Contrails

Spectral Graph-Theoretic Air Traffic Network Analysis

Navtej Singh

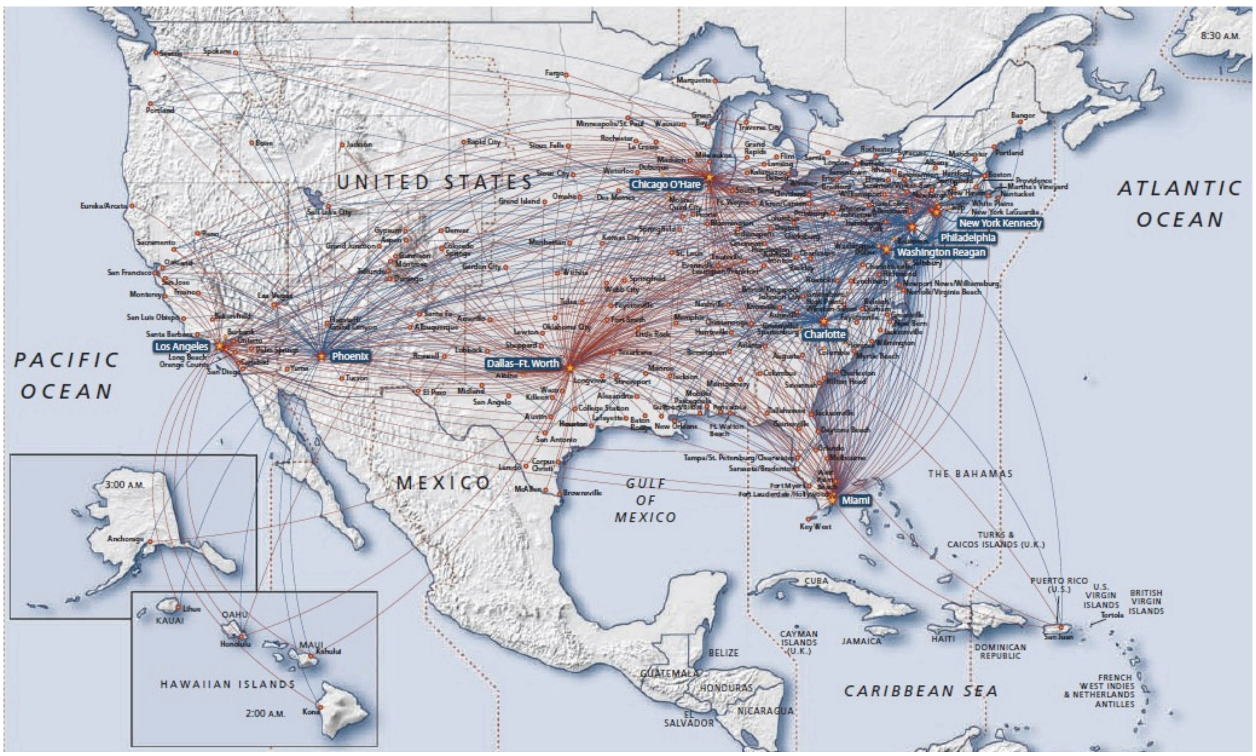
Data Science Assignment 1
Monday, January 12, 2015

ABSTRACT

Spectral analysis can concisely characterize a network's most salient features. This study implements such an analysis on 27 years of US air traffic data obtained from TranStats, a service of the US Department of Transportation's Bureau of Transportation Statistics. Using packet sniffing, an ad-hoc interface to TranStats is created to efficiently download the data contained in over 300 files. An Extraction-Transformation-Loading pipeline prunes and prepares the data for analysis. Next, graph representations of each dataset are instantiated and the associated Adjacency and Laplacian matrices are constructed; computing these matrices' eigenvalues, appropriate measures of connectedness, path redundancy, and robustness are calculated. These results are returned, along with a qualitative description of the model estimated to best describe the network and a summarization of the graph's topology. Finally, select legal, logistical, and financial data help to contextualize the longitudinal results and further uses and improvements of this approach are delineated.

INTRODUCTION

A graph G(V,E) is a collection of vertices (points) and edges (connections between them); it can be used to represent relationships in an interconnected system. For example, airline route maps such as the one below are often well expressed as graphs. A graph is *directed* if its edges relate vertices in a particular direction (e.g. a one-way route) and *weighted* if its edges are associated with a respective weight (e.g. a route graded by some metric).



Spectral Graph Theory involves the study of the properties of a graph through the analysis of values characterizing certain matrices associated to it; herein, we consider the eigenvalues of a graph's associated Adjacency and Laplacian matrices. These values and further derived quantities constitute a powerful language for describing a graph's inherent properties. Of the aforementioned values, we consider three in particular when characterizing a network: algebraic connectivity as a measure of connectedness, natural connectivity as a measure of path redundancy, and the minimum bound of a metric called the Cheeger constant as a measure of robustness.

METHODOLOGY

I. Model Selection

Taking the list of flights flown every month, we can represent each flight as an origin-destination pair; this, in turn, constitutes an edge in the directed, unweighted graph representation of a dataset. This provides a natural and intuitive abstraction because of the amenability of modeling airports (either origin or destination) as vertices and routes taken by flights connecting them as edges with a direction (origin to destination) but not a weight; this representation thus exhausts the meaning we wish to capture about how (and how well) the flights flown connect the nation's airports at various points in time. A weighting based on frequency of flights per route can also be considered.

II. Metric Determination

A. Algebraic Connectivity

The algebraic connectivity of a graph is the second-smallest eigenvalue of the Laplacian matrix associated to a graph. For connected graphs (graphs which connect all vertices), this value is greater than 0 and less than or equal to the vertex connectivity of a graph (the minimum number of vertices which render a graph disconnected upon removal). With this natural relation to connectedness, algebraic connectivity has additionally been shown to have the "tightest bound to the network robustness...and it is the most computational efficient network robustness metric". [1] (Pg. 13)

B. Natural Connectivity

The natural connectivity of a graph is the natural log of the arithmetic mean of the eigenvalues of the Adjacency matrix of a graph. This additional measure of connectivity quantifies "the weighted number of closed walks of all lengths" [2] (Pg. 3) in a graph, is relatively insensitive to small changes in the graph (as opposed to the algebraic connectivity) and accommodates unconnected graphs. Furthermore, this metric is monotonic with respect to graph modification [2] (Pg. 2); that fact further allows this metric to be used for optimal air traffic rerouting by modeling rerouting (e.g. due to inclement weather or emergency) as an edge deletion/insertion problem.

C. Minimum Bound of the Cheeger Constant

The Cheeger Constant of a graph is a measure of robustness insofar as it measures "bottleneckedness" – that is, the larger the Cheeger constant, the greater the number of links between any possible division of the vertex set into two subsets. This fact allows us to model resilience against regional shutdowns (e.g. inclement weather affecting many geographically close airports) and class disruptions (e.g. security threats most relevant to the airports of the most populous cities) quite naturally; additionally, generic reliability of a small number of nodes in a network (e.g. a few airports spontaneously serving as hubs for incoming traffic rerouted from elsewhere) can be tested with this approach. As computing the Cheeger constant is NP-hard, even with more relaxed constraints than ours such as k-regularity and undirectedness, we rely on a minimum bound by the Cheeger Inequality. [3] (Pg. 756)

D. Network Model Estimate

Using algebraic connectivity with a fixed tolerance of 0.1 (selected to minimize false positives based on heuristics from empirical assessment), a network model traditionally associated with air traffic networks is assigned to each dataset. An algebraic connectivity within the tolerance of 0 describes a "Relatively Disconnected" network; one within the tolerance of the theoretical maximum determined by Fiedler's Bound [4] (Pg. 694) describes a "Point to Point" network; one between the first and second ratings can be well described as a "Hub and Spoke" network.

E. Topology

Extremal values of the topology of the graph demonstrate the highest to lowest vertex hierarchy of the graph; in a directed graph, this may be thought of intuitively as the list of most source-like and most sink-like vertices of the network, respectively.

III. Programming

Haskell (Haskell Platform 2014.2.0.0) was used exclusively in this study, along with some standard web (http-conduit), archive (zip-archive), and matrix manipulation (hmatrix) packages. Haskell is a lazy, statically-typed, pure functional programming language with denotational semantics making it apt for mathematical programming. The results of the analyses were output as "txt" and "csv" files. The GraphViz library and corresponding Haskell package can be utilized to visualize graphs, though they are not output by default.

RESULTS

I. Summary Statistics

	Algebraic Connectivity	Mean Potential Algebraic Connectivity	AC/ MPAC	Natural Connectivity	Min Bound of Cheeger Constant
Min	0	13.05732	0	88.05323	0
Avg	0.266748839	14.49568872	1.842556258	118.7564048	0
Max	0.50431	16.32517	3.53971	172.30279	0

N.B. Mean Potential Algebraic Connectivity := Fiedler's Upper Bound / 2

	Passenger Load Factor	Revenue Passenger Miles (Mils)
Min	55%	163147.546
Avg	67%	480201.8679
Max	83%	693372.479

Model	Count
Relatively Disconnected	71
Hub and Spoke	261
Point to Point	0

II. Data

See Files in Supplemental/Results

III. Figures

See Files in Supplemental/Figures

DISCUSSION

Next steps include mathematical proof of correctness of the aforementioned implementations as well as determination of complexity estimates. Additionally, GraphViz implementation along with a more sophisticated plotting library could be utilized to generate visually stunning and informative graph visualizations. Finally, along with a more expressive yet equally economical Cheeger constant estimator, implementation of an iterative Natural Connectivity maximization algorithm which can recommend new edges to make after some others are broken can be used for reroute handling by aviation authorities. A candidate is the MIOBI-MakeEdge algorithm whose pseudocode is posted here. [2] A live version of this may also be put online as a demonstration and exercise.

Input: Graph $G(V, E)$, its adj. matrix $\mathbf{A}$, and int. budget k
Output: Set S of k edges to be added

$S = \emptyset$
Compute the top t (eigenvalue, eigenvector) pairs $(\lambda_j, \mathbf{u_j})$ of $\mathbf{A}$, $1 \leq j \leq t$
for *step* $= 1$ to k **do**
  Compute the largest degree $d_{\max}$ of $\mathbf{A}$
  Find the candidate subset $\mathcal{C}$ of $d_{\max}$ nodes with the highest $\mathbf{u_1}$ eigen-scores
  Select the edge $(\bar{p}, \bar{r})$ out of $\forall (p, r) \notin E, p \in \mathcal{C},\ r \in \mathcal{C},\ p \neq r$, that maximizes Equ. (4.10) for top t eigenvectors, i.e.

$$\max_{\substack{(p,r)\notin E \\ p\in\mathcal{C}, r\in\mathcal{C}}} \quad c_1 \left(e^{2\mathbf{u_{p1}}\mathbf{u_{r1}}} + c_2 e^{2\mathbf{u_{p2}}\mathbf{u_{r2}}} + \ldots + c_t e^{2\mathbf{u_{pt}}\mathbf{u_{rt}}} \right)$$

  $S := S \cup (\bar{p}, \bar{r})$, $E := E \cup (\bar{p}, \bar{r})$
  Update $\mathbf{A}$; $\mathbf{A}(\bar{p}, \bar{r}) = 1$ and $\mathbf{A}(\bar{r}, \bar{p}) = 1$
  Update top t eigenvalues of $\mathbf{A}$ by Equ. (4.3)
  Update top t eigenvectors of $\mathbf{A}$ by Equ. (4.7)
end for
Return S

References

1. Wei, P., L. Chen, and D. Sun. "Algebraic Connectivity Maximization of an Air Transportation Network: The Flight Routes' Addition/deletion Problem." *Transportation Research Part E: Logistics and Transportation Review*: 13-27. Print.

2. Chan, H., L. Akoglu, and H. Tong "Make It or Break it: Manipulating Robustness in Large Networks" SDM 2014.

3. Chung, F. "Four Cheeger-type Inequalities for Graph Partitioning Algorithms" ICCM 2007.

4. Ghosh, A., and S. Boyd. "Upper Bounds on Algebraic Connectivity via Convex Optimization." *Linear Algebra and Its Applications*: 693-707. Print.