Pattern Detection in Computer Networks Using Robust Principal Component Analysis

Randy Paffenroth

Associate Professor of Mathematical Sciences, and Associate Professor of Computer Science
Data Science Program
Worcester Polytechnic Institute

CS525 URBAN NETWORKS: METHODS AND ANALYSIS
4-19-2017
Urban networks vs. Computer networks
What am I going to talk about?

- *Robust Principal Component Analysis* as applied to analysis of computer networks.
  - In effect, I am interested in “semi-supervised” learning where much of the data is unlabeled and has to “speak for itself”

- Attempt to justify why I think this is an interesting way to think about network analysis.

- Show some examples in this area.

- Beware! I am a mathematician and, morally, I can’t give a talk without any equations :-)
Theory and Practice
Where we find our inspiration for practice...

- Stuxnet, Flame, Target Inc., Neiman Marcus, Affinity Gaming, Dairy Queen...
  - *Et tu, Dairy Queen!? This is when things got personal...*

- "Axiom" 1: Unless some sensor, or collection of sensors, is *effected by an attack* then you can't detect it.
  - I.e. either the marginal or joint probability density function of the sensors must be different in a statistically meaningful way, conditioned on the absence or presence of an attack.

- "Axiom" 2: The most dangerous attacks are those for which you *don't have a signature*.
  - Virus detection and intrusion detection systems (IDS) do a good job of detecting attacks for which a signature is known, but have nothing to say if the attack has no signature

- "Theorem": Therefore the most dangerous attacks can only be detected by sensors which were *not designed to detect that threat*.
  - You have to get lucky and have a sensor that detects the new attack even though it was not designed to do so.

- "Corollary": You want lots of sensors!
  - But, how do you fuse them? Even once you have a way of fusing the data, how do you avoid being overwhelmed with *false alarms!*
Advanced Persistent Threats

Reconnaissance

Command and control

Botnet

Point of entry

Exfiltration

Pivoting
What do we mean by a sensor?
What do we mean by a sensor?
What do we mean by a sensor?
What kinds of sensors?

- Already talked about packet rates.
- Port, CPU, memory activity, etc.
- Intrusion Detection Systems
  - Bro, Snort, Suricata, etc.
- More "complicated" sensors such as those inspired by information theory.
  - Packet payload entropy

Best with an example!
Data matrix

- Given a graph $G = \{E, V\}$ we assign to each vertex $v_i \in V$ a discrete vector valued time series $y_i \in \mathbb{R}^{l_i \times n}$ of dimension $l_i$ and length $n$.

- We then construct a signal matrix $Y \in \mathbb{R}^{m \times n}$ with $m = \sum_{i=1}^{|V|} l_i$.

\[ Y = \begin{bmatrix} y_1 \\ \vdots \\ y_{|V|} \end{bmatrix} \]
First order anomaly
Sparse correlations?
Latent signal model...

- We study the data using a latent time series model.

\[ Y = m A U + m B V + m N \]

Where \( A \in \mathbb{R}^{m \times l} \) is low rank, \( B \in \mathbb{R}^{m \times k} \) is sparse, and the matrices \( U \in \mathbb{R}^{k \times n} \), \( V \in \mathbb{R}^{l \times n} \), and \( N \in \mathbb{R}^{m \times n} \) have (approximately) mutually orthogonal rows.

- What is the point of the model?

  - \( AU \) models those traces that influence a large part of \( Y \).
  - \( BV \) models those traces that influence only a few rows of \( Y \).
  - \( N \) models those traces (e.g. noise) that influence individual rows of \( Y \).
A simple second order anomaly
Second order theory!

In our work we focused on analyzing the second order statistics of $Y$ by way of its covariance or normalized cross correlation matrix $M$, such as

$$M = \frac{1}{n-1} (Y - \mu_Y)^T (Y - \mu_Y)$$

Interesting questions:
- Correlation versus covariance?
- More refined calculations such as Maximum Likelihood covariance estimation (e.g. using convex optimization).

Well defined for missing data and different data types (e.g. point-biserial correlation).
Second order anomaly
Standing on the shoulders of giants

- Over the past 4-5 years there has been a flurry of activity on this problem, much of which we suspect the current audience is aware of.
- Ideas such as matrix completion, robust principal component analysis, and robust matrix completion have generated a lot of interest, including among us!

Matrix completion: The Netflix problem!

\[ L = \arg \min_{L_0} \| L_0 \|_* \]
\[ \text{s.t. } P_{\Omega}(L_0) = P_{\Omega}(M) \]


\[ L = \arg \min_{L_0} \| L_0 \|_* \]
\[ \text{s.t. } \| P_{\Omega}(L_0) - P_{\Omega}(M) \|_F < \delta \]


Robust principal component analysis

\[ L, S = \arg \min_{L_0, S_0} \| L_0 \|_* + \lambda \| S_0 \|_1 \]
\[ \text{s.t. } \| M - L_0 - S_0 \|_F \leq \delta \]


\[ L, S = \arg \min_{L_0, S_0} \| L_0 \|_* + \lambda \| S_0 \|_1 \]
\[ \text{s.t. } P_{\Omega}(L_0 + S_0) = P_{\Omega}(M) \]


What I am interested in :-)

\[ L, S = \arg \min_{L_0, S_0} \| L_0 \|_* + \lambda \| S_0 \|_1 \]
\[ \text{s.t. } \| P_{\Omega}(L_0 + S_0) - P_{\Omega}(M) \| \leq \epsilon \]

R. Paffenroth, P. Du Toit, R. Nong, L. Scharf, A. Jayasumana and V. Bandara
Space-time signal processing for distributed pattern detection in sensor networks

\[ M = L + S \]
The appropriate structures appear all over the place in real data!

Singular Values of Matrices

Insurance Satisfaction Surveys
The appropriate structures appear all over the place in real data!

Singular Values of Matrices

Amazon product communities

SKAION Internet Attack (e.g., DDoS) simulations
Abilene Internet2 Backbone
Abilene Internet2 Backbone
Abilene Internet2 Backbone
Abilene Internet2 Backbone
Abilene Internet2 Backbone
Enough math for the moment, let's try a really practical example

- **DARPA Lincoln Lab Intrusion Detection Evaluation Data Set**
  - IP sweep of the AFB from a remote site
  - Probe of live IP's to look for the sadmind daemon running on Solaris hosts
  - Breakins via the sadmind vulnerability, both successful and unsuccessful on those hosts
  - Installation of the trojan mstream DDoS software on three hosts at the AFB
  - Launching the DDoS

https://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html
Feature generation

Raw PCAP files

<table>
<thead>
<tr>
<th>Derived features</th>
<th>27 0-1 features</th>
<th>27 0-1 features</th>
<th>13 0-1 features</th>
<th>13 0-1 features</th>
<th>2 0-1 features</th>
<th>2 0-1 features</th>
<th>1 0-1 feature</th>
<th>1 0-1 feature</th>
<th>7 0-1 feature</th>
<th>1 numerical feature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unique source IP addresses</td>
<td>unique destination IP addresses</td>
<td>important system ports on the source side</td>
<td>important system ports on the destination side</td>
<td>distinguish ports below 1024 from those above 1024 on the source side</td>
<td>distinguish ports below 1024 from those above 1024 on the destination side</td>
<td>designate missing port number on the source side</td>
<td>designate missing port number on the destination side</td>
<td>various protocols (ICMP, sadmind, Portmap, TELNET, TCP, FTP, and HTTP)</td>
<td>number of bytes in the packet</td>
</tr>
</tbody>
</table>
Important idea... don't blindly follow theory

$$L, S = \arg \min_{L_0, S_0} \| L_0 \|_* + \lambda \| S \|_1$$

$$s.t. \| P_{\Omega} (L_0 + S_0) - P_{\Omega} (M) \|$$
Lincoln Labs DARPA Intrusion Detection Data Set - PCA

- IP sweep from a remote site,
- a probe of live IP addresses looking for a running Sadmind daemon,
- and then an exploitation of a Sadmind vulnerability.
Lincoln Labs DARPA Intrusion Detection Data Set - Comparison

RPCA – Too many false positives

PCA – Too many false negatives
Lincoln Labs DARPA Intrusion Detection Data Set - Comparison

Too “thick”

Too “thin”

Just right $\lambda$
Key idea

- Semi-supervised learning
  - PCA and RPCA have many parameters
  - Far too many to train on reasonably sized collections of attacks
  - Only train a **few important parameters** on supervised training data
    - Like $\lambda$
  - Gives better generalization and less over-fitting
Key idea

- Semi-supervised learning

Training data for $\lambda$

Algorithm not trained on this attack vector!
Other fun problems: LANDER

The LANDER project measures the number of “active” (i.e. respond to pings) on subnets across the Internet.

Same structure appears!

Can be used to pick out all LG DACOM subnets in Europe.

Subnets in anomaly:
[1, 210, 44, 0]
[1, 210, 173, 0]
[1, 219, 34, 0]
[1, 219, 206, 0]
[1, 218, 60, 0]
[1, 218, 121, 0]
[1, 218, 173, 0]
Other fun problems: CAIDA

Here is a small section of the 1.1 petabyte (and growing) CAIDA data set. It contains measurements of the worldwide Internet connectivity and latency (traceroute).

Same structure appears!
Big Data

Computer Science

Math

Equivalent formulation

\[ L, S = \arg \min_{L_0, S_0} \|L_0\|_* + \lambda \|S_0\|_1 \]
\[ \text{s.t. } \|P_\Omega(L_0 + S_0) - P_\Omega(M)\| \leq \varepsilon \]

\[ L_1, S_1 = \arg \min_{L_0, S_0} \|L_0\|_* + \lambda S_\varepsilon(S_0) \|_1 \]
\[ \text{s.t. } P_\Omega(L_0 + S_0) = P_\Omega(M) \]
\[ L = L_1, S = S_\varepsilon(S_1) \]
Big Data

Original algorithm. Rank=2, probability of corruption=2%, observations=10m and new algorithm!

Big Data

Hey, wait a minute...
How can this be? Math helps...
How can this be? Implementation helps...

Think about as distributed databases.
Distributed databases.
One meta-thought: The "Iron man" approach
Person AND Machine

http://www.independent.co.uk/arts-entertainment/films/reviews/iron-man-3-review-a-big-hand-for-downey-jr-but-movie-lacks-dramatic-mettle-8588873.html
Questions?

\[ Y = AU + BV + N \]

\[ YY^T = L + S + E \]

\[ YY^T \approx A\Sigma_{UU}A^T + B\Sigma_{VV}B^T + \Sigma_{NN} \]

\[ \min_{L,S} \|L\|_* + \lambda\|S\|_1 \]

s.t. \(|P_\Omega(M) - P_\Omega(L + S)| \leq \bar{\epsilon}\)