FluMapper: An Interactive CyberGIS Environment for Massive Location-based Social Media Data Analysis

Anand Padmanabhan\textsuperscript{1,4}, Shaowen Wang\textsuperscript{1,2,3,4}, Guofeng Cao\textsuperscript{1}, MyungHwa Hwang\textsuperscript{1}, Yanli Zhao\textsuperscript{1}, Zhenhua Zhang\textsuperscript{1,5}, Yizhao Gao\textsuperscript{1}, Kiumars Soltani\textsuperscript{1}

\textsuperscript{1}CyberInfrastructure and Geospatial Information Laboratory (CIGI) Department of Geography and Geographic Information Science School of Earth, Society and Environment

\textsuperscript{2}Department of Computer Science

\textsuperscript{3}Department of Urban and Regional Planning

\textsuperscript{4}National Center for Supercomputing Applications (NCSA) University of Illinois at Urbana-Champaign

\textsuperscript{5}Computer Network Information Center, Chinese Academy of Sciences
Outline

• Motivations
• Background
• FluMapper
  o Spatial Data Handling
  o Spatial Data Analysis
• Demo
• Conclusions and Future work
Motivations

• **Current influenza like illnesses (ILI) surveillances focus on a certain spatiotemporal scale, usually at a state (or city) level and are typically updated weekly**
  - Early detection and progression ILI are important in public health research
  - By exploiting social media data we can do much better

• **Recent studies with significant societal impacts have been conducted with social networking and media data**
  - E.g. predictions of the stock market (Bollen et al. 2011), disaster responses (Goodchild and Glennon 2010) to infectious disease tracking (Signorini and Segre 2011, Wang et al. 2013)

• **Dramatic increase in the volume of spatial data available via social media**
  - 2-4% of the 340M daily tweets sent via Twitter have GPS information available
  - This number is growing with increase in usage of location-aware mobile devices
Background

- Conventional approaches to disease surveillance use indicators like daily visits to physician and hospital visits, and laboratory-test requests
  - Though possibly more accurate, this type of information is difficult to collect and hence there is a potential time lag between an outbreak and when it gets reported

- Alternative approaches to disease surveillance like search (Ginsberg et al. 2008), online news and documents (Keller et al. 2009) and location-based social media (LBSMD) data (Wang et al. 2013) have been investigated
Using LBSMD

**Advantages**
- Rich information about various social themes and captures the dynamic process of how humans interact over time and space (Tsou et al. 2013)
- Unlike traditional data, like from surveys and clinical reports, that produce retrospective information and are cost prohibitive, LBSMD can be collected in near real-time with much less cost (Signorini et al. 2011)
- Individual-level spatial and temporal observations allow exploratory analysis of spatiotemporal dynamics of disease processes on varying scales (Wang et al. 2013)
- Far less restriction on use of LBSMD compared to clinical or survey data

**Challenges**
- Data sizes quickly grow beyond the capabilities of conventional geographic information systems (GIS) (Wang et al. 2013)
- Data are in unstructured textual form so intensive data modeling and processing is required to transform LBSMD into structured spatial data entities (e.g., point patterns and trajectories) on which spatiotemporal analysis could be applied (Li et al. 2013)
- Sampled natured of LBSMD data raises questions of quality and statistical validity
CyberGIS Software Ecosystem

- CyberGIS Social Media Analytics
- CyberGIS Gateway and Applications
- CyberGIS Toolkit
- GISolve Middleware

Spatially Explicit Agent-Based Modeling of Disease Spread

Viewshed Analysis

FluMapper

Cyberinfrastructure and Geospatial Information Laboratory
CyberGIS Gateway

- Online collaborative geospatial problem solving environment
- Enables easy access to CyberGIS analytics and data sources
- Provides transparent access to a rich set of cyberinfrastructure environments
- Represents a broad approach to CyberGIS
  - Widely accessible
FluMapper Goals

• Provide early ILI activity detection to public health researchers and practitioners
• Efficiently manage the large volumes of LBSMD
• Capture the dynamics of flu risk across multiple spatiotemporal scales
  o Provide ILI risk maps from national scale to very fine local scales
• Identify mobility patterns of population
  o Aggregate individual user trajectory across different spatial and temporal scales to generate flows
  o Capture travel patterns across the country, regional, state, city and local levels
• Facilitate interactive and exploratory analysis of LBSMD
  o Update ILI activity in near real-time for the conterminous United States
  o Provide an integrated view of flu risk indicators and mobility of users
FluMapper Components

• **Data collection and processing**
  - Collects, processes and stores data from *Twitter* in near real time
  - Efficient and scalable services to query raw and derived data developed

• **Spatiotemporal data model**
  - Provides aggregated data and statistics at multiple scales for efficient spatial query
  - At the finest scale, the conterminous United States is represented as a uniform grid with cell size of 30 arc seconds

• **Exploratory spatial data analysis**
  - Kernel density estimation (KDE), a spatial cluster detection technique, is used to detect unusual concentration of flu events
  - Monte-Carlo simulations with random labeling are conducted to improve our confidence in analysis results

• **Flow mapping analysis**
  - Single-source flow mapping is applied for mapping the travel pattern of population from or to a specific area
Overall Architecture

---

After (Wang et al. 2013)
Data Collection and Processing

• Provides near real-time access to data from Twitter
  o Data from Twitter collected using streaming application program interface (API)

• Transforms tweets into structured spatial data entities

• Facilitates multiscale exploratory spatiotemporal analytical methods of flu risks and the related population movements
  o Interactive analysis uses a 7-day sliding window
  o Provides spatiotemporal surveillance of flu by processing tweets continuously and aggregating summary flow data in a hierarchical multi-scale structure

• Provides scalable service interface to access the massive data stream
  o MongoDB, a scalable and open source NoSQL database used
  o Set of distributed databases for managing spatial data entities of tweets and flows
Data Pipeline Architecture

1. Crawler
2. Tweet Importer
3. Cube Generator
4. Data Filter Service

Tweets Cache

Twitter's Streaming API

Tweets DB
Space-time Cube DB

[Diagram showing the flow of data through the pipeline.]
Data Pipeline Components

- **Crawler**
  - Use Twitter’s streaming API
  - ~2.5M tweets daily (Data has been collected for a year)
  - A spatial filter (no keyword filter) of North America is applied when data is collected

- **Tweet importer**
  - Reads in raw tweets and converts them into spatial events
  - Uses text mining techniques to identify flu related tweets (Signorini et al. 2011)

- **Cube generator**
  - Generates trajectories of individual users
  - Aggregates these trajectories into flows at multiple spatiotemporal scale
  - Cube consisted of ten layers of spatial grids, and at the finest level contained a spatial grid of $3072 \times 7168$ cells of about 30 arc-seconds ($\approx 1$ km) width

- **Data filter service**
  - Provides spatio-temporal query capabilities to the flu and flow databases
Data Characteristics

Daily volumes of tweets
(May 23 ~ June 5, 2013)

Spatial distribution of Twitter users
(May 31~ June 6, 2013)
Spatiotemporal Data Model

• Spatiotemporal data cube is developed to support efficient query of aggregated statistics
  o Data cube decomposes the spatiotemporal space into a lattice of multi-scale, hierarchical cuboids
  o Provides multiple scales of spatial indices for fast and efficient spatial query
    • Given a region, statistics such as the number of flu occurrences, and number of trajectories traveling out and in from this region are efficiently retrieved
Data in Spatiotemporal Cube

- Specific measures/statistics of geo-locate tweets are defined and computed for each cuboid
- For each cuboid at multiple scales, the following measures are computed recursively:
  - Number of twitters
  - Number of tweets
  - Number of flu tweets
  - Number of trips traveling out of a cuboid
  - Number of trips traveling to the cuboid
- For each pair of cuboids at multiple scales, the following measures are computed recursively:
  - Number of trips traveling from a cuboid to the other.
  - Number of flu related trips traveling from a cuboid to the other
- At the finest scale, the conterminous United States is represented as a uniform grid with cell size of 30 arc seconds
An Example Spatiotemporal Data Cube

Sizes of a collection of spatiotemporal data cube DB (May 23 ~ June 5, 2013)
Exploratory Spatial Data Analysis

• **Input**
  - Individual tweets in the Twitter feeds are indexed by both space and time
  - Flu-related tweets represent infection case, while all tweets together form the background population

• **Result**
  - Spatiotemporal maps exhibiting flu risk and unusual concentration of flu events

• **Technique**
  - Kernel density estimation (KDE), an exploratory spatial data analysis technique to identify spatial clusters, applied to detect concentration of flu events (Shi. 2010)
  - Adaptive bandwidth is applied to account for inhomogeneous background population
  - Monte-Carlo simulations with random labeling are conducted to improve our confidence in analysis results
  - Scale is a crucial factor that influences exploratory spatial data analysis results and so we conduct our analysis at multiple scales to discover spatial patterns
  - Code developed to run on GPUs (Graphical Processing Units) and executed on XSEDE
Adaptive Kernel Density Estimation

- Estimate risk surface by using a kernel whose bandwidth is adapted to constant population points (Shi 2010)

- Calculate adaptive bandwidth of each pixel

- Estimate risk surface using KDE estimator

- Monte Carlo simulation

- $O(NMk)$
  - $N$ - the number of cells in risk surface
  - $M$ - the number of points
  - $k$ - the $k$th neighbor to be searched

- Computationally intensive
  - Spatially sort points into tiles
  - Search neighboring tiles
  - Partial sort tiles to find $k$ neighbors

- $k$ nearest neighbor search (KNN search)
Scalability Challenges and Spatial Strategies

- **Challenges**
  - Load imbalance
  - Intensive memory access

- **Spatial Strategies**
  - Coarse-scale spatial computational domain decomposition
  - Fine-scale spatial data domain decomposition
Addressing Load Imbalance

**Problem**
- Traditional approach to domain decomposition is based on spatial data domain without considering computation
  - Causes load imbalance

**Solution**
- Construct spatial computational domain (Wang et al. 2009)
  - Perform coarse-scale domain decomposition on spatial computational domain
- Decomposition on spatial computational domain, guided by computational intensity
  - Properties of data and operations in spatial domain transformed to computational intensity in spatial computational domain
  - A transform function defined for a particular spatial analysis
  - Account for heterogeneity of underlying computing architecture
- Group decomposed domain into parallel tasks
  - Achieve homogeneity between these parallel tasks using the computational intensity values
Coarse-scale Decomposition for KDE

- **Transform Function for KDE**
  - As the number/density of points in a cell and its intermediate neighbors decreases, the computation intensity of a task associated with that cell increases.

- **Coarse decomposition uses adaptive quadtree and applies transform function to estimate computational intensity**
  - Using a space-filling curve tiles are grouped together to form parallel tasks.
  - The goal is to minimize the difference in computational intensity values among the parallel tasks.

![Computational intensity value](image)
Evaluating Performance of Coarse-scale Spatial Computational Domain Decomposition

<table>
<thead>
<tr>
<th>Task number</th>
<th>Regular decomposition</th>
<th>Our strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11 12 15 16</td>
<td>14 14 16 16</td>
</tr>
<tr>
<td></td>
<td>9 10 13 14</td>
<td>13 13 15 15</td>
</tr>
<tr>
<td></td>
<td>3 4 7 8</td>
<td>5 5 10 10</td>
</tr>
<tr>
<td></td>
<td>1 2 5 6</td>
<td>3 4 4 6</td>
</tr>
</tbody>
</table>

k=500, Keeneland cluster

<table>
<thead>
<tr>
<th>Task</th>
<th>16GPUs (regular)</th>
<th>16GPUs (our strategy)</th>
<th>1GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>168sec</td>
<td>65sec</td>
<td>890sec</td>
</tr>
<tr>
<td>Task1</td>
<td>166sec</td>
<td>43sec</td>
<td></td>
</tr>
<tr>
<td>Task2</td>
<td>30sec</td>
<td>40sec</td>
<td></td>
</tr>
<tr>
<td>Task3</td>
<td>50sec</td>
<td>32sec</td>
<td></td>
</tr>
<tr>
<td>Task4</td>
<td>13sec</td>
<td>49sec</td>
<td></td>
</tr>
<tr>
<td>Task5</td>
<td>26sec</td>
<td>63sec</td>
<td></td>
</tr>
<tr>
<td>Task6</td>
<td>103sec</td>
<td>40sec</td>
<td></td>
</tr>
<tr>
<td>Task7</td>
<td>11sec</td>
<td>41sec</td>
<td></td>
</tr>
<tr>
<td>Task8</td>
<td>86sec</td>
<td>14sec</td>
<td></td>
</tr>
<tr>
<td>Task9</td>
<td>26sec</td>
<td>36sec</td>
<td></td>
</tr>
<tr>
<td>Task10</td>
<td>13sec</td>
<td>15sec</td>
<td></td>
</tr>
<tr>
<td>Task11</td>
<td>21sec</td>
<td>39sec</td>
<td></td>
</tr>
<tr>
<td>Task12</td>
<td>7sec</td>
<td>45sec</td>
<td></td>
</tr>
<tr>
<td>Task13</td>
<td>13sec</td>
<td>39sec</td>
<td></td>
</tr>
<tr>
<td>Task14</td>
<td>35sec</td>
<td>28sec</td>
<td></td>
</tr>
<tr>
<td>Task15</td>
<td>13sec</td>
<td>48sec</td>
<td></td>
</tr>
<tr>
<td>Task16</td>
<td>30sec</td>
<td>43sec</td>
<td></td>
</tr>
</tbody>
</table>
Addressing Intensive Memory Access

- **Problem**
  - Many threads directly access to long-latency global memory

- **Solution**
  - Fine-scale spatial domain decomposition applied
    - Exploits spatial properties
    - Spatial locality - data could be shared among neighbors
    - Spatial dependency - dependent data can be identified by scope of spatial operations, e.g., neighbor scope
  - Decompose space into fine-scale tiles
    - If tile size is too large, the it may involve many irrelevant data
    - If tile size is too small, then it may reduce data reuse
    - Tile size is also constraint by hardware resources
    - Tile size is usually tuned experimentally
  - Spatially sort data into tiles
  - Group memory access of parallel threads in a tile by coalescing them into a thread block
  - Search, pre-fetch and cache dependent data of a thread block into shared memory
Fine-scale Decomposition for KDE

A thread block

Search dependent data

Threads directly access to fast shared memory while computation and reuse data

Approach shifts access from slow global memory to fast shared memory
Evaluating Performance of Fine-scale Spatial Data Domain Decomposition

Tile size

- no decomposition
- 2km*2km
- 4km*4km
- 8km*8km
- 12km*12km
- 16km*16km
- 20km*20km
- 24km*24km
- 32km*32km

Computing time (seconds)

- k=500, keeneland cluster, 1GPUs
Flow Mapping Analysis

- **Movement data visualization and analysis method**
  - Each edge represents a movement (flow) between a pair of geographical regions and thickness of each edge is corresponding to the flow magnitude on this edge

- **Flow mapping in FluMapper**
  - Single-source flow mapping is applied for mapping the travel pattern of population from or to a specific area
  - Relies on spatiotemporal data cube model
    - Individual trajectories are identified from Twitter feeds
      - Each trajectory provides the spatiotemporal footprint of a Twitter user
    - Trajectories are aggregated by the spatiotemporal data cube at multiple scales
  - Supports visualization multiple scales of Twitter flu users mobility
  - Flow information could help contextualize disease spread within and between regions
Flow Mapping: Challenges and Solutions

• Visualizing flows contained in large volumes of LBSMD data are both compute- and data-intensive
  o CyberGIS environment resolves the computational challenges

• Visualizing flow between a large number of origin-destination can lead to cluttered display
  o Spiral tree-based algorithm adopted to minimize visual clutter (Verbeek et al. 2011)
    • Properties of spiral tree avoid intersections

• Remaining Problems
  o Does not consider geography of people movement
    • People usually move on the roads, or by airplane
  o Difficult to fully capture the regional activity
Flow Mapping – Considering Geography

- **Consider ground and air based movements separately**
  - Ground movement uses routing service
  - Air routes use spiral-tree approach

- **Solutions**
  - Online Routing Services
    - Google Direction API
    - Mapquest Direction API with OSM data
    - Drawbacks
      - Call times and frequency restriction
      - Performance challenges owing to throttling, connection stability and competition with other applications
  - Locally deployed routing service
    - OSRM (Open Source Routing Machine)
      - [https://github.com/DennisOSRM/Project-OSRM/wiki](https://github.com/DennisOSRM/Project-OSRM/wiki)
      - Uses OSM (Open Street Map) data
    - Advantages
      - No restriction
      - Configurable and adaptable
  - Locally deployed routing service was chosen
## Performance Evaluation

<table>
<thead>
<tr>
<th>Source Place</th>
<th>MapQuest(ms) (Posting requests with multi-threads in Client side)</th>
<th>Local OSRM(ms) (Posting requests with multi-threads in Client side)</th>
<th>Local OSRM(ms) (with multi-threads in server side)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texas</td>
<td>3000</td>
<td>113</td>
<td>883</td>
</tr>
<tr>
<td>Florida</td>
<td>8000</td>
<td>2150</td>
<td>1615</td>
</tr>
<tr>
<td>Kansas</td>
<td>6000</td>
<td>2331</td>
<td>1286</td>
</tr>
<tr>
<td>New York</td>
<td>10000</td>
<td>2551</td>
<td>1735</td>
</tr>
<tr>
<td>California</td>
<td>22000</td>
<td>1763</td>
<td>1365</td>
</tr>
<tr>
<td>Chicago</td>
<td>10000</td>
<td>7644</td>
<td>5724</td>
</tr>
</tbody>
</table>
Flow Mapping – M to N

- Visualize all the activities (all source to all target) in the observation window
- Tackles large data size, around 500k points at finest level
- Applies visualization strategies to reduce clutter
- Major components
  - Backend: (a) reads data from MongoDB database; (b) transforms data into a graph format; and (c) identifies important nodes in each region
    - Computationally intensive step – hence it is pre-computed
  - Sevlet: (a) processes request from our web frontend; (b) queries the result from the backend processing; (c) runs Force Directed Edge Bundling algorithm on the query output to produce a visually desirable result
    - This component processes request in real time and produces a weighted directed graph
  - Web frontend: Use weighted graph returned by the servlet to categorize graph edges and color them
Demo

http://flumapper.org/
Summary

• Providing early ILI activity detection to public health researchers and practitioners is crucial
  o LBSMD are well suited to provide insights into this issue

• FluMapper handles big data effectively
  o Spatiotemporal data model supports efficient spatial queries
  o Data processing pipeline handle large volumes of LBSMD

• FluMapper provides interactive spatial analysis on LBSMD
  o Kernel density estimation is used to identify unusual risk patterns
  o Flow mapping provides additional exploratory tool to contextualize disease spread patterns

• CyberGIS provides a suitable environment for resolving computational challenges of FluMapper
  • CyberGIS integrates computation, data and communication technologies from XSEDE to resolve computational challenges
Future Work

• Expand the spatial extent of FluMapper to North America
  ○ This will test the scalability of the data pipeline and query efficiency

• Studies to compare FluMapper results with those from CDC and Google Flu Trends are underway

• Improve multi-source to multi-destination flow mapping performance
  ○ Map-reduce approach being investigated
Acknowledgements

• National Science Foundation
  o BCS-0846655
  o OCI-1047916
  o OCI-0503697
  o TeraGrid/XSEDE SES070004

• Colleagues
  o http://www.cigi.illinois.edu/doku.php/people/index
Thank You

• Questions?