GEODA WEB – ENHANCING WEB-BASED MAPPING WITH SPATIAL ANALYTICS

Xun Li, Luc Anselin and Julia Koschinsky

GeoDa Center for Geospatial Analysis and Computation Arizona State University
WHAT IS GEODA-WEB?

• A cloud mapping platform that integrates spatial analytics and web mapping
  – builds on strengths of GeoDa Center programs
    • interactive analysis through linked statistical and spatial data views
    • specialized spatial functionality like cluster maps and spatial regression
    • user-friendly and GUI-based
  – connects these spatial tools to maps and non-spatial analytics
    • functionality of GeoDa-Web can be extended through APIs
  – scales to larger datasets with potential to analyze data streams
  – delivered in flexible formats
    • access through browser on desktop, tablet or mobile phone
    • called from within Python or R
    • publish results seamlessly through social media
WHY GEODA-WEB?

• Big Data
• Open Data
• Social Media
SOFTWARE ARCHITECTURE

• cloud-to-cloud solution
• applies the latest web technologies
  – HTML5 Canvas, Local Storage, Web Sockets
• integrates various cloud-based software services and application programming interfaces (API)
  – access to data, analytical functions, mapping and social media
GeoDa-Web.js

GeoDa-Web

PySAL REST API

Cloud Mapping Services
- Cloud SpatialDB
- Cloud Routing
- Cloud Geocoding

GeoDa-Web Cloud

Server

Client

Browser

Terminals

System Design

Web Socket
ILLUSTRATIONS: BROWSER ACCESS

1. Fast local Moran cluster maps
2. Spatial regression
3. LISA clusters for points on street networks
1. Fast Local Moran Cluster Maps with Google Map Data

Precinct map of hot and cold spots of car thefts linked to scatter plot matrix of different crime types
Identifying clusters and hot spots of events

GeoDa Web

CartoDB

- request data via SQL API
- send back data (zip, csv)
- create and draw maps/plots
- request add new column in table
- request spatial aggregation
- create map/plots using data in new column
- create new column if not exists
- run spatial aggregation SQL
GeoDa Web

- request spatial weights creation
- request variable data
- return weights file and variable data (csv file)
- using weights and variables to do Moran test and LISA map
- upload LISA results as a new column in CartoDB table
- create and publish cartocss vizjson
- publish results in CartoDB

CartoDB

- serve viz.json
- request tiles
- serve tiles

Tumblr
2. Spatial Regression: Assessing spatial statistical relationships between multiple factors

Cluster map of car thefts with spatial regression results: Crimes vs. liquor stores
Assessing spatial statistical relationships between multiple factors
3. LISA clusters for points on street networks

Road map of accident hotspots linked to scatterplot of built environment characteristics
Snapping points to roads & running LISA

GeoDa Web

- request new column for road snapping results
- Request snapping points to roads
- request road weights creation

CartoDB

- create new column if not exists
- run spatial aggregation SQL
- create spatial weights for roads
- create cartoCSS vizJSON

PySAL

- serve viz.json
- serve tiles

Tumblr

- spatial regression
- using weights and variables to do Moran test and LISA map
- create new column if not exists
- run spatial aggregation SQL
- create spatial weights for roads
ILLUSTRATION: PYTHON + R ACCESS

1. Python
   (R under development)
Python (or R) integrated with GeoDa-Web through Javascript
Python (PySAL) Integrated with GeoDa-Web

```
In [ ]: import pysal

In [ ]: shp = pysal.open(pysal.examples.get_path('NAT.shp'), 'r')
dbf = pysal.open(pysal.examples.get_path('NAT.dbf'), 'r')

In [ ]: import d3viz

---

There is no dependent library, no extra command lines.
Just call setup() to start a websocket server and a http server,
and open a main web page.

Note: if websocket/http server has already been started,
this function will close these servers first.

---

In [ ]: d3viz.setup()

---

show a base map of NAT.shp.
Note: A geojson file will be created using the shapefile and dbf file.
The web page will use this geojson file to show map.

---

d3viz.show_map(shp)

---

User can also call select() function in Python to trigger selection on the map.
For example:

```
select_polygons = [i for i, v in enumerate(dbf.by_col['HC60']) if v <= 0.001]
d3viz.select(shp, ids=select_ids)
```

---

User can create and show a quantile map in a pop-up web page.
Note: the quantile computation is done using pysal.esda.mapclassify.Quantile

---

Quantile map for variable [HR90], k=5

---

Scatter plot matrix for variables [HR90, UE90]

---

Change the bandwidth of fitted LOESS curve: 0.4
```
FUTURE DIRECTIONS

• Launch on robust production platform
• Refine functionality
• Implement open API
• Share URL of linked graphs and maps
Mapping & Statistics Programs: Desktop & Cloud

Mapping
- ArcGIS
- qGIS
- ...

Statistics
- R Packages
- Python (PySAL)
- ...

Desktop

Cloud
- CartoDB
- MapZen
- MapBox
- Google Maps
- MapSense
- ArcGIS Online
- ...

- D3 Stats Viz
- GeoTurf
- PySAL Rest API
- ...

R Packages
- Python (PySAL)
- ...

Cloud
Integration Examples: GeoDa

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>Cloud</td>
</tr>
</tbody>
</table>

Cloud
Integration Examples: ggmap, sparkR...
Integration Examples: GeoDa-Web

Mapping

Statistics

Desktop

Cloud