Efficiently Identify Local Frequent Keyword Co-occurrence Patterns in Geo-tagged Twitter Stream

Xiaoyang Wang[‡], Ying Zhang[♮,‡], Wenjie Zhang[‡], Xuemin Lin[†,‡] *
[†]East China Normal University, China [‡]The University of New South Wales, Australia
[♮]University of Technology, Sydney, Australia
{xiaoyangw, yingz, zhangw, lxue}@cse.unsw.edu.au

ABSTRACT

With the prevalence of the geo-position enabled devices and services, a rapidly growing amount of tweets are associated with geo-tags. Consequently, the real time search on geo-tagged Twitter streams has attracted great attentions. In this paper, we advocate the significance of the co-occurrence of keywords for the geo-tagged tweets data analytics, which is overlooked by existing studies. Particularly, we formally introduce the problem of identifying local frequent keyword co-occurrence patterns over the geo-tagged Twitter streams, namely LFP query. To accommodate the high volume and the rapid updates of the Twitter stream, we develop an inverted KMV sketch (IK sketch for short) structure to capture the co-occurrence of keywords in limited space. Then efficient algorithms are developed based on IK sketch to support LFP queries as well as its variant. The extensive empirical study on real Twitter dataset confirms the effectiveness and efficiency of our approaches.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services

Keywords

Keyword Co-occurrence Pattern; Geo-tagged; Twitter Stream

1. INTRODUCTION

Nowadays, Twitter is becoming one of the most important online social media because it can convey information to people much faster than traditional media. With the proliferation of geo-position enabled device, a large amount of tweets are geo-tagged. For instance, recently it is reported [1] that there are about 30 millions people sending out geo-tagged data into the Twitterverse, and 2.2 percent of tweets (about 4.4 million tweets a day) provide location data together with the text of their posts. As a result, there is an emerging call for effective and efficient data analytics techniques to make sense of this geo-tagged Twitter stream.

Motivation. Due to its concise feature, tweets with geo-tag are studied as the human sensor of social events that are happening in the specific area. Therefore, geographic factor is a big concern for users in the Twitter stream analytics. Moreover, interesting/important events are usually associated with high frequent keywords. Consequently, the problem of identifying the local frequent keywords over Twitter stream has received growing interests in the literature. Nevertheless, the existing works (e.g., [3,5]) overlook the co-occurrence of the keywords (i.e., the keywords appear in the same tweet) which may fail to delivery interesting patterns or even mislead users. Following is a motivating example.

Example 1. *Many people send tweets about the local new events to inform or share with friends. Suppose there are two exciting events in the downtown on the same day: 1) there is a new Apple store opened and 2) a popular restaurant named "Sokyo" provides discount for customers. We may expect that there are lots of tweets about these two local events, where "Apple", "Sokyo", and "discount" will be frequent keywords in this downtown area. Nevertheless, without the co-occurrence knowledge of these keywords, customers may not be able to capture the important information (e.g., discount at Sokyo) or even be misled since customers may conclude there is a promotion in Apple store. Therefore, besides the individual frequent keywords in a particular region, it is critical to identify the frequent keywords co-occurrence (e.g., "Sokyo" and "discount" in this example) so that users can obtain richer and more accurate information.*

Motivated by the above facts, in this paper we investigate the problem of identifying local frequent keyword co-occurrence in geo-tagged Twitter stream. Specifically, given a region R and a minimal support θ, we aim to efficiently identify a set of local frequent keyword co-occurrence patterns (**LFP** for short) whose number of occurrence in the tweets within the region R exceeds θ. Moreover, since in practice users are usually interested in the recent tweets, we apply the sliding window model and the outdated tweets will not be considered in the query processing. We further show that the techniques developed in the paper can be extended to identify dense regions (**DR** for short) regarding a particular keyword co-occurrence pattern, which is useful for users to track the hot areas regarding a particular topic.

Challenges and Contributions. To the best of our knowledge, this is the first work to systematically investigate the

*Corresponding author
[1]http://www.futurity.org/tweets-give-info-location

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609548.

problem of identifying local frequent keywords co-occurrence patterns over geo-tagged Twitter stream. The main challenges of the problem lie on two aspects. Firstly, to accommodate the rapid arrival and expiration of geo-tagged tweets, it is essential to build a concise summary for streaming data to support LFP queries with high accuracy and low query latency. Moreover, the summary of tweets stream should be rapidly updated and be able to capture the co-occurrence of keywords. Secondly, we need to exploit both spatial and frequency of the keywords for the query processing. It is desirable to effectively incorporate the spatial location and frequency of the keyword in the summary of the Twitter stream.

To address the above challenges, we propose a hybrid sketch structure as well as efficient query processing algorithms. Our principle contributions are summarized as follows. 1) We formally define the problem of identifying local frequent keyword co-occurrence patterns over geo-tagged Twitter stream. 2) We propose an inverted KMV sketch (IK sketch for short) strucutre, which is a summary of tweets stream and can properly capture both spatial and frequency information with limited space. 3) Efficient algorithms are proposed to support LFP query and its variant based on IK sketch structure. 4) Extensive empirical study demonstrates the efficiency and effectiveness of our techniques proposed in the paper.

Road Map. The problem studied in this paper is formally presented in section 2. The preliminary work is introduced in Section 3. Section 4 overviews the IK sketch structure and its maintenance algorithm. The effectiveness and efficiency of the techniques are demonstrated in Section 5. Section 6 concludes this paper.

Related Work. Recently, much efforts have been devoted to identify spatial frequent or burst keywords in geo-temporal Twitter stream, which has proven useful in various applications. Existing works consider the spatial and temporal dimensions, but the co-occurrence of keywords is ignored. For instance, the frequency of each individual keyword is counted regarding particular regions in [3,5,9]. In [1], the correlation of the keywords is considered by the EvenTweet system where keywords with close spatial proximity are grouped together. However, the co-occurrence of keywords in the *same* tweet is not considered. On the other hand, the approximate frequent pattern mining over data streams has attracted significant attention in the literature (See survey in [7]) where the co-occurrence of keywords in each transaction is considered by some existing works. Nevertheless, the spatial location is not considered and their techniques cannot be trivially extended to support the problem studied in this paper. There are some existing work on spatial co-location pattern mining (See survey in [4]). However, this problem is inherent different with ours because they do not consider the co-occurrence of keywords in the *same* tweet. Moreover, their techniques are not developed in the context of data streams.

2. PROBLEM DEFINITION

In this section, we first introduce some important notations and definitions. Without loss of generality, we adopt the count-based sliding window model in this paper; that is, the Twitter stream is denoted as $\mathcal{D} = \{d_1, d_2, ..., d_n\}$ where n is the sliding window size and d_n is the latest arrived tweet. Each tweet $d \in D$ consists of the tweet's spatial location (denoted by $loc(d)$) and a set of keywords (denoted by $key(d)$) from a vocabulary $\mathcal{V}$. Let P denote a keywords co-occurrence pattern where $P = \{k_i\}$ with $k_i \in \mathcal{V}$. The frequency of P regarding a region R, denoted by $f(P, R)$, is the number of occurrence of P within the region R. Formally, we have $f(P, R) = |\{d \in \mathcal{D} \mid P \subseteq key(d) \text{ and } loc(d) \in R\}|$. For presentation simplicity, local frequent pattern thereafter is referred to the local frequent keyword co-occurrence pattern. Following is a formal definition.

Definition 1. *(Local Frequent Pattern, LFP) Given a region R and a minimal support θ, we say a pattern P is local frequent pattern over the Twitter stream $\mathcal{D}$ if $f(P, R) \geq \theta$.*

A LFP is *maximal*, if there is no superset of the pattern that is frequent.

Problem Statement. In this paper, we investigate the problem of identifying the maximal LFP over the geo-tagged Twitter stream. Particularly, we build a small summary of the geo-tagged Twitter stream $\mathcal{D}$ which consists of the most recent n tweets posted. We aim to approximately identify all maximal local frequent patterns, denoted as LFP query, with high accuracy and a small space.

3. PRELIMINARIES

In this section, we briefly introduce the KMV synopsis [2], which is used in our IK sketch to capture the co-occurrence of keywords. KMV synopsis is designed for estimating the number of distinct value in a multiset. Assume N points are uniformly distributed over [0,1], the expected distance for two adjacent points is $1/(N+1) \approx 1/N$. Consider that h is a uniform random hash function. Each distinct value v_i in set is hashed to [0,1] and $h(v_i) \neq h(v_j)$ if $i \neq j$. KMV synopsis consists of k smallest hash values, so the number of distinct value can be estimated with equation 1.

$$\hat{D}_k = \frac{k-1}{U^k} \tag{1}$$

Based on the analysis in [2], the expected relative error of the estimator is $\sqrt{\frac{2}{\pi(k-2)}}$.

Intersection Operation: Given two sets $\mathcal{A}$ and $\mathcal{B}$, with corresponding KMV synopses $\mathcal{L}_\mathcal{A}$ and $\mathcal{L}_\mathcal{B}$ of size k_A and k_B, respectively. In this paper, we use $\cup_m$ ($\cap_m$) to denote the union (intersect) operation which removes the duplicate values. In [2], $\mathcal{L}_\mathcal{A} \oplus \mathcal{L}_\mathcal{B}$ is used to denote the set compromising the k smallest distinct hash values in $\mathcal{L}_\mathcal{A} \cup_m \mathcal{L}_\mathcal{B}$ where $k = min(k_A, k_B)$. Then $\mathcal{L} = \mathcal{L}_\mathcal{A} \oplus \mathcal{L}_\mathcal{B}$ is the KMV synopses of $\mathcal{A} \cup \mathcal{B}$. Let $K_\cap$ denote the number of common distinct hash values in $\mathcal{L}$ regarding $\mathcal{L}_\mathcal{A}$ and $\mathcal{L}_\mathcal{B}$, i.e., $K_\cap = |\{v \in \mathcal{L} : v \in \mathcal{L}_\mathcal{A} \cap_m \mathcal{L}_\mathcal{B}\}|$. We can estimate the number of distinct values in $\mathcal{A} \cap \mathcal{B}$, denoted by $\hat{D}_\cap$, as follows.

$$\hat{D}_\cap = \frac{K_\cap}{k} \times \frac{k-1}{U^k} \tag{2}$$

Note that the above intersection operation can be immediately extended to support intersection operation on multiple sets following Equation 2.

4. IDENTIFY LFP

In this section, we first introduce the IK sketch, which is a hybrid structure of KMV synopsis and Z-order technique [8] as well as the sketch graph, followed by the corresponding maintenance algorithms. Then we develop efficient algorithms to support LFP query and its variant based on IK sketch technique.

4.1 IK Sketch and Sketch Graph

IK Sketch Structure. An IK sketch $\mathcal{L}$ is an inverted index structure for KMV synopsis of tweets objects. Specifically,

we continuously maintain k objects with the k smallest hash values over the Twitter stream $\mathcal{D}$. For each keyword t in the sampled tweet objects, we build a posting list, denoted by $\mathcal{L}(t)$, where $\mathcal{L}(t) = \{d | d \in \mathcal{L} \text{ and } t \in key(d)\}$. Let $z(d)$ represent the Z-order value [8] of the object d according to $loc(d)$. The objects in each posting list are sorted by their Z-order values. This can significantly reduce the cost of spatial range search (i.e., identify objects within the search region R) since the spatial proximity of objects are preserved by Z-order values.

Sketch Graph. To facilitate the LFP queries which need to count the co-occurrence of keywords, we also maintain a sketch graph (SG for short) for the keywords in IK sketch. A node of the SG corresponds to a keyword in IK sketch $\mathcal{L}$, and there is an edge between two keywords (nodes) t_1 and t_2 if there is a co-occurrence of t_1 and t_2 in an object $d \in \mathcal{L}$. The weight of an edge is the number of co-occurrences of two keywords in $\mathcal{L}$. Figure 1 illustrates an example of sketch graph where there are five keywords $A \sim E$ in $\mathcal{L}$. To facilitate the range search, for each edge in sketch graph SG, we also maintain a minimal bounding box of the objects which contribute to the weight of the edge.

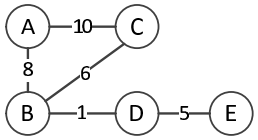


Figure 1: Sketch Graph

IK sketch and sketch graph maintenance. Since the Twitter stream $\mathcal{D}$ is continuously updated due to the arrival and expiration of tweets data, we cannot guarantee the k objects with smallest hash values are continuously maintained without retrieving objects in $\mathcal{D} \backslash \mathcal{L}$. Therefore, besides objects in $\mathcal{L}$, we also maintain a set of candidate objects, denoted by $\mathcal{L}_c$, which may become objects of $\mathcal{L}$ in the future. By taking advantage of the property of skyband [10], we can efficiently maintain a candidate set with much smaller size than n. Specifically, we say an object d_1 dominates another object d_2 if d_1 arrives later than d_2 and $h(d_1) \leq h(d_2)$. It is safe to exclude an object d if it is dominated by more than k other objects in $\mathcal{L}_c$ since d will never join $\mathcal{L}$. Consequently, we only need to continuously maintain the skyband of $\mathcal{D}$, i.e., objects survived from the above dominance check, and efficient skyband maintenance algorithm is developed in [10].

Algorithm 1 illustrates the details of the continuous maintenance of IK sketch and sketch graph SG. For each new incoming object o, we will update the IK sketch $\mathcal{L}$ and the sketch graph SG if $h(o)$ is smaller than the current k-th smallest hash value in $\mathcal{L}$. Note that $w(t_1, t_2)$ represents the weight of an edge between two nodes t_1 and t_2 in SG. On the other hand, the IK sketch and sketch graph are updated when an object in $\mathcal{L}$ is expired or replaced by others. Note that an edge of SG is removed if its weight becomes zero. Meanwhile, the skyband $\mathcal{L}_c$ is continuously maintained to provide candidate for $\mathcal{L}$.

4.2 LFP Query Processing

We develop an efficient algorithm to support LFP query based on the IK sketch and sketch graph structures, where details are illustrated in Algorithm 2. Let L_i represent a set of local frequent patterns with size i and C_i keeps candidate patterns for L_i. Line 2 retrieves the relevant samples in $\mathcal{L}$ falling in the region R. Then the local frequent patterns with single keyword are identified based on Equation 1 (Lines 3-5). Instead of directly applying the Apriori property, Line 8 takes advantage of the SG graph to generate a smaller number of candidate patterns. Specifically, a keyword t is considered for the growth of a local frequent pattern P *iff* the node t has an edge with each keyword in P and the minimal bounding box of each edge overlaps the region R. Lines 9-11

Algorithm 1: Continuous Maintenance Algorithm

Input : o : a new coming object.
Output: Updated IK sketch $\mathcal{L}$ and sketch graph SG.

```
1  v := h(o) ;
2  if o is inserted into L then
3      foreach keyword t ∈ key(o) do
4          L(t) := L(t) ∪ o ;
5      foreach pair of keywords (t1, t2) in o do
6          increase w(t1, t2) by one ;
7  if an object d ∈ L is expired or replaced then
8      remove d from L ; update L based on Lc ;
9      foreach pair of keywords (t1, t2) in d do
10         decrease w(t1, t2) by one ;
11 update Lc ;
```

Algorithm 2: LFP Query Algorithm

Input : R : query region, θ : minimum support.
Output: Maximal local frequent patterns regarding R.

```
1  P := ∅ ; L1 := ∅ ;
2  Lr := samples of L within the region R ;
3  foreach t ∈ Lr do
4      estimate f(t, R) based on the Equation 1 ;
5      L1 := L1 ∪ t if f(t, R) ≥ θ;
6  i := 1 ;
7  while Li ≠ ∅ do
8      Ci+1 ← generate candidate patterns from Li and SG;
9      foreach pattern P ∈ Ci+1 do
10         estimate f(P, R) based on Equation 2 ;
11         Li+1 := Li+1 ∪ P if f(P, R) ≥ θ ;
12     P := P ∪ Li+1; i := i + 1 ;
13 return maximum patterns in P ;
```

verify candidate patterns based on Equation 2. Finally, we return the *maximal* local frequent patterns.

4.3 Extension.

Besides LFP queries, sometimes users may be interested in identifying dense regions (DR) for a given pattern, which are useful for users to track the hot areas for a certain topic. Suppose the space is partitioned into unit grid cells (regions). For a given pattern P which users are interested in (e.g., "iPhone" and "promotion"), we aim to identify the dense regions regarding P; that is, find the cells $\{C\}$ with $f(P, C) \geq \theta$ where θ is a pre-given minimal support. The computation of $f(P, C)$ is similar to that of Algorithm 2 by utilizing IK sketch techniques. Then we may further merge the connected dense regions for a better understanding of the distribution of the dense regions regarding a pattern P.

5. EXPERIMENTAL EVALUATION

In this section, we demonstrate an empirical performance study to evaluate the effectiveness and efficiency of the IK sketch and query processing algorithms (denoted by **IK**) on Twitter data. Since there is no existing work for the problems studied in this paper, we use a uniform sampling based approach (denoted by **UN**) as the baseline method. Specifically, a uniform sample of $\mathcal{D}$ is continuously maintained and we have $f(P, R) = \frac{\hat{f}(P,R)}{s}$ where s is the sample rate and $\hat{f}(P, R)$ is the local frequency of P derived from the samples.

Dataset. One real Twitter dataset from [6] is employed in the empirical study, which contains 13 million geo-tagged tweets collected from May 2012 to August 2012.

Experiment Setting. All the algorithms in the experiments are implemented in C++ and experiments are run on a PC with Intel Xeon 2.40GHz dual CPU and 4G memory with Debian system. The slide window size n is set as 0.5

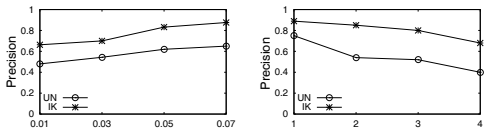
million and k is set to $0.05 \times n$ as default. The minimal support θ is set to $0.1 \times n \times frac\_region$, where $frac\_region$ is the fraction of query region size to the whole space size. The search regions in LFP queries are randomly chosen, and the region size varies from 0.01 to 0.07 of the space. The space is partitioned into 40×40 grids for DR queries. 500 queries are issued for both LFP and DR queries, and the average precision, recall and response time are used for performance evaluation.

Experimental Results. We next present our findings.

1) Effectiveness. As expected, there are many interesting local frequent keyword co-occurrence patterns detected by issuing LFP queries. For instance, we find two frequent patterns in Taipei: 1) *mac* and *on sale* in Taipei World Trade Center. 2) *Eslite*, which is a popular bookstore close to Taipei World Trade Center. Clearly, if we ignore the co-occurrence of the keywords, one may think there is a *on sale* at *Eslite* bookstore.

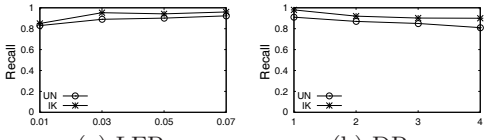
2) Precision and Recall. To evaluate the quality of our approximate approaches, Figures 2 and 3 depict the precision and recall for LFP queries and dense region (**DR**) queries. For LFP queries, we evaluate the performance of the algorithms by varying the size of query in terms of the percentage of space; for DR queries, we vary the size of the query pattern. Through the experiments, we observe that: 1) IK sketch has similar recall with UN method, but significantly outperforms the UN in terms of precision for both queries. This is because KMV synopsis is more suitable for set intersection operation compared with uniform sampling approach. Moreover, it is reported that the UN approach often overestimates the frequency of the patterns which may result in low precision compared with IK sketch. 2) it is shown in Figure 2(a) that the precision of two methods decreases when the search region becomes small in LFP queries. This is because there are less sample objects within the search region, and hence lead to a large relative error according to the analysis in [2]. Similar trend is observed in Figure 2(b) for DR queries when the size of the pattern increases.

Figures 4(a) and 4(b) report the precision of two queries when the sample rate grows. As expected, the performance of both algorithms improves with the growth of sample rate. Nevertheless, the performance of IK sketch significantly outperforms UN method under all settings.



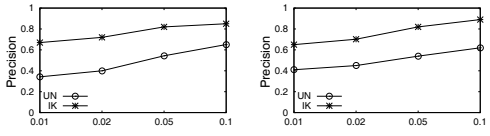
(a) LFP (b) DR

Figure 2: Precision Evaluation



(a) LFP (b) DR

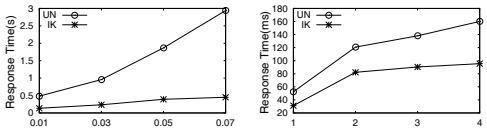
Figure 3: Recall Evaluation



(a) LFP (b) DR

Figure 4: Sample Rate Evaluation

3) Efficiency. To evaluate the efficiency of IK sketch, we examine the query response time and the sketch update time. The query response time is reported in Figure 3. For LFP query, we measure the running time by varying the size of the query region. With the increase of the region size, the query time of two algorithms increases. While IK sketch significantly outperforms UN method because the former approach can take advantage of SG and Z-order. Particularly, SG can significantly reduce the number of candidate patterns, and Z-order technique can speed up the retrieval of relevant objects. For DR query, we vary the size of the query pattern, and similar trend is observed for DR query. We also evaluate the update time of IK sketch by changing the size of k. It takes $67.8ms$ and $89.3ms$ when k is set to $0.005 \times n$ and $0.1 \times n$ respectively. Recall that n is the size of the Twitter stream. While it takes $66.1ms$ and $87.5ms$ respectively for UN approach.



(a) LFP (b) DR

Figure 5: Efficiency Evaluation

6. CONCLUSIONS

In this paper, we investigate the problem of identifying local frequent keyword co-occurrence patterns over geo-tagged Twitter streams. A novel sketch technique is proposed to summarize the recent geo-tagged Twitter streams. Then efficient algorithms are developed to support local frequent pattern query studied in the paper. The empirical study has demonstrated the efficiency and effectiveness of our approaches.

7. ACKNOWLEDGMENTS

Ying Zhang is supported by ARC DE140100679 and DP13 0103245. Wenjie Zhang is supported by ARC DE120102144 and DP120104168. Xuemin Lin is supported by NSFC612320 06, NSFC61021004, ARC DP120104168 and DP110102937.

8. REFERENCES

[1] H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *PVLDB*, 6(12), 2013.

[2] K. S. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla. On synopses for distinct-value estimation under multiset operations. In *SIGMOD Conference*, 2007.

[3] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi. Geoscope: Online detection of geo-correlated information trends in social networks. *PVLDB*, 7(4), 2013.

[4] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15(1), 2007.

[5] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras. On the spatiotemporal burstiness of terms. *PVLDB*, 5(9), 2012.

[6] G. Li, Y. Wang, T. Wang, and J. Feng. Location-aware publish/subscribe. In *KDD*, 2013.

[7] H. Liu, Y. Lin, and J. Han. Methods for mining frequent items in data streams: an overview. *Knowl. Inf. Syst.*, 26(1), 2011.

[8] G. M. Morton. A computer oriented geodetic data base and a new technique in file sequencing. *Technical Report, Ottawa, Canada: IBM Ltd*, 1966.

[9] K. Watanabe, M. Ochi, M. Okabe, and R. Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *CIKM*, 2011.

[10] Y. Zhang, X. Lin, Y. Yuan, M. Kitsuregawa, X. Zhou, and J. X. Yu. Duplicate-insensitive order statistics computation over data streams. *TKDE*, 22(4), 2010.