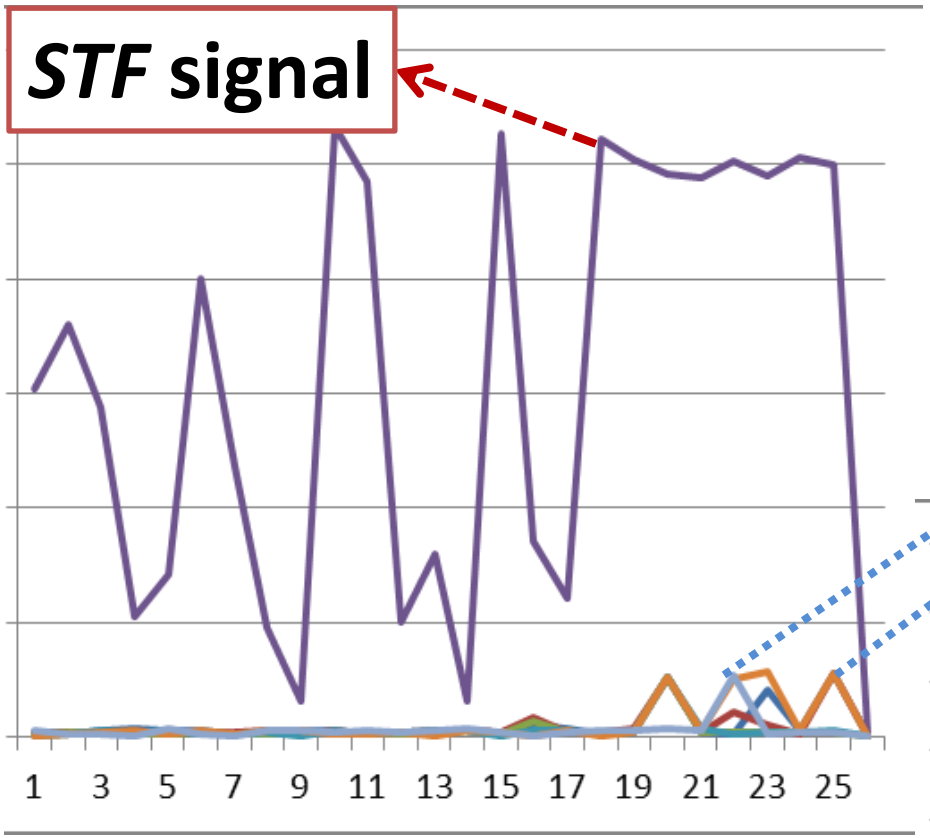
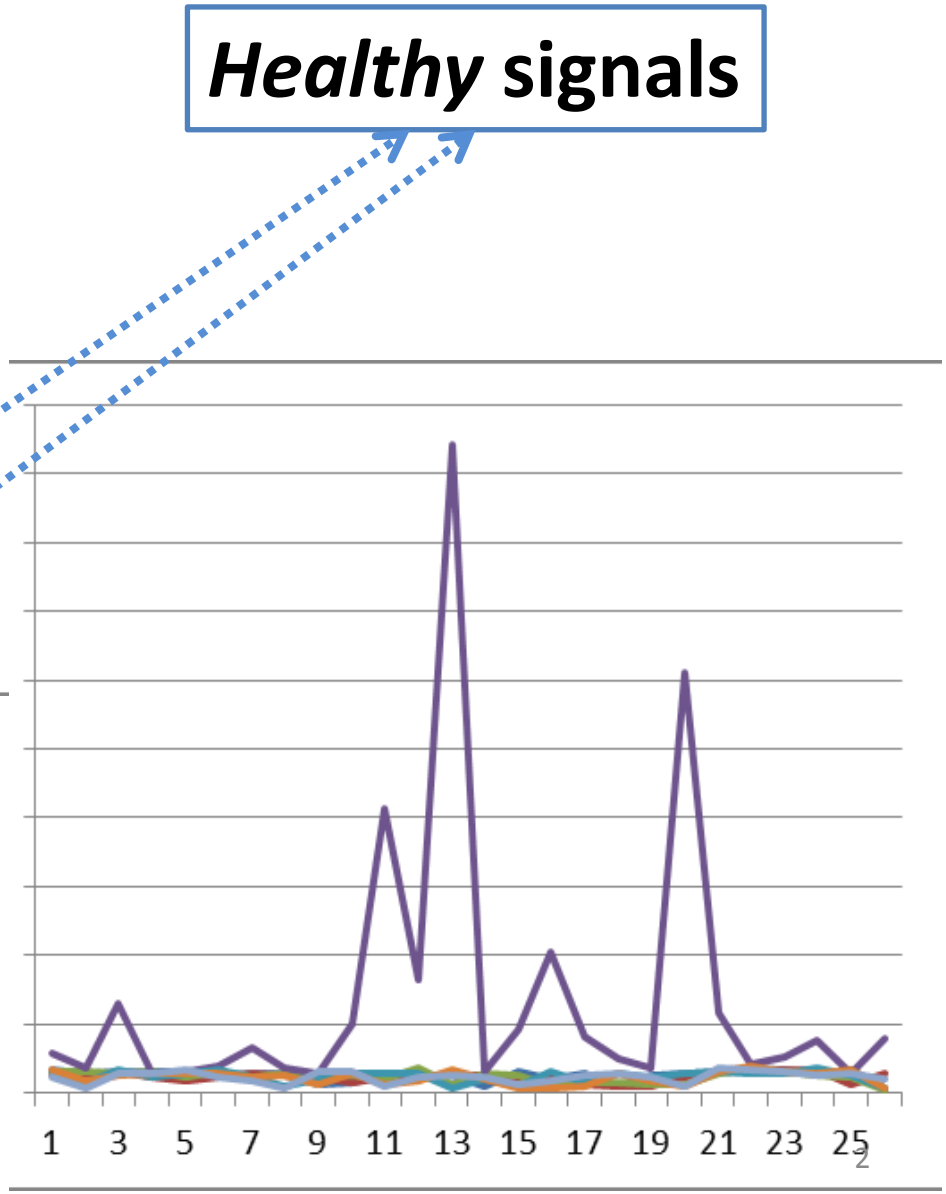
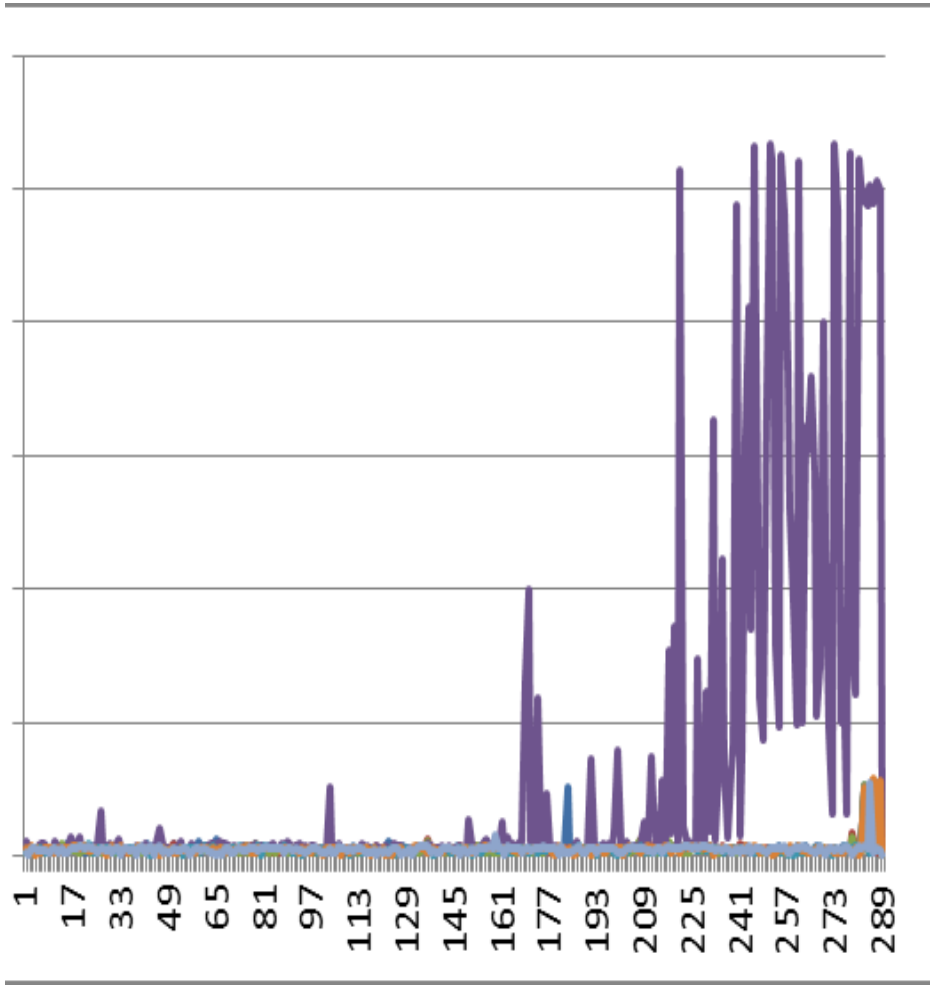
Finding soon-to-fail disks in a haystack

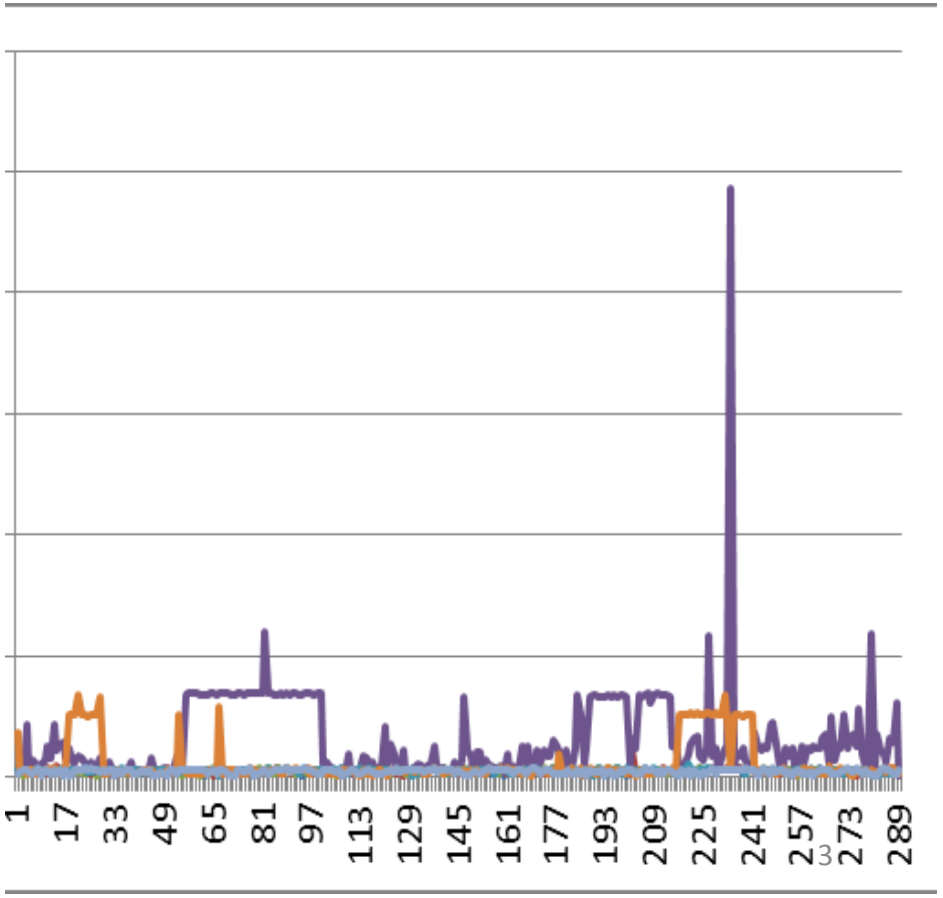
Moises Goldszmidt
Microsoft Research


STF signal
1 3 5 7 9 11 13 15 17 19 21 23 25


Healthy signals
1 3 5 7 9 11 13 15 17 19 21 23 25



4 days before failure



Objectives….

- Illustrate/demystify *ONE* statistical machine learning approach to build a detector for these signals
- Quantify the worth of performance signals for prediction
- Inspire further research

Set up

- You want to predict disk failures (candy)

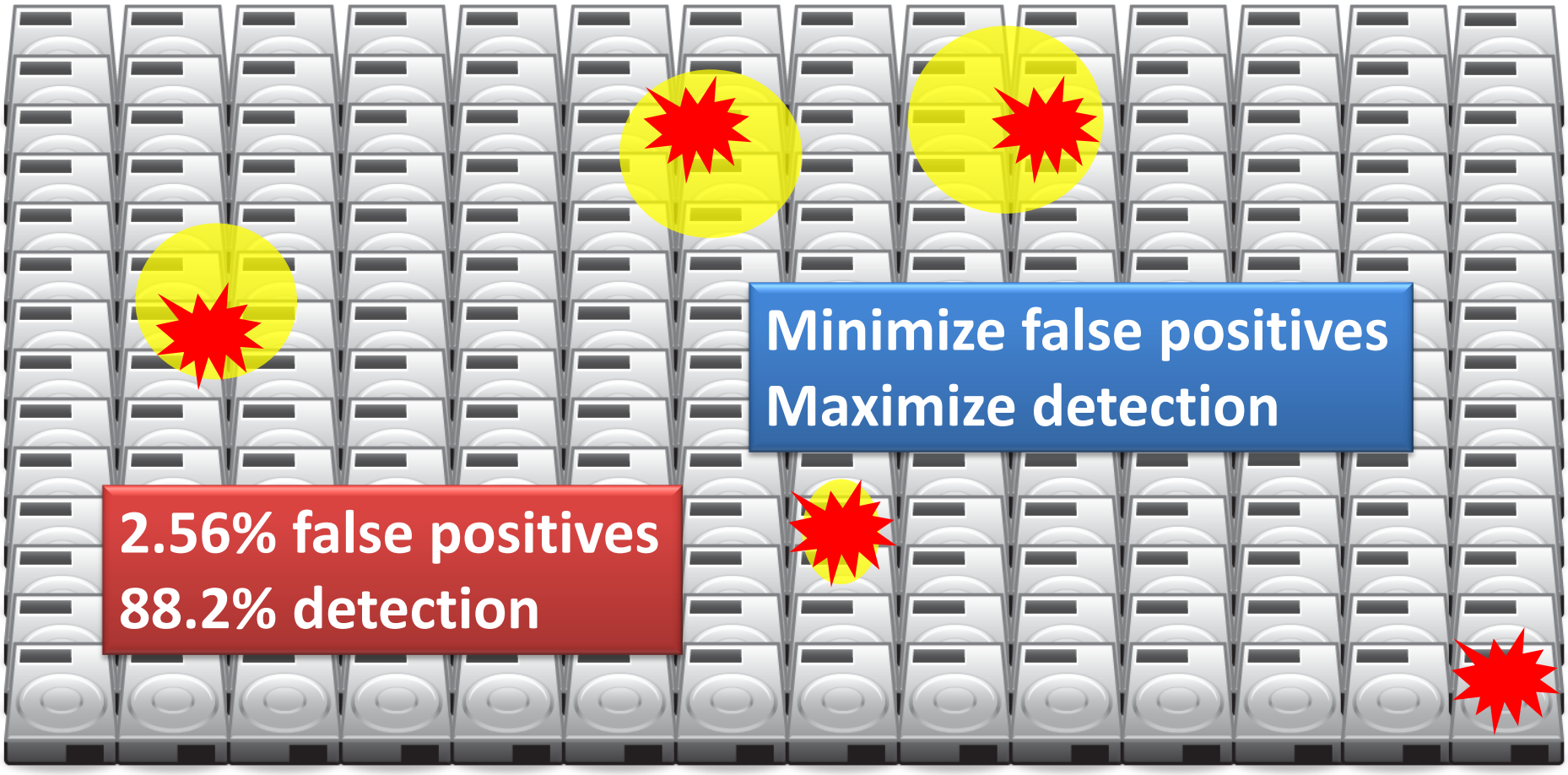
Disk failure prediction is important !!!

- Prediction involves a tax (false positives)
- Decision on whether and how to act depends on the tradeoff
  How much tax for the candy…

Initial state….

Manufacturer says → XXX% will fail

Wanted!


Minimize false positives
Maximize detection
2.56% false positives
88.2% detection

Related work….

But Google said:

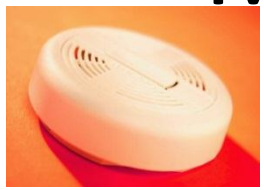
"It is unlikely that we can find predictive signal in SMART diagnostic data"

We are going to use performance data

Preliminaries

- The Disks
  - Cluster of 144 servers
  - Each server has 34 drives in an array
  - Drives = 750GB SATA
  - One month of data
- Signals collected every 20 minutes
- **Sol**: Average-Max-Latency (AML)
- **Failure**: Disk serial-number changed

Methodology

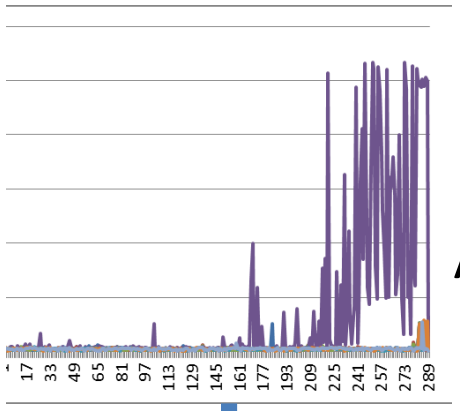


Train

Test

Estimate model parameters

Estimate model performance



AML signal

$P_{stf}(D)$

Model Comparison

$P_H(D)$

Threshold

Peak Counter

$Log(P_{stf}(D)/P_H(D))$

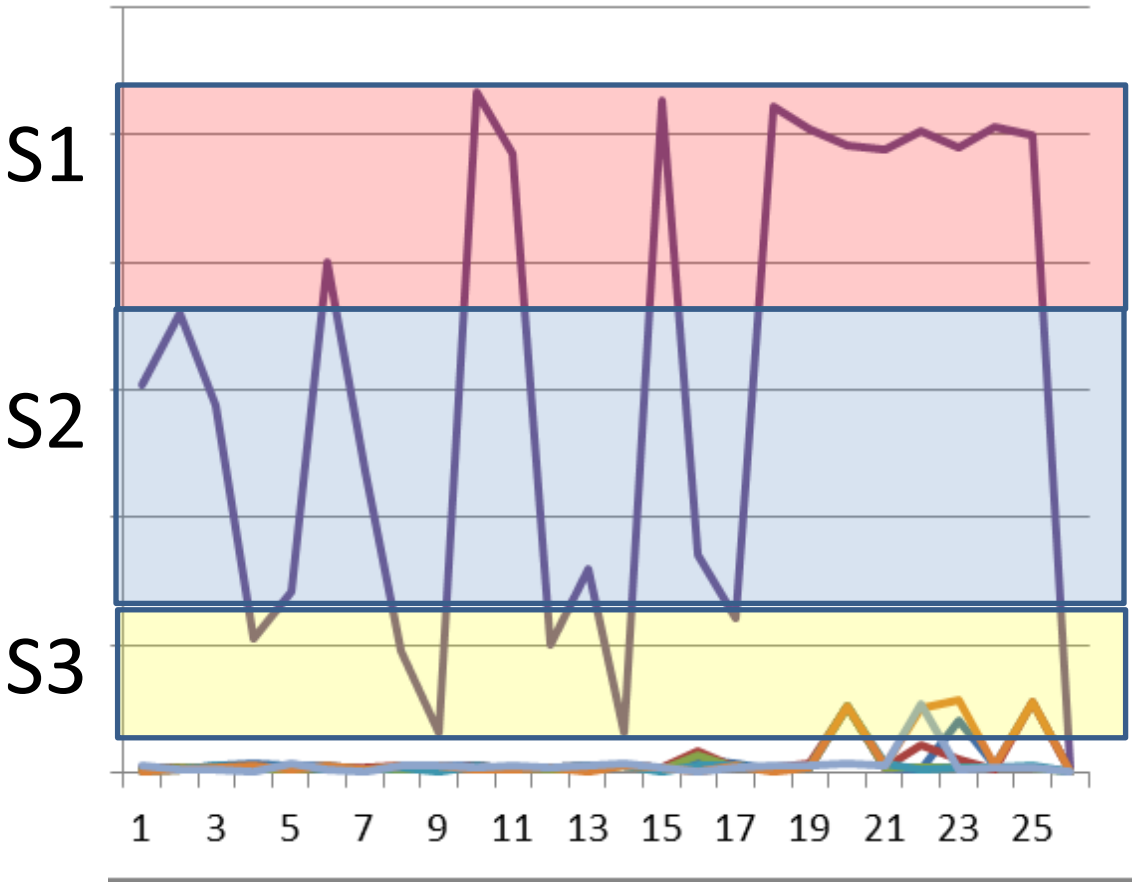
How-Many

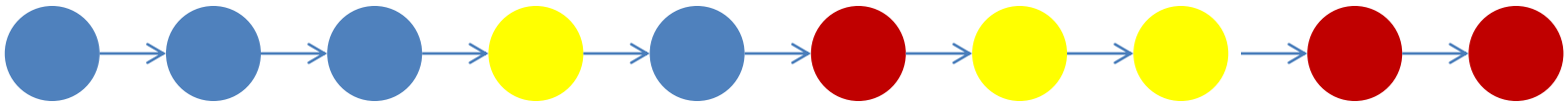
Fusion

Logistic Regression

Alarm

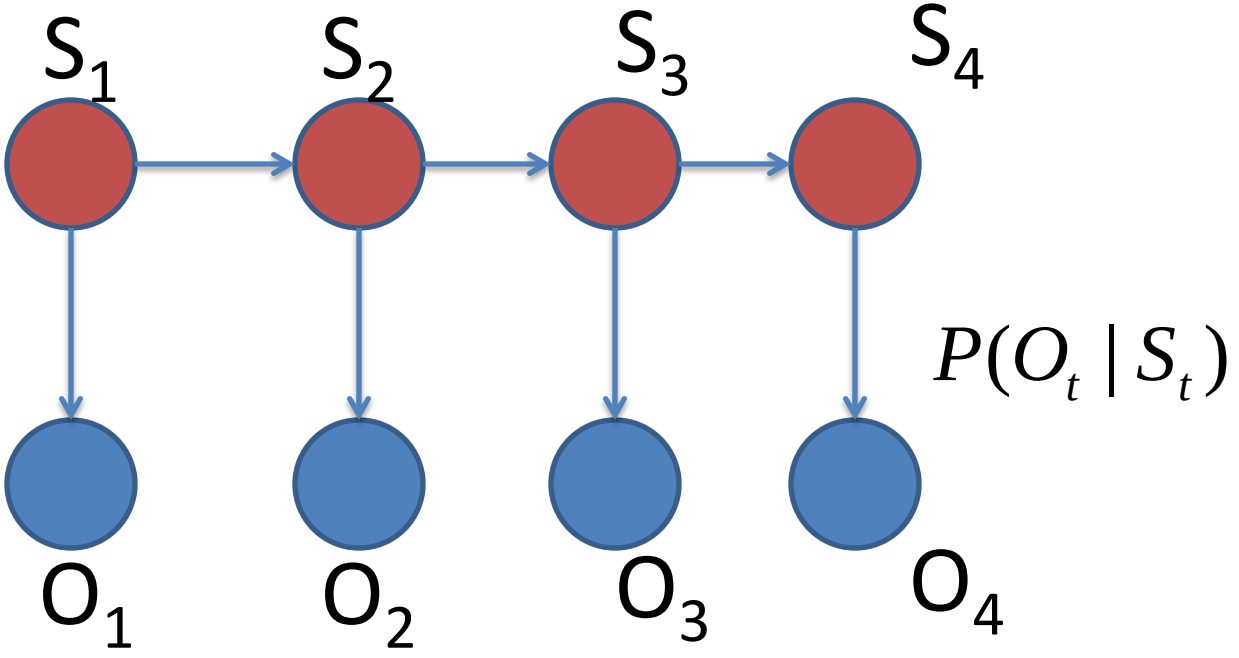
Intuition about the HMM model…





Hidden Markov Models

$P(S_t \mid S_{t-1})$



$P(O_t \mid S_t)$

Parameters

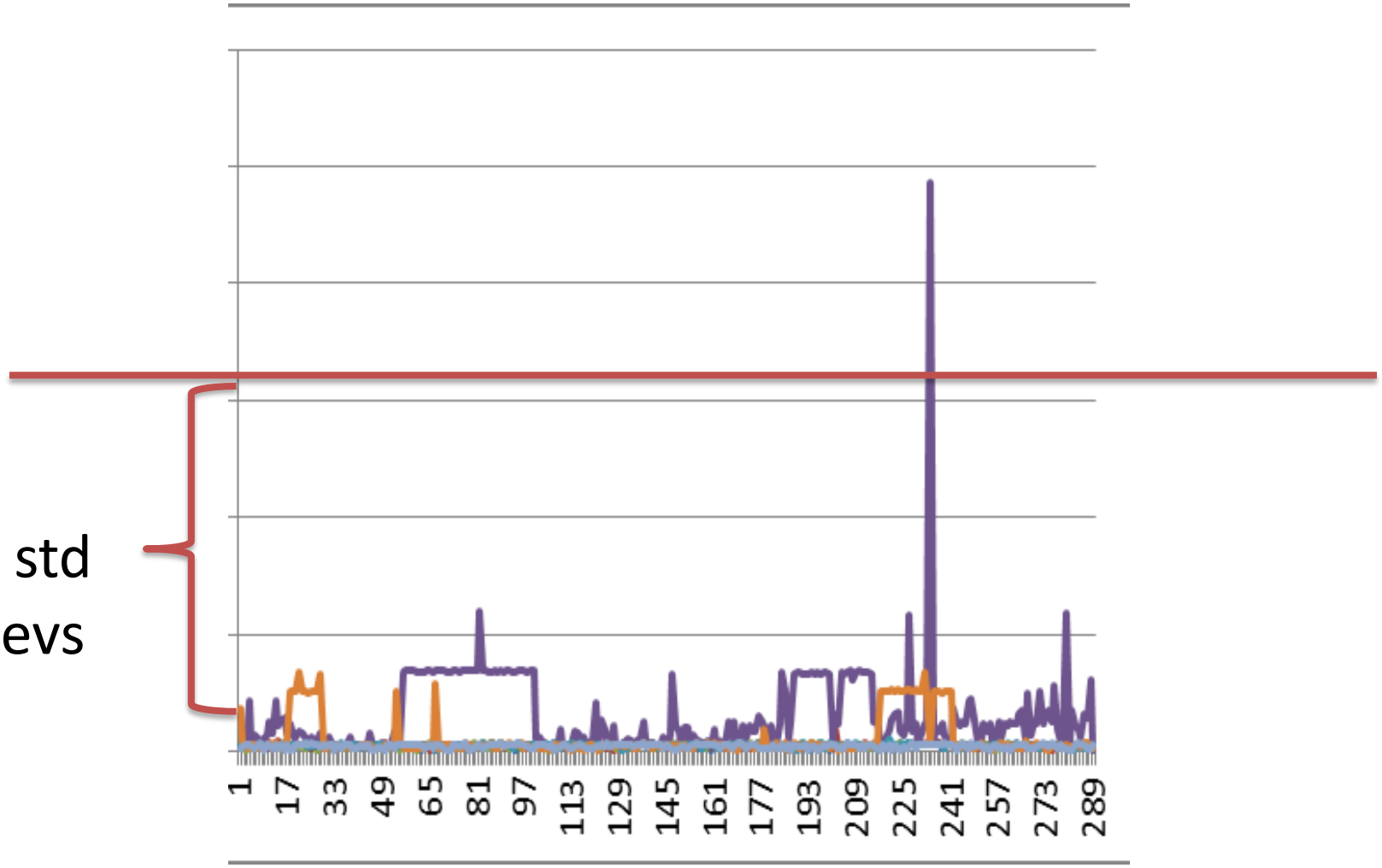
$$P(O_t \mid S_t) \sim N(\mu_S, \sigma_S)$$

$$P(S_{t+1} \mid S_t) \sim Multi(\theta_{S_{t+1}, S_t})$$

# of states – use BIC score and linear search

	Healthy	STF	
S1	16,5	99,9	
S2	2,5	18,5	
S3	0,4	4,2	
Healthy	S1	S2	S3
S1	76%	23%	1%
S2	0.80%	99%	0.20%
S3	0.50%	2.50%	97%
STF	S1	S2	S3
S1	85%	7%	8%
S2	7%	63%	30%
S3	4%	35%	61%

Peak detector



Logistic regression

$$\log\left(\frac{P(STF)}{P(H)}\right) = \beta_0 + \beta_1 * LogOdds + \beta_3 * HowMany$$

Putting it all together…

- Divide the data into 2 sets of 1190 disks with 17 failed disks (each)
- Train HMM with 24 hours of 12 Failed and 48 Healthy disks
- Add 200 disks and train Logistic Regression
- Add 900 to set threshold to minimize FPs
- Run on the test set of 1190 disks

How the whole thing works

24 hours

4 hours

Counting

- Predicted: If detector sounds alarm and disk is changed before the end of the month
- False Positive: If detector sounds alarm and disk is not changed before the end of the month

Results

- Test data (estimate):
  - 15 out of 17 - 88.2% detection
  - 30 – 2.56% false positive
  - Between 2 weeks and 8 hours
- Workload stable
- No solid correlation to SMART

Life Ref Time	Reset Ref Time	Perf Collection Interval
Life Sectors Read	Reset Sectors Read	Perf Read Requests
Life Hard Reads	Reset Hard Reads	Perf Write Requests
Life Retry Reads	Reset Retry Reads	Perf Max Queue Depth
Life ECC Reads	Reset ECC Reads	Perf Average Latency
Life Sectors Written	Reset Sectors Written	Perf Maximum Latency
Life Hard Writes	Reset Hard Writes	Perf Maximum Wait Time
Life Retry Writes	Reset Retry Writes	Error Log Total Errors
Life Used Reallocs	Reset Used Reallocs	SMART Reallocated Sectors
Life Timeouts	Reset Timeouts	SMART Pending Reallocs
Life Pred Fail	Reset Pred Fail	SMART ECC Errors

Summary

- Presented a detector of soon-to-fail disks
  - 88.2% of detection with 2.56% of false positives
  - Input: performance signal – Output: alarm
- Future
  - Why doesn't it correlate?
  - Other class of disks?
  - Complete the system (migration, etc)