2012 IEEE 12th International Conference on Data Mining Workshops

Geo-referenced Time-series Summarization Using k-Full Trees: A Summary of Results

Dev Oliver[*], Shashi Shekhar[*], James M. Kang[†], Renee Laubscher[†], Veronica Carlan[†], Michael R. Evans[*]
[*]*Department of Computer Science, University of Minnesota, Minnesota, USA*
Email: {oliver, shekhar, evans}@cs.umn.edu
[†] *National Geospatial-Intelligence Agency*
Email: {James.M.Kang, Renee.B.Laubscher, Veronica.M.Carlan}@nga.mil

***Abstract*—Given a set of regions with activity counts at each time instant (e.g., a listing of countries with number of mass protests or disease cases over time) and a spatial neighbor relation, geo-referenced time-series summarization (GTS) finds k-full trees that maximize activity coverage. GTS has important potential societal applications such as understanding the spread of political unrest, disease, crimes, fires, pollutants, etc. However, GTS is computationally challenging because (1) there are a large number of subsets of k-full trees due to the potential overlap of trees and (2) a region with no activity may be a part of a larger region with maximum activity coverage, making apriori-based pruning inapplicable. Previous approaches for spatio-temporal data mining detect anomalous or unusual areas and do not summarize activities. We propose a k-full tree (kFT) approach for GTS which features an algorithmic refinement for partitioning regions that leads to computational savings without affecting result quality. Experimental results show that our algorithmic refinement substantially reduces the computational cost. We also present a case study that shows the output of our approach on Arab Spring data.**

***Keywords*-Spatial Data Mining; Summarization; Geo-referenced Time-series; Full Trees**

I. Introduction

Geo-referenced or geographic Time-series (GTs) allow us to observe the evolution in time of some phenomenon in a fixed location, i.e., the time-changing value of an observed property [1]. Examples of GTs include the CIA World Factbook data for each year from 1989 to 2011 [2] and West Nile Virus cases for each of the 50 states in the US from 1999 to 2002. The ability to summarize geo-referenced time-series has potentially important applications for understanding the spread of protests, diseases, crimes, fires, pollutants, etc. For example, the geopolitical implications of the Arab Spring protests have drawn global attention [3, 4] and summarization of data related to these events is of interest to political geographers, peace and conflict researchers, economists, etc. In epidemiology, geo-refereced time series summarization may be useful for understanding the spread of disease so that patterns of progression can be established [5].

We define the problem of geo-refereced time series summarization (GTS) as follows: given a set of regions with activity counts at each time instant (e.g., a listing of countries with number of protests or disease cases over time), a spatial neighbor relation, and a positive integer k, find k trees that maximize activity coverage and impose a partition on the GTS. Section II presents formal definitions of the basic concepts and problem statement of GTS.

GTS is challenging for the following reasons. First, the computational complexity is high because of the large number of subsets of k trees due to the potential overlap between trees; there are an exponential number of choices when selecting k subsets from a set of trees. Second, a region with no activity may belong to a larger region with maximum activity coverage, making apriori-based pruning inapplicable. Determining appropriate interest measures for capturing event spread is important in capturing domain semantics. Examples of possible interest measures include maximum activity coverage and minimum average distance. Summarization based on maximizing activity coverage works well in domains such as epidemiology and geo-politics because it captures areas with the most activity (e.g., disease outbreak, protest). However, a region with maximum coverage might be composed of sub-regions with no activity which means that maximum activity coverage does not exhibit the anti-monotone property [6]. This makes pruning subsets based on maximum coverage more challenging since a region with no activity may be a part of a larger region with maximum activity coverage.

A. Data Summarization

Data summarization is an important concept in data mining that entails techniques for finding a compact description or representation of a dataset. The process typically involves defining a set of groups, finding a representative for each group, and reporting a statistic for each group (e.g., sum, mean, standard deviation). These notions differ depending on the genre of the data being summarized. Table I presents a summarization framework for four genres of data. Summarization methods listed for three of the genres already exist. An example of the first, relational table summarization, is the GROUP BY clause in SQL that is used to group rows having common values to report SQL aggregation functions such as mean and standard deviation. The group definition in this case is a partition of rows and the group representation is distinct values of attributes such as age-groups, citizenship, income-group, etc.

978-0-7695-4925-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDMW.2012.64

Table I
SUMMARIZATION FRAMEWORK FOR VARIOUS DATA GENRES

Data Genre	Group Definition (Partitioning Criteria)	Group Representation Choices	Statistic
Relational Table (a set of rows)	a partition of rows	Distinct values of attributes (e.g., age-group)	sum, count, mean, etc.
Spatial (Euclidean Space)	a partition of space	points, polygons, ellipses, line-strings	sum, count, mean, etc.
Spatial Graph (Neighbor Relationship)	a partition of a graph	node, path, tree, MST, heaviest full tree	sum, count, mean, etc.
Geo-referenced Time-series (Fixed Neighbor)	a partition of a geo-referenced time-series	ST-node, ST-path, ST-heaviest full tree	sum, count, mean, etc.

Spatial Euclidean summarization includes heat maps and hotspot analysis. Heat maps provide a graphical representation of data in which individual values contained in a matrix are represented as colors. The group definition for heat maps might include a set of pixels and the group representation may be a subset of these pixels. Hotspots are a special kind of partitioned pattern where objects in hotspot regions have high similarity in comparison to one another and are dissimilar to all the objects outside the hotspot [7]. These spatial summarizations are based on spatial point locations where the group definition is a partition of space and the groups could be represented by points, polygons, ellipses, or line-strings. An example of a spatial summarization is given in Figure 1(a) where incidents of crime in a major US city are shown as dots and the spatial (Euclidean) summarization is represented using ten ellipses. The spatial summarization technique used in this case was K-Means [8].

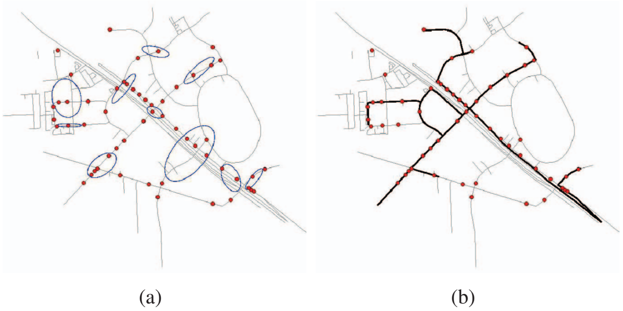


Figure 1. An example of (a) spatial summarization (ellipses) and (b) spatial graph summarization (paths).

Spatial graph summarization defines groups based on a partition of a graph and represents groups using nodes, paths, trees, etc. For example, the thicker lines in Figure 1(b) illustrate a summarization technique that uses linear representatives or paths to represent each partition [9]. Here ten paths are used to summarize the incidents of crime (dots) that are on the spatial graph (i.e., the road network).

Summarization of the last data genre listed in the table, geo-referenced time-series summarization, has been largely unexplored until now. It defines groups based on a partition of a geo-referenced time series and may represent groups using spatio-temporal (ST)-nodes, ST-paths, or ST-heaviest full trees as is proposed in this work.

Finding a representative for each group and reporting a statistic for each group gives only a partial picture of the group. In many cases they give an effective summary of the group but not in all cases. This may lead to issues such as Simpson's paradox [10]. Thus one should examine groups (partitions) in addition to reviewing the statistics and group representatives for a complete story.

B. Related Work and their Limitations

Geo-referenced time-series summarization may be thought of as either anomaly-based or summarization-based. In anomaly-based approaches, anomalous hotspots or unusual regions are flagged. Anomaly-based approaches include the spatial and space-time (ST) scan statistic [11, 12, 13], emerging partition detection [14], irregular partition discovery [15], and STPC [16]. The spatial scan statistic uses a scanning window that varies its center and radius in order to scan the region under study when detecting spatial partitions. Emerging partition detection extends this idea to determine whether an observed increase in cases in a region is significant. Irregular partition discovery focuses on detecting partitions with irregular shapes where geographical boundary information is used to construct a graph in which a partition growing process is performed based on likelihood function maximization. STPC is a density-based technique for partitioning spatial and temporal polygons. Each polygon is required to have a minimum number of polygons in its neighborhood. Anomaly-based approaches do not summarize geo-referenced time-series. Instead they detect anomalous or unusual areas.

Summarization-based approaches, on the other hand, detect popular areas and aim to cover as many activities as possible.

C. Contributions

To the best of our knowledge, we are the first to propose techniques to summarize geo-referenced time-series. To address the challenges of the problem of GTS we propose the k-full tree (kFT) algorithm. In summary, our contributions are as follows:

- We formulate the problem of summarizing Geo-referenced Time Series (GTS) using k-full trees
- We introduce a basic algorithm for the GTS problem. For reducing computational costs, we also introduce an algorithmic refinement called voronoi partition assignment (VPA) for partitioning regions.

- Experiments on real and synthetic datasets show that kFT with VPA leads to computational savings without affecting result quality.
- We present a case study on real data showing the output of our approach on Arab Spring data.

D. Scope and Outline

In this paper, we use spatio-temporal full trees as representatives for partitions of GTS; alternate representations such as directed acyclic graphs or paths are not pursued in the present research. In a spatio-temporal full tree, every node other than the leaves has the same number of children and every leaf has the same depth. The trees are descriptive in nature and predictive notions such as causality are not explored.

The rest of this paper is organized as follows: Section II presents the basic concepts and problem statement of GTS. Our proposed kFT algorithm and its algorithmic refinement are detailed in Section III. Section IV outlines the experimental evaluation. Section V presents a case study that shows the output of our approach on Arab Spring data. Section VI concludes the paper and previews future work.

II. Basic Concepts and Problem Statement

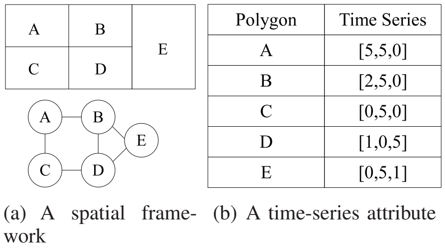
In this section, we introduce several key concepts in geo-referenced time-series summarization (GTS), and give a formal problem statement.

A. Basic Concepts

Relevant definitions to our problem statement and proposed approaches are as follows:

A **spatial framework, S**, is a partition of a region of geographic space, forming a finite tesselation of spatial objects [17]. In the plane, the elements of a spatial framework will be polygons. A **spatial neighbor relation, SN** $\subseteq S \times S$ may be defined by the sharing of boundaries.

An example of a spatial framework are country boundaries and an example of spatial neighbors as defined by SN are neighboring countries such as the United States and Canada. For the spatial framework shown in Figure 2(a), polygons A and B are spatial neighbors while polygons C and E are not.



Polygon	Time Series
A	[5,5,0]
B	[2,5,0]
C	[0,5,0]
D	[1,0,5]
E	[0,5,1]

(a) A spatial framework (b) A time-series attribute

Figure 2. A geo-referenced time-series.

A **temporal framework, T**, is a partition of a time interval into sub-intervals. A **time-series** is a computable function from T to a finite attribute domain, A_i. A **temporal neighbor relation, TN** $\subseteq T \times T$ and a time interval, t_i **immediately-follows** t_j iff $TN(t_i, t_j)$ and $t_i = t_j + 1$.

Months in a year is an example of a temporal framework where a calendar year is partitioned into months January, February, March, etc. January and February are temporal neighbors and February immediately follows January. Figure 2(b) shows another example of a temporal framework for each polygon, where $T \rightarrow Integers$. Polygon A has the time-series $[5, 5, 0]$ where each integer in the time-series may represent number of protests or disease cases.

A **Geo-referenced Time-series, GT** = $S \times T$. A **spatio-temporal node** in GT is denoted $\langle s_i, t_i \rangle$. A **spatio-temporal directed neighbor relationship, STN($\langle s_i, t_i \rangle$, $\langle s_j, t_j \rangle$)** if and only if s_i and s_j are spatial neighbors and t_i immediately-follows t_j. s_i's value at t_i is the **activity count,** $c_{i,j}$, of s_i at t_i.

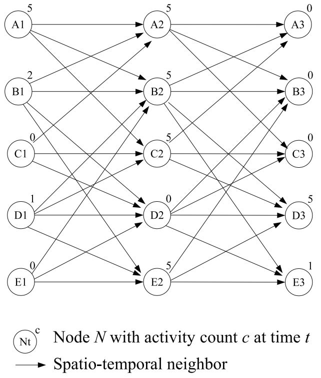


Figure 3. Spatio-temporal directed neighbor relationships.

Figure 2 shows an example of a GT where the spatial framework comprising five regions (nodes), is shown on the left and the temporal framework for each polygon is shown on the right. Figure 3 shows a graph representation of Figure 2 with all the spatio-temporal directed neighbor relationships. Spatio-temporal nodes $A1$ and $B2$ are spatio-temporal directed neighbors because they are spatial neighbors and t_2 immediately follows t_1.

A **spatio-temporal (ST)-full tree,** f is a tree of degree d and depth h such that

- $f \subseteq STN$
- non-leaves have degree d
- every leaf has the same depth h.

Examples of spatio-temporal (ST) full trees are shown in Figure 4. $\langle A1, A2, B2, C2 \rangle$ is a spatio-temporal full tree with degree 3.

A **Spatio-Temporal Field, STF** is a function from GT to a finite attribute domain, A_i.

Examples of STF include $GT \rightarrow Integers$, which denotes activity counts, such as number of disease cases and $GT \rightarrow Boolean$, which denotes event presence or absence such as whether there were protests in a country at a given time during the Arab Spring revolutions.

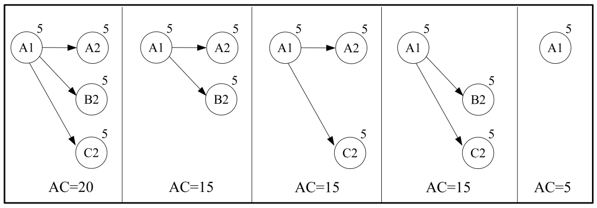


Figure 4. Examples of spatio-temporal (ST) full trees from Figure 3 rooted at $A1$.

The **activity coverage (AC)** of a ST-full tree, f, is the sum of activity counts, $\sum c_{i,j}$, for all spatio-temporal nodes $n \in f$.

In Figure 4, $AC(A1, A2, B2, C2)$ is 20 because the activity count for each of the four nodes in the full tree is 5, i.e., $AC(A1) + AC(A2) + AC(B2) + AC(C2) = 5 + 5 + 5 + 5 = 20$.

A **summary tree set** F is a collection of summary trees where each tree $f \in F$ is a full tree. A **summary tree** imposes a partitioning on a set of spatio-temporal nodes, N, such that $distance(n, f) \leq distance(n, f')\ \forall f' \in F$, $\forall n \in N$. The height of a summary tree is bounded by the number of time instants in N.

Figure 5(b) shows a summary tree $\langle A1, A2, B2, C2 \rangle$ that imposes a partitioning on spatio-temporal nodes $A1$, $A2$, $A3$, $B1$, $B2$, $B3$, $C1$, $C2$, and $C3$.

B. Problem Formulation and Statement

The problem of geo-referenced time-series summarization (GTS) can be expressed as follows:

Given:
- A geo-referenced time-series
- A spatial neighbor relation, R
- A positive integer, k

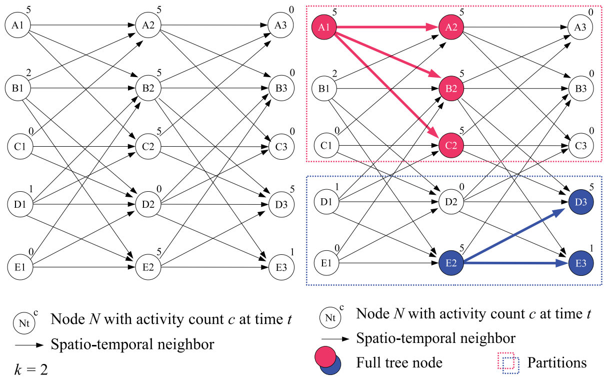
Find:
- k trees
- A partitioning of spatio-temporal nodes

Objective:
- Maximize the activity coverage (AC) across k trees

Constraints:
- Each tree is a spatio-temporal full tree and represents one source rooted at spatio-temporal node n
- Trees are mutually non-overlapping

The geo-referenced time-series input for GTS is defined in Section II-A, R defines the spatial relationship between the regions of the geo-referenced time-series, and k represents the desired number of trees. The output for GTS is a set of k trees that maximize activity coverage and impose a partition on the GTS. The constraints are that the k trees are spatio-temporal full trees (Section II-A), represent one source, and are mutually non-overlapping.

Example. Figure 5 shows input and output examples of GTS. For the input, the GTS and spatial neighbor relation from Figure 2 is used. Figure 5(a) adds all the spatio-temporal edges for each node and the number of trees k is set to 2. The output in Figure 5(b) shows 2 full trees that maximize activity coverage.



(a) GTS Input (b) GTS Output

Figure 5. Example (a) input and (b) output of GTS (Best in color).

III. Proposed Approach

This section describes the k-full tree (kFT) algorithm and our algorithmic refinement, voronoi partition assignment (VPA).

A. Basic k-Full Tree (kFT) Algorithm

Algorithm 1 presents the pseudocode for kFT, whose inputs are a geo-referenced time-series, a spatial neighbor relation, and a positive integer k, and whose outputs is a set of k full trees that maximize activity coverage and impose a partition on the GTS. The basic structure of kFT resembles that of K-Means [8] in terms of selecting initial seeds, forming k partitions and updating the representative of each partition until the assignments no longer change. Line 1 of Algorithm 1 randomly selects k full trees as initial summary trees, which are the "seeds" for kFT. The algorithm then proceeds in two main phases. First, it forms k partitions by assigning each spatio-temporal node to its closest summary tree, i.e., the assignment step (line 3). Second, it calculates the summary tree of each partition that maximizes activity coverage, i.e., the update step (line 4). These two phases are repeated until the summary trees no longer change and the final summary trees and partitions are returned.

1) Phase 1: Form k partitions by assigning each ST node to its closest summary tree: In phase 1, each spatio-temporal node, $n \in N$, is assigned to each of the given k summary trees, $f_i \in F$, to form k partitions. The assignment of n to a partition is based on the spatio-temporal distance from n to each spatio-temporal node u_i in f_i. The distance between n and f_i is the minimum distance between n and each $u_i \in f_i$. The distance between n and every f_i's is calculated and n is assigned to the closest summary tree. A basic method for phase 1 calculates a different shortest path between each spatio-temporal node n and all k trees

Algorithm 1 Basic k-Full Tree (kFT) Algorithm

```
Input:
    1) A geo-referenced time-series,
    2) A spatial neighbor relation, $R$,
    3) A positive integer, $k$
Output:
    $k$ full trees that maximize activity coverage and impose
    a partition on the GTS
Algorithm:
Select $k$ full trees as initial summary trees
repeat
    Phase 1: Form $k$ partitions by assigning each ST
        node to its closest summary tree
    Phase 2: Recompute the summary tree of each
        partition
until summary trees do not change
```

for all $n \in N$. Section III-C describes the voronoi partition assignment algorithm (VPA) that improves the runtime of phase 1.

2) Phase 2: Recompute the summary tree of each partition: Phase 2 computes and returns the summary tree with maximum activity coverage for each partition. All possible full trees that are rooted at each spatio-temporal node for each degree are calculated and the summary tree, f_{max}, with the highest activity coverage is stored. f_{max} is compared with the summary tree of the current partition and if it is found to have higher coverage, then the summary tree of the current partition is updated to f_{max}; otherwise the summary tree remains unchanged. A basic method for phase 2 enumerates all possible full trees within each partition and selects the one with maximum activity coverage.

B. Execution Trace

Figure 6 provides an execution trace of kFT. The input includes (1) a GTS with five nodes and three time instants (Figure 6(a)), (2) the spatial neighbor relationships shown in Figure 2, and (3) $k = 2$. kFT starts by selecting two trees as initial summary trees, namely $\langle A1 \rangle$ and $\langle E1 \rangle$ (Figure 6(b)). Next, two partitions are formed by assigning each spatio-temporal node to the nearest summary tree. The partitions formed are shown in Figure 6(c) and include nodes $A1, A2, A3, B1, B2, B3, C1, C2, C3$ being assigned to $\langle A1 \rangle$, and nodes $D1, D2, D3, E1, E2, E3$ being assigned to $\langle E1 \rangle$. For example, $A2$ is assigned to $\langle A1 \rangle$ since its distance to $\langle A1 \rangle$ is 1 and its distance to $\langle E1 \rangle$ is 3.

Within each partition, kFT selects the full tree with the maximum activity coverage to be the new summary tree (Figure 6(d)). If the current summary tree has coverage greater than or equal to the tree with maximum coverage, then the summary tree remains unchanged; otherwise, the summary tree is updated. In this case, summary tree $\langle A1 \rangle$ changes to $\langle A1, A2, B2, C3 \rangle$, and summary tree $\langle E1 \rangle$ changes to $\langle E2, E3, D3 \rangle$. Figure 4 shows the possible summary trees rooted at $A1$ and their respective activity coverages. Every possible summary tree for each spatio-temporal node is considered.

Phases 1 and 2 of kFT, which are depicted in Figures 6(c) and 6(d), are repeated until the summary trees do not change. In this example, phases 1 and 2 are repeated once and since the summary trees do not change, the algorithm terminates, returning the partitions shown in Figure 6(d).

C. Refinement 1: Voronoi Partition Assignment

VPA aims to improve the runtime of phase 1, which forms k partitions by assigning ST nodes to full trees. The pseudocode for VPA is presented in Algorithm 2. The basic idea of VPA is to perform one distance calculation between the nodes of the summary trees and every other spatio-temporal node versus performing multiple calculations as described earlier. Figure 7 shows an example of VPA. A virtual node V is connected to each node of the summary trees $\langle A1 \rangle$ and $\langle E1 \rangle$. $A2$ is assigned to $\langle A1 \rangle$ since $A1$ is a node on the shortest path from V to $A2$, i.e., $V, A1, A2$, whereas $E2$ is assigned to $\langle E1 \rangle$ since $E1$ is a node on the shortest path from V to $E2$, i.e., $V, E1, E2$.

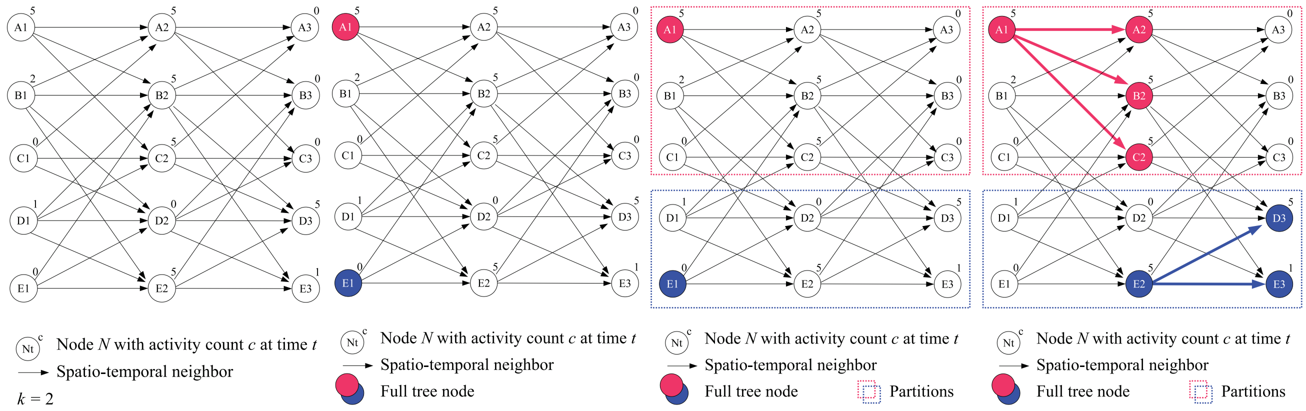
Algorithm 2 Voronoi Partition Assignment

```
Input:
    A set of $k$ summary trees
Output:
    A set of $k$ partitions formed by assigning each spatio-
    temporal node, $n$, to one summary tree $f \in F$, where
    $distance(n, f) \leq distance(n, f') \ \forall f' \in F, \ \forall n \in N$.
Algorithm:
$Open \leftarrow$ all nodes of all $f_i \in F$
$Closed \leftarrow \emptyset$
$T_{nodes} \leftarrow$ all nodes of all $f_i \in F$
repeat
    $n \leftarrow$ node in $Open$ that is nearest to a summary tree
    $Closed \leftarrow n$
    for each $x_i \in Adj[n]$ do
        if $x_i \notin Closed$ then
            update $x_i.prev, x_i.distance, x_i.f$ in $T_{nodes}$
            assign $x_i$ to the closest summary tree based
                on $T_{nodes}$
            if $x_i \notin Open$ then
                $Open \leftarrow x_i$
until all nodes $\in Closed$
```

IV. Experimental Evaluation

We experimentally evaluated kFT with and without the VPA algorithmic refinement on both synthetic and real-world data. For each workload experiment, we ran two versions of kFT:

- Basic kFT (kft)
- kFT with VPA (kft_v)



(a) GTS Input

(b) Select k initial full trees

(c) Assign ST nodes to trees

(d) Compute new trees

Figure 6. kFT Execution Trace (Best in color).

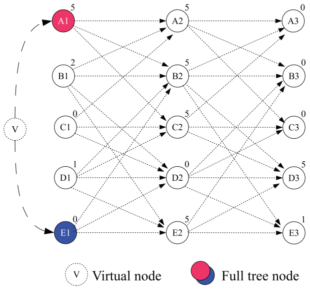


Figure 7. Voronoi Partition Assignment Example. $A2$ is assigned to $\langle A1 \rangle$ since $A1$ is a node on the shortest path from V to $A1$, i.e., $V, A1, A2$ (Best in color).

The goal of our experiments was a comparative analysis in which we investigated:

- How do candidate algorithms compare on computational cost? Subquestions included the following:
  - What is the effect of number of nodes?
  - What is the effect of number of time-intervals?
  - What is the effect of number of trees, k?

All experiments were performed on a Mac Pro with a 2 x Xeon Quad Core 2.26 GHz processor and 16 GB RAM.

A. Experiment Data Sets

Our experiments were performed on both synthetic and real-world data. To generate our synthetic geo-referenced time-series, we used the US Census Bureau's 2011 TIGER/Line Shapefiles for Hennepin County, Minnesota [18]; the nodes were the intersections of the road network. For each node, activity counts for each time instant were set to a number between 0 and 100.

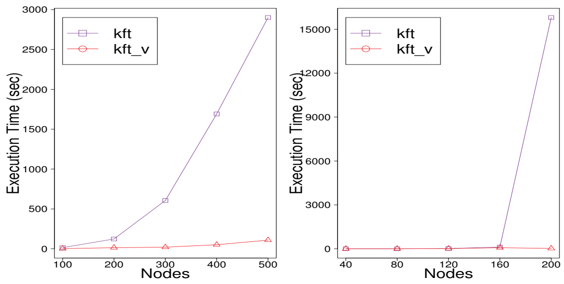
The real-world data set we used was from the CIA World Factbook, which provides information on the history, people, government, economy, geography, communications, transportation, military, and transnational issues for 267 world entities [2]. This factbook is an excellent source of geo-referenced time-series because for each country we are given its spatial location and time-changing values of many quantitative variables of interest. For our experiments, we used the unemployment rate variable for each country. The spatial relationships for each country were derived using the sharing of boundaries.

Understanding the human terrain of culture, politics, and economics of a region is of global importance and datasets such as the CIA World Factbook contain a wealth of information that could be explored or summarized analytically. New methods to explore this data can potentially deepen our understanding of world events such as the Arab Spring [4].

B. Experimental Results

Our experimental results are as follows:

What is the effect of number of nodes? Both synthetic and real-world data sets were used to observe the effect of increasing the number of nodes on execution time. For the synthetic data, the number of time instants was set to 3 and k was set to 2. For the real-world data, the number of time instants was also set to 4 and k was also set to 2. Figure 8 gives the execution times for both data sets. As can be seen, computational savings increases as the number of nodes increases due to VPA.

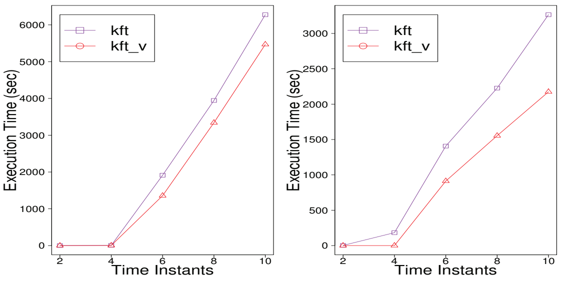


(a)

(b)

Figure 8. Effect of number of nodes on (a) synthetic and (b) real data

What is the effect of number of time instants? Again, both synthetic and real-world data sets were used to observe the effect of increasing the number of time instants on execution time. For the synthetic data, the number of nodes was set to 50 and k was set to 2. For the real-world data, the number of nodes was also set to 50 and k was also set to 2. Figure 9 gives the execution times for both data sets. As before, computational savings increase when the VPA algorithmic refinement is introduced.

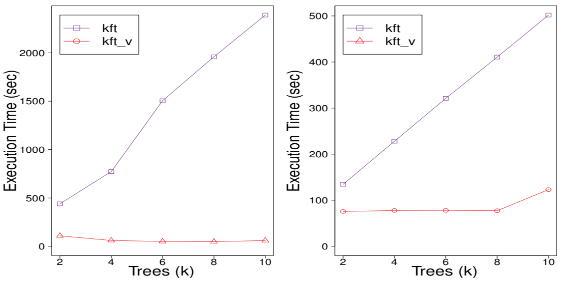


(a)

(b)

Figure 9. Effect of number of time instants on (a) synthetic and (b) real data

What is the effect of number of trees, k? The effect of increasing the number of trees on execution time was observed using both synthetic and real-world data sets. For the synthetic data, the number of nodes was set to 500 and the number of time instants was set to 3. For the real-world data, the number of nodes was set to 100 and the number of time instants was set to 3. Figure 10 gives the execution times for both data sets. As can be seen, computational savings increase when VPA is introduced.



(a)

(b)

Figure 10. Effect of number of trees on (a) synthetic and (b) real data

Interpretation: The results of the series of experiments are uniformly that kft with voronoi partition assignment clearly gives the best results, much better than kft alone.

V. Case Study

We also conducted a qualitative evaluation of kFT that illustrates its output in applications involving geo-referenced time-series. Figure 11 presents the results of our case study on Arab Spring data [3].

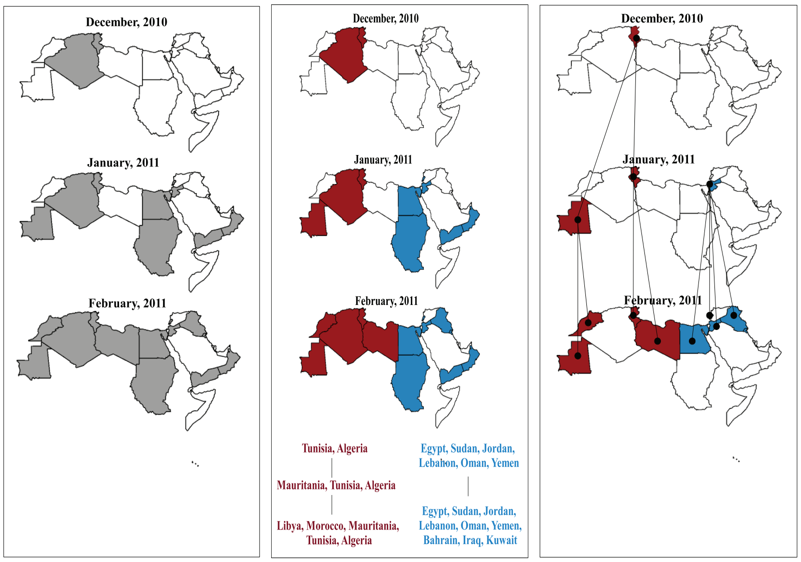
The input is shown in Figure 11(a) and entails the geo-referenced time-series comprised of countries of the Arab League with boolean values for each country indicating whether there were protests from December 2010 to February 2011 [3]. The activity count for each country was set to 1 if protests occurred during the month (0 otherwise) and k was set to 2. December's data for each country is shown on top, January in the middle, and February on the bottom. The spatial relationships were derived using a distance threshold.

The output of kFT includes a partitioning of spatio-temporal nodes shown in Figure 11(b). The two groups or partitions that kFT produces are represented by different colors. For example, the maroon partition (partition 1) has members (Tunisia, Algeria) - (Mauritania, Tunisia, Algeria) - (Libya, Morocco, Mauritania, Tunisia, Algeria) and the blue partition has members (Egypt, Sudan, Jordan, Lebanon, Oman, Yemen) - (Egypt, Sudan, Jordan, Lebanon, Oman, Yemen, Bahrain, Iraq, Kuwait). The 2 trees or representatives for the partitions are shown in Figure 11(c), which is a sparser representation of the groups in Figure 11(b). For example, the maroon partition shows more members in Figure 11(b) but the representative in Figure 11(c) is a full tree. The kFT algorithm provides groups and group representatives of the actual protest data and the output shows how geo-referenced time-series may be summarized using this approach.

The lines in Figure 11(c) give examples of how the components are connected. As discussed, the goal of GTS is to summarize in a descriptive manner. GTS is not predictive and notions such as causality are not explored. Additionally, the temporal resolution of the data was a month. For example, although protests in Tunisia started on December 18, 2010 and protests in Algeria started on December 28, 2010, both are recorded as having protests in December 2010, with no distinction as to which started first based on the granularity of a month. Finer temporal resolutions (e.g., weekly) and accounting for activity frequencies may provide a better summarization and we plan to evaluate this in the future. We also plan to get user feedback as to their preference of groups (Figure 11(b)) versus representatives (Figure 11(c)) and we will be evaluating kFT on other real data sets such as GDP in CIA world factbook data [2].

VI. Conclusions

This work introduced the problem of geo-referenced time-series summarization (GTS) using k full trees. This problem is important for summarizing the spread of events over space and time such as outbreaks of disease or the Arab Spring uprisings. However, this problem is computationally challenging because there are a large number of subsets of k-full trees due to the potential overlap of trees. We proposed a k full tree (kFT) algorithm to solve GTS. kFT is novel because it finds popular areas (i.e., full trees) with maximum coverage rather than anomalous or unusual areas. kFT uses Voronoi Partition Assignment to speed up summary tree


December, 2010
January, 2011
February, 2011
December, 2010
January, 2011
February, 2011
Tunisia, Algeria
Mauritania, Tunisia, Algeria
Libya, Morocco, Mauritania, Tunisia, Algeria
Egypt, Sudan, Jordan, Lebanon, Oman, Yemen
Egypt, Sudan, Jordan, Lebanon, Oman, Yemen, Bahrain, Iraq, Kuwait
December, 2010
January, 2011
February, 2011

(a) Actual Protests

(b) Groups ($k = 2$)

(c) Group Representation - k full trees ($k = 2$)

Figure 11. kFT output on Arab Spring data from December 2010 to February 2011 [3] (Best viewed in color).

calculation without reducing activity coverage. kFT was validated using experimental evaluation on real and synthetic data and a case study on Arab Spring data. Experiments demonstrate the computational savings of running kFT with VPA.

In future work, we plan to develop an algebraic cost model and investigate alternate interest measures (e.g., distance, density, etc). We plan to explore other partition representatives such as directed acyclic graphs. We also plan on refining our approach to attain more effective summaries by accounting for finer temporal resolutions and frequencies. Choosing the right k and determining kFT's usefulness to domain professionals will also be explored.

Acknowledgment

This work has been supported by the following grants: NSF III-CXT IIS-0713214, NSF IGERT DGE-0504195, NSF CRI:IAD CNS-0708604, USDOD HM1582-08-1-0017, USDOD HM1582-07-1-2035, and the University of Minnesota. We would like to thank Kim Koffolt and the members of the University of Minnesota Spatial Database and Spatial Data Mining Research Group for their comments.

References

[1] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo, "Spatio-temporal clustering," *Data mining and knowledge discovery handbook*, pp. 855–874, 2010.

[2] CIA World Factbook, https://www.cia.gov/library/publications/the-world-factbook/.

[3] Arab Spring, en.wikipedia.org/wiki/Arab_Spring/.

[4] Foreign Affairs, *The New Arab Revolt: What Happened, What It Means, and What Comes Next*. Council on Foreign Relations, 2011.

[5] Disease Surveillance, http://en.wikipedia.org/wiki/Disease_surveillance/.

[6] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[7] S. Shekhar, M. Evans, J. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 193–214, 2011.

[8] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 281-297. California, USA, 1967, p. 14.

[9] D. Oliver, A. Bannur, J. M. Kang, S. Shekhar, and R. Bousselaire, "A K-Main Routes Approach to Spatial Network Activity Summarization: A Summary of Results," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, 2010, pp. 265–272.

[10] Simpson's paradox, http://en.wikipedia.org/wiki/Simpson's_paradox.

[11] M. Kulldorff, "A spatial scan statistic," *Communications in statistics-theory and methods*, vol. 26, no. 6, pp. 1481–1496, 1997.

[12] M. Kulldorff, W. Athas, E. Feurer, B. Miller, and C. Key, "Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico." *American Journal of Public Health*, vol. 88, no. 9, pp. 1377–1380, 1998.

[13] M. Kulldorff, R. Heffernan, J. Hartman, R. Assunção, and F. Mostashari, "A space–time permutation scan statistic for disease outbreak detection," *PLoS medicine*, vol. 2, no. 3, p. e59, 2005.

[14] D. Neill, A. Moore, M. Sabhnani, and K. Daniel, "Detection of emerging space-time clusters," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 218–227.

[15] M. Costa, M. Kulldorff, and R. Assuncao, "A space time permutation scan statistic with irregular shape for disease outbreak detection," *Advances in Disease Surveillance*, vol. 4, p. 243, 2007.

[16] D. Joshi, "Polygonal spatial clustering," Ph.D. dissertation, University of Nebraska, 2011.

[17] M. Worboys and M. Duckham, *GIS: A computing perspective*. CRC, 2004.

[18] US Census Bureau 2011 Census TIGER/Line Shapefiles, http://www.census.gov/geo/www/tiger/tgrshp2011/tgrshp2011.html/.