Online Anomaly Detection for Optical Networks

Tarem Ahmed and Mark Coates

McGill University

tahmed@mail.mcgill.ca, coates@tsp.ece.mcgill.ca

Introduction

- High-speed optical backbones are constantly hit by wide variety of types/classes of network anomalies, from DOS attacks and viruses to large data transfers and equipment failures.
- Need online and instantaneous, anomaly detection method.
- We propose an online algorithm based on kernel version of Recursive Least-Squares (RLS) algorithm.
- Different anomalies affect network in different ways; it is not always possible to know *a priori*, how a potential anomaly would exhibit itself.
- Our algorithm learns characteristics of normal traffic behavior, then raises alarm immediately upon encountering a deviation from the norm.
- We test on data from the Abilene backbone network, and compare with the offline, block-based algorithm based on Principal Component Analysis (PCA) [1].

The Kernel Approach

- Data sequence is of form:

  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots\}, \quad \mathbf{x}_i \in \mathrm{X}, y_i \in R$

  where $\{\mathbf{x}_i\}_{i=1}^{t-1}$ are training vectors, and y_t is a function of $\{\mathbf{x}_i\}_{i=1}^{t-1}$.
- Represent predictor as

  $$\hat{y} = \sum_{i=1}^{t} \alpha_i \cdot \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}) \rangle = \sum_{i=1}^{t} \alpha_i \cdot \mathtt{kernel}(\mathbf{x}_i, \mathbf{x})$$

  where φ represents the mapping from input space to feature space, and $\boldsymbol{\alpha}$ is the weight vector that minimizes the L2 error.
- Direct knowledge of φ is **not** required. Only inner product of ***mapped*** feature vectors is required, which is provided by the chosen kernel function.
- **Problem**: once new data $\mathbf{x}_t$ arrives, dimension of $\{\varphi(\mathbf{x}_1) \cdots \varphi(\mathbf{x}_t)\}$ may keep increasing over time, unless $\varphi(\mathbf{x}_t)$ is linearly dependent on $\{\varphi(\mathbf{x})_i\}_{i=1}^{t-1}$.
- **Solution**: define notion of **approximate** linear independence as in [2]:

  $\varphi(\mathbf{x}_t)$ is said to be ***approximately*** linear dependent on $\{\varphi(\mathbf{x}_1) \cdots \varphi(\mathbf{x}_t)\}$, if the following condition holds:

  $$\min_{a} \left\| \sum_{i=1}^{t-1} a_i \cdot \phi(\mathbf{x}_j) - \phi(\mathbf{x}_t) \right\|^2 \leq \nu \qquad (1)$$

  for some optimum **sparsification** coefficient vector **a**.

- **Objective**: Build a **dictionary** of input vectors $D = \{\tilde{\mathbf{x}}_j\}_{j=1}^{m}$, such that $\{\varphi(\tilde{\mathbf{x}}_j)\}_{j=1}^{m}$ forms an approx. linearly independent basis for the feature space.

Application to optical networks

- Data is normalized timeseries of number of packets or bytes, in flows between core routers in Abilene network.
- $\mathbf{x}_t$ defined to be vector giving number of packets or bytes in each flow at time *t*.
- y_t defined to be the total number of packets or bytes in the network at time *t*.

The Detection Algorithm

- **Initialize** at t = 1, by entering $\mathbf{x}_1$ into dictionary.
- **Iterate** for t = 2,3,…

Step 1: New dat arrives. Evaluate the approximate linear dependence of $\varphi(\mathbf{x}_t)$ on the dictionary at time t:

$$\delta_t = \min_{a} \left\| \sum_{j=1}^{m_{t-1}} a_j \phi(\tilde{\mathbf{x}}_j) - \phi(\mathbf{x}_t) \right\|^2 \qquad (2)$$

Step 2: Compare δ_t with thresholds ν_1 and ν_2, where $\nu_1 < \nu_2$:
- If $\delta_t > \nu_2$, new input vector is **very** far away from space spanned by dictionary. Conclude that this is an anomaly, raise **red alarm**. **No change** to dictionary.
- If $\nu_1 < \delta_t < \nu_2$, new input vector not sufficiently explained by dictionary. **Add** $\mathbf{x}_t$ to dictionary, raise **orange alarm**.
- If $\delta_t < \nu_1$, new input vector falls into normal subspace. **No alarm**. **No change** to dictionary.

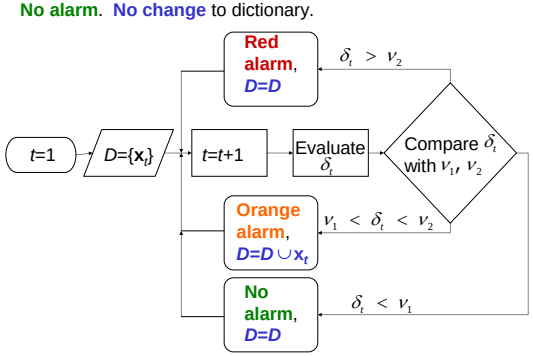

Fig. 1: Outline of online anomaly detection algorithm.

Key idea: After sufficient training, dictionary should span normal subspace.

Results

- Example: Data from 11 core Abilene routers, for 1152 5-min intervals from 0000hrs Sep. 1, 2005 to 2359hrs Sep. 4, 2005.
- Using linear kernel: $\mathtt{kernel}(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$.

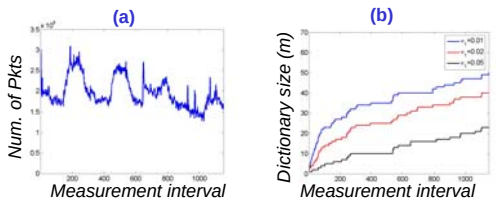

Fig. 2: **(a)** Variation in total number of packets in network; **(b)** growth in *D* for various values of ν_1, with $\nu_2 = 6\nu_1$.

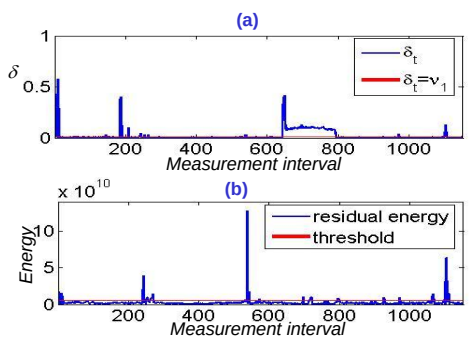

Fig. 3: Comparing **(a)** δ_t in proposed algorithm with $\nu_1 = 0.01$ and $\nu_2 = 6\nu_1$, with **(b)** energy in residual subspace using block-based PCA from [1]. Spikes represent anomalies.

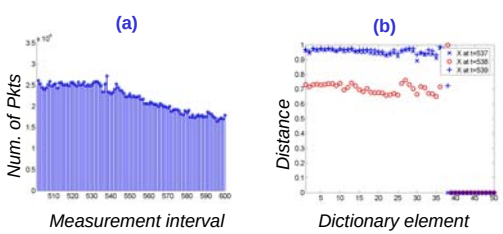

Fig. 4: Example anomaly at *t* = 538. Not easily seen in **(a)** timeseries of packets, but **(b)** obvious by observing distance of $\mathbf{x}_t$ from each dictionary member.

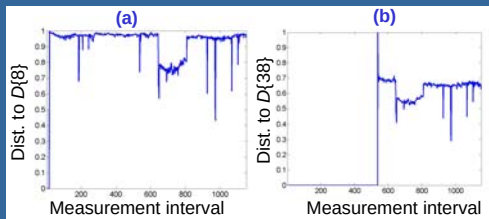

Fig. 5: Distance of $\mathbf{x}_t$ from **(a)** a normal dictionary member, and **(b)** an anomalous input vector that was admitted to dictionary.

Conclusions and Future Work

- Algorithm is recursive, there is no need to relearn from scratch when new data arrive.
- Dictionary size (*m*) is seen to stabilize after about 500 timesteps.
- Storage requirement and complexity bounded by $O(m^2)$, i.e. independent of time.
- Performance comparable to accepted offline, block-based PCA method in [1].
- Work in progress includes controlling dictionary by enabling dropping of obsolete or anomalous elements, and confirming anomaly in case of **orange alarm** if relevant $\mathbf{x}_t$ is distant from subsequent input vectors.
- Future work involves letting the data determine the thresholds ν_1 and ν_2.

Acknowledgements

R. Summerhill and M. Fullmer at Abilene provided access to raw data.

References

[1] A. Lakhina, M. Crovella and C. Diot, *Diagnosing Network-Wide Traffic Anomalies*, ACM SIGCOMM, Portland, OR, August 2004.

[2] Y. Engel, S. Mannor and R. Meir, *The Kernel Recursive Least Squares Algorithm*, IEEE Trans. on Sig. Proc., 52(8), pp.2275--2285, 2004.