Web Graph Similarity for Anomaly Detection

Pallavi Gudipati
Aritra Ghosh

IIT Madras

November 14, 2014

Motivation

- Webgraphs are approximate snapshots of the Web.
- How can we tell an important part of the web is missing?
- Very hard to detect problems simply by examining a single snapshot or instance.
- **Consequences:** Can affect ranking of search results.
- Monitoring all the infrastructure and verifying its proper function is expensive and has no guaranteed overall solution.
- It takes lots of time and resources to build a web graph. It is useful to ensure that the graph is still "okay" before rushing to a costly rebuilding.

Motivation

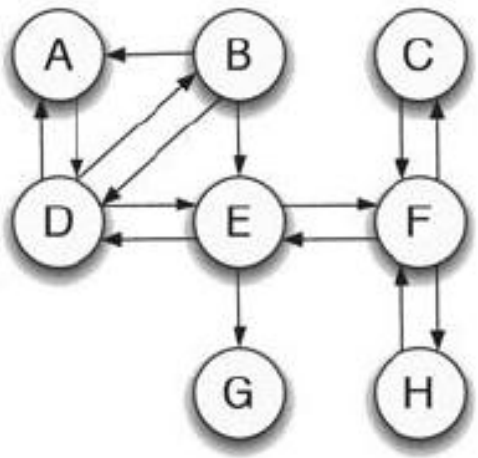
- **Solution:** More practical to identify anomalies based on differences with previous snapshots.
- **Anomaly:** "Factors that may result in web graphs with poor Web representation".
- Types of Anomalies:
  - Failures of web hosts that do not allow the crawler to access their content.
  - Hardware/Software problems in the search engine infrastructure which can corrupt parts of the crawled data.

Applications

- Similarity between two graphs can be used to tune frequency of crawls required.
- Bugs in crawler code if any can be monitored by introducing correct similarity measures.
- Gives an idea how web is evolving with time.

Background

- Webgraph: A (host level) webgraph is a directed weighted graph whose vertices correspond to active hosts of the web and whose weighted edges aggregate the hyperlinks of webpages in these hosts.
- An anomaly occurs when the stored graph representation does not reflect the topology and the properties of the actual graph at crawl time.



Machine	Vertex	Outlinks	PageRank
1	A	D	0.56
	B	A, D, E	0.43
2	C	F	0.51
	D	A, B, E	1.01
3	E	D, F, G	0.93
	F	C, E, H	1.29
4	G		0.41
	H	F	0.51

Problem Statement

- A sequence of web graphs $G_1, ..., G_n$ built consecutively.
- Quantify the changes from one web graph to the next by computing multiple similarity scores between two consecutive web graphs.
- Detect anomalies by comparing these similarity scores against their respective threshold.

Anomalies

- **Missing Random Vertices:** This anomaly usually occurs when one of the machines on which the data is stored fails. Random vertices disappear from the graph.

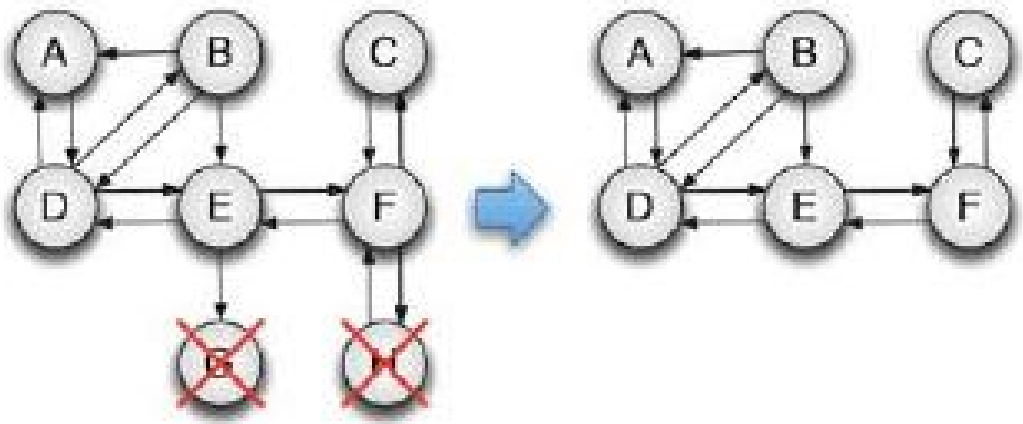


Figure: Missing Vertices

Anomalies

- **Connectivity Change:** This anomaly occurs due to bugs in code, crawler errors etc. The scalar properties of the graph (total number of nodes, total number of edges etc.) remain constant, but the neighbors of some vertices change.

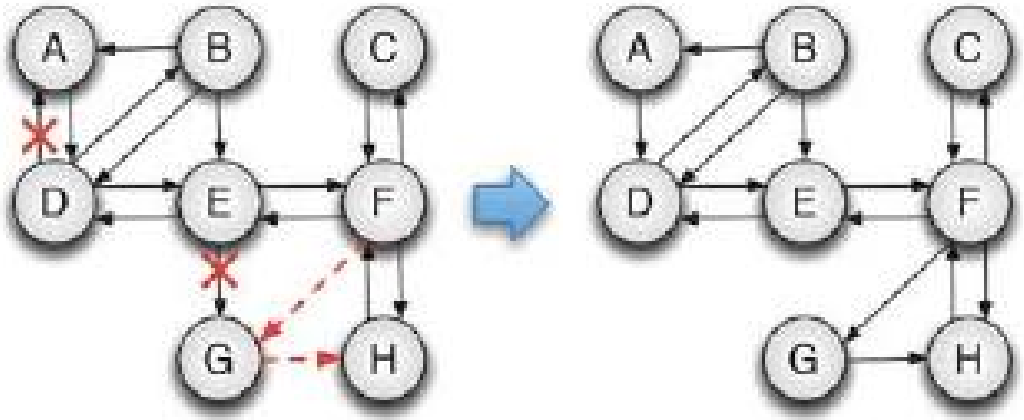


Figure: Connectivity Change

Anomalies

- **Vertex Label Exchange:** This anomaly usually occurs due to wrong DNS records. In this anomaly, the scalar properties of the graph remain the same, but the labels of the vertices are exchanged.
- **Missing Random Connections:** This anomaly usually occurs when a web crawler receives responses with incomplete headers. In this anomaly, some of the edges disappear from the graph.

Datasets

Stanford's SNAP Library: Dataset corresponds to pages from Stanford University (www.stanford.edu)

Table: Statistics of *stanford.edu* Webgraph

Nodes	281903
Edges	2312497
Average Clustering Coefficient	0.5976
Diameter	674

Dataset Generation

Generated Webgraphs where the vertices are affected with the following probabilities: 0.01, 0.05, 0.1, 0.2, 0.5 and 0.8.

- **Missing Random Vertices Dataset:** Generated by randomly knocking off vertices and their corresponding edges.
- **Connectivity Change Dataset:** Generated by randomly picking nodes and transferring the edge to one of its neighbor to a pair of random nodes.
- **Vertex Label Exchange Dataset:** Generated by choosing nodes with similar PageRank scores and swapping their vertex labels.
- **Missing Random Connections Dataset:** Generated by randomly choosing nodes and deleting the edge between them.

Baseline

Similarity Measures

- **Vertex/Edge Overlap**

$$sim_{VEO}(G, G') = 2\frac{|V \cap V'| + |E \cap E'|}{|V + |V'| + |E| + |E'|}$$

- **Vertex ranking**

$$\rho = 1 - \frac{2\sum_i d_i^2}{n(n^2-1)/3} \in [-1, 1]$$

An extension is used for graphs with different number of vertices.

Baseline

Signature Similarity

SimHash

- **L1** = $[(A, 1.4), (B, 2.1), (C, 0.5), (AB, 0.7), (AC, 0.7), (BC, 2.1), (CA, 0.5)]$
- **L2** = $[(0001), (0010), (0100), (0011), (0101), (0110), (1000)]$
- **L3** = $[(-1.4, -1.4, -1.4, 1.4), (-2.1, -2.1, 2.1, 2.1), (-0.5, 0.5, -0.5, -0.5), (-0.7, 0.7, 0.7, 0.7), (-0.7, 0.7, -0.7, -0.7), (-2.1, 2.1, 2.1, -2.1), (0.5, -0.5, -0.5, -0.5)]$
- **L4** = $[-7, -1.4, 1.8, -3.8]$
- **S** = $[0010]$

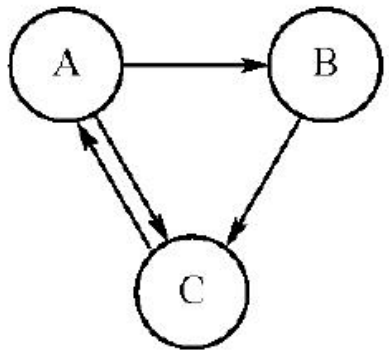
$$sim_{SimHash}(L, L') = 1 - \frac{Hamming(h, h')}{b} \quad (1)$$

Baseline

Signature Similarity

SimHash

- Used for high-dimensional vectors and documents.



Feature	Outlinks		Quality
A	B	C	1.4
B	C		2.1
C	A		0.5
AB			1.4*1/2
AC			1.4*1/2
BC			2.1*1/1
CA			0.5*1/1

Baseline

Signature Similarity

- "Two objects are similar if their signatures are similar". Each graph is represented by a set of features.
- Idea of SimHash used: (Hashing Function: SHA-512)

$$sim_{SS}(G.G') = sim_{SimHash}(\phi(G), \phi(G')) \tag{2}$$

- Vertices and edges from the graph are taken as features.
- The weight of a vertex is its PageRank, the weight of an edge (u, v) is the PageRank of u normalized by the out-degree of u.

Signature Similarity

with Vertex Out-Degree

- Signature Similarity works well for Missing Vertices Anomaly and Exchanged Connections Anomaly.
- Fails in detecting Vertex Label Exchange Anomaly.
- Reason?
- Weight of a vertex is defined as follows:

$$Weight_{SSV}(u) = \alpha PageRank(u) + (1 - \alpha) OutDegree(u)$$

where α lies between 0 and 1. α controls the trade-off between vertex quality and vertex neighborhood structure.

Signature Similarity

with Average Neighbor Out-Degree

- SS with Vertex Out-Degree performs well for Vertex Label Exchange Anomaly.
- Fails for Connection Change Anomaly as well as Missing Edges Anomaly.
-

$$Weight_{SSA}(u) = \alpha PageRank + (1 - \alpha)\frac{\sum_{v \in N_u} OutDegree(v)}{|N_u|}$$

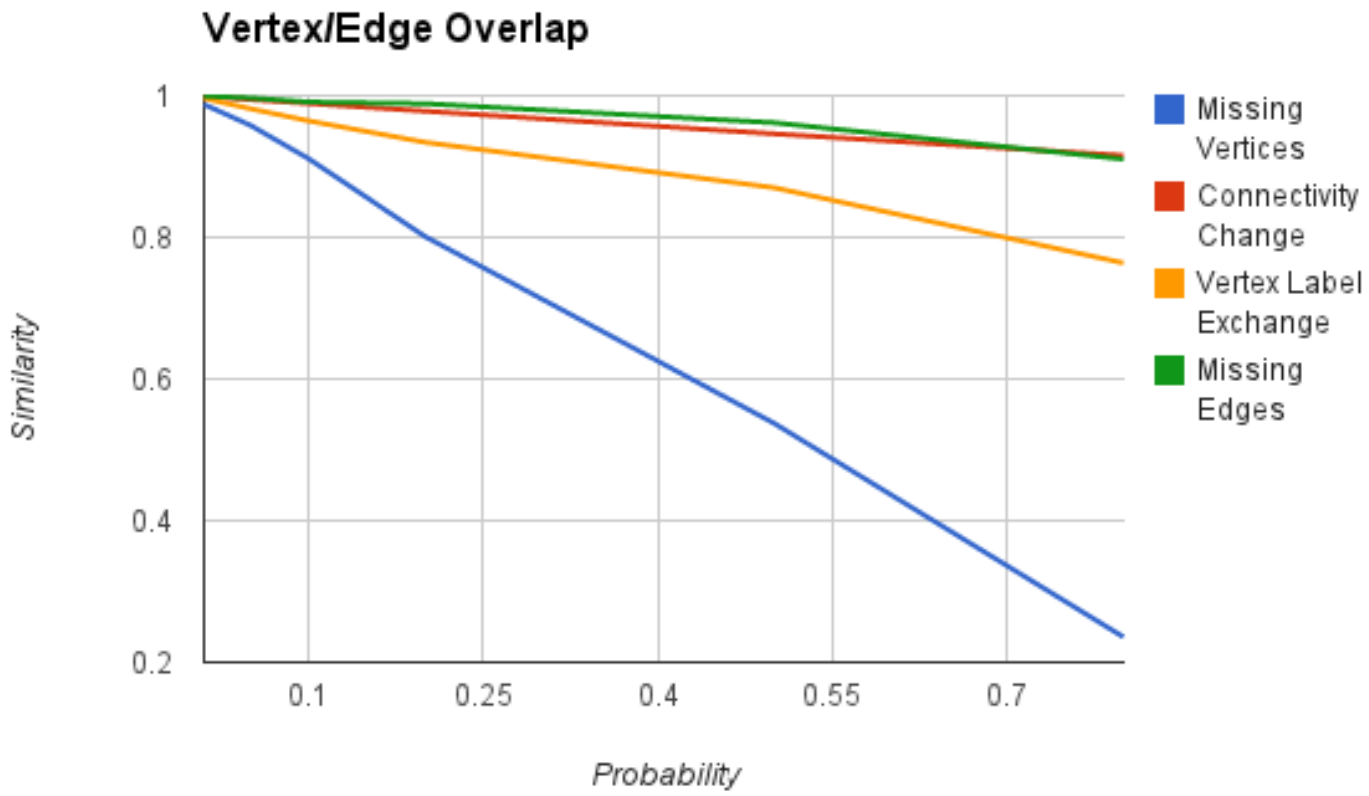
where α lies between 0 and 1, and N_u is the set of all the out-neighbors of u. α controls the trade-off between vertex quality and vertex neighborhood structure.

Experimental Results

- **Computer Configurations:**
  - 4th Generation Intel Core i5
  - 8 GB RAM
  - 1.60 GHz
- Most of the time is spent in scanning the edge lists.
- Time taken to calculate the similarity is $\approx$ 1 min.

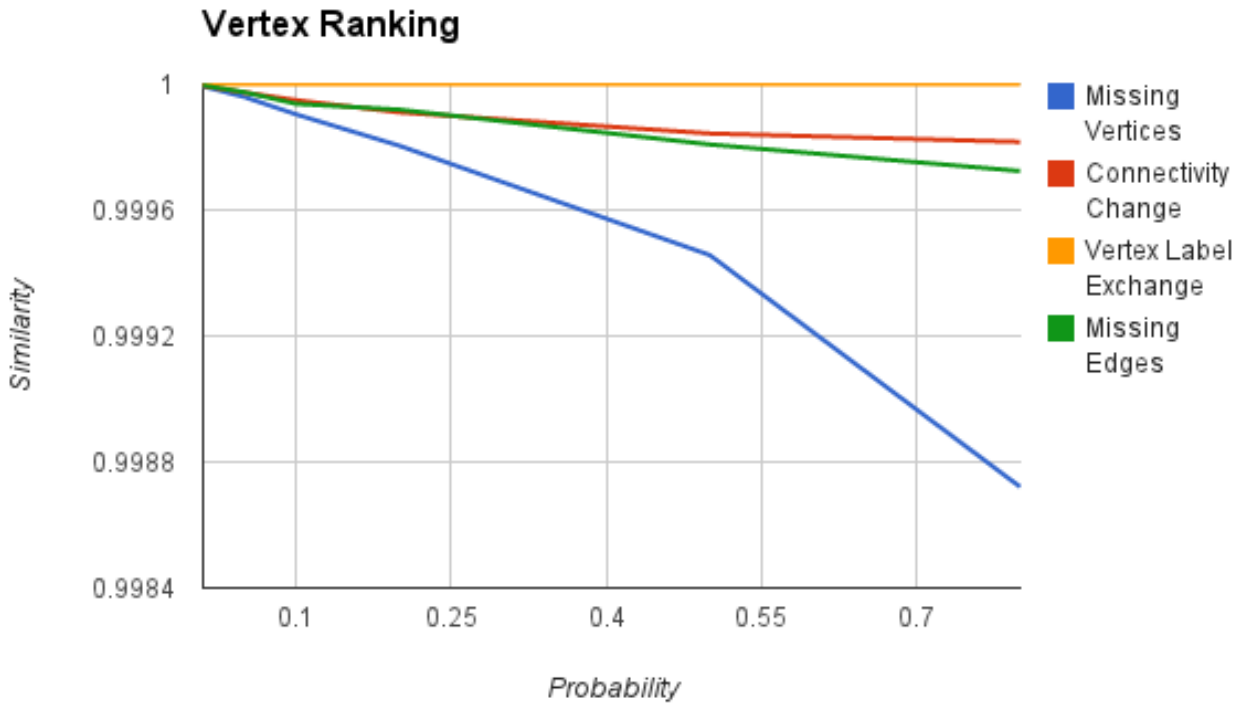
Experimental Results

Vertex/Edge Overlap



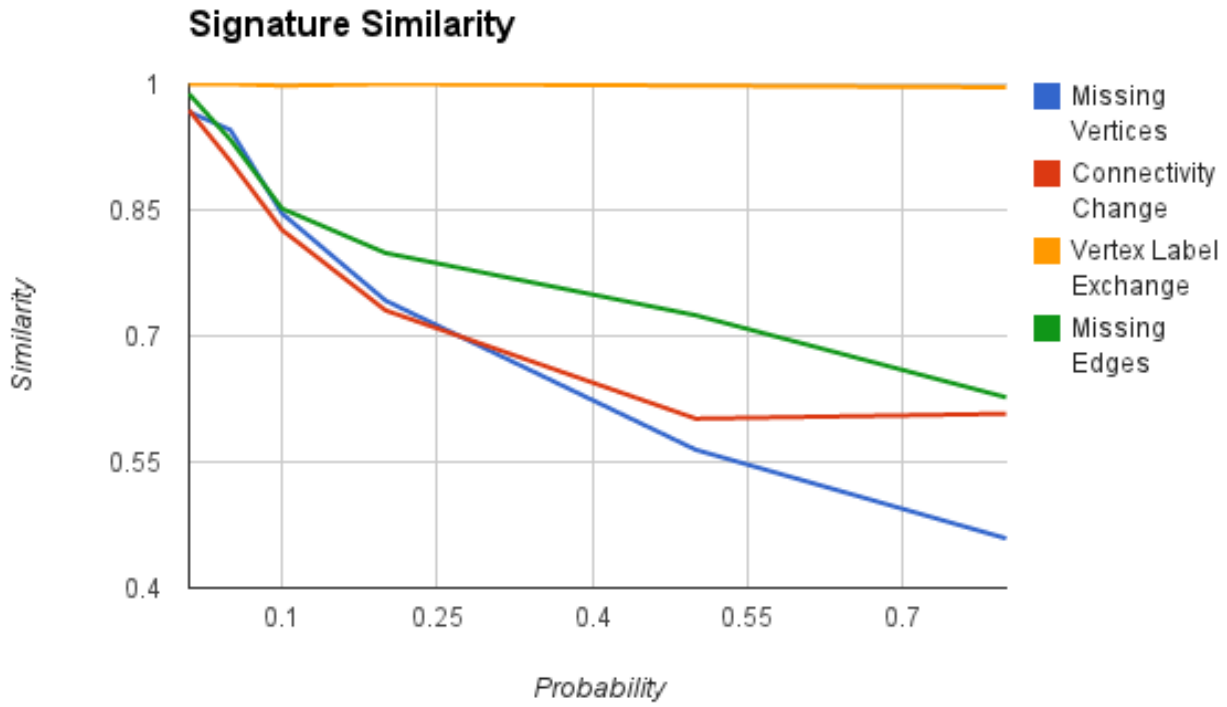
Experimental Results

Vertex Ranking



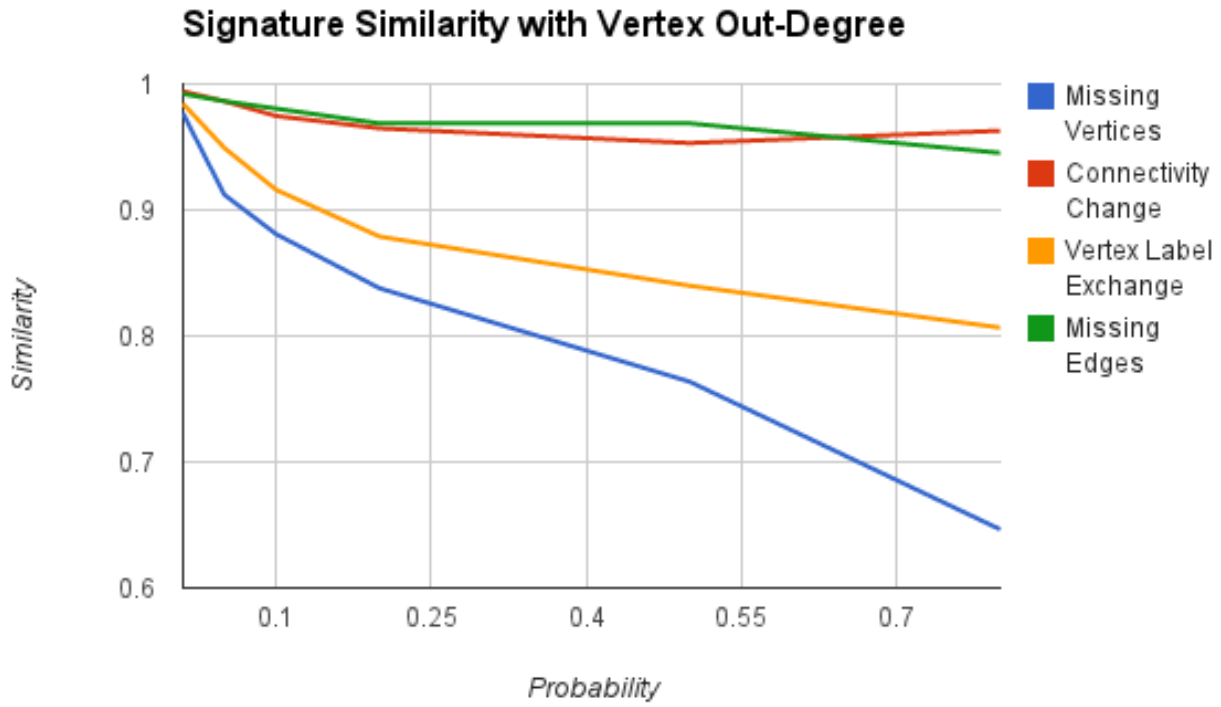
Experimental Results

Signature Similarity



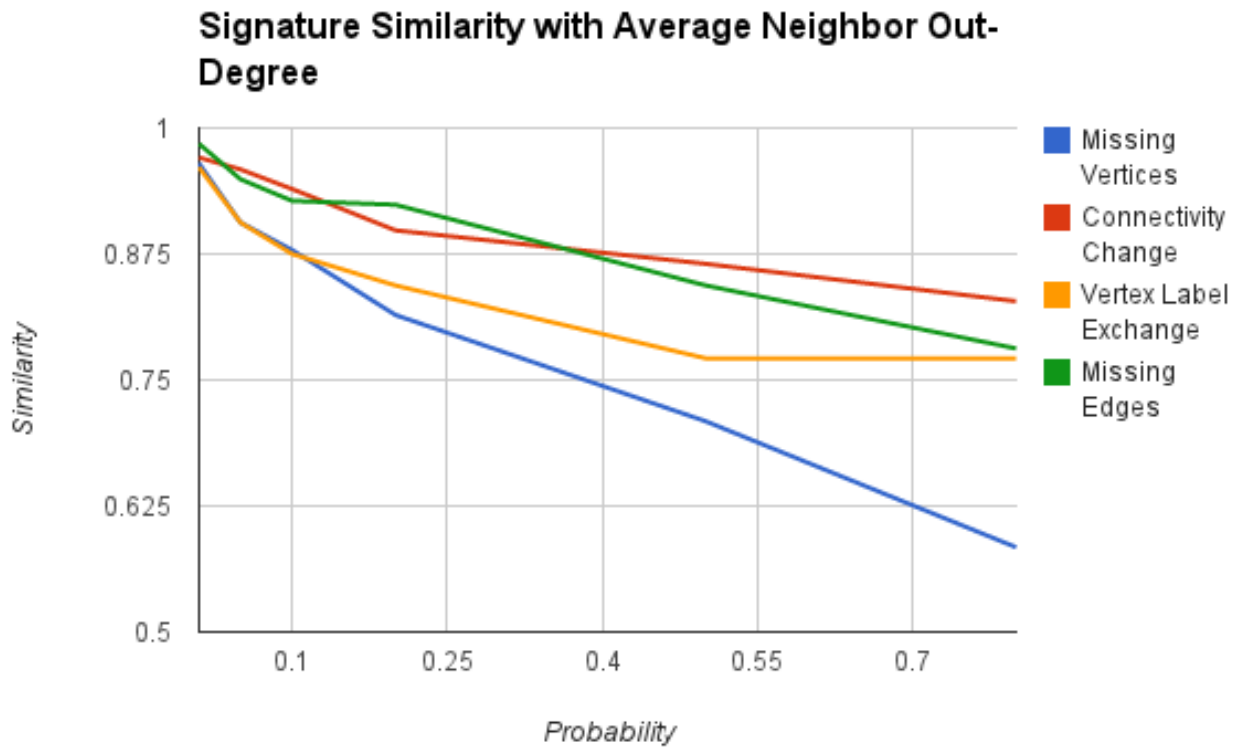
Experimental Results

Signature Similarity with Vertex Out-Degree



Experimental Results

Signature Similarity with Average Neighbor Out-Degree



Conclusions and Future Work

Summary

- Studied the problem of detecting anomalous Webgraphs.
- Introduced two new types of anomalies: Vertex Label Exchange and Missing Random Connections.
- Proposed two modifications to Signature Similarity which takes into account vertex neighbourhood.

Future Work

- Is it enough to label a whole graph as anomalous? - Anomalous subgraphs.
- Detecting type and extent of anomalies present.

Thank You!

Any Questions or Comments?