# Dominant and Deviant Pattern Detection in Event Traces for Intrusion Detection

**Ananth Grama**

CERIAS and Department of Computer Sciences
Purdue University.
ayg@cs.purdue.edu, http://www.cs.purdue.edu/people/ayg

Work done with Paul Ruth and Ioannis Ioannidis*.

(*Supported by a grant from CERIAS).

# The Problem:

In large audit files, it is useful to derive a notion of dominant (and thereby) deviant behavior. Dominant behavior is useful in provisioning resources and deviant behavior is important for flagging (possible) intrusions.

Deviants can be processed further using conventional intrusion detection techniques (tree-based classifiers, neural nets, etc.) or using thresholding techniques developed by Szpankowski et al. [See accompanying poster].

# Problem Formulation:

1. Compute dominant patterns.

2. Use the dominant patterns to determine patterns that are orthogonal (deviant) from these dominant patterns.

   This can be done by computing the angle (or cosine of the angle) and checking to see if it is close to orthogonal (or the dot product of event set and <u>every</u> dominant behavior set is close to zero).

# Computing Dominant Patterns: Problem Formulation.

Input:

Let $T_i$ be the set of events associated with user $i$ (or time interval $i$ / resource $i$ for alternate formulations).

The input $I$ is the set $\{T_1, T_2, ..., T_n\}$.
(set of events for each modeled entity).

Output:

Set $O = \{R_1, R_2, ..., R_m\}$, where $R_i$ is a set of events that are dominant in the input set.

Here, $m << n$ and a specified majority of input event sets are within prescribed distance from some representative in the output set.

# Computing Dominant Patterns: Notes.

- The problem as posed is NP complete for minimizing set $O$ over $m$.
- People have explored similar problems in such problems as frequent set computations in association rule mining and term co-occurrences in information retrieval.
- Most of these techniques are exponential in the dimension of the dominant pattern (the number of events in the dominant pattern).

## Computing Dominant Patterns: Algebraic Underpinnings.

Consider each event set ($T_i$ or $R_i$) as a binary (0/1) vector.

The objective is to determine a (much) smaller number of representative binary attributed vectors such that every vector in the original set is within a bounded (given) Hamming distance $\varepsilon$ from some vector in the representative vector set.

- The problem can also be thought of as error-bounded clustering or vector quantization over discrete (binary) spaces.
- However, current clustering techniques do not work on discrete spaces.

# Compressing Binary-Attributed Vectors:

## Example:

Attribute vector

$$\begin{array}{c} 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

## Can simply be represented as:

2: [0 1 1]

1: [1 1 1]

Notice that this induces a mapping of each vector to one of the representative vectors based on the Hamming distance. The grouping of vectors is inherent in this mapping. Also note that the representative vectors themselves quantify dominant group behavior.

# Compressing Event Vectors:

Consider the following rank-1 set of vectors (stacked together as a matrix in which each row represents an event vector):

$$T = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Since the order of the vectors is not important, this is equivalent to 4 event vectors each being [0,1,0,1,1].

<u>But</u>: Event vector sets are never rank-1.

# Compressing Event Vectors:

Decompose matrix into sequence of rank-1 matrices (Singular Value Decomposition or SVD)!

However:

- Singular vectors are orthogonal. This introduces negative values into the representative vectors!
- Singular vectors (rows) contain non-integral (non-discrete or continuous) values that often do not make physical sense for binary attributed vectors (what does it mean to have a 0.55 ping on a port).
- Non-integral column vectors are not interpretable either.

# Compressing Event Vectors:

- Use modified semi-discrete transforms for approximating the matrix.
- These transforms take a positive integer valued matrix and decompose it into an outer product of two 0/1 valued vectors.
- The corresponding singular values give the strength of the pattern.
- A single outer product induces a binary partitioning of the event vectors into those that are well approximated by the singular row vector (those rows that have a 1 in the corresponding column vector), and those that are not.
- The event set is recursively subdivided until the required fraction of event sets are well modeled.

# Computing Modified Semi-Discrete Decompositions.

$$F_k(d, x, y) \equiv \left\| A - dxy^T \right\|_F^2$$

$$F_k(d, x, y) = \|A\|_F^2 - 2dx^T Ay + d^2 \|x\|_2^2 \|y\|_2^2$$

At the optimal solution, $\frac{\partial F}{\partial d} = 0$ ,therefore,

$$d^{min} = \frac{x^T Ay}{\|x\|_2^2 \|y\|_2^2}$$

Substituting in previous equation, this yields:

$$F_k(d^{min}, x, y) = \|A\|_F^2 - \frac{(x^T Ay)^2}{\|x\|_2^2 \|y\|_2^2}$$

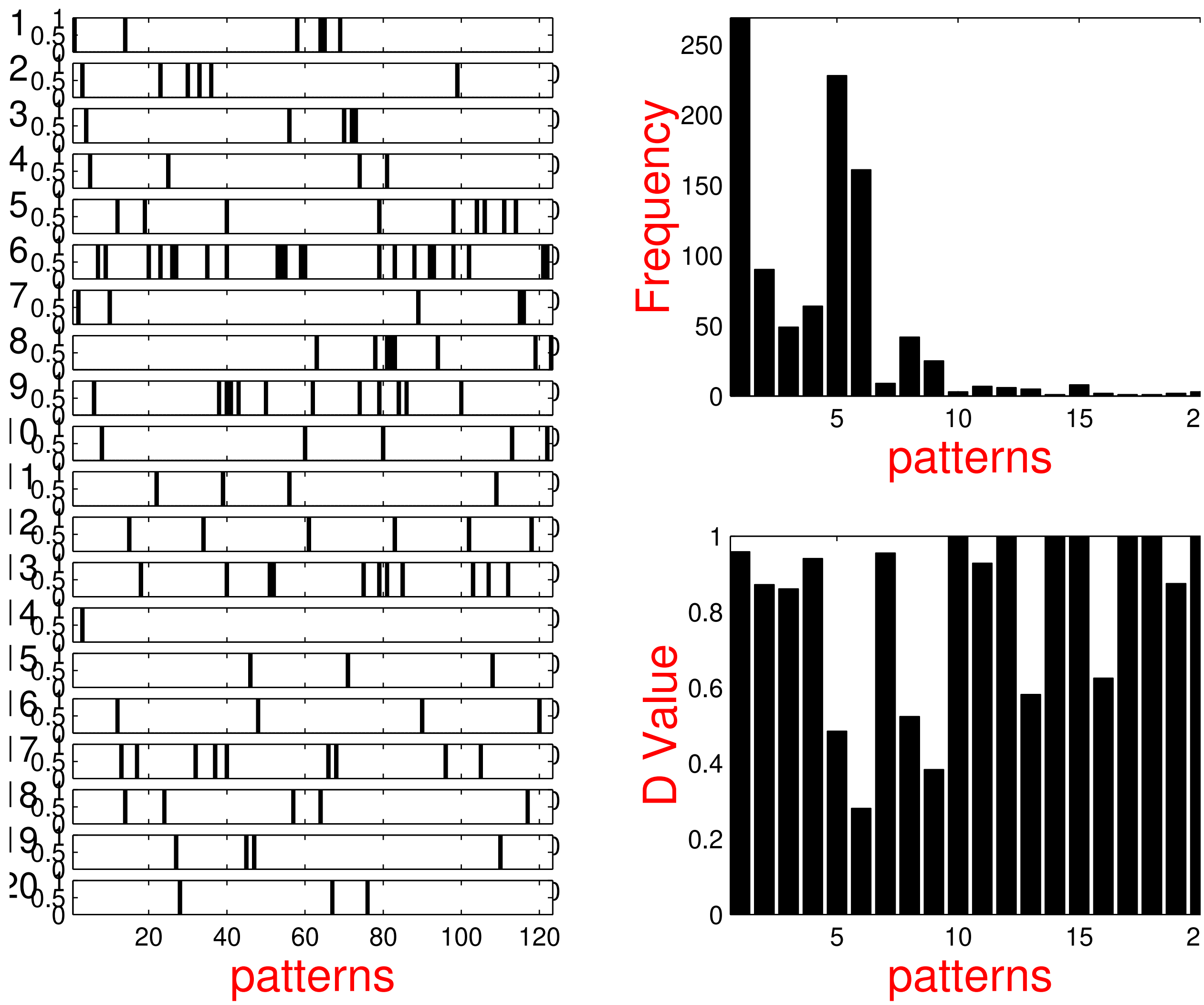
or, we need to maximize:

$$max\tilde{F}_K(x, y) \equiv max\frac{(x^T Ay)^2}{\|x\|_2^2 \|y\|_2^2}$$

# Computing Modified Semi-Discrete Decompositions.

- Unfortunately, computing this maximization is very difficult.
- Conventional methods rely on alternating procedures (fix $x$, compute $y$, use this $y$ to compute $x$, alternate until convergence).
- However, this minima is not guaranteed to be a global minima.
- This process can be improved by using alternate techniques to compute good starting points for $x$.
- Thresholded singular vectors as starting points for modified semi-discrete vectors.
- Use clustering and cluster centeroids as starting points.
- Using these, we have developed algorithms and software that can compute modified semi-discrete vectors of dimension 100K or more with 1M event sets in a few seconds.

# Experimental Evaluation.




Dataset generated from IBMs synthetic embedded pattern data generator (www.almaden.ibm.com).

# Ongoing work.

- Post-processing deviants to signal intrusion.
- Optimized algorithms for dynamically maintaining representative event sets.
- Associating deviants with likelihoods of their occurrence.
- Full scale production system.