Signal Processing Based Intrusion Detection Using PCA

By Jeff Terrell
For COMP 290 - IDS - Spring 2005

General Introduction

- Several drawbacks to signature-based detection
  - Human intervention
  - Not adaptive; can't learn
  - Can be evaded by small changes
  - Fundamentally *can't* catch some attacks (like what?)

General Introduction

- Signal Processing (SP)-based methods:
  - Are more adaptive
  - Require less human intervention
  - Detect a broader range of attacks
  - Are much harder to apply!
- A real-time solution is an even bigger challenge

Outline

- Introduction to Principal Components Analysis (PCA)
- Singular Value Decomposition
- Eigenflows
- Detrending
- Subspace Method
- Characterization of Anomalies
- Conclusions

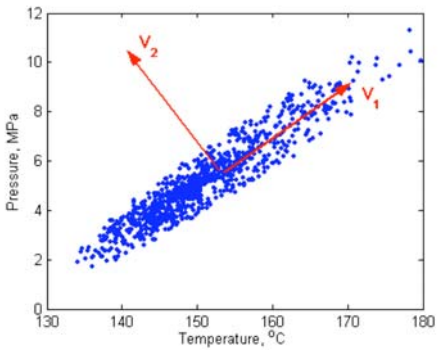
Introduction to PCA
Motivation

- Many ID/networking problems are high dimensional
  - Many studies stick to single end-to-end pair to keep dimensionality low
- The "curse of dimensionality": high dimensional problems are harder
- Decomposition into "normal" and "anomalous" components
  - Theme of signal-processing-based methods

Introduction to PCA
High-Level Overview

- PCA is like rotation in k-dimensional space
- New axes are most appropriate for data
- Lower-order axes capture most variation in data
  - Why is this (or more precisely its inverse) important?
    - Throw out the high-order axes!
    - Reduces dimensionality

Introduction to PCA
2-D example [1]



Introduction to PCA
Intuitive Examples

- Football - 1st axis along the length
- Piece of paper - "intrinsically" ~2D
- Faces - A 100x100 bitmap is 10,000D, but how many dimensions would we need optimally?
  - Answer: 42

Introduction to PCA
Geometric Details

- 1st axis captures greatest variation
  - In 2-D, what will the 1st axis be?
- 2nd axis captures greatest *remaining* variation
  - Remove 1st axis by "collapsing" data points into orthogonal (hyper) plane
- Rinse and repeat
- All axes must be orthogonal
  - Last axis is easy
- End result: rotation in k-D space

Introduction to PCA
Demonstrations

- http://www.uwlax.edu/faculty/will/svd/perpframes/index.html
- http://www.cac.sci.kun.nl/people/philipg/nfo-6/

Setup (from [2])

- Abilene traffic data used
- 11 Points of Presence (PoPs)
- 11^2 = 121 Origin-Destination (OD) flows
- Aggregation at 5 minutes for 1 week (2,016 intervals)

Setup (from [2])

- Measurement is the number of flows
- Thus X is 2016x121 data matrix
  - Column *i* is timeseries of *i*-th OD flow
  - Row *j* is vector of measurements at *j*-th interval
- Note the high dimensionality (121D)

Singular Value Decomposition

- Any matrix can be decomposed into 3 matrices: $U*S*V^T$
- V^T, 121x121, is PCA's rotation matrix (a *frame*)
- S, 121x121, is diagonal and contains ordered *singular values* λ_k
- U, 2016x121, contains our *eigenflows*

Singular Value Decomposition

- An eigenflow, U_i, is a 2016-vector, and there are 121 of them
- Each U_i is a component of the data
- Each OD-flow timeseries can be *completely* represented with a weighted sum of eigenflows
  - The weights are given in V^T

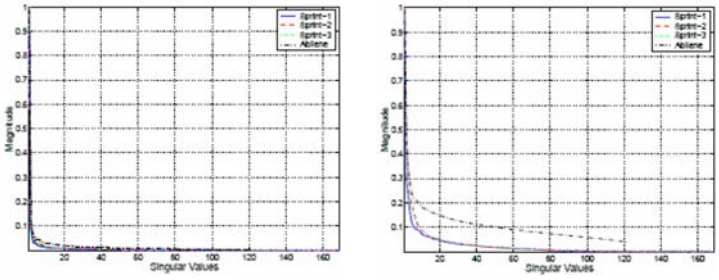
Singular Value Decomposition

- Recall: S, diagonal, contains λ_1 - λ_{121}
- λ_i's are arranged in decreasing order
- They are sqrt(eigenvalues) of $V*V^T$
- They represent amount of energy explained by component *i*
  - What does this say about our eigenflows?
    - They are arranged in decreasing order of importance

SVD - Scree Plots

- A *scree plot* is a plot of *i* vs. λ_i^2
- Useful for portraying relative importance of each λ_i

SVD - Scree Plots



SVD - Recap

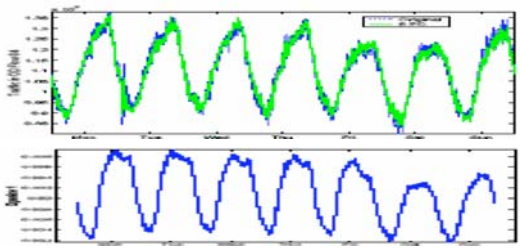
- $X = U*S*V^T$
- U_i = column of U = eigenflow
- S, diagonal, is singular values λ_i
- V^T is PCA's rotation matrix
- Singular Value *i* represents amount of energy captured by U_i

A Taxonomy of Eigenflows

- Deterministic (D-) eigenflows
  - Large trends
  - Periodic
  - Defined heuristically as having maximum frequency component at 12 or 24 hours

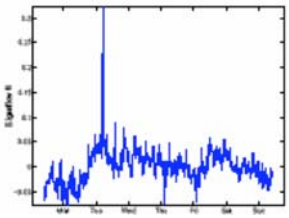
A Taxonomy of Eigenflows

D-eigenflow example



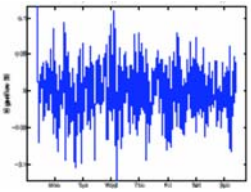
A Taxonomy of Eigenflows

- Spike (S-) eigenflows
  - Major element is at least 1 large spike
  - Defined heuristically as having at least 1 value more than 5 standard deviations from the mean

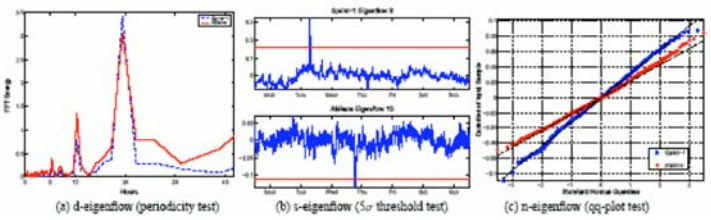


A Taxonomy of Eigenflows

- Noise (N-) eigenflows
  - Resembles Gaussian noise
  - Think of these as making up the leftover energy
  - Defined heuristically with a qq-plot



A Taxonomy of Eigenflows



(a) d-eigenflow (periodicity test) (b) s-eigenflow (5σ threshold test) (c) n-eigenflow (qq-plot test)

A Taxonomy of Eigenflows

- Where are we going with this?
  - We can now decompose each OD flow in terms of how deterministic, spiky, or noisy it is
  - Detrending
  - Forecasting

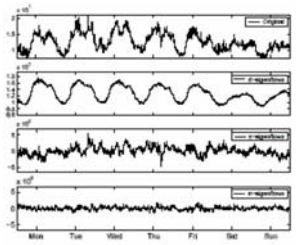


Figure 10: Decomposition of OD flow timeseries into the sum of its three constituent eigenflows.

A Brief Note on Stability

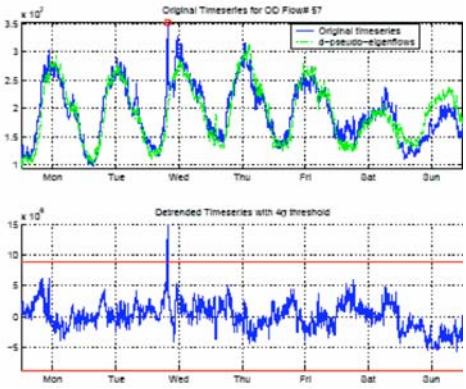
- Why would thresholding alone fail to detect anomalies?
  - We'd never detect an anomaly at 4 A.M.
  - We'd detect lots of anomalies at noon
- The timeseries is not *stable*...yet

Discussion

- Detrending: remove D-eigenflows from an OD flow
  - Now the timeseries is *stable*, so we can use simple thresholding to detect anomalies
- Forecasting: use most significant eigenflows of one trace to predict, say, next week's traffic
  - Identify anomalies this way

Discussion

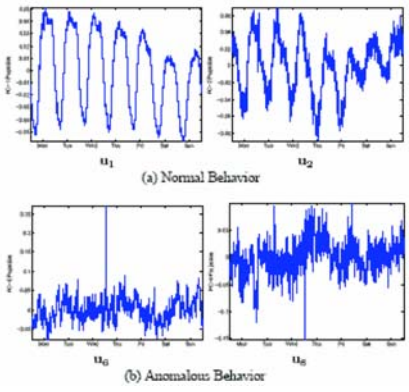
Or, do both at the same time!



Introduction to the Subspace Method (from [3])

- Very similar to detrending
  - Separation of "normal" from "anomalous"
- Mark first eigenflow with a value > 3 standard deviations from the mean
- This is the beginning of the "anomalous subspace"
- Everything prior is the "normal subspace"

Application of Subspace Method



Introduction to the Subspace Method

- Each OD-flow is completely characterized by normal and anomalous components
- So, we can remove the normal components, and examine the residuals

Applying the Subspace Method

- Let ***N*** be projection of data onto normal subspace (the modeled part)
- Let ***A*** be projection onto anomalous subspace (the residual part)

Applying the Subspace Method

- Similar to detrending, we can now just threshold on ***A*** to detect anomalies
  - Project each 121-D point onto ***A***
  - How could we tell how anomalous this projection is?
    - Euclidean distance from origin
- Heavy on statistics, but confidence intervals and such are involved

Discussion

- False positive rate and detection rate
  - False positive rate estimated with EWMA and other techniques
  - Detection rate estimated by injecting anomalies
- Feasibility of deployment onto actual networks

Setup (from [4])

- Same setup as before
- Except, now perform subspace method on byte, packet, *and* flow matrices
- Objective: after detection, characterize (and quantify) anomalies

Setup (from [4])

- We've seen how to catch anomalies by thresholding residuals
- This time, also catch anomalies in normal subspace with use of the t^2 statistic

Characterization of Anomalies

- By detecting coinciding anomalies in bytes, packets, and flows, can crudely classify the type of anomaly
  - Coinciding spike in bytes & packets may mean large transfer
  - Coinciding spike in flows & packets might be a network scan

Characterization of Anomalies

- By also checking for dominant sources or destinations, we can do better
  - DDoS is manifest as spike in F, P, or FP counts with a dominant destination
  - Most worms will manifest as spike in F counts with a dominant port

Discussion

- How might we distinguish between DDoS attack and flash crowd?
  - Paper says flash crowds usually dominated by a single OD-flow
- Even without bulletproof characterization, this is still a big help to network administrators

Concluding Remarks

- PCA and the subspace method are better in many ways than signature-based means of detection
  - Adaptive
  - No human intervention
- However, there are still plenty of improvements to be made

Concluding Remarks

- PCA and the subspace method are not the only signal-processing based methods of intrusion detection
- Others include:
  - Spectral analysis
  - Wavelet decomposition
  - Other SVD techniques

Questions?

1. http://www.mech.uq.edu.au/courses/mech4710/pca/s1.htm
2. "Structural Analysis of Network Traffic Flows" by A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft
3. "Diagnosing Network-Wide Traffic Anomalies" by A. Lakhina, M. Crovella, and C. Diot
4. "Characterization of Network-Wide Anomalies in Traffic Flows" by A. Lakhina, M. Crovella, and C. Diot