Region-Based Message Exploration over Spatio-Temporal Data Streams

Lisi Chen,[1] **Shuo Shang**[2*]
[2] UESTC, China [1,2]Inception Institute of Artificial Intelligence, UAE
[1]chenlisi.cs@gmail.com [2]jedi.shang@gmail.com

Abstract

Massive amount of spatio-temporal data that contain location and text content are being generated by location-based social media. These spatio-temporal messages cover a wide range of topics. It is of great significance to discover local trending topics based on users' location-based and topic-based requirements. We develop a region-based message exploration mechanism that retrieve spatio-temporal message clusters from a stream of spatio-temporal messages based on users' preferences on message topic and message spatial distribution. Additionally, we propose a region summarization algorithm that finds a subset of representative messages in a cluster to summarize the topics and the spatial attributes of messages in the cluster. We evaluate the efficacy and efficiency of our proposal on two real-world datasets and the results demonstrate that our solution is capable of high efficiency and effectiveness compared with baselines.

Introduction

Massive amount of spatio-temporal data containing location, text, and time information are being generated on an unprecedented scale. Such type of data, which can be modelled as multimodal data streams, offer first-hand information for various kinds of local breaking news, bursty events, and general public concerns.

Due to the high arrival rate of spatio-temporal data streams and their vast topic coverage, it is of great interest for users to discover and monitor trending events and topics based on their preferred spatial and textual attributes. Additionally, end-users may want to effectively grab the key points of the spatio-temporal messages within a particular region. The problem of online spatial keyword search, which allows users to retrieve spatio-temporal messages (e.g., geotagged tweets) relevant to their queries has been studied by a number of research projects (Rocha-Junior et al. 2011; Hu et al. 2015a; Guo et al. 2015; Li et al. 2013). Through online spatial keyword search, users stay informed about local events, trending activities, public concerns, and so forth that happen around them. Moreover, users browse various types of location-aware news and information in a real-time fashion.

The problems of existing studies of online spatial keyword search aim at finding a set of spatio-temporal messages as the result. In other words, users may receive all messages satisfying their pre-defined spatial and textual requirements. However, such item-based spatial keyword search has the following limitations. First of all, most of the users prefer receiving summary information that satisfies their interested topics and location-based patterns rather than receiving a list of raw items (Farzindar and Khreich 2015; Tobler 1970). Secondly, users may receive a set of near-duplicate messages (Ozsoy, Onal, and Altingovde 2014), which greatly reduce the result diversity and topic coverage. Thirdly, it is difficult for users to understand the key topics and the local distribution of a large number of result spatio-temporal messages within a few seconds. In particular, besides informative messages, many messages on data streams are not related to any particular real-world events and are often exhibit low quality (e.g., spam messages) (He et al. 2007).

To address the above limitations, we develop a region-based message exploration mechanism that retrieve spatio-temporal message clusters (i.e., cluster regions) based on users' preferences on message topic and message spatial distribution (i.e., subscription region). Next, we devise a region summarization algorithm that finds a subset of representative messages in a cluster to summarize the topics and the spatial attributes of messages in the cluster.

Two challenges exist in our proposal. First, we need to define a metric that can effectively measure the relevancy between a cluster region and a subscription region. Second, we need to develop an efficient and effective algorithm to generate a region summary for each cluster region, which is an NP-hard problem. To address the first challenge, we leverage triplet network, which is a deep metric learning model, to learn a metric that measures the relevancy between a subscription region and a cluster region. We use Convolutional Neural Network (CNN) to extract spatial and topic features. In addition, we propose a novel method to train our triplet to make our relevancy metric robust to noise and skewness of spatio-temporal data. To address the second challenge, we investigate the region summarization problem and prove that the problem is NP-hard. We develop an efficient greedy algorithm for region summarization problem. We conduct extensive experiments to evaluate the effectiveness of our rele-

[*]Corresponding author.

vancy metric and the efficiency of our region summarization algorithm.

Problem Statement

We introduce the spatio-temporal message, the problem of region-aware publish/subscribe, and the problem of region summarization.

Definition 1: Spatio-Temporal Message. A spatio-temporal message is denoted by $m = \langle\psi, \rho, t_c\rangle$, where ψ is textual information (i.e., a set of terms), ρ is a coordinate represented by latitude and longitude on map, and t_c is the timestamp indicating the publish time of m. □

Spatio-temporal messages are ubiquitous in modern social media. For instance, they can be tweets with location information, geo-tagged photos in Instagram, check-ins with textual messages in Foursquare, web news with geographical information, etc. In our settings, the arrival rate of spatio-temporal messages is very high.

Region-aware publish/subscribe

Definition 2: Cluster Region. A cluster region c_r is the minimum bounding rectangle of a spatio-temporal message cluster generated by a clustering algorithm. □

Note that our proposal does not depend on a specific clustering algorithm because it is important for applications to cope with the data from different types of resources and meet the requirements in various scenarios.

Definition 3: Subscription Region. A subscription region s_r is represented by a rectangular region on the map. It retrieves top-K cluster regions (i.e., a set of spatio-temporal messages) based on a relevancy metric (i.e., $rel(\cdot)$). □

We define the relevancy between a subscription region s_r and a cluster region c_r as $rel(s_r, c_r)$. The relevancy metric $rel(\cdot)$ should consider both the similarity of textual content and the relative locations of the messages in a region. Hence, we aim to develop an effective approach to learning the relevancy metric between a subscription region and a cluster region.

Region summarization

When a relevant cluster region is delivered to a subscription region, we propose to summarize the region of the cluster by selecting a subset of representative spatio-temporal messages from the cluster region and display them on the map. In particular, the selection objective is to generate a summary set by considering both representative and concise aspects. It remains a challenge on selecting such a representative set from messages in the region of each delivered cluster. Specifically, if we select a large number of messages as the summary set, users may be difficult in discovering the outlined information of a region; Otherwise, it will be impossible to generate representative information because of data sparsity. We also need to avoid two messages located too close to each other (i.e., cartographic diversification (Sarma et al. 2012)) and avoid two messages who are temporally close to each other (i.e., query result diversification (Chen and Cong 2015)).

As a consequence, we need to consider the following selection criteria: (1) *Summary set cardinality*; (2) *Summary representativeness*; (3) *Message spatial proximity constraint*; (4) *Message temporal proximity constraint*. To address criterion (1), we fix the cardinality of the summary set as k. For (2), we set it to be an optimization criterion. Specifically, we define the *information coverage* to measure the *strength in summarizing* of a summary set. We define criteria (3) and (4) as constraints by enforcing the spatial proximity and temporal proximity between any two messages in a summary set should no less than a spatial proximity threshold ζ and a temporal proximity threshold δ, respectively.

Definition 4: Region Summarization (RS) Problem. Given a cluster region c_r and an integer n, let M be a set of spatio-temporal messages located in c_r and ζ be a spatial proximity threshold. The RS problem finds a subset N of M such that: (1) $|N| = n$; (2) $\forall\, m_i, m_j \in N$, $d(m_i, m_j) \geq \zeta$; (3) $\forall\, m_i, m_j \in N$, $|m_i.t_c - m_j.t_c| \geq \delta$; (4) The *information coverage* of N for M is maximized. □

Note that we use *information coverage* to measure how well N can represent M, which is defined by Equation 1.

$$I_C(N, M) = \frac{1}{|M|} \times \sum_{m_i \in M} max\{m_j \in N | S(m_i, m_j)\}, \tag{1}$$

where $S(m_i, m_j)$ represents the spatio-textual similarity between m_i and m_j (Equation 2).

$$\begin{aligned} S(m_i, m_j) \quad = \quad & \alpha \times P(m_i.\rho, m_j.\rho) + (1 - \alpha) \\ & \times T(m_i.\psi, m_j.\psi), \end{aligned} \tag{2}$$

where $P(m_i.\rho, m_j.\rho)$ denotes the spatial proximity score between $m_i.\rho$ and $m_j.\rho$, $T(m_i.\psi, m_j.\psi)$ denotes the textual similarity between $m_i.\psi$ and $m_j.\psi$, and α is a preference parameter ranging from 0 to 1 that balances the weight of spatial proximity and textual similarity. Equation 2 is a widely applied similarity measurement that takes both spatial and textual aspects into consideration (Chen et al. 2013).

Framework

Figure 1 illustrates the framework of our region-based message exploration mechanism. We have two types of input data: (1) A stream of spatio-temporal messages published by location-based social media; (2) A set of subscription regions registered by users.

In the first stage, continuously arriving spatio-temporal messages are clustered by a general spatio-temporal online clustering algorithm. Next, each new cluster is regarded as a "query" and we traverse the subscription index to find a subset of subscription regions that *match* the new cluster region based on a relevancy metric between subscription region and cluster region (i.e., $Rel(\cdot)$), which is learned by a deep metric learning model, namely the triplet network. To visualize each delivered cluster we select a subset of representative spatio-temporal messages from the cluster region (i.e., region summarization). Specifically, the selected messages are expected to be representative of the spatial and textual information of all messages in the cluster region.

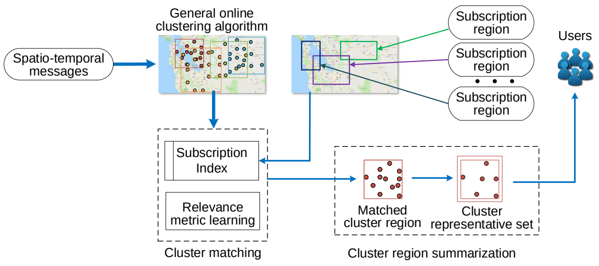


Figure 1: Framework Overview

Relevancy Metric Learning

Inspired by Convolutional Neural Networks (CNNs) based metric learning, which exhibits excellent performance in image classification, we propose to use CNN for capturing regional spatial correlation among spatial-temporal messages.

In particular, we develop a relevancy metric learning model based on triplet network, which learns a metric for comparing the spatio-textual relevancy between two regions by considering both spatial distribution and text similarity of messages in the regions (Hoffer and Ailon 2015; Liu, Zhao, and Cong 2018). To apply triplet network on spatio-temporal messages, we design an approach to generating training data with hard negative example mining (Wang, Lan, and Zhang 2017). Additionally, to make our model robust to data skewness, we use a normalized training loss, which bound the loss of each training tuple within $[0, \tau]$.

Settings of triplet network

This section presents our settings of the triplet network for learning a relevancy metric.

A triplet network, denoted by $TN(\cdot)$, contains the following three instances of a shared CNN (Krizhevsky, Sutskever, and Hinton 2012): (1) A subscription instance a; (2) A relevant instance a^+; (3) An irrelevant instance a^-. While inputting the above three instances, $TN(\cdot)$ calculates the following two values: (1) $D(TN(a), TN(a^+))$; (2) $D(TN(a), TN(a^-))$. Note that $D(\cdot)$ represents the Euclidean distance between the two instances and $TN(a_i)$ indicates the feature map of a_i in the last layer of the CNN. Because that we are only interested in a feature embedding, it is not necessary for us to maintain the fully-connected layer in the CNN.

Existing study suggests that the objective of training $TN(\cdot)$ is to enforce $D(TN(a), TN(a^+)) < D(TN(a), TN(a^-))$ (Hoffer and Ailon 2015). The objective function for training $TN(\cdot)$ can be formulated as follows:

$$\begin{aligned}&\lambda\|TN(\cdot)\|_2 + \sum_{i=1}^{N} max\{0,\\ &D(TN(a_i), TN(a_i^+)) - D(TN(a_i), TN(a_i^-)) + g\},\end{aligned} \tag{3}$$

where $\|TN(\cdot)\|_2$ denotes a L_2 regularization for $TN(\cdot)$, λ is a parameter indicating the weight decay, N is the number of the triplets of samples, and g represents the gap parameter between two distances.

Input of triplet network We discuss how to feed a region R (i.e., R can be either a subscription region or a cluster region) into the triplet network $TR(\cdot)$.

We represent R by a set of grid squares, where the size of each square is pre-defined (e.g., 100 m^2). Each square is associated with a vector that sums up the attribute vector of each message located in the square. Note that the attribute vector contain textual (topic) and temporal information of a message. Here, R can be regarded as a 3-dimensional tensor $R \in \mathbb{R}^{x \times y \times t}$, where $x \times y$ denotes the spatial dimensions of the grid squares and t demotes the attribute dimension.

Output of triplet network The output of the triplet network $\mathcal{R}$ is a feature map of $TN(R)$, which can be represented as a 3-dimensional tensor as well (i.e., $\mathcal{R} \in \mathbb{R}^{x' \times y' \times d_k}$), where d_k denotes the number of dimensions of the output features. Each dimension of the output feature is considered to be a latent feature with spatial information.

Since difference regions may have different values of x and y, how to compare feature maps with different sizes remains a challenge. To solve the problem, we add a feature aggregation layer (i.e., $f_a(\cdot)$) at the end of $TN(\cdot)$. Such layer can aggregate all feature maps to their corresponding feature vectors with d_k dimensions. Let $\mathbf{w}$ be the output of $f_a(\mathcal{R})$, we have $\mathbf{w} \in \mathbb{R}^{d_k}$. After applying $f_a(\cdot)$, the output distances, namely $D(TN(a), TN(a^+))$ and $D(TN(a), TN(a^-))$, can be formulated as follow:

$$D(TN(a), TN(a^+)) = \|\mathbf{w} - \mathbf{w}^+\|_2, \tag{4}$$

$$D(TN(a), TN(a^-)) = \|\mathbf{w} - \mathbf{w}^-\|_2. \tag{5}$$

Based on Equation 4, Equation 5, and the learned metric, we formulate the relevancy between a subscription region s_r and the region of a cluster c_r as a value in $[0, 1]$, which is presented as follows:

$$rel(s_r, c_r) = \frac{1}{1 + \|\mathbf{w}_s - \mathbf{w}_c\|_2}, \tag{6}$$

where $\mathbf{w}_s$ and $\mathbf{w}_c$ represents $f_a(s_r)$ and $f_a(c_r)$, respectively.

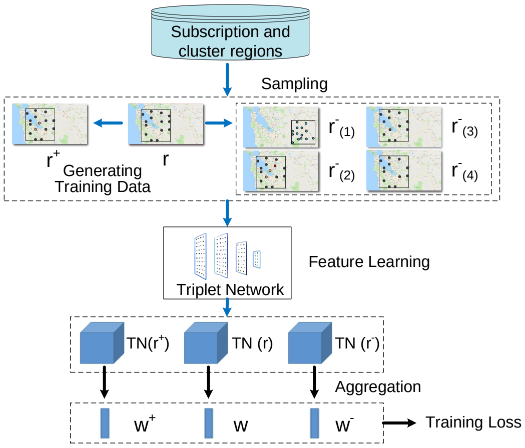


Figure 2: Workflow of relevancy metric learning

Training triplet network

This section presents how to train our triplet network on subscription and cluster regions. Two technical challenges exist in training triplet network. Firstly, we do not have groundtruth (labeled data) for $rel(\cdot)$. Secondly, the trained network must be able to resist small noise. To solve the first challenge, we generate "labeled region" from unlabeled training data (i.e., sampled regions) and learn their corresponding *self-relevancy*. To solve the second challenge, we develop a self-supervised learning mechanism that learns robust $rel(\cdot)$ directly on the basis of the self-relevancy. Specifically, robust $rel(\cdot)$ is expected to regard regions that have slight differences as relevant. We proceed to present how to generate labeled regions as training data.

Generation of labeled regions At first, we randomly sample existing subscription and cluster regions from the underlying space. For each sampled region r, we generate r^+ and r^- respectively.

To generate r^+, we add four types of noise to r: (1) Random message insertion (i.e., inserting $\kappa \times |r|$ messages into r at a random location); (2) Random message deletion (i.e., randomly removing $\kappa \times |r|$ messages from r); (3) Random message move (i.e., moving each message in r along a random direction by a random distance smaller than υ); (4) Random message time shift (i.e., changing the creation time of each message by a random value smaller than σ). Here $|r|$ denotes the number of spatio-temporal messages in r, $\kappa \in (0, 1)$, υ, and σ are three noise indicators.

We generate four groups of r^-. The first group of r^-, denoted by $r^-_{(1)}$, is generated by sampling regions from the underlying space that do not have overlapping area with r. The second group $r^-_{(2)}$ is generated by adding message insertion/deletion noise to r^+. The third group $r^-_{(3)}$ is generated by adding message move noise to r^+. The fourth group $r^-_{(4)}$, is generated by adding time shift noise to r^+. The rationale of this quad generation is that most of the regions in $r^-_{(1)}$ bear little resemblance to r, which may lower the determinativeness of learned features, while the regions in $r^-_{(2)}$, $r^-_{(3)}$, and $r^-_{(4)}$ do have some relevancy towards r even if it is smaller than r^+. As a result, such quad generation can enhance the determinativeness of features learned by the model, which can substantially lower the loss.

Loss function Due to the skewness of spatial distribution for spatio-temporal messages, the loss of our triplet network can be highly skewed. Specifically, some training instances may have extremely large loss, which is inevitable to overwhelm other instances. A straightforward approach to resolving this issue is normalising the cardinality of different regions. However, we are unable to acquire the number of messages in a region afterwards, which is regarded as very important information. For example, if we do not consider the region cardinality, a region containing 1,000 crime alert messages would be unreasonably regarded as similar to a region with 5 crime alert messages. Hence, we propose to modify the loss function by introducing a "normalization step", which is presented as follows.

$$Loss = \lambda \|TN(\cdot)\|_2 + \sum_{i=1}^{N} max\{0, \frac{D(TN(a_i), TN(a_i^+))}{D(TN(a_i), TN(a_i^+)) + D(TN(a_i), TN(a_i^-))} - g\}. \quad (7)$$

Algorithm for Region Summarization

In this section, we first show our proof that the RS problem is NP-hard. Next, we present our proposed algorithm for solving the RS problem. Finally, we present the complexity and approximation analyses of our proposed algorithm.

Theorem 1: The RS problem is NP-hard.

Proof. The RS problem can be reduced from an existing NP-hard problem, the dominating set problem on a graph $G(V, E)$, which outputs whether there exists a k-subset of vertices $V_d \subseteq V$ such that: (1) $\forall v_i \in V$, $v_i \in V_d$; or (2) $\exists v_j \in V_d$ s.t. $v_i \in Neighbors(v_j)$.

Let M be a set of spatio-temporal messages. We assume that $\forall m_i, m_j \in M$, $d(m_i, m_j) \geq \zeta$. We build a RS problem to resolve a dominating set problem as follows. Given the graph $G(V, E)$, we map each vertex $v_i \in V$ to a spatio-temporal message m_i. Specifically, if v_i and v_j are neighbors, we set $S(m_i, m_j) = 1$; Otherwise, we set $S(m_i, m_j) = 0$.

Let $V_d = \{v_0, v_1, ...v_{n-1}\}$ be the result of an instance of dominating set problem, and $M_s = \{m_0, m_1, ...m_{n-1}\}$ be the mapped result set, which can be considered to be the result of the RS problem. Now we assume that $|\mathcal{C}| = \sum_{m_i \in \mathcal{C}} max\{m_j \in M_s | S(m_i, m_j)\}$, $\forall m_i \in \mathcal{C}$ we have $max\{m_j \in M_s | S(m_i, m_j)\} = 1$. Consequently, we have the following two situations: (1) $m_i \in M_s$; (2) $\exists m_j \in \mathcal{C}$ s.t. $S(m_i, m_j) = 1$. Because that for each vertex v_i we have: (1) $v_i \in V_d$; or (2) $v_i \in Neighbors(v_j)$ and $v_j \in V_d$, we can deduce that V_d is the result of the dominating set problem. As a result, the RS problem is NP-hard.

□

Because of the NP-hardness of the RS problem, it is impossible to develop an efficient exact algorithm to resolve the problem. Nevertheless, it is possible to propose a approximate algorithm with a bounded ratio. Now we present our proposed approximate algorithm for the RS problem with a proved approximate ratio.

Our high-level idea is inspired by a greedy algorithm for the SOS problem proposed by Guo et al. (Guo et al. 2018). However, they do not consider the message temporal proximity constraint. We generate the result set $\mathcal{R}$ by greedily and iteratively selecting spatio-temporal messages from a cluster of messages $\mathcal{C}$. In particular, for each iteration we select the spatio-temporal message m_n with the maximum increment of information coverage and add it into $\mathcal{R}$. Next, we remove the existing spatio-temporal messages in $\mathcal{R}$ whose distance to m_n is smaller than ζ. The algorithm ends when we the cardinality of $\mathcal{R}$ is equal to k. Specifically, the major technical challenge here is how to find the message with the maximum increment of information coverage. An straightforward method is to calculate the increment of information coverage by selecting each $m_n \in \mathcal{C} \setminus \mathcal{R}$. However, this method is computationally prohibitive. To address this challenge, we develop an effective pruning strategy to filter out unwanted candidates, which is defined by Lemma 1.

Lemma 1: Let $\mathcal{R}$ and $\mathcal{S}$ be two sets of spatio-temporal messages where $\mathcal{R} \subseteq \mathcal{S}$ and m_n be a new message. Assume that $\mathcal{R}'$ denotes $\mathcal{R}$ with m_n inserted and $\mathcal{S}'$ denotes $\mathcal{S}$ with m_n inserted, we have the following inequation:

$$I_C(\mathcal{R}',\mathcal{C}) - I_C(\mathcal{R},\mathcal{C}) \geq I_C(\mathcal{S}',\mathcal{C}) - I_C(\mathcal{S},\mathcal{C}).$$

Proof. Let's assume that $S(m_n, m_i) > max\{m_j \in \mathcal{R}|S(m_i, m_j)\}$ and $S(m_n, m_i) > max\{m_j \in \mathcal{S}|S(m_i, m_j)\}$. Because that $I_C(\mathcal{R},\mathcal{C}) \leq I_C(\mathcal{S},\mathcal{C})$, we have $S(m_n, m_i) - I_C(\mathcal{R},\mathcal{C}) \geq S(m_n, m_i) - I_C(\mathcal{S},\mathcal{C})$. Hence, we have $max\{m_j \in \mathcal{R}'|S(m_i, m_j)\} - max\{m_j \in \mathcal{R}|S(m_i, m_j)\} \geq max\{m_j \in \mathcal{R}'|S(m_i, m_j)\} - max\{m_j \in \mathcal{R}|S(m_i, m_j)\}$. Thus, we complete the proof. □

Based on Lemma 1, we find that the increment of information coverage when we insert m_n into $\mathcal{S}$ cannot be greater than the increment of information coverage when we insert m_n into $\mathcal{R}$. In other words, the increment of information coverage will decrease as we proceed to execute the iteration. Therefore, to prune unnecessary candidate messages we propose to use the pruning strategy that works as follows: For each message m, we record its increment of information coverage in each iteration by generating a triple entry $\{\varrho, \delta, \imath\}$. In particular, $m.\varrho$ denotes the pointer/id of m, $m.\delta$ indicates the increment of information coverage when we insert m_n into $\mathcal{R}$, and $m.\imath$ is the iteration count.

Algorithm 1: GreedyRegionSummary ($\mathcal{C}$, k, ζ, δ)

```
1  $\mathcal{R} \leftarrow$ empty;
2  $PQ \leftarrow$ empty;
3  for each $m_i \in \mathcal{C}$ do
4      $entry \leftarrow \{m_i, I_C(\{m_i\},\mathcal{C}), 0\}$;
5      $PQ$.Push($entry$);
6  while $PQ$ is not empty and $|\mathcal{R}| < k$ do
7      $cur \leftarrow PQ$.top();
8      $PQ$.pop();
9      while $cur.\imath \neq |\mathcal{R}|$ do
10         $cur.\delta \leftarrow$ SSInc($\mathcal{C}$,$\mathcal{R}$,$m_n$);
11         $cur.\imath \leftarrow |\mathcal{R}|$;
12         $PQ$.push($cur$);
13         $cur \leftarrow PQ$.top();
14         $PQ$.pop();
15     $\mathcal{R}$.add($cur.\varrho$);
16     for each $m_j$ in $PQ$ do
17     if $d(cur,m_j) < \zeta$ or $|cur.t_c - m_j.t_c| < \delta$ then
18         $PQ$.remove($m_j$);
19 return $\mathcal{R}$;
```

Algorithm 2 presents the corresponding pseudo code of our proposed greedy algorithm for the RS problem. First, we initialize the result set $\mathcal{R}$ and the priority queue PQ (lines 1–2). Next, for each spatio-temporal message m_i we generate its corresponding triple $\{m_i, I_C(\{m_i, \mathcal{C}), 0\}$ and push it into PQ (lines 3–5). Note that $I_C(\{m_i, \mathcal{C}), 0\}$ denotes the increment of information coverage when we insert m_i into the empty set $\mathcal{R}$. After that, we generate the cluster summary set in an iterative fashion (lines 6–18). Specifically, at the beginning of each iteration we fetch and pop the top triple from PQ (lines 7–8). If the increment of information coverage induced by the top triple cur has not been computed yet (line 9), we need to compute it (line 10), update the iteration count ($cur.\imath$) (line 11), and push it to PQ (line 12). Then we proceed to fetch and pop the top tripe (lines 13–14). If the increment of information coverage induced by the top triple cur has already been computed, we add the corresponding message into the result set (line 15). At the same time, we need to remove the existing messages in PQ whose distance to the current message is smaller than ζ or whose temporal proximity is smaller than δ (lines 16–18).

Experiments

We present the experiments with two datasets that offer insight into the effectiveness and efficiency of baselines and our proposed algorithms.

Experiment Setup

Dataset Our experiments are conducted on two real-life datasets: FS and TT. FS is a dataset collected from Foursquare, which contains 1.2 million POIs in North America with location information (i.e., latitude and longitude). The dataset TT is a larger dataset that contains 40 million geo-tagged tweets in the U.S.A. with geographical point locations. Each POI or geo-tagged tweet is regarded as a spatio-temporal message. We train an Online Latent Dirichlet Allocation (OLDA) (AlSumait, Barbará, and Domeniconi 2008) model using our datasets. Each message is associated with a topic distribution vector.

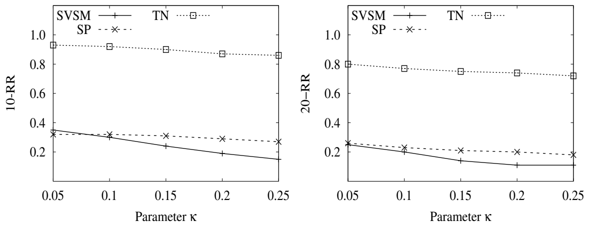
Baselines We evaluate the effectiveness and efficiency of our triplet network for learning relevancy metric (denoted by TN) and our greedy algorithm for region summarization (denoted by GRS) by comparing against the following baselines.

(1) SVSM (Sheng et al. 2010): SVSM can be applied for solving our problem. While computing the relevancy of two regions, we first derive the spatial feature vector for each region, which is calculated based on the message topics in the region, the message locations, and the average distance between messages and region reference points (i.e., vertices and center). Next, we computes the cosine similarity between their vectors.

(2) SP (Lazebnik, Schmid, and Ponce 2006): SP, which is designed for image categorization, can be used for solving our problem as well. SP indexes the underlying space by a Quad-tree that recursively splits the space into four cells. To compute the relevancy between two regions, we firstly calculate the average cell-wise similarities in each level. Next, we aggregate the average similarities in all levels.

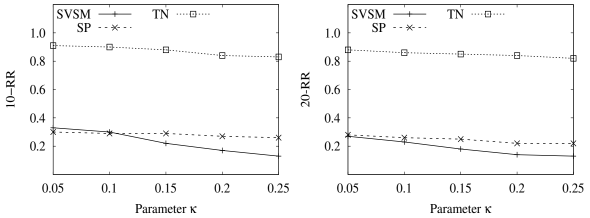
(3) Max-Sum, Max-Min (Drosou and Pitoura 2014): These two result diversification methods are widely used for selecting a set of most diverse messages. We compare GRS against Max-Sum and Max-Min. Note that the result set returned by the two methods may not satisfy our constraints defined in Definition 4.

(4) Rand: This method generates n representative messages of a region by randomly selecting a spatio-temporal



(a) FS (b) TT

Figure 3: Effect of κ (FS)



(a) FS (b) TT

Figure 4: Effect of κ (TT)

message m from the messages located in the region. If inserting m into the current result set will break the constraints in Definition 4, we discard m and continue selecting the next message; Otherwise, we insert m into the result set. Once the size of the result set reaches n, we return the set as the result.

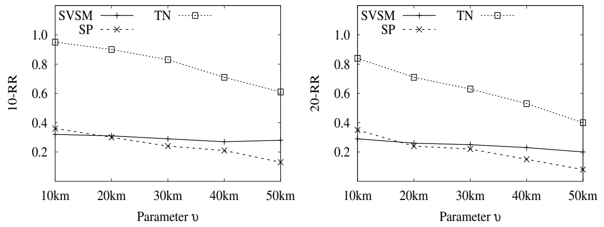
Settings For evaluating TN, we build a CNN that consists of 3 convolutional layers. The stride is set as 2×2. We use a ReLU non-linearity between two adjacent layers. The weight decay parameter λ is set to be 5×10^{-5}. The gap parameter g is 0.25. We train our model on 5000 generated instances. For generating each relevant instance, we randomly set the two noise indicators, namely κ and υ, between the ranges $(0.05, 0.35)$ and $[1km, 50km]$, respectively.

All of the algorithms are run in memory. We report the cpu time for efficiency evaluation and report "relevant ratio of top-k result (k-RR)" for efficacy evaluation. In particular, the k-RR measures the proportion of positive instances among the result set.

Experimental Result

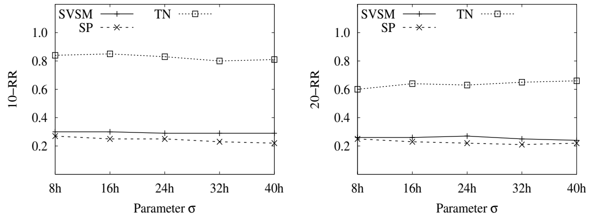
Effect of noise indicators In this set of experiments, we evaluate the effectiveness of SVSM, SP, and TN, by varying noise indicators κ from 0.05 to 0.25, υ from 10km to 50km, and σ from 8h to 40h.

Based on Figures 3 and 4, We can see that the TN performs substantially better than SVSM and SP on both datasets in terms of 10-RR and 20-RR when we vary κ. In addition, we find that TN is more robust to noise compared with the other two baselines. Specifically, when we increase κ from 0.05 to 0.25, the 10-RR and 20-RR are only slightly reduced by 10% and 7.8%, respectively. While for SP, the 10-RR and 20-RR are moderately reduced by 22% and 31%, respectively. SVSM produces the worst performance among the three methods, with 10-RR and 20-RR reduced by 53% and 39%, respectively. Furthermore, we also notice that all of the three methods produce better performances on dataset TT in comparision to FS. The reason is that the message density of TT is higher than that of FS.



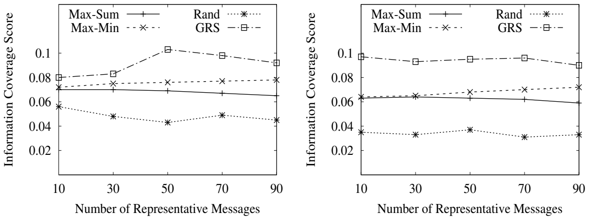
(a) FS (b) TT

Figure 5: Effect of υ



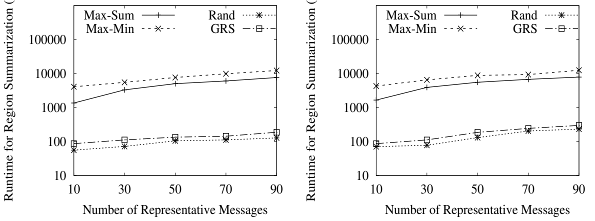
(a) FS (b) TT

Figure 6: Effect of σ



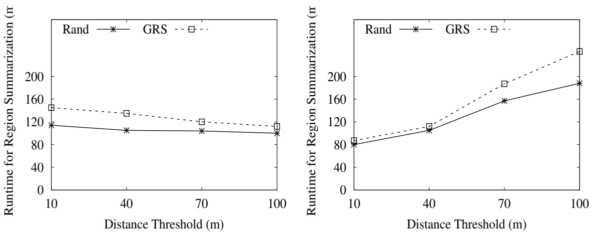
(a) FS (b) TT

Figure 7: Effect of # representative messages w.r.t. efficacy



(a) FS (b) TT

Figure 8: Effect of # representative messages w.r.t. efficiency



(a) FS

(b) TT

Figure 9: Effect of distance threshold

Figure 5 shows the effectiveness performances of the three methods as we vary the other noise indicator υ. TN still has the best performance. Additionally, we find that 10-RR and 20-RR significantly decrease as we increase υ. This is because that the increment of υ will change the relative locations of messages, which may lower the relevancy. We also notice that SVSM does not exhibit a performance decreasing trend as we increase υ. The reason is that the relevancy metric of SVSM does not take the relative locations of messages into account.

Effect of the number of representative messages In this set of experiments, we investigate the effectiveness and efficiency with regard to the cardinality of representative message set. From Figure 7, we can see that all of the four methods demonstrate a stable trend for information coverage score when we increase the size of representative message set. In particular, GRS produces the highest information coverage score. As for efficiency aspect (Figure 8), we can find that all the methods perform worse when we increase the number of representative messages in a region. The runtime performance of GRS is at least an order of magnitude better than Max-Min and Max-Sum, and it is only slightly worse than the random selection method (Rand).

Effect of distance threshold We proceed to evaluate the effect of the distance threshold (i.e., ζ) for region summarization algorithms. Figure 9 shows that when we increase the distance threshold for dataset FS, both GRS and Rand exhibit a slight decreasing trend regarding the runtime, while for dataset TT both methods exhibit a significant increasing trend. This can be explained by the fact that the density of messages in TT is much more higher than that in FS.

Related Work

Continuous query processing over spatio-temporal data streams

Our problem is relevant to the location-based publish/subscribe problem. Given a new spatio-temporal message m and a set of subscriptions, the location-based publich/subscribe problem aims at finding a subset of subscriptions whose spatial and textual predicates match m. Specifically, the subscriptions defined by some literature (Wang et al. 2015; Chen, Cong, and Cao 2013; Chen et al. 2014; Chen et al. 2017) require that m falls in a subscription region or m has overlapping area with the subscription region (Li et al. 2013). While for others, a score that measures the spatial proximity between the query location and the location of a new spatio-temporal message m (Hu et al. 2015a; Hu et al. 2015b; Chen et al. 2015; Chen and Shang 2018; Chen et al. 2018), or a score that measures the spatial overlap between a continuous query and the region of a new spatio-temporal message (Yu et al. 2015), is calculated for matching process.

However, existing studies on location-based publish/subscribe let subscriptions receive single-granularity items from data streams based on Boolean predicates or similarity function, which may not reflect the actual user preferences.

Online clustering and visualization algorithms for text and geo-text streams

Threshold-based incremental clustering algorithm (Allan et al. 1998) is commonly used for detecting and tracking news, bursty events, and trending topics over textual data streams (e.g., tweets) (Farzindar and Khreich 2015; Becker, Naaman, and Gravano 2011; Petrovic, Osborne, and Lavrenko 2010; Phuvipadawat and Murata 2010; Aggarwal and Yu 2006; Zhao, Chen, and Cong 2016; Tsur, Littman, and Rappoport 2013; Yin 2013). Specifically, given an existing set of clusters, a stream of textual items (i.e., messages), and a similarity threshold, the algorithm sequentially evaluates each new message and find a cluster that has the highest similarity score towards each message. However, if the similarities between a message and all existing clusters are less than the threshold, the message will form a new cluster. Compared to the other online text clustering algorithms (e.g., online k-means algorithm), the threshold-based incremental clustering algorithm often produces higher efficiency. Additionally, it guarantees a real-time clustering result, which is important for modern online social media apps (Zhang, Chan, and Tan 2014). Zhong (Zhong 2005) develops an online text clustering algorithm based on the classic *k-means* named the online spherical k-means (OSKM) algorithm. The key idea of OSKM algorithm is to partition a data stream into segments. Because that the size of each segment is relatively small, it can be processed efficiently in main memory. Nevertheless, the k-means based clustering algorithms require the number of clusters to be defined in advance, which is unreasonable for streaming data since the number of bursty events and trending topics is unknown. Furthermore, k-means based clustering algorithms are impossible to generate real-time results.

Conclusion

We develop a region-based message exploration mechanism that retrieve spatio-temporal message clusters from a stream of spatio-temporal messages based on users preferences on message topic and message spatial distribution. We also propose a region summarization algorithm that finds a subset of representative messages in a cluster to summarize the topics and the spatial attributes of messages in the cluster. Experimental results show that our proposal is capable of both the efficacy and efficiency.

References

Aggarwal, C. C., and Yu, P. S. 2006. A framework for clustering massive text and categorical data streams. In *SDM*, 479–483.

Allan, J.; Carbonell, J. G.; Doddington, G.; Yamron, J.; and Yang, Y. 1998. Topic detection and tracking pilot study final report.

AlSumait, L.; Barbará, D.; and Domeniconi, C. 2008. On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, 3–12.

Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*.

Chen, L., and Cong, G. 2015. Diversity-aware top-k publish/subscribe for text stream. In *SIGMOD*, 347–362.

Chen, L., and Shang, S. 2018. Approximate spatio-temporal top-k publish/subscribe. *World Wide Web* 1–23.

Chen, L.; Cong, G.; Jensen, C. S.; and Wu, D. 2013. Spatial keyword query processing: an experimental evaluation. In *PVLDB*, 217–228.

Chen, L.; Cui, Y.; Cong, G.; and Cao, X. 2014. SOPS: A system for efficient processing of spatial-keyword publish/subscribe. *PVLDB* 7(13):1601–1604.

Chen, L.; Cong, G.; Cao, X.; and Tan, K. 2015. Temporal spatial-keyword top-k publish/subscribe. In *ICDE*, 255–266.

Chen, Z.; Cong, G.; Zhang, Z.; Fu, T. Z. J.; and Chen, L. 2017. Distributed publish/subscribe query processing on the spatio-textual data stream. In *ICDE*, 1095–1106.

Chen, L.; Shang, S.; Zhang, Z.; Cao, X.; Jensen, C. S.; and Kalnis, P. 2018. Location-aware top-k term publish/subscribe. In *ICDE*, 749–760.

Chen, L.; Cong, G.; and Cao, X. 2013. An efficient query indexing mechanism for filtering geo-textual data. In *SIGMOD*, 749–760.

Drosou, M., and Pitoura, E. 2014. Diverse set selection over dynamic data. *IEEE Trans. Knowl. Data Eng.* 26(5):1102–1116.

Farzindar, A., and Khreich, W. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence* 31(1):132–164.

Guo, L.; Zhang, D.; Li, G.; Tan, K.; and Bao, Z. 2015. Location-aware pub/sub system: When continuous moving queries meet dynamic event streams. In *SIGMOD*, 843–857.

Guo, T.; Feng, K.; Cong, G.; and Bao, Z. 2018. Efficient selection of geospatial data on maps for interactive and visualized exploration. In *SIGMOD*, 567–582.

He, Q.; Chang, K.; Lim, E.; and Zhang, J. 2007. Bursty feature representation for clustering text streams. In *SDM*, 491–496.

Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *SIMBAD*, 84–92.

Hu, H.; Liu, Y.; Li, G.; Feng, J.; and Tan, K. 2015a. A location-aware publish/subscribe framework for parameterized spatio-textual subscriptions. In *ICDE*, 711–722.

Hu, J.; Cheng, R.; Wu, D.; and Jin, B. 2015b. Efficient top-k subscription matching for location-aware publish/subscribe. In *SSTD*, 333–351.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1106–1114.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2169–2178.

Li, G.; Wang, Y.; Wang, T.; and Feng, J. 2013. Location-aware publish/subscribe. In *KDD*, 802–810.

Liu, Y.; Zhao, K.; and Cong, G. 2018. Efficient similar region search with deep metric learning. In *KDD*, 1850–1859.

Ozsoy, M. G.; Onal, K. D.; and Altingovde, I. S. 2014. Result diversification for tweet search. In *WISE*, 78–89.

Petrovic, S.; Osborne, M.; and Lavrenko, V. 2010. Streaming first story detection with application to twitter. In *HLT-NAACL*, 181–189.

Phuvipadawat, S., and Murata, T. 2010. Breaking news detection and tracking in twitter. In *Web Intelligence/IAT Workshops*, 120–123.

Rocha-Junior, J. B.; Gkorgkas, O.; Jonassen, S.; and Nørvåg, K. 2011. Efficient processing of top-k spatial keyword queries. In *SSTD*, 205–222.

Sarma, A. D.; Lee, H.; Gonzalez, H.; Madhavan, J.; and Halevy, A. Y. 2012. Efficient spatial sampling of large geographical tables. In *SIGMOD*, 193–204.

Sheng, C.; Zheng, Y.; Hsu, W.; Lee, M.; and Xie, X. 2010. Answering top-*k* similar region queries. In *DASFAA*, 186–201.

Tobler, W. R. 1970. A computer movie simulating urban growth in the detroit region. In *Economic geography 46*, 234–240.

Tsur, O.; Littman, A.; and Rappoport, A. 2013. Efficient clustering of short messages into general domains. In *ICWSM*.

Wang, X.; Zhang, Y.; Zhang, W.; Lin, X.; and Wang, W. 2015. Ap-tree: Efficiently support continuous spatial-keyword queries over stream. In *ICDE*, 1107–1118.

Wang, C.; Lan, X.; and Zhang, X. 2017. How to train triplet networks with 100k identities? In *ICCV Workshops*, 1907–1915.

Yin, J. 2013. Clustering microtext streams for event identification. In *IJCNLP*, 719–725.

Yu, M.; Li, G.; Wang, T.; Feng, J.; and Gong, Z. 2015. Efficient filtering algorithms for location-aware publish/subscribe. *IEEE Trans. Knowl. Data Eng.* 27(4):950–963.

Zhang, D.; Chan, C.; and Tan, K. 2014. An efficient publish/subscribe index for ecommerce databases. *PVLDB* 7(8):613–624.

Zhao, K.; Chen, L.; and Cong, G. 2016. Topic exploration in spatio-temporal document collections. In *SIGMOD*, 985–998.

Zhong, S. 2005. Efficient streaming text clustering. *Neural Networks* 18(5-6):790–798.