFRP-Mining (TFRP-M)* that quickly discovers the desired patterns from the top-k list.

TFRP-Initialization (TFRP-I). *TFRP-I* builds the top-k list which is simply a linked-list associated with a hash table (for efficient updating). Each entry of the top-k list contains five informations: pattern (item or itemset) I, support s^I, regularity r^I, support-regularity value sr^I and a set of transaction-ids (tidset) T^I where pattern I occurs in database.

To construct the top-k list, *TFRP-I* first creates the hash table for all single items. Subsequently, each transaction of the database is sequentially scanned and

then each item in the transaction is regarded. For each item, the top-k list is accessed once via the hash table to seek the entry of the item. If the entry does not exist, a new entry for the item is created with its support, regularity, and tidset. Otherwise, its information is updated. The scanning step is repeated until the last transaction of the database. After all scanning, the regularity and support-regularity value of each item in the top-k list are calculated. Finally, all entries in the top-k list are sorted by descending order of support-regularity values and the entries after the k^{th} entry are eliminated (they cannot belong to the result).

Following definition 4 we can notice that the support-regularity value has the downward-closure property (*i.e.* any superset of a low support-regularity pattern is also a low support-regularity pattern). Thus, the support-regularity measure can be used to prune the supersets of low support-regularity value during the mining process.

TFRP-Mining (TFRP-M, see Algorithm 1). *TFRP-M* applies a best-first search strategy to pairs of patterns with highest support-regularity values first to the lowest ones, since two patterns with high support-regularity values tend to frequently and/or regularity appear together in database. However, even if the the two consider patterns do not frequently and/or regularity appear together, the performance of applying the best-first search is still acceptable. It will consider at most $O(k^2)$ pairs of patterns to mine the complete set of top-k frequent regular patterns. To mine patterns, two patterns in the top-k list are joined to generate a new pattern if they meet the following constraints: (i) they have the same number of items, and (ii) they have the same prefix (*i.e.* each item from both patterns is the same, except only the last item). These two constraints can help to reduce the redundancy of regarding pairs of patterns. Whenever both patterns satisfy the above constraints, their sets of transaction-ids(tidsets) are sequentially intersected and collected in order to calculate support, regularity, and support-regularity value of the new generated pattern. If the support-regularity value of the new pattern is greater than that of the k^{th} pattern then it is inserted at its place into the top-k list and the k^{th} element is removed from the top-k list. Joining and intersection steps are repeated for all pairs of patterns in the top-k list.

Table 1. A transactional database

Transaction-id	items	Transaction-id	items
1	a b c d	6	a e
2	a c d	7	a b c
3	a b d	8	b c d e
4	b c d e	9	a b d e
5	a b c e	10	a e

Example of TFRP-Mine. Let consider the database with 10 transactions of Table 1. Suppose our task is to find the top-5 frequent/regular patterns under weight of interest on support $\lambda = 0.4(40\%)$ (to save space the hash table is not presented).

Algorithm 1. *TFRP-Mining (TFRP-M)*

Input:
- Top-k list
- The number of desired result (k)
- The weights of interest on support (λ) and regularity (ϕ)

Output:
- Top-k regular/frequent patterns

for each pattern X in the top-k list **do**
  for each pattern Y in the top-k list ($X \neq Y$) **do**
    if X and Y satisfy the two joining constraints (mentioned above) **then**
      - merge patterns X and Y to be the pattern $Z = \{i_1^X, \ldots, i_{|X|}^X, i_{|Y|}^Y\}$
      - initialize values of Z, $r^Z = 0, s^Z = 0$ and $T^Z = \{\}$

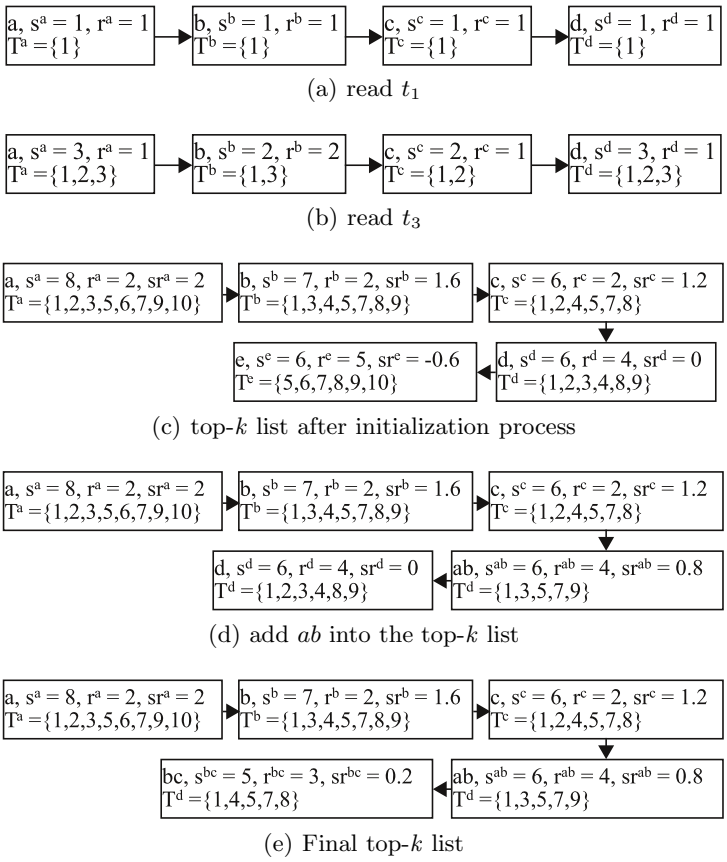
      for each t_p in T^X and t_q in T^Y (where $p \in [1, |T^X|]$ and $q \in [1, |T^Y|]$) **do**
        if $t_p = t_q$ **then**
          - calculate rtt_j^Z by t_p (where j is the number of tids in T^Z)
          - add the support s^Z by 1
          - collect t_p as the last tid of T^Z

      - calculate regularity of Z, $r^Z = max(fr^Z, rtt_2^Z, rtt_3^Z, \ldots, rtt_{|T^Z|}^Z, lr^Z)$
      - calculate support-regularity value of Z, $sr^Z = (\lambda)(s^Z) - (\phi)(r^Z)$

      if $sr^Z \geq sr^K$ (where K is the current k^{th} pattern in the top-k list) **then**
        - remove k^{th} entry and then insert the pattern Z into top-k list (with its values: r^Z, s^Z, sr^Z and T^Z)

TFRP-I first creates the hash table and calculates the weight of interest on regularity, $\phi = 0.6$. As shown in Fig. 1(a), the entries for items a, b, c and d are created by reading the first transaction t_1. Their supports, regularities and tidsets are initialized to be 1, 1 and $\{1\}$, respectively. Next, transaction $t_2 = \{a, c, d\}$ is read and the support of items a, c and d are increased to 2 whereas their tidsets are updated to $\{1, 2\}$. With the scanning of transaction $t_3 = \{a, b, d\}$, the supports and the tidsets of items a, b and d are adjusted as shown in Fig. 1(b), and so on for transactions t_4 to t_{10}. After scanning the entire database, the regularity and the support-regularity value of each item in the top-k list are calculated using Definition 1 and Definition 4. Finally, the top-k list is sorted by descending order of the support-regularity value. As illustrated in Fig. 1(c), the final top-k list, is composed of five entries (each entry consists of *item, support, regularity, support-regularity value, and tidset.*

To mine patterns with the best-first search strategy, item b is first joined with item a since they have highest support-regularity values. Consequently, their tidsets, T^a and T^b, are sequentially intersected to compute $s^{ab} = 5$, $r^{ab} = 2$ and $sr^{ab} = (0.4)(5) - (0.6)(2) = 0.8$ and $T^{ab} = \{1, 3, 5, 7, 9\}$ (transactions id where ab occurs). By comparing sr^{ab} with that of the $k^{th}(5^{th})$ pattern, $sr^e = -0.6$, *TFRP-M* removes pattern e and then inserts pattern ab into the top-k list



(a) read t_1

(b) read t_3

(c) top-k list after initialization process

(d) add ab into the top-k list

(e) Final top-k list

Fig. 1. Top-$k$list during mining

(see Fig. 1(d)). Next, the pattern c is merged with a and b. After joining and intersecting steps, we get the $s^{ac} = 4, r^{ac} = 3, sr^{ac} = -0.2$ and $T^{ac} = \{1, 2, 5, 7\}$ for the pattern ac. Since the support-regularity value sr^{ac} is less than that of d (the current k^{th} pattern), pattern ac is not added into the top-k list and then ac is eliminated from the mining process. For the pattern bc, its support, regularity, support-regularity value and tidset are $s^{bc} = 5, r^{bc} = 3, sr^{bc} = 0.2$, and $T^{bc} = \{1, 4, 5, 7, 8\}$. With the investigation of sr^{bc} with sr^d, the pattern d is removed and the pattern bc is inserted into the top-k list. *TFRP-M* proceeds the remaining patterns in the top-k list with the same manner and we finally obtain the k patterns with highest support-regularity values as illustrated in Fig. 1(e).

5 Performance Evaluation

We here report some experimental studies done to investigate the performance of the proposed *TFRP-Mine* algorithm. From the best of our knowledge, there is

no approach which aims to avoid difficulties of setting regularity threshold. Thus, no comparative study can be provided. However, the *TFRP-Mine*'s performance can be used later as an initial baseline.

Four datasets are used: the synthetic *T10I4D100K* (a sparse dataset generated by the IBM synthetic market-basket data generator containing 100,000 transactions of size 10 in average, and with potential maximal large itemsets of size 4) and three real datasets *Chess, Connect* and *Retail* (two dense and one sparse datasets retrieved at `http://fimi.ua.ac.be/data/` with 3,196, 67,557, 88,122 transactions and 75, 129, 16,469 distinct items, respectively).

Two kind of experiments are done with 7 values for k (50, 100, 200, 500, 1,000, 2,000, 5,000 and 10,000) and 5 values for the weight of support (0, 0.25, 0.5, 0.75 and 1) to measure: (*i*) effects of the user-given weight of interest, and (*ii*) computational time and memory usage of *TFRP-Mine* algorithm.

Effects of the User-Given Weights
As shown in Fig. 2, the average support computed on the set of results decreases as the value of k increases: obviously, when the value of k increases, we can gain more patterns in which support of such patterns in the top-k set are likely to decrease. In addition, higher weight for support value (the weight of regularity is decreased) leads to patterns with higher support as well (in Fig. 2 support values have to be multiply by 10^3;). From Def. 4, the higher weight of support will cause the patterns with high support having also high support-regularity values. These patterns will have the highest support-regularity values and thus will be in the results set. Then, we can said that the average support is increased as the weight on support increases. Notice that when the weight of support is equal to 1 then we obtain classical top-k patterns as in [14].

From Fig. 3, the average regularity of patterns increases with increasing value of k. Obviously, with the higher value of k, the user can gain more patterns with less frequency and/or high regularity values. Moreover, the average regularity also increases with the increasing of weight of support. In this case, there may be patterns with frequently appearance that have a large gap of disappear included in the set of results (*i.e.* the new included patterns may appear frequently at the beginning and then disappear for a long time (this will cause patterns have high regularities). Another case is the case that new patterns disappear for a long time at the beginning and then frequently occur in database. From these situations, we can claim that with the higher value of k and higher value of weight on support, we will gain more patterns with more irregularly occurrence.

Computational Time and Memory Usage
Fig. 4 shows the runtime on the four datasets. Obviously, runtime increases with k. Runtime also increases with the weight of support. This is due to the fact that *TFRP-Mine* then needs more time to collecting and intersecting larger tidset for each pattern. However, we can see from the figure that even with very low support weight (*i.e.* the results may have low support and/or low regularity), *TFRP-Mine* is efficient.

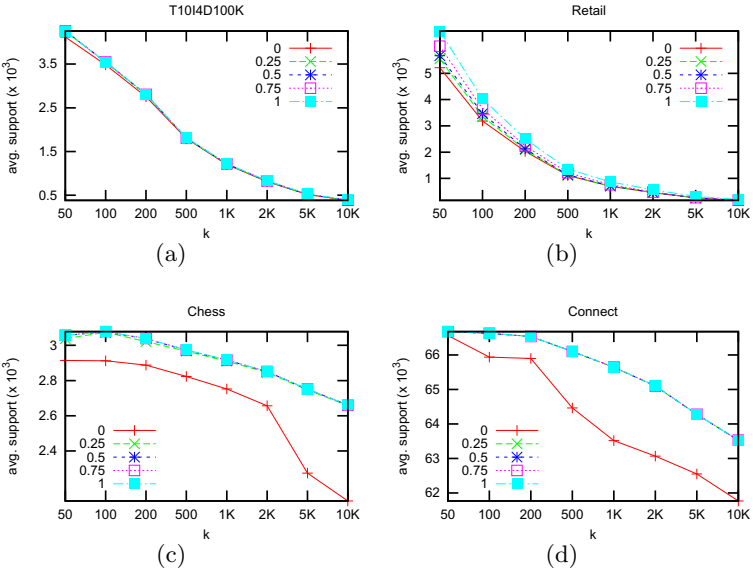


Fig. 2. Average support of results from TFRP-Mine

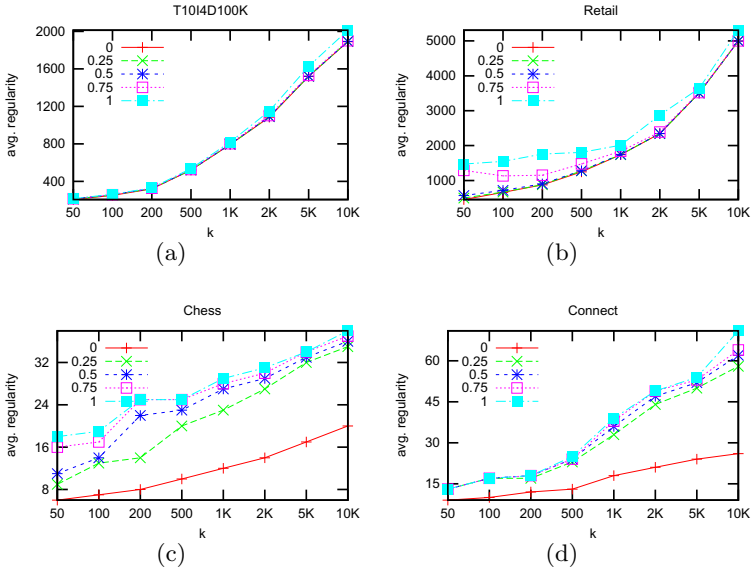


Fig. 3. Average regularity of results from TFRP-Mine

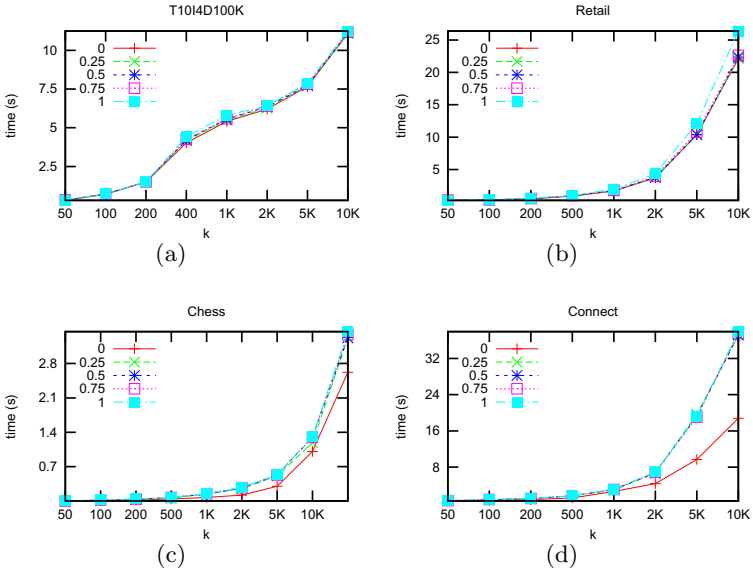


Fig. 4. Computational time of TFRP-Mine

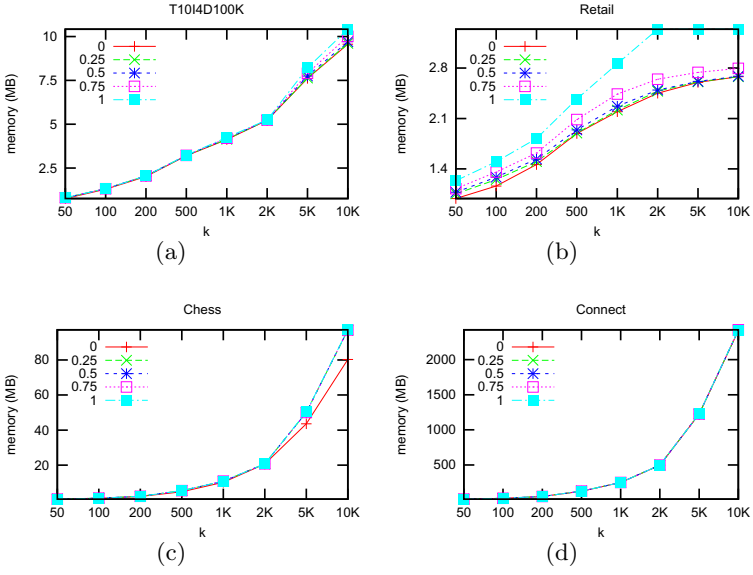


Fig. 5. Memory consumption of TFRP-Mine

The investigation of memory consumption is presented Fig. 5. It increases with k (obviously). It increases also with the weight of support. This is due to the fact that *TFRP-Mine* then needs to collect more tids in main memory to compute the support, regularity and support-regularity values of patterns.

As a whole when compared to classical top-k patterns mining as described in [14] *i.e.* when weight of support is equal to 1 in each figures one can notice that *TFRP-Mine* is very competitive: performance are slightly degraded and *TFRP-Mine* offers more flexibility between frequency and regularity.

6 Conclusion

Regular patterns and top-k regular patterns have recently attracted attention. Mining frequent and regular patterns require a support and a regularity thresholds. However, it is well-known that setting these thresholds is not easy. We thus propose to mine top-k regular/frequent patterns with one weight to balance between frequency and regularity depending on the user interest and the number k of desired patterns. This allows also to balance between the two criteria in comparison with the classical approach that requires two strict thresholds. An efficient single-pass algorithm, *TFRP-Mine*, is proposed. Experiments show that it can discover a wide range of patterns with low, middle and high value of supports and regularities.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, pp. 207–216 (1993)
2. Goethals, B.: Frequent set mining. In: The Data Mining and Knowledge Discovery Handbook, pp. 377–397. Springer (2005)
3. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Discov. 15(1), 55–86 (2007)
4. Chang, J.: Mining weighted sequential patterns in a sequence database with a time-interval weight. Knowledge Based Systems 24(1), 1–9 (2011)
5. Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K.: Discovering periodic-frequent patterns in transactional databases. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 242–253. Springer, Heidelberg (2009)
6. Heierman, E.O., Youngblood, G.M., Cook, D.J.: Mining temporal sequences to discover interesting patterns. In: KDD Workshop on Mining Temporal and Sequential Data (2004)
7. Fu, A.W.-C., Kwong, R.W.-W., Tang, J.: Mining N-most Interesting Itemsets. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 59–67. Springer, Heidelberg (2000)
8. Wang, J., Han, J., Lu, Y., Tzvetkov, P.: Tfp: an efficient algorithm for mining top-k frequent closed itemsets. Proceeding of the IEEE Transactions on Knowledge and Data Engineering 17, 652–664 (2005)

9. Yang, B., Huang, H., Wu, Z.: Topsis: Finding top-k significant n-itemsets in sliding windows adaptively. Knowl.-Based Syst. 21(6), 443–449 (2008)
10. Li, H.F.: Mining top-k maximal reference sequences from streaming web click-sequences with a damped sliding window. Expert Systems with Applications 36(8), 11304–11311 (2009)
11. Ke, Y., Cheng, J., Yu, J.X.: Top-k correlative graph mining. SDM, 1038–1049 (2009)
12. Webb, G.I.: Filtered-top-k association discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(3), 183–192 (2011)
13. Fournier-Viger, P., Tseng, V.S.: Tns: mining top-k non-redundant sequential rules. In: Shin, S.Y., Maldonado, J.C. (eds.) SAC, pp. 164–166. ACM (2013)
14. Amphawan, K., Lenca, P., Surarerks, A.: Mining top-k periodic-frequent patterns without support threshold. In: IAIT 2009. CCIS, vol. 55, pp. 18–29. Springer, Heidelberg (2009)
15. Amphawan, K., Lenca, P., Surarerks, A.: Efficient mining top-k regular-frequent itemset using compressed tidsets. In: Cao, L., Huang, J.Z., Bailey, J., Koh, Y.S., Luo, J. (eds.) PAKDD Workshops 2011. LNCS, vol. 7104, pp. 124–135. Springer, Heidelberg (2012)
16. Amphawan, K., Lenca, P., Surarerks, A.: Mining top-k regular-frequent itemsets using database partitioning and support estimation. Expert Systems with Applications 39(2), 1924–1936 (2012)
17. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S.: Mining regular patterns in incremental transactional databases. In: Int. Asia-Pacific Web Conference, pp. 375–377. IEEE Computer Society (2010)
18. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S.: Mining regular patterns in data streams. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 399–413. Springer, Heidelberg (2010)
19. Surana, A., Kiran, R.U., Reddy, P.K.: An efficient approach to mine periodic-frequent patterns in transactional databases. In: Cao, L., Huang, J.Z., Bailey, J., Koh, Y.S., Luo, J. (eds.) PAKDD Workshops 2011. LNCS, vol. 7104, pp. 254–266. Springer, Heidelberg (2012)
20. Kiran, R.U., Reddy, P.K.: Mining periodic-frequent patterns with maximum items' support constraints. In: Proceedings of the Third Annual ACM Bangalore Conference, COMPUTE 2010, pp. 1–8 (2010)