Implementation & Application of Origin-Destination Flow Data Smoothing & Mapping

BDSS IGERT Team, Cohort 2014-2016
Pennsylvania State University
State College, PA 16801

Abstract

An implementation of the visualization algorithm developed in [1] is presented. This procedure is then applied to an extended dataset of migration flows ranging from the years of 1978 to 2011. Preliminary results, primarily in the form of visualizations, are presented. We also discuss several next steps in improving or using this algorithm for visualization of flow maps.

1 Overview

Often, visualization of time-based network data is difficult, and it is only getting more difficult as data-sets grow larger in size given improvements in sensing technology. Data collected regarding locations across time can prove to be invaluable, allowing for studies of important phenomena that can be viewed as networks of various kinds, ranging from time-varying social networks [3] to temporal protein networks that govern organ development [2].

In particular, geographic mobility data is one domain where the particular challenge of visualization is clearly present, where massive bundles of visually over-lapping connections are present and map space with which to display network/connection information is limited. When data is plentiful, flow maps, which are useful for observing directed pathways between origin and destination locations, often suffer from the "curse of density" (as we coin the problem). In this case, massive intersections and overlaps, grounded by fixed-point locations (necessary for context) yield a cluttered and uninformative, if not utterly incomprehensible, map-image.

The problem of visualization (for flow maps), however, has seen significant previous study, and, although still a challenging issue to address in modern day analytic problems via computational methods, can be decomposed into three objectives: (1) the cluttering Problem, (2) the modifiable area unit problem, and (3) the normalization problem. The first problem entails the issue of information loss, where bundling or re-routing approaches often simplify connections that lead to a difficult-to-interpret map render or require heavy user interaction (i.e., "human-in-the-loop"). The second problem summarizes the challenge of aggregation–arbitrary aggregation (at a non-informed scale) can lead to a degradation of spatial resolution or generate incorrect or incomplete patterns. The third and final problem deals with fundamental units–selecting a reasonable atomic geographic unit, such as the "county", means selecting a unit that varies widely in either size (i.e., population) or area (i.e., geographic coverage) creates invalid comparisons of flows/connections. The goal of any method designed to generate an interpretable visualization of a flow-map must be able to solve the above three problems while still "faithfully" preserving the critical, important flow patterns and regularities within the complex data-driven network.

While the above problem above certainly requires further research efforts to develop efficient, scalable methods for flow-map visualization, fortunately, [1] recently proposed a novel approach that attempts to address all the three critical objectives of flow-map visualization. The approach essentially combines a kernel-density estimation method with generalization method for removing spurious variance in the data and smooth and normalize the flows to a controllable neighborhood size. More importantly, the method is able to preserve and bring out the key “high-level” patterns in the data, useful for facilitating exploratory analysis.

We decided to implement the approach in [1] and test it on actual IGERT-held data (from previous hackathon events). In Section 3 we will present some basic, preliminary results depicting the capability of the model.

2 Implementation of the Flow-Map Visualization Algorithm

In this section, we describe the original algorithm we chose to implement (in the R statistical programming language).

2.1 Method Design

As mentioned earlier, we implemented the approach developed by [1] to extract (hard-to-detect) underlying patterns in large-scale geographic mobility data to generate informative visualizations of such data. These visualizations are ultimately useful in developing a meaningful understanding of complex systems and their essential space-time dynamics, such as complicated flow trends or migration patterns of freshly graduated college students to areas of work.

3 Experimental Results

We used our implementation of the approach in [1] to generate a visualization of the large-scale County Migration data-set, which contains 1702436 samples, each with dimensionality equal to 18 (i.e., 18 covariates or features). This data is composed of samples collected across the time-range of 1978 to 2011 and contains features including ID’s, string names, and geocodes for both origin and destination points of each flow (among other useful statistics that the visualization algorithm would not make use of, such as number of tax filings).

The algorithm was run on a 40-core machine, using Parallel R, resulting in an approximate total run-time of 2 hours. Scripts (using scripting languages such as JavaScript) were written to generate the actual visualizations that appear in this paper from the flow algorithm’s output. The code, scripts, and preliminary program output (which serves a sort of sample of the program’s capability) are available at the following link to GitHub:

```
https://github.com/flinder/mig_flows
```

.

Figure 1 depicts a holistic view of the flow-map generated by our implementation while Figure 2 shows zoomed insets that displays the finer details of areas of interest in the map.

4 Conclusions

Let’s see what happens....hopefully something publishable!

Note, the self-referential link for this file can be found at

```
https://www.overleaf.com/2678227dnjnvj#/7100270/
```

where those who wish to edit this document may do so from the WriteLatex inferace AGO set up...

References

[1] Guo, D., and Zhu, X. Origin-destination flow data smoothing and mapping. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2043–2052.

[2] Lage, K., Mllgrd, K., Greenway, S., Wakimoto, H., Gorham, J. M., Workman, C. T., Bendsen, E., Hansen, N. T., Rigina, O., Roque, F. S., Wiese, C., Christoffels, V. M., Roberts, A. E., Smoot, L. B., Pu, W. T., Donahoe, P. K., Tommerup, N., Brunak, S., Seidman, C. E.,

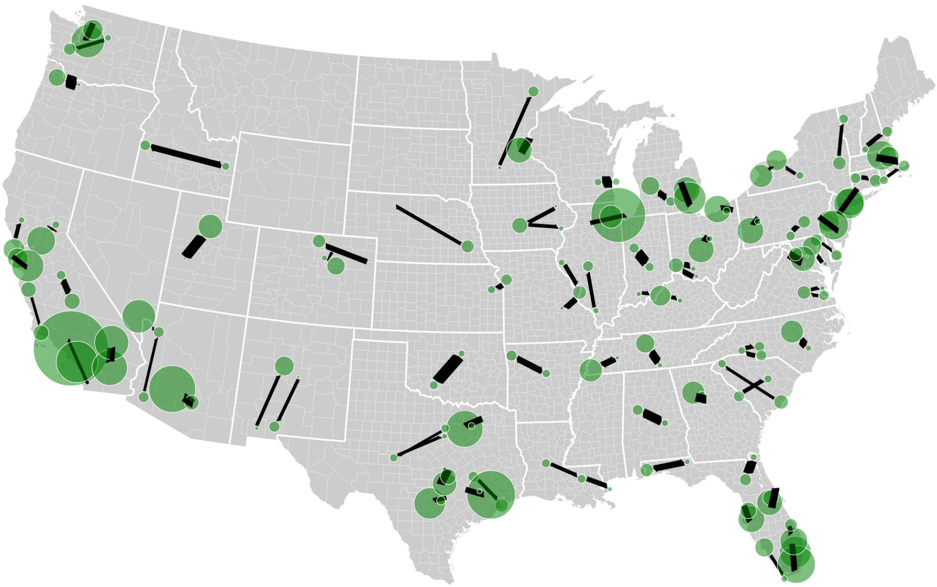
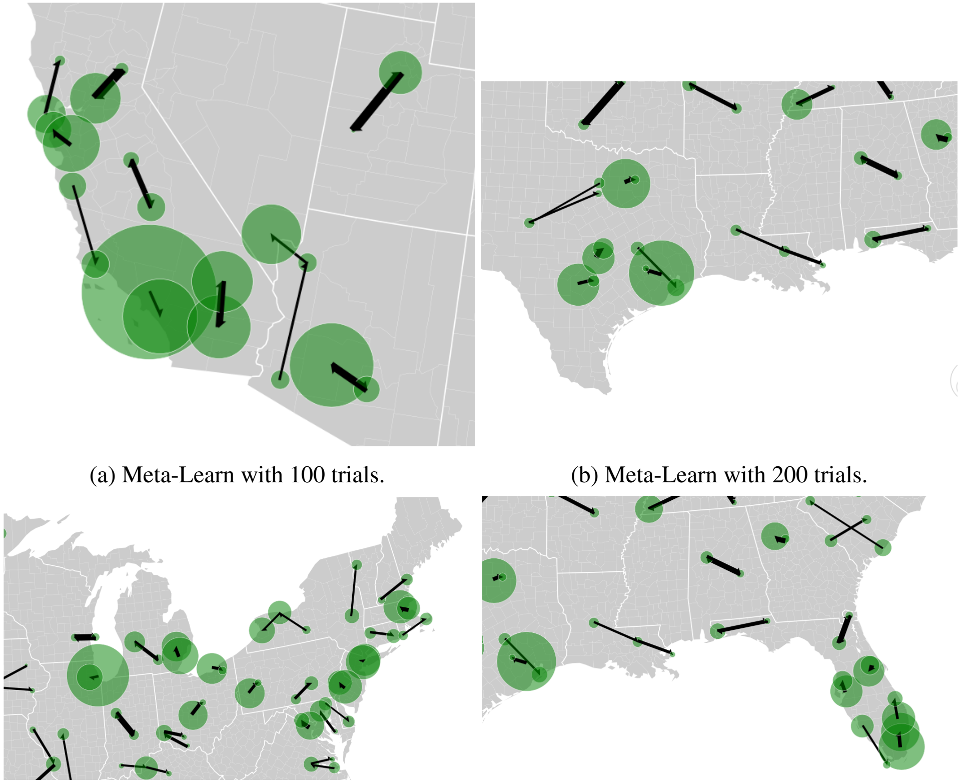


Figure 1: A flow-map generated from our implementation of the visualization algorithm.

SEIDMAN, J. G., AND LARSEN, L. A. Dissecting spatio-temporal protein networks driving human heart development and related disorders. *Molecular Systems Biology 6*, 1 (2010).

[3] SANTORO, N., QUATTROCIOCCHI, W., FLOCCHINI, P., CASTEIGTS, A., AND AMBLARD, F. Time-varying graphs and social network analysis: Temporal indicators and metrics. *arXiv:1102.0629 [physics]* (2011).



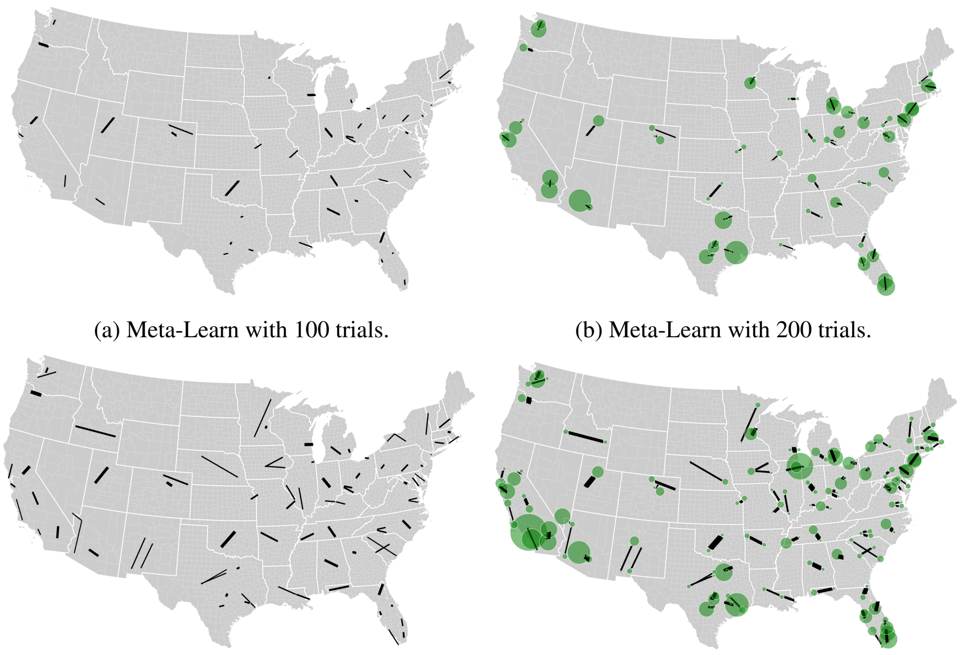
(a) Meta-Learn with 100 trials.

(b) Meta-Learn with 200 trials.

(c) Meta-Learn with 300 trials.

(d) The actual error landscape.

Figure 2: Several zoomed-in shots of several sub-sections of our generated map. These highlight interesting regions that are effectively pulled out by the algorithm, ideally zones that contain interesting regularities within the data-set. Also note the directional nature of the flow arrows, which indicate direction of flow or migration movement.



(a) Meta-Learn with 100 trials.

(b) Meta-Learn with 200 trials.

(c) Meta-Learn with 300 trials.

(d) The actual error landscape.

Figure 3: Several zoomed-in shots of several sub-sections of our generated map. These highlight interesting regions that are effectively pulled out by the algorithm, ideally zones that contain interesting regularities within the data-set. Also note the directional nature of the flow arrows, which indicate direction of flow or migration movement.