NetSpot: Spotting Significant Anomalous Regions on Dynamic Networks - Supplemental material

Misael Mongiovì[*] Petko Bogdanov[*] Razvan Ranca[*] Evangelos E. Papalexakis[†]
Christos Faloutsos[†] Ambuj K. Singh[*]

1 Preliminaries

1.1 Example of dynamic network Fig. 1 shows an example of dynamic network and some typical regions. The network is a simple path. The horizontal axis represents the time. Each region spans a connected sub-network and a time interval. A region may contain negative edges (see region 5), provided that its total score is high.

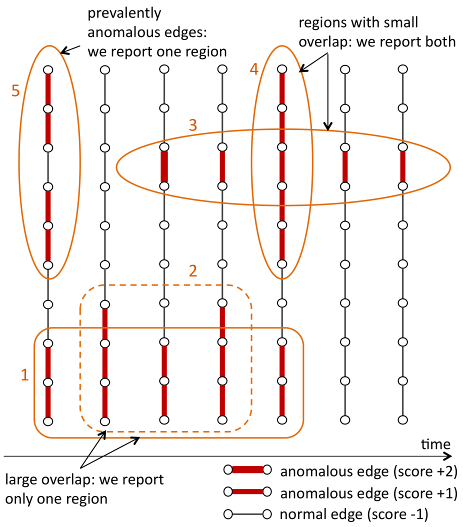

prevalently anomalous edges: we report one region
regions with small overlap: we report both
5
4
3
2
1
time
large overlap: we report only one region
anomalous edge (score +2)
anomalous edge (score +1)
normal edge (score -1)

Figure 1: We want to report all regions whose total score is higher than a threshold and do not have much overlap. The network is a simple vertical path. The horizontal axis represents the time. We omit region 2 since it overlaps significantly with region 1.

1.2 Special cases of HDS Some special cases of HDS are listed in Table 1 along with their complexity. We briefly discuss these problems.

Input	Name	Complexity	Rooted
Single edge:	MSS	$O(T)$	$O(T)$
Single slice:			
Graphs	HS	NPhard	NPhard
Trees	tree-HS	$O(\|E\|)$	$O(\|E\|)$
Paths	MSS	$O(\|E\|)$	$O(\|E\|)$

Table 1: Special cases of Heaviest Dynamic Subgraph (HDS). The rooted version of each problem (last column) constraints a given node/edge to belong to the solution.

Given a sequence of positive and negative values, The Maximum Score Subsequence (MSS) problem calls for finding the contiguous subsequence that maximizes its score. Its rooted version (last column in Table 1) constraints a given time instant (root) to belong to the solution. This problem has a linear time solution [5] since the highest score of any subsequence ending at t can be derived by the highest score of any subsequence ending at $t-1$.

Given a graph whose edges are weighted with positive and negative values, the Heaviest Subgraph (HS) [1] problem calls for finding the connected subgraph such that the sum of its edge weights is the highest. HS is equivalent to Prize Collecting Steiner Tree (PCST) [3], which is NP-hard. Its rooted version (in which a special "root" node is constrained to belong to the solution) does not admit any constant factor approximation. An efficient heuristic for HS, namely TopDown, has been recently proposed [1]. The main idea is to first compute a spanning tree of the graph and then find the optimal subtree in linear time.

2 Problem definition

2.1 Intuitive problem definition

Definition 2.1. Intuitive problem definition*: given a dynamic network whose edges are annotated with their degree of anomaly, we want to report a comprehensive (spanning all regions of interest) set of all anomalous regions that are significant and have low overlap.*

Admitting a small overlap between regions in the

[*] Dept. of Computer Science. UC Santa Barbara
[†] Dept. of Computer Science. Carnegie Mellon University

answer set is advantageous since partially overlapping regions can capture separate nearby processes, e.g. multiple congestions in the same locality of a road network. On the other hand, admitting unrestricted overlap may lead to finding many regions that are pairwise similar to each other, since small changes to a high-score region are also likely to give high-score regions. For example, in Fig. 1, among regions 1 and 2 we want to report only 1 (with score 10). According to our problem definition, the score of region 2 does not consider the contribution from edges that overlap with region 1, and hence its score is 2. In contrast, we want to report both regions 3 and 4. In this case the score of region 4 is discounted by 1.

3 Method

3.1 Alternative seed generation strategies We describe three alternative approaches for seed generation, in increasing level of complexity and computational requirements. These approaches have been compared with the proposed approach and perform worse.

- **Random edge in a random slice (Rand).** The simplest and most intuitive strategy is that of sampling random edge/time seeds. Although in some cases this approach leads to a good solution, a bad starting point can affect considerably the quality of the result.

- **Maximum edge seed (Max).** This strategy selects the edge/timestamp with maximum weight. Although this strategy outperforms Rand, it does not consider the extension of anomaly. Therefore its rate of failure in identifying the highest score region is high.

- **Matrix Factorization (MF).** Another alternative is to adopt matrix factorization for seed selection by considering the edge-by-time matrix of scores. Decomposing this matrix in sparse, low-rank components [4] would produce a set of edges and time instants that may be used as seeds. The efficiency of this approach, however, degrades as the score matrix gets denser. Moreover, matrix factorization loses the topological information encoded in the graph structure and the time order. Therefore, it is not guaranteed to produce connected and contiguous seeds.

3.1.1 Incremental update for seed generation All rooted HS and All rooted MSS need to be re-computed at every iteration of NetSpot. To improve the overall efficiency, we keep track of the part of network that has been affected by the changes and needs re-computation. The idea is that if an edge (u, v) is affected, then a neighbor (x, u) is affected only if $s_{\leftarrow}(u, v) > 0$, while a neighbor (v, y) is affected only if $s_{\rightarrow}(u, v) > 0$.

3.2 Assessing significance The score threshold $\mathcal{T}$ discriminates significant from non-significant regions. How to set this threshold is not trivial, since it depends on the topology of the considered network. Low values may bloat the set of returned regions, by including regions whose scores can be obtained by chance. On the other hand, high values may result in missing interesting regions.

In order to estimate a good score for $\mathcal{T}$ we propose to compute a number of trials by random shuffling of the values on individual edges in time. Each trial is computed by conserving the network topology and computing, for each edge, a permutation of the time series of edge values. Such permutation is different for every edges. For each trial, its highest score region is computed and finally a list of scores is built and ordered in descending order. This list defines a correspondence between scores and p-values. The score threshold $\mathcal{T}$ is chosen as the element of the list that is at the position corresponding to a chosen p-value multiplied by the number of trials. A significance level commonly used in literature is p-value$= 0.01$, though other values can also be used.

4 Experimental analysis

4.1 Datasets

4.1.1 Vehicular traffic The graph structure of our vehicular traffic datasets corresponds to the highway network of Los Angeles. Edges are highway segments and their values are based on the average speeds at 5 minutes resolution. This 854 MB dataset spans one month and is obtained from the *PeMS* project. The p-value of an edge is computed by considering the distribution of average speeds along the edge at the same time of the day. Hence, anomalous connected edges correspond to locations observing speed lower than expected. They may correspond to unexpected congestions induced by car accidents. Our smaller traffic dataset is a connected subgraph of the full one.

We match the set of regions computed by NetSpot with a list of reported accidents provided by the California Highway Patrol. The accident data is available for download from the *PeMS* system. We consider all accidents that caused injuries or fatalities with reported duration at least one hour. We obtain a set of 695 accidents.

4.1.2 Enron email communication

The ENRON dataset consists of a corpus of email messages exchanged among employees of the Enron corporation [2]. The link structure of this network spans pairs of email accounts that exchanged at least one message over the timeline between 1999–2001. Abnormal edges in this network are the ones that observe unexpected high rate of activity during a day, while abnormal regions correspond to communication backbones of correspondence that are employed more than normal.

4.1.3 Wikipedia page views

We also experiment with the a snowball sample of the Wikipedia information network. The nodes in this networks correspond to Wikipedia pages, while the structure of the network is based on links among them (two nodes are linked if their corresponding pages have at least 8 links to each other). We track the daily page views of pages in 2008 and 2009 and assign anomaly scores based on the p-value of daily views in the article's empirical distribution. Scores of nodes are averaged to obtain link scores.

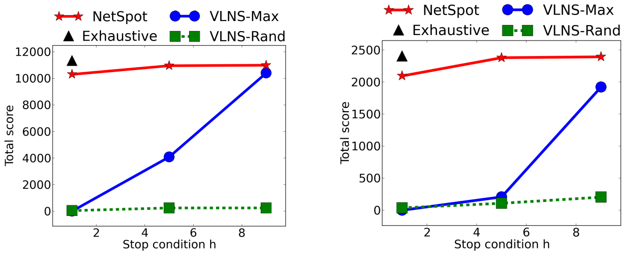
4.1.4 Synthetic datasets

We generate synthetic datasets to test the scalability of NetSpot for increasing graph size and time interval length; as well as for quality evaluation in detecting synthetically "injected" anomalies. Our synthetic network has a fixed topology, obtained by a Delaunay triangulation of uniformly sampled 2-D points. The dynamic behavior of the network edges is produced by injecting random anomalies that diffuse in time and network locality. To inject an anomaly, we first choose a random set of anomalous network vertices that covers 1% of nodes. The adjacent neighboring edges of the selected vertices (up to several hops) are uniformly sampled for inclusion in the anomalous region. The participating edges are assigned with anomalous scores in a random time interval that overlaps with the original seed node. The diffusion strength in time and network hops is controlled by parameters. In order to create a realistic scenario, we also introduce uniform noise in both normal and anomalous regions.

Each synthetic network is accompanied by the positions and time of the injected anomalies. We use this data to evaluate the ability of our method to discover anomalies that diffuse in the dynamic network.

4.2 Results

Accuracy results on Enron and Wikipedia are presented in Fig. 2. As for Traffic, NetSpot consistently produces high quality regions, achieving more than 96% relative quality with respect to Exhaustive on real networks.



(a) Enron, $\mathcal{T}$=10, μ=0.01 (b) Wikipedia, $\mathcal{T}$=10, μ=0.01

Figure 2: Quality of our algorithm, compared to Exhaustive on Enron and Wikipedia. The HSMS seed generation (NetSpot), combined with our NetAmoeba procedure, produces good quality regions in all networks

In order to assess the effectiveness of the three seed generation strategies, we examine the behavior of NetSpot and Exhaustive during their execution. Fig. 3 shows the total score of found regions at each iteration of the algorithm. A good performance requires that best patterns are discovered fast and as early as possible. Although NetSpot does not discover patterns in strictly decreasing order, it is able to find high-score patterns relatively soon, while VLNS–Max and VLNS-Rand tend to show higher delay. Indeed the score of NetSpot is very close to baseline at the beginning, and slightly separates from it during the execution. In contrast, the scores of VLNS–Max and VLNS-Rand grow much slowly.

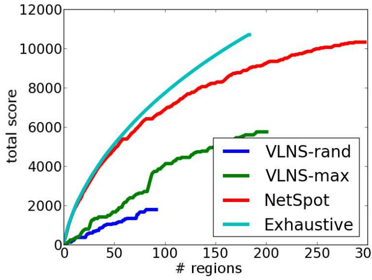


Figure 3: Among the seed generation methods, HSMS (the seed generation of NetSpot) reaches the highest quality (total score) in the same number of iterations. The plot shows the total score of found regions in order of execution. NetSpot (close to Exhaustive) largely outperforms the other methods.

Acknowledgements

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] P. Bogdanov, M. Mongiovi, and A. K. Singh, *Mining heavy subgraphs in time-evolving networks*, in ICDM, 2011.

[2] J. Diesner, T. L. Frantz, and K. M. Carley, *Communication Networks from the Enron Email Corpus It's Always About the People. Enron is no Different*, J. of Computational and Mathematical Organization Theory, (2006).

[3] D. S. Johnson, M. Minkoff, and S. Phillips, *The Prize Collecting Steiner Tree Problem : Theory and Practice*, Proc. of SODA, (2000).

[4] E. Papalexakis, N. Sidiropoulos, and M. Garofalakis, *Reviewer profiling using sparse matrix regression*, in Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, IEEE, 2010, pp. 1214–1219.

[5] W. L. Ruzzo and M. Tompa, *A linear time algorithm for finding all maximal scoring subsequences*, in ISMB, 1999.