On Predictability of System Anomalies in Real World

Yongmin Tan, Xiaohui (Helen) Gu
Department of Computer Science
North Carolina State University
{ytan2@ncsu.edu, gu@csc.ncsu.edu}
Motivation

- SLO violations
- Hotspots
- S/H failures
- Bottleneck
- ·····

Challenge: Can we provide automatic anomaly management?
Existing Approaches

- **Reactive approach**
  - Take corrective actions *after* an anomaly happens

- **Proactive approach**
  - Take preventive actions on *all* system components

- **Predictive approach**
  - Predict system anomalies *before* they happen

[ICDCS08,ICDE09,PODC10]

**Question:** Whether real system anomalies do exhibit predictability?
Approach Overview

- Combine anomaly classifier with feature value predictor
- Prediction is application-agnostic

Anomaly detection

Anomaly can be:
- Performance SLO violations
- Host ping failures
- Hard drive failures

Exemplary anomaly predicates:
- Throughput < 10 requests/s
- User response time > 1 s
- Ping response time > 10 s
**System Design**

- **Feature evolving model**
  - Predict future value with probabilities

- **Statistical classifier**
  - Identify anomaly symptoms

- **Integrated predictor**
  - Estimate probability of future system normal/abnormal state
Feature Evolving Pattern Model

- **Goal**: Predict the values of system metrics at a future time
- **Discrete-time Markov chain (DTMC)**
  - Build one model for each metric
  - A set of discrete bins

- **Output**: Multi-step state transition matrix

CPU usage (%) ranging from 0 to 100 with three discrete states
Data Preprocessing - Discretization

- **Equal-width approach**
  - Divide feature values into a set of equal-width bins
  - Preserve original value distribution
  - Sensitive to outliers

- **Equal-depth approach**
  - Put equal number of samples in each bin
  - Balance sample distribution in different bins
  - May lose original value distribution
Discretization - A Hybrid Approach

- Hybrid approach
  - Combine equal-width and equal-depth approach

Step 1: equal-width (M bins) → Step 2: equal-width (2M bins) → Step 3: merge small bins → End
Statistical Anomaly Classification

- **Naïve Bayesian Classifier**
  - Naïve assumption:
    Each metric is independent given the anomaly class label C

  \[
P(\text{C} = \text{abnormal/normal})
  \]

  \[
P(\text{C} = \text{normal}) \rightarrow \prod P(x_i | \text{C}) \cdot P(\text{C})
  \]

  \[
P(\text{C} = \text{abnormal}) \rightarrow \prod P(x_i | \text{C})
  \]

  - Example:
    \[
P(\text{normal} | \mathbf{x}) = 0.05
    \]
    \[
P(\text{abnormal} | \mathbf{x}) = 0.2
    \]
    raise alert!

- **Tree-augmented Bayesian Network**
  - Relax the naïve assumption:
    Consider inter-metric dependencies but with at most one parent other than C

  \[
P(\text{C} = \text{abnormal/normal})
  \]

  \[
P(\text{C} = \text{normal}) \rightarrow \prod P(x_i | \text{C}) \cdot P(\text{C})
  \]

  \[
P(\text{C} = \text{abnormal}) \rightarrow \prod P(x_i | \text{C})
  \]

  - Example:
    \[
P(\text{normal} | \mathbf{x}) = 0.3
    \]
    \[
P(\text{abnormal} | \mathbf{x}) = 0.1
    \]
    Don’t raise alert!
**Integrated Anomaly Prediction**

- **Goal**: Perform classification on future predicted data
- Deterministic values -> probabilistic values, i.e. sum up all possible paths

Naïve Bayesian Classifier with lead time $T=3$; metric $X,Y$ have three states ($S=3$)

- $P(0|x)[T] = 0.67$
- $P(1|x)[T] = 0.33$

Do not raise alert.
Real-world Systems for Evaluation

- **PlanetLab**
  - Host ping failures, SSH failures, Sensor program failures
  - 66 host metrics such as CPU load, virtual memory states

- **SMART**
  - 369 hard disk records, 191 of them are “failed” disk
  - 59 SMART attributes

- **IBM System S**
  - Performance anomalies (SLO violation)
  - 21 system-level metrics
Evaluation Metrics

- **Mean prediction error (MPE)**
  - Future value prediction accuracy
- **ROC curve**
  - Performance of the anomaly classifier
- **True positive /false positive rate against lead time**
  - Performance of the integrated anomaly predictor
  - TP: predict an anomaly and it actually happens within the lead time.
  - FP: predicted anomaly does NOT happen within the lead time.
Experimental Results - PlanetLab: 1. Raw Failure Data Trace

ping failure snapshot

![Graph showing Normal and Abnormal data over time for Availability and Load1 metrics](image)

NC STATE UNIVERSITY Computer Science
Experimental Results - PlanetLab:

2. Markov Prediction Error

discretization scheme

discretization granularity
Experimental Results - PlanetLab: 3. Classification ROC Curves

Naïve Bayesian

TAN
Experimental Results - PlanetLab:
4. Anomaly Prediction Accuracy

Naïve Bayesian + Markov

TAN + Markov
Experimental Results - SMART: 1. Raw Failure Data Trace

- **ReadError3**
  - Data sample
  - Normal
  - Abnormal

- **Servo10**
  - Data sample
  - Normal
  - Abnormal

- **ReadError18**
  - Data sample
  - Normal
  - Abnormal

- **Write**
  - Data sample
  - Normal
  - Abnormal
Experimental Results - SMART:
2. Markov Prediction Error

discretization scheme

discretization granularity
Experimental Results - SMART:
3. Classification ROC Curves

Naïve Bayesian

TAN
Experimental Results - SMART:
4. Anomaly Prediction Accuracy

Naïve Bayesian + Markov

TAN + Markov
Overhead Measurement

training time

prediction time
Conclusions

- Integrated anomaly prediction scheme combining feature evolving model and anomaly classifier
- First attempt to quantify the predictability of real world system anomalies

• Several findings:
  – Real world system anomalies exhibit predictability
  – Markov predictor can achieve reasonable accuracy
  – Intelligent discretization scheme improves accuracy
  – Simple classifiers can achieve good accuracy
Acknowledgement

- This work was sponsored in part by NSF CNS0915567 grant, NSF CNS0915861 grant, U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI), IBM Exploratory Stream Analytics Award, and IBM Faculty Award.

- Our source code and dataset are available upon email request.

- Project webpage: http://dance.csc.ncsu.edu/projects/sysMD/