Space-time Cluster Detection Using Network-Constrained Minimum Spanning Trees on GPU

K. A. Stofan[1]

[1]University of South Florida, School of Geosciences, 4202 E. Fowler Avenue, NES 107, Tampa, FL 33620 USA
Email: stofank@mail.usf.edu

Abstract

This study proposes a network-constrained minimum spanning tree space-time cluster detection method. The method utilizes parallel programming techniques on a graphics processing unit to improve computational performance and avoid race conditions when used in near real-time systems. The method is tested using observed and simulated data. Results and computational performance gains are also reported.

Keywords: minimum spanning tree; network constrained distance; GPU.

1. Introduction

Space-time cluster detection has recently emerged as an important tool for statistically-based early warning systems in several disciplines including epidemiology and crime geography (Unkel et al. 2012, Shiode and Shiode 2014). Recent advances in computing and parallel processing have enabled the analysis of increasingly larger spatial datasets. Common geocomputational operations such as shortest path calculations, inter-observation distances (e.g. distance matrices), and Markov chain Monte Carlo (MCMC) simulations have been improved through the use of distributed processing and parallel computing techniques such as those using graphics processing units (GPU) (Bolz et al. 2003, Li et al. 2010).

This study presents a network-constrained minimum spanning tree (NcMST) method to detect clusters in spatial point data on GPUs. GPU parallel processing is utilized for several processes within the workflow including shortest path calculations (Dijkstra's algorithm), minimum spanning tree calculations (Kruskal's algorithm), and distance matrix construction. The method is assessed for accuracy using a homogenous Poisson process to create synthetic clusters within the study area. Additionally, processing time is compared to non-GPU assisted execution of the methodology where applicable. Finally, the methodology is tested on an observed dataset consisting of georeferenced social media photograph postings for a one month period.

2. Background

Preliminary work using the minimum spanning tree (MST) method adapted from Wieland et al. (2007) has been used in a near real-time early warning system for detecting spatial clusters in geotagged social media. The method has been used successfully on sample sizes of 100 observations or less at temporal intervals of one minute or greater. Computational constraints of CPU-based processing have prevented the use of the method on larger datasets. For example, observation sizes of n = 100 require the

calculation of approximately 5 million inter-event distances for observed data and significance testing. Additionally, these calculations are conducted for $n - 1$ sample sizes to detect clusters in subtrees. In order to avoid race conditions in the system, parallel processing on GPUs has been identified as a solution to the massive number of computations needed for larger datasets.

In addition to computational constraints associated with the MST method mentioned above, cluster detection was limited to temporal cross-sections of the data (i.e. spatial clusters). Observed clusters within a temporal cross-section were compared to future cross-sections in order to determine temporal persistence of spatial clusters within a study period. Despite the utility of tracking spatial clusters within cross-sections over time, the approach is not statistical in nature. The NcMST methodology proposed in this paper improves on the MST method described above by assessing the distribution of events in both space and time.

Inter-event distances used to calculate spanning trees within the MST method described above were calculated using Euclidean distance. Shiode and Shiode (2013) suggest that applications of cluster detection within highly urbanized environments where inter-event travel is restricted to street networks may benefit from using network-constrained inter-event distances. Network distance may more accurately reflect space-time distances in near real-time systems and applications where temporal scales are on the order of seconds or minutes or where temporal distances are measured in travel time. The NcMST methodology improves on Euclidean-based MST method described above by utilizing network space-time distance for spanning tree calculations.

3. Methods

The NcMST methodology proposed in this study (Figure 1) is conducted using a 64-bit dual core processor at 3.2 GHz with 4mb of memory. The graphics card used is an NVIDIA GeForce GTX 650 GPU with 192 compute unified device architecture (CUDA) cores and CUDA compute level three. The workflow is written in Python and utilizes NumPy/SciPy sparse matrices for network distance and minimum spanning tree calculations. The methodology also utilizes the Continuum Analytics NumbaPro Python library for GPU optimization on NumPy/SciPy matrix inputs. The methodology consists of several steps including point to network snapping, shortest network path, distance matrix, and minimum spanning tree operations. These operations are conducted on both observed and experimental simulation data.

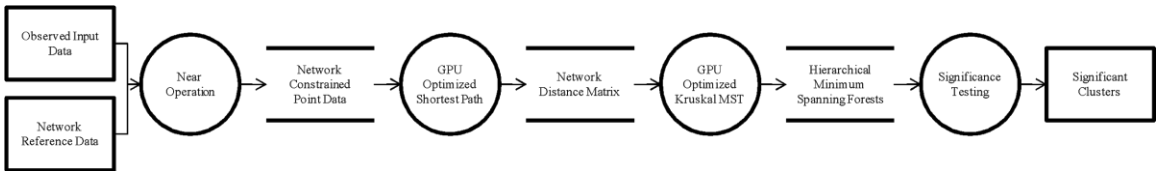

Figure 1. Network constrained minimum spanning tree methodology data flow diagram.

Observed events are analyzed within a moving space-time window to approximate conditions observed in a near real-time system monitoring events over indefinite study periods. Moving space-time windows limit the number observations and experiments to a sample size within the capabilities of the system and avoid unbounded calculations over

long study periods. Distance calculations below refer to space-time distances within a moving space-time window.

Network replacement of points in Euclidean space is accomplished using a near operation consisting of observed or experimental points as an input and street network data as a set of edges and nodes. The shortest distance between observed points and edges within the street network are determined by calculating the perpendicular between them. The operation is well suited to GPU-optimization due to the large number of perpendicular line and distance calculations between points and edges needed to determine the closest network-constrained point. Individual points are replaced in this manner until an exhaustive network-constrained set of observed or experimental points are obtained (Figure 2).

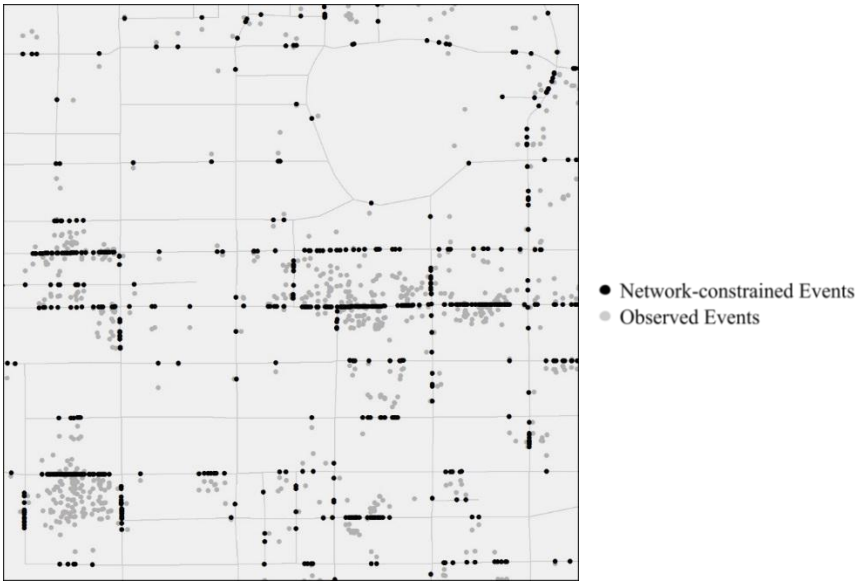

Figure 2. Network corrected social media image observations.

Network-constrained point data are used to create network constrained distance matrices for observed and experimental point datasets using a GPU-assisted Dijkstra's shortest path algorithm to calculate inter-event distances. Network constrained distance matrices are then calculated for observed and experimental data for use in minimum spanning tree calculations. Once again, these operations are well suited to GPU-assisted execution due to the large number parallel distance and shortest path calculations needed to derive their products.

Network-constrained distance matrices are then converted to sparse graphs and used to calculate minimum spanning forests using a GPU-assisted minimum spanning tree algorithm based on Kruskal's algorithm (Figure 3). The algorithm uses a greedy deletion approach to iteratively calculate minimum spanning forests and remove the largest path on successive iterations. The results of this stage in the workflow are n - 1 network constrained minimum spanning forests (NcMSF). Finally, simulated NcMSFs are calculated using simulated spatial point experiments for use in pseudo-significance testing of observed NcMSTs.

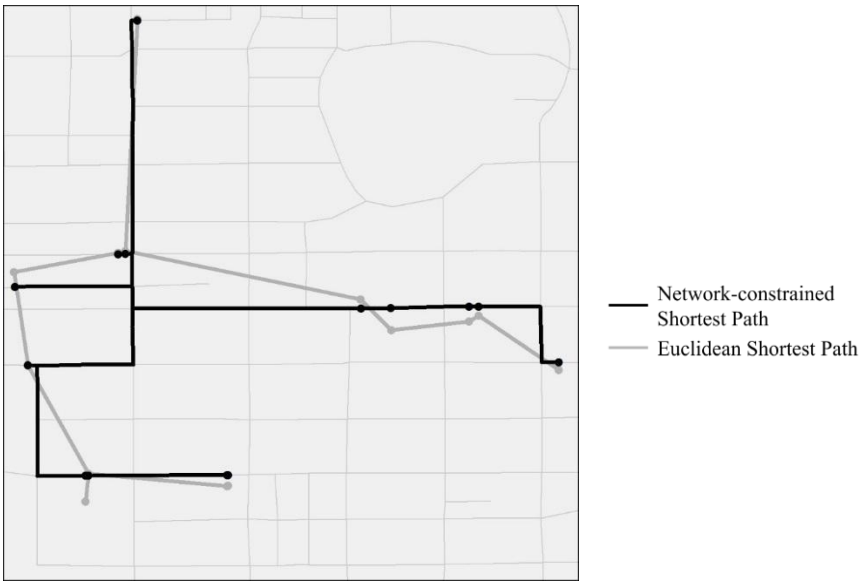

Figure 3. Regular and Network Minimum Spanning Trees

4. References

Bolz, J., Farmer, I., Grinspun, E., & Schröoder, P., 2003, Sparse matrix solvers on the GPU: conjugate gradients and multigrid. *ACM Transactions on Graphics (TOG)*, 22(3):917-924.

Li, Q., Kecman, V., & Salman, R., 2010, A chunking method for euclidean distance matrix calculation on large dataset using multi-gpu. *IEEE 2010 Ninth International Conference Machine Learning and Applications (ICMLA)*, 208-213.

Shiode, S., & Shiode, N., 2013, Network-based space-time search-window technique for hotspot detection of street-level crime incidents. *International Journal of Geographical Information Science*, *27*(5), 866-882.

Shiode, S., & Shiode, N., 2014, Microscale Prediction of Near-Future Crime Concentrations with Street-Level Geosurveillance. *Geographical Analysis*, 46(4), 435-455.

Unkel, S., Farrington, C., Garthwaite, P. H., Robertson, C., & Andrews, N., 2012, Statistical methods for the prospective detection of infectious disease outbreaks: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(1), 49-82.

Wieland, S. C., Brownstein, J. S., Berger, B., & Mandl, K. D., 2007, Density-equalizing Euclidean minimum spanning trees for the detection of all disease cluster shapes. *Proceedings of the National Academy of Sciences*, *104*(22), 9404-9409.