COSI-*tree*: Identification of Cosine Interesting Patterns Based on FP-tree

Xiaojing Huang[1] Junjie Wu[2] Shiwei Zhu[3] Hui Xiong[4]

[1,2,3]School of Economics and Management, Beihang University, China
[4]Rutgers Business School, Rutgers University, USA

Abstract

The cosine similarity, also known as uncentered Pearson Correlation, has been widely used for mining association patterns, which contain objects strongly related to each other. However, it is often used as a post-evaluation measure and is computationally prohibitive for large data. To this end, we develop an FP-tree like algorithm, named COSI-*tree*, for finding association patterns based on the cosine measure. A key idea is to combine the strength of the FP-tree structure and the Conditional Anti-Monotone Property of the cosine measure. Experimental results on real-world data demonstrate the effectiveness of COSI-*tree*, in particular for finding rare but interesting patterns at extremely low support levels.

Keywords: Interestingness Measure; Cosine Measure; Conditional Anti-Monotone Property; FP-tree

1. Introduction

Recent years have witnessed an increased interest in association analysis, which is concerned with the identification of strongly related objects. However, the traditional support-confidence framework may not disclose truly interesting relationships [1, 2, 3]. This is usually illustrated by the well-known "coffee-tea" example [4]. To meet this challenge, many interestingness measures have been developed for mining really interesting patterns. Among them, the cosine measure receives particular attention. As a widely used measure, the cosine similarity is suitable for high-dimensional item vectors, has the important null-invariant property [5], and often produces high quality results across different domains. In this paper, we have the focus on mining association patterns based on the cosine measure.

In the literature, many existing studies limit their scope to the *post-evaluation* of interesting patterns [5, 6]. This post-evaluation scheme can be computationally expensive and often misses some interesting but infrequent patterns. A feasible solution is to employ an *in-evaluation* scheme, which uses the interestingness measures to generate the interesting patterns directly. For the cosine measure, however, the lack of the important anti-monotone property makes it extremely hard to realize the in-evaluation scheme. This indeed motivates our work: Can we find an alternative to the anti-monotone property?

The main contributions of this paper are summarized as follows.

First, we define the novel concepts of the "Conditional Anti-Monotone Property" (CAMP) and the "Support-Ascending Set Enumeration Tree" (SA-SET). We prove that the cosine measure possesses the CAMP, and therefore can serve as an in-evaluation measure for interesting pattern discovery if the itemset traversal sequence is defined by the SA-SET. Second, we argue that the FP-tree based depth-first strategy is better than the Apriori-like breath-first strategy in taking use of CAMP for the cosine measure. Therefore, an FP-growth-like algorithm called COSI-*tree* is developed to mine cosine interesting patterns. Finally, COSI-*tree* is tested on various real-world data sets. Experimental results show that compared with the post-evaluation scheme, COSI-*tree* is much more efficient, and specializes in finding non-trivial interesting patterns even at extremely low levels of support.

2. Problem Definition

In this section, we briefly introduce the well-known cosine interestingness measure and its extension to the multi-itemset case. Then we define our problem.

2.1. Cosine Interestingness Measure

In the field of association analysis, cosine similarity is defined as an interestingness measure to a specific itemset. Let i_1, i_2 be two items in a transaction data set, we have

Definition 1 (Cosine Measure of 2-Itemsets) For a 2-itemset $X = \{i_2, i_1\}$, the cosine measure of X is defined as

$$\cos(X) = \frac{supp(\{i_2 i_1\})}{\sqrt{supp(\{i_2\})supp(\{i_1\})}}. \quad (1)$$

According to Definition 1, we can reasonably extend the 2-itemset case to the multi-itemset case as follows [7]:

Definition 2 (Cosine Measure of Multi-Itemsets) For a K-itemset $X = \{i_K, \cdots, i_2, i_1\}$, $K = 2, 3, \cdots$, the cosine measure of X is defined as

$$\cos(X) = \frac{supp(X)}{\sqrt[K]{\prod_{k=1}^{K} supp(\{i_k\})}}. \quad (2)$$

2.2. Problem Definition

We first give a formal definition for the "cosine interesting patterns" as follows:

Definition 3 (Cosine Interesting Patterns) Let $\mathcal{D}$ be a transaction database over a set of items $\mathcal{I}$, *min_supp* be the minimum support threshold, and *min_cos* be the minimum cosine threshold. The collection of the cosine interesting patterns in $\mathcal{D}$ w.r.t. *min_supp* and *min_cos* is defined by

$$\mathcal{F} = \{X \subseteq \mathcal{I} | supp(X) \geq min\_supp, cos(X) \geq min\_cos\}.$$

Based on Def. 3, we define our problem studied in this paper as follows:

Problem Definition. Given *min_supp* and *min_cos*, find all the cosine interesting patterns $\mathcal{F}$, with the restriction that interesting but rare patterns can be found even when $min\_supp = 0$.

One may argue that a simple solution to the above problem is to employ a *post-evaluation* scheme. That is, we first use the well-established tools to find the frequent patterns, which are then subject to *min_cos* to find the interesting patterns. This scheme, however, suffers from the dilemma of setting an appropriate *min_supp*: A higher threshold will result in missing the rare but interesting patterns (i.e.,

violates the restriction in Problem Definition), while a lower threshold may lead to the generation of too many spurious patterns exceeding the capacity of the mining system.

We therefore should turn to the *in-evaluation* scheme. That is, we should use the cosine measure together with support in the pattern mining process. It is well recognized that to incorporate an interestingness measure into the mining process, the measure should have the anti-monotone property as support. The cosine measure, however, does not hold this property. In what follows, we solve this problem by proposing a novel conditional anti-monotone property.

3. The Conditional Anti-Monotone Property

In this section, we first introduce the Conditional Anti-Monotone Property (CAMP). Then we show how to use CAMP to find cosine interesting patterns from an FP-tree.

3.1. Introducing CAMP

Definition 4 (CAMP) Let $\mathcal{I}$ be a set of items. A measure f is conditionally anti-monotone if $\forall X, Y \subseteq \mathcal{I}$, given that (1) $X \subseteq Y$, and (2) if $Y \setminus X \neq \emptyset, \forall i \in X$ and $i' \in Y \setminus X$, $supp(\{i\}) \leq supp(\{i'\})$, we have $f(X) \geq f(Y)$.

Theorem 1 *The cosine measure possesses CAMP.*

Proof. Without loss of generality, assume $X = \{i_K, \cdots, i_1\}$ is a K-itemset ($K \geq 2$), and the superset $Y = \{i_{K+L}, \cdots, i_{K+1}, i_K, \cdots, i_1\}$ is a $(K+L)$-itemset ($L \geq 1$). Suppose $\forall\ 1 \leq l \leq L$ and $1 \leq k \leq K$, $supp(\{i_{K+l}\}) \geq supp(\{i_k\})$. We need to show $\cos(X) \geq \cos(Y)$. We have

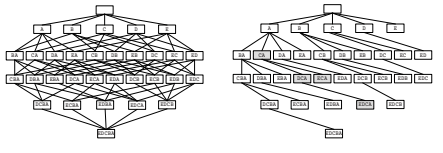
$$\cos(X) = \frac{supp(X)}{\sqrt[K]{\prod_{k=1}^{K} supp(\{i_k\})}} \geq \frac{supp(Y)}{\sqrt[K]{\prod_{k=1}^{K} supp(\{i_k\})}} \geq \frac{supp(Y)}{\sqrt[K+L]{\prod_{k=1}^{K+L} supp(\{i_k\})}} = \cos(Y).$$

Thus we complete the proof. ■

Compared with the well-known anti-monotone property, CAMP makes a special demand on the items in the difference set of the superset and the subset. That is, any item in the difference set must have a higher support than any item in the subset. To achieve this, we must adopt a special itemset traversal sequence. We detail this below.

3.2. Using CAMP

Now we explore how to use CAMP to mine cosine interesting patterns. Let us begin with a lattice view of itemsets.



(a) A Subset Lattice of five items. (b) A SA-SET of five items.

Fig. 1: Illustration of Subset Lattice and Set enumeration tree.

As we know, given the universal itemset $\mathcal{I}$ of a market-basket database, the search space consists of $2^{|\mathcal{I}|}$ different subsets, as shown by Fig. 1(a). In what follows, we propose the Support-Ascending Set Enumeration Tree (SA-SET), a special Set Enumeration Tree (SET) [8], to simplify the lattice yet keeping the completeness of the search. Note in SA-SET, the children of a node N enumerate those itemsets that can be formed by appending a single item of $\mathcal{I}$ to N, with the restriction that this single item must follow every item already in N according to the support-ascending order.

Example 1 *Fig. 1(b) is an example of the SA-SET of five items A, B, C, D and E with* $supp(\{A\}) \leq supp(\{B\}) \leq supp(\{C\}) \leq supp(\{D\}) \leq supp(\{E\})$.

Given the notion of SA-SET, we have the following theorem:

Theorem 2 *The cosine measure can serve as an in-evaluation measure for interesting pattern discovery if the itemset traversal structure is defined by SA-SET.*

PROOF. Suppose Y is an immediate superset of X in the *SA-SET*, i.e., $Y = X \bigcup \{i\}$. According to the notion of the *SA-SET*, $supp(\{i\}) \geq supp(\{i'\}), \forall i' \in X$. Furthermore, since the cosine measure is conditionally anti-monotone, we have $cos(X) \geq cos(Y)$. This means that if X is not an interesting itemset ($cos(X) < min\_cos$), its immediate superset Y will be definitely not an interesting itemset either. Thus the cosine measure can serve as an in-evaluation measure for the early uninteresting superset pruning. ■

Example 2 *Let us look at the SA-SET in Fig. 1(b). We assume that the itemset* $\{C, A\}$ *is uninteresting* ($\cos(\{C, A\}) < min\_cos$). *Then all its supersets* $\{D, C, A\}$, $\{E, C, A\}$ *and* $\{E, D, C, A\}$ *(the shaded nodes in Fig. 1(b)) are guaranteed to be uninteresting and can be pruned safely.*

3.3. Breath-First versus Depth-First

Here we choose the traversal sequence in SA-SET for cosine interesting pattern mining.

Typically, we have two options, either the Apriori-like breath-first strategy or the FP-tree-based depth-first strategy [9]. Since the cosine measure possesses CAMP rather than the anti-monotone property, the well-known candidate generation method $\mathcal{F}_k \times \mathcal{F}_k$ for Apriori is no longer applicable. For example, in Fig. 1(b), the fact that the itemset $\{C, A\}$ is uninteresting does not necessarily mean that $\{C, B, A\}$ is uninteresting, so we may miss the candidate $\{C, B, A\}$ if $\mathcal{F}_k \times \mathcal{F}_k$ is used. In [6], Zhu et al. proposed a novel $\mathcal{F}_k \times (\mathcal{F}_k \bigcup \mathcal{T}_k)$ strategy to handle this problem. Nevertheless, the introduction of $\mathcal{T}_k$ may increase the memory consumption greatly for generating too many uninteresting candidates.

So we turn to the depth-first method. Indeed, the FP-tree is more suitable for the cosine interesting pattern discovery due to its projection mechanism of conditional FP-trees. For example, in Fig. 1(b), if $\{C, A\}$ is uninteresting, not only $\{C, A\}$ but also the interesting patterns ending in $\{C, A\}$ can be pruned directly, which is similar to the case of support for frequent pattern discovery. Furthermore, it has been shown that the FP-growth algorithm itself may outperform the standard Apriori algorithm by several orders of magnitude in discovering frequent patterns. Therefore, in what follows, we focus on mining cosine interesting patterns using the FP-tree.

4. COSI-*tree*: The Algorithmic Issues

In this section, we propose a novel FP-tree-based algorithm: COSI-*tree* to mine cosine interesting patterns.

COSI-*tree* consists of two main procedures: First, FP-tree is constructed in the *support-descending* order; Then, the algorithm *search* is called to mine the interesting patterns from the FP-tree. Specifically, the algorithm *chain*

Algorithm 1: $search(\mathcal{F}, X, Tree, min\_supp, min\_cos)$

```
1  if Tree contains a single prefix path then
2  |  let P be the single prefix of Tree;
3  |  let node be the last item of P;
4  |  chain(F, node, X, min_cos);
5  |  let Q be the multipath part;
6  else
7  |  let Q be Tree;
8  end
9  if Q ≠ ∅ then
10 |  for each item a_i in Q do
11 |  |  generate pattern Y = a_i ∪ X;
12 |  |  if cos(Y) ≥ min_cos then
13 |  |  |  F = F ∪ Y;
14 |  |  |  construct Y's conditional FP-tree Tree_Y;
15 |  |  |  if Tree_Y ≠ ∅ then
16 |  |  |  |  search(F, Y, Tree_Y, min_supp, min_cos);
17 |  |  |  end
18 |  |  end
19 |  end
20 end
```

Algorithm 2: $chain(\mathcal{F}, node, X, min\_cos)$

```
1  while the item of the node is not the root node do
2  |  generate Y = node ∪ X;
3  |  node=node's parent;
4  |  if cos(Y) ≥ min_cos then
5  |  |  F = F ∪ Y;
6  |  |  chain(F, node, Y, min_cos);
7  |  end
8  end
```

is used to deal with the *single prefix-path FP-tree* for optimization. More details are shown below.

4.1. The Overview of COSI-tree

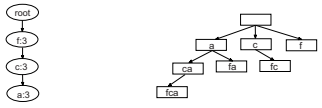
Alg. 1 shows the pseudocode of *search*, which is called recursively for each constructed conditional FP-tree. There are five input parameters in *search*: $\mathcal{F}$ is used to save the cosine interesting patterns, X is an intermediate variable storing the frequent patterns, $Tree$ is the conditional FP-tree to mine, and min_supp and min_cos are the thresholds of the interestingness measures. In each recursion, Alg. 1 firstly divides $Tree$ into two parts, i.e., the single prefix-path portion P and the multipath portion Q. P is processed in Line 4 by calling *chain*, which is shown in Alg. 2. Q is processed in Lines 9-20. Then candidate itemset Y is generated and verified in Line 10-12. If Y is interesting, the conditional FP-tree for Y is constructed and processed recursively in Lines 14 and 16, respectively. However, if Y is not interesting, Y's supersets are uninteresting either by Theorem 1, and thus we can start traversing another item of Q. Note that unlike the FP-growth algorithm introduced by Han et al. [9], $Tree_Y$ is constructed on the whole $Tree$ rather than Q — only the multipath portion.

4.2. Generating Interesting Patterns of *Single-Path* tree

For the single-path FP-tree, the frequent patterns are mined by enumerating all the combinations of the subpaths [9]. In our algorithm, however, the enumeration will be made in a specific order to meet the requirements of cosine interesting pattern discovery.

As can be seen in Fig 2(a), if we have the single-path tree as the figure shows, *chain* will employ the depth-first search strategy, and enumerate frequent item-

sets in a bottom-up fashion, from the bottom node a to the top node f, which is shown in Fig. 2(b).



(a) single-path tree (b) frequent itemsets of single-path tree

Fig. 2: Mining Frequent Patterns From Single-path Tree.

Alg. 2 shows the procedure of dealing with a single-path FP-tree. Lines 1-8 traverse the items from the bottom to the top and generate the corresponding candidate itemset Y. If Y is interesting, *chain* is called recursively to find interesting patterns ending with Y. If not, we traverse the parent of the current *node* in Line 3.

5. Experimental Results

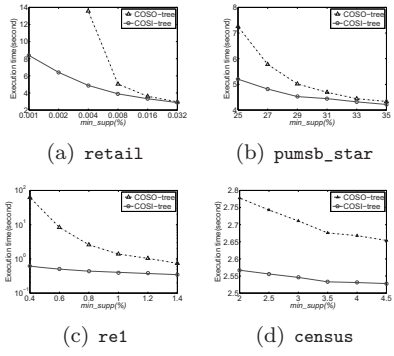
5.1. Experimental Setup

For evaluation purposes, we have performed cosine interesting pattern discovery on various real data sets whose characteristics are summarized in Tab. 1. Specifically, our validation mainly focuses on two aspects: (1) The comparison between the in-evaluation strategy (COSI-*tree*) and the post-evaluation strategy (COSO-*tree*); (2) The interestingness of the patterns found by COSI-*tree*.

Data set	#Item	#Record	Source
`retail`	16470	88162	Belgian Retail store
`pumsb_star`	2088	49046	IBM Almaden
`re1`	3758	1657	Reuters-21578
`census`	132	488422	UCI

Table 1: Some Characteristics of Experimental Data Sets.

All the programs are written in Microsoft Visual C++ 2008. Our implementations of COSI-*tree* and COSO-*tree* algorithms are based on the "FP-growth" source codes provided by Borgelt[1]. We modified the codes so as to incorporate the cosine measure as (1) an in-evaluation measure for COSI-*tree* and (2) a post-evaluation measure for COSO-*tree*. Note that in COSO-*tree*, the FP-tree is constructed in the *support-descending* order too. Although the algorithm doesn't depend on this specific order, experiments show that it leads to much shorter execution time than a random order. Note that the *execution time* used here means the total execution time including the time of constructing the FP-tree from the original database.



(a) `retail` (b) `pumsb_star`

(c) `re1` (d) `census`

Fig. 3: The Execution Time Comparison Between COSO-*tree* and COSI-*tree*: The Impact of *min_supp*.

5.2. COSI-*tree* Versus COSO-*tree*

In this subsection, we compare the execution time between COSI-*tree* and COSO-*tree*.

Fig. 3 shows the execution time over different (absolute) minimum support thresholds. A value of 0.6 is set as the minimum cosine threshold. As can be seen, COSI-*tree* is obviously much more efficient than COSO-*tree* no matter what the *min_supp* is. More im-

[1] Available at http://www.borgelt.net.

portantly, with *min_cos*, we can use COSI-*tree* to identify interesting patterns even when $min\_supp = 0$. On the contrary, COSO-*tree* seems to be very sensitive to *min_supp*. That is, when *min_supp* goes down, the execution time of COSO-*tree* increases rapidly. In the extreme case, e.g., $min\_supp < 0.004\%$ for `retail`, the execution time of COSO-tree increases dramatically to over 100 seconds.

5.3. The Interestingness of the Mined Patterns

We examine the interestingness of the patterns found by COSI-*tree* as follows.

Tab. 2 shows some of the interesting patterns extracted from the `re1` data set. As can be seen, these patterns contain closely related words at a very low support level. For example, for pattern `{victor, bolivian, von, estenssoro, comibol, paz}`, "Victor Paz Estenssoro" was ever elected as the Bolivian president four times, and "comibol" was a Bolivian state-owned mineral enterprize.

Data set	Interesting Patterns	Support	Cosine
re1	victor, bolivian, von, estenssoro, comibol, paz	0.302%	0.812
re1	cuenca, martinez, alejandro, nicaraguan	0.302%	0.913
re1	welland, seawai, erie, lawrence	0.241%	0.855

Table 2: Interesting Patterns in `re1`.

In summary, by using COSI-*tree*, we can discover rare but really interesting patterns at an extremely low support level. This is often computationally prohibitive for a post-evaluation scheme such as COSO-*tree*.

6. Conclusion

In this paper, we provided a study of mining association patterns based on the cosine measure from large-scale data. Specifically, we showed that the cosine measure possesses the conditional anti-monotone property, and thus can be used as an in-evaluation measure for pattern discovery. By exploiting this property, we have developed an efficient COSI-*tree* algorithm for mining interesting patterns. Finally, as demonstrated by the experimental results, COSI-*tree* is suitable for finding rare but real interesting patterns at extremely low levels of support.

References

[1] H. Xiong, P-N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *ICDM*, pages 387–394, 2003.

[2] J. Han and M Kamber. *Data Mining: Concepts and Techniques, 2nd edn.* Morgan Kaufmann, 2005.

[3] S. Zhu, J. Wu, H. Xiong, and G. Xia. Scaling up top-k cosine similarity searcy. *DKE*, In press, 2010.

[4] P-N. Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining.* Addison-Wesley, 2005.

[5] P-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *KDD'02*.

[6] S. Zhu, J. Wu, and G. Xia. Cosine interesting pattern discovery. *Technical Report*, Beihang University, Beijing, China, 2009.

[7] T. Wu, Y. Chen, and J. Han. Association mining in large databases: A re-examination of its measures. In *PKDD'07*.

[8] R. Rymon. Search through systematic set enumeration. In *KR'92*.

[9] J. Han, J. Pei, Y. Yin, and R.Y. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *DMKD*, 8:53–87, 2004.