

Bilinear Logistic Regression for Factored Diagnosis Problems

Sumit Basu¹, John Dunagan^{1,2}, Kevin Duh^{1,3}, and Kiran-Kumar Munuswamy-Reddy^{1,4}

¹Microsoft Research

²Microsoft

³NTT Labs

⁴Harvard University

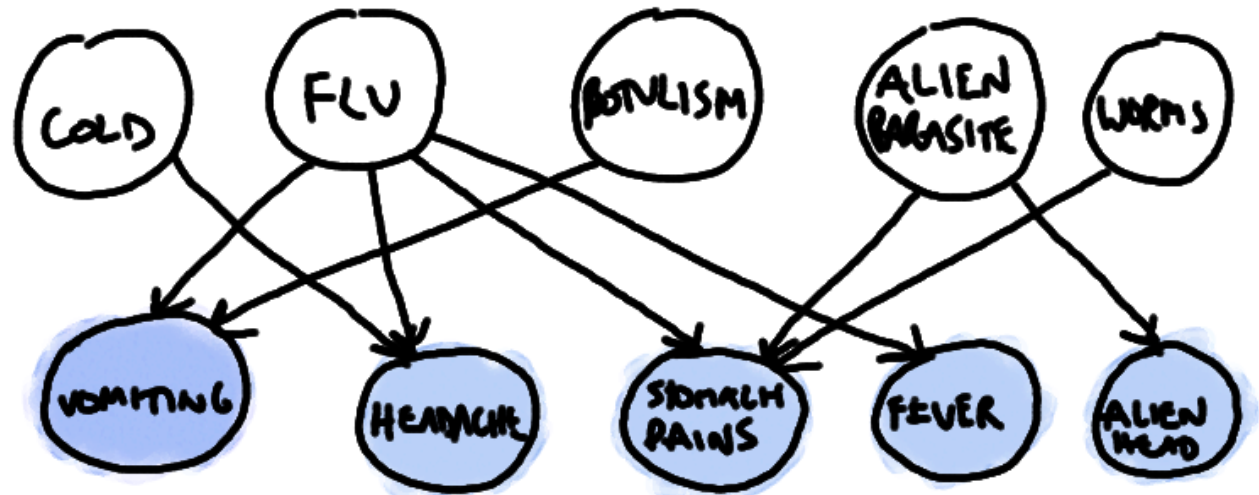
Note: if you use content from these slides in your presentations/papers, please attribute it to: S. Basu, J. Dunagan, K. Duh, and K-K. Munuswamy-Reddy. "Bilinear Logistic Regression for Factored Diagnosis Problems." In *Proceedings of SLAML 2011*. Cascais, Portugal. October, 2011.

Goals of this Talk

- A New Way of Looking at Diagnosis
 - For problems with a large number of uniform entities with uniform features that fail as a whole
 - “Factored Diagnosis”
 - A method, BLR-D, for approaching such problems
- Some Useful Statistical Tools (for any method)
 - Figuring out which parameters matter
 - Estimating false alarm rates **without labels**

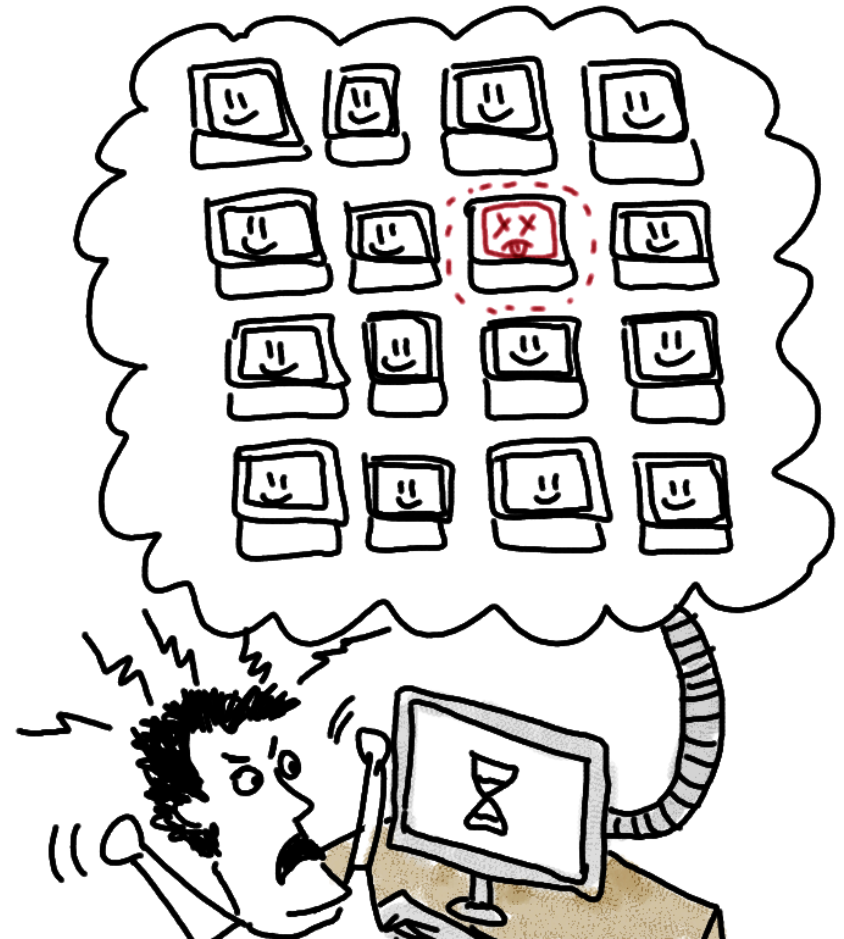
Forms of Diagnosis Problems

- “Clinical” Diagnosis
 - “Bob has stomach cramps and a high fever”
 - J diseases and K symptoms
 - **Goal: given symptoms, compute posterior over diseases**



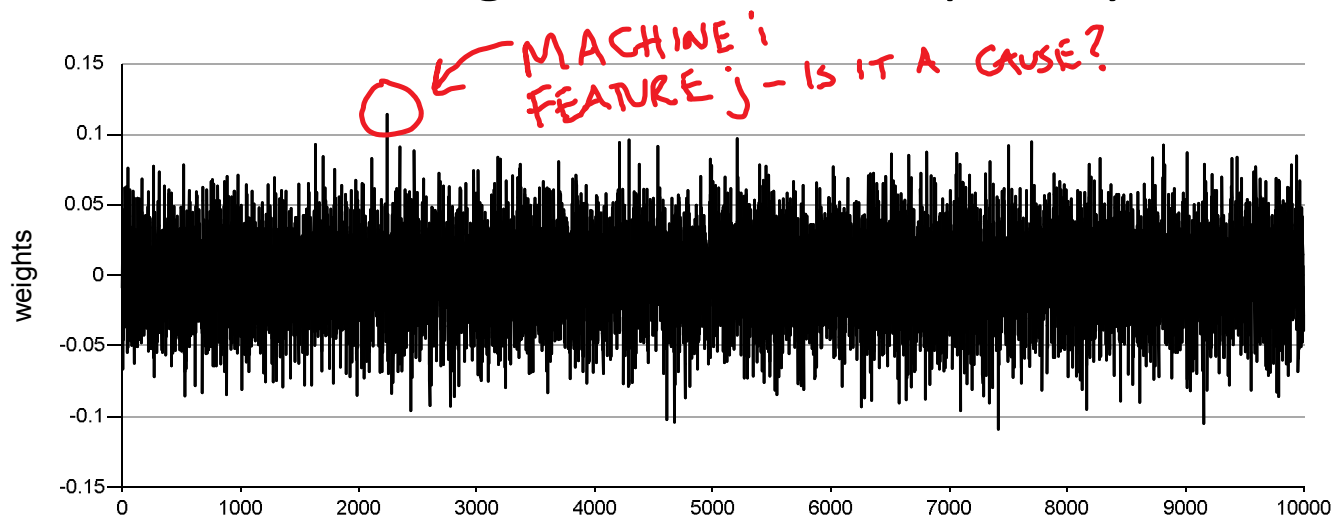
Forms of Diagnosis Problems 2

- “Factored” Diagnosis
 - J entities, each with the same K features (**J*K features**)
 - Hundreds of machines in a datacenter, each with the same performance counters, occasional faults
 - Hundreds of processes on a machine, each with the same performance counters, occasional hangs
 - Occasional **labels on the ensemble**
 - **Goal: given labels, find the true causes of the faults**



How Can We Solve Such Problems?

- Naïve Approach: train a classifier on the faults and try to interpret the feature weights
 - Logistic Regression – each weight is a parameter
 - Problem: $J \cdot K$ parameters w_i (10,000's)
 - Only hundreds of labels
 - Use L1 regularization for sparsity?



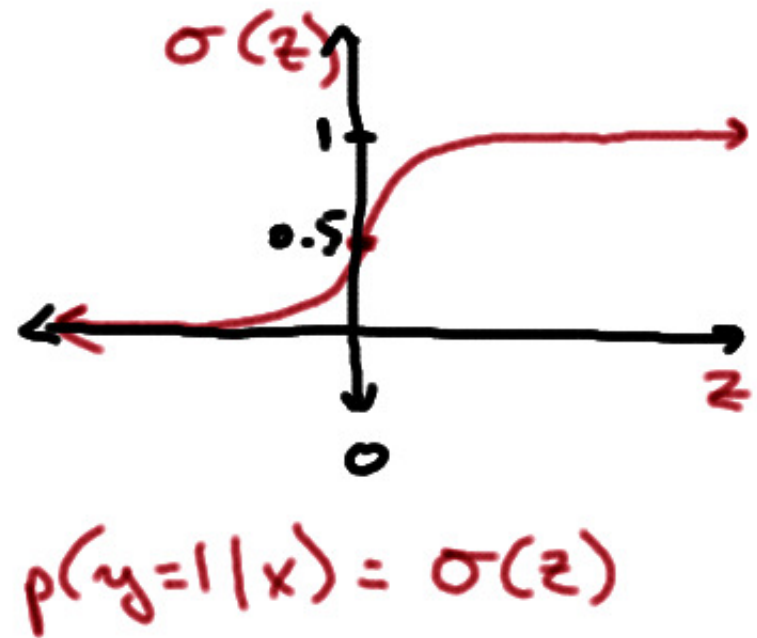
An Alternative Approach: Factorize!

- Leverage factored nature of the problem
 - Parameterize $J \cdot K$ parameters as the product of J entity weights α_j and K feature weights β_k
 - Only $J+K$ parameters!
 - So: $w_{jK+k} = \alpha_j \beta_k$
 - (more intuition coming soon...)

Highlights of Prior Work

- Long history of diagnosis work in ML, including using Logistic Regression along with Wald's Test for significance
- Bilinear Logistic Regression for Classification (Dyrhom et al. 2007)
- Diagnosis in Systems
 - Heuristics (Engler et al. 2003)
 - Hierarchical Clustering (Chen et al. 2002)
 - Metric Attribution (Cohen et al. 2005)
 - Bayesian Techniques (Wang et al. 2004)
 - Factor Graphs (Kremenek et al. 2006)
 - Many, many more...
- Our contribution: leveraging factored structure for diagnosis problems

Ordinary Logistic Regression: Intuition



Ordinary Logistic Regression

- Probability Model

$$P(y_i) = \frac{1}{1 + e^{-z_i}} = \sigma(z_i) \quad z_i = \sum_j \alpha_j f_{ij} + \delta$$

- Likelihood

$$P(Y) = \prod_i (\sigma(z_i))^{y_i} (1 - \sigma(z_i))^{1-y_i}$$

- Negative Log Likelihood

$$-\log P(Y) = -\sum_i y_i \log \sigma(z_i) - \sum_i (1 - y_i) \log(1 - \sigma(z_i))$$

Bilinear Logistic Regression: Intuition

$$\begin{aligned} & \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{3k} \end{bmatrix} \rightarrow \begin{bmatrix} w_0 & w_1 & w_2 & \dots & w_k \\ w_{k+1} & \dots & \dots & \dots & w_{2k} \\ \vdots & & & & \vdots \\ w_{(j-1)k+1} & \dots & \dots & \dots & w_{3k} \end{bmatrix} \approx \begin{bmatrix} \alpha_0 \beta_0 & \alpha_0 \beta_1 & \dots & \alpha_0 \beta_k \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \alpha_j \beta_0 & \dots & \dots & \alpha_j \beta_k \end{bmatrix} \\ & = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_j \end{bmatrix} [\beta_0 \beta_1 \dots \beta_k] = \alpha \beta^T \end{aligned}$$

Bilinear Logistic Regression

- Probability Model

$$P(y_i) = \frac{1}{1 + e^{-z_i}} = \sigma(z_i) \quad z_i = \sum_j \sum_k \alpha_j \beta_k f_{ijk} + \delta$$

- Likelihood

$$P(Y) = \prod_i (\sigma(z_i))^{y_i} (1 - \sigma(z_i))^{1-y_i}$$

- Negative Log Likelihood

$$-\log P(Y) = -\sum_i y_i \log \sigma(z_i) - \sum_i (1 - y_i) \log(1 - \sigma(z_i))$$

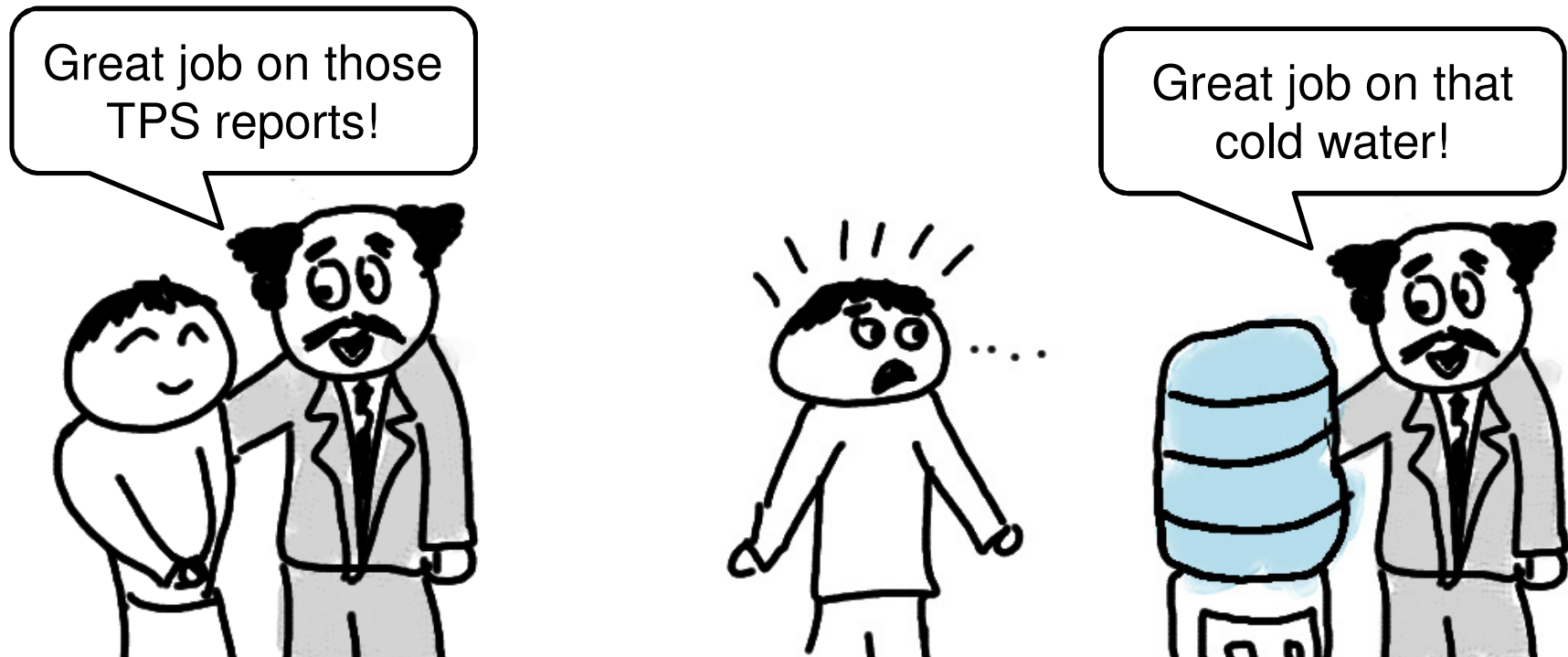
- Enforce Positive α_j for interpretability

$$\alpha_j = \gamma_j^2$$

Now for the Statistics

- **Question 1:** How can we determine whether a parameter is significant?
- **Question 2:** How can we tell how valid our “discovered” causes are if we don’t have ground truth labels for causes?
- These questions come up in many, many problems, so even if you never use BLR-D, this will be useful in your future

Common Principle for Both Questions: the “Does my boss like me?” Problem

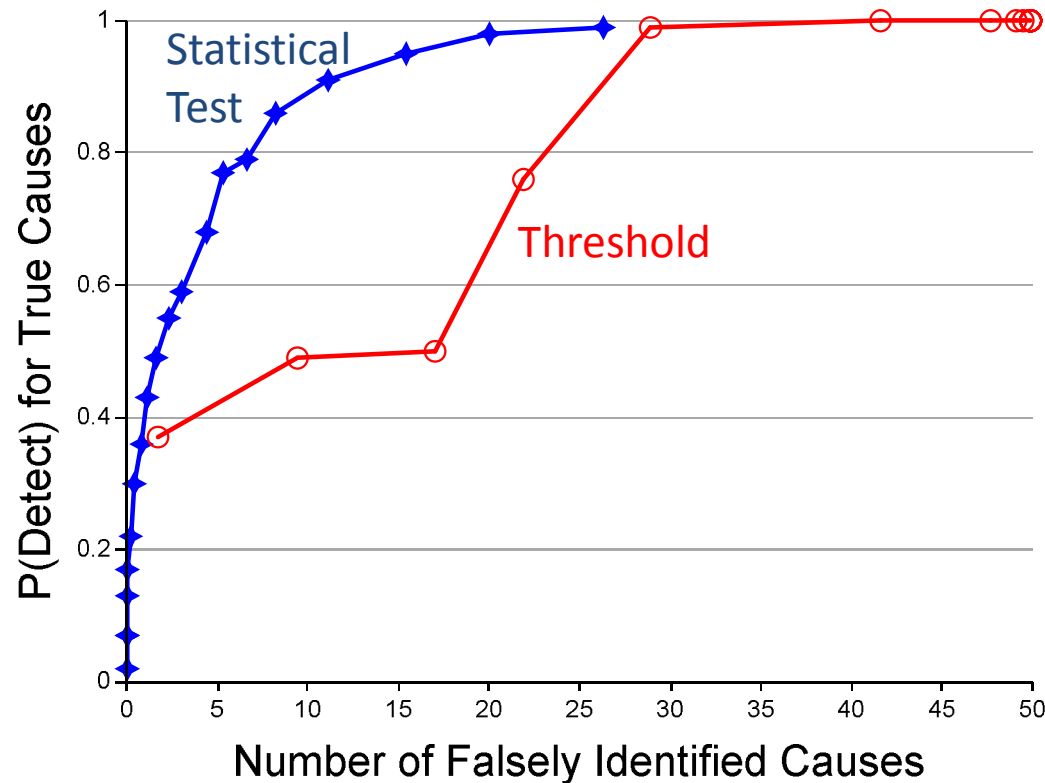


The data world's equivalent of seeing the difference in how your boss will act with you and with other people:

Efron's Bootstrap and **False Labels**

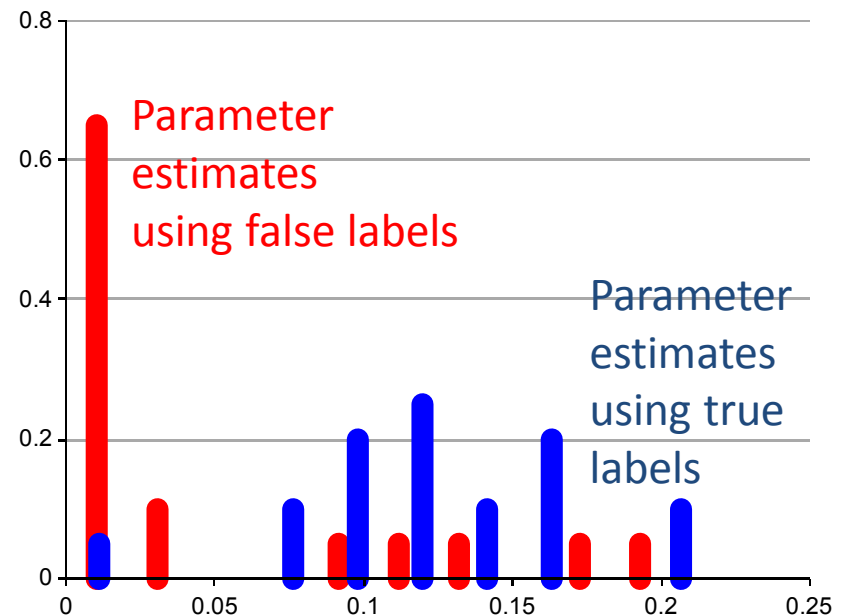
Question 1: When are Parameters Significant?

- Why not just use a threshold?
- Friends don't let friends use thresholds



What's the Statistical Approach?

- Compute population of parameter values under both true and false labels
 - True labels: perform multiple bootstraps
 - False labels: multiple bootstraps, permute labels
- Compare the two populations with a statistical test (Mann-Whitney)
- Yes, it's expensive!



Question 2: Are the Discoveries Meaningful?

- How can you tell if you're getting false alarms without labels for the true causes?
- Intuition: what would the method do when given random labels?
 - Consider the algorithm “a” which reports a certain number of parameters as “guilty”
 - Compute how often “a” reports guilty parameters under false vs. true labels
 - Formally, the “False Discovery Rate” (FDR):

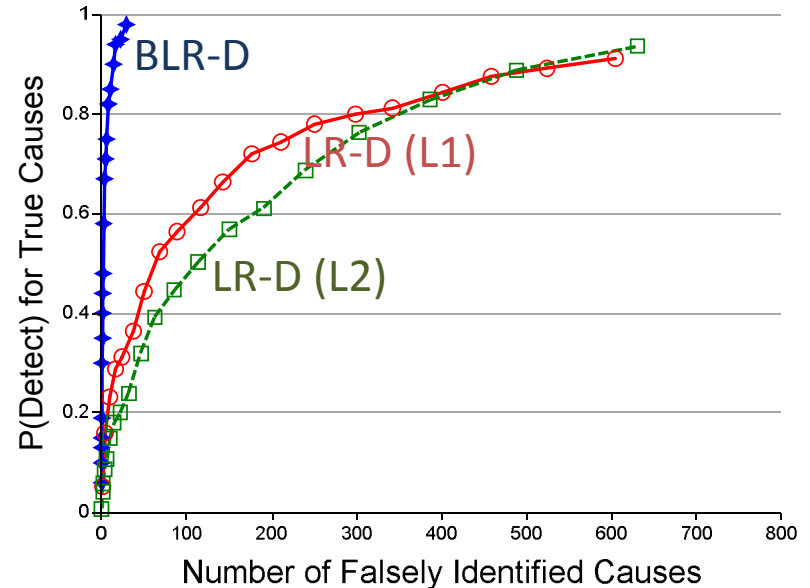
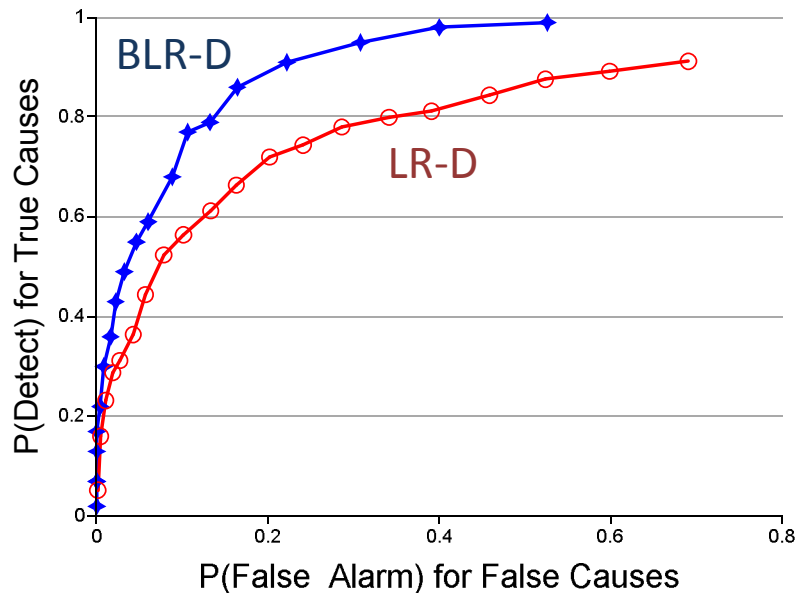
$$FDR(a) = E \left[\frac{F(a)}{S(a)} \right] \cong \frac{E[F(a)]}{E[S(a)]} \cong \frac{\sum_{q=1}^Q \frac{N(D^q, a)}{Q}}{N(D, a)}$$

The Overall Procedure: BLR-D

- Bilinear Logistic Regression for Diagnosis
 - Factor parameters into bilinear form
 - Train BLR classifier with overall faults as labels
 - Test individual parameters for significance with bootstrap and Mann-Whitney Test
 - Estimate False Discovery Rate (when ground truth labels on causes are not available)
 - Adjust Mann-Whitney threshold until FDR is reasonable
 - Report significant parameters

P(FA) vs. Number of False Alarms

- The probability of False Alarms doesn't capture the true cost to the analyst when the number of parameters/causes is very large



Experiment 1: Machines in a Datacenter

- Synthetic Model of Datacenter
 - J machines (base: 30)
 - Each has K normally-distributed features (base: 30), some of which are fault-causing (5)
 - Some machines are fault-prone (base: 5)
 - When a fault-prone machine has a fault-causing feature exceed a probability threshold, a system fault (label) is generated)
 - Data publicly available (see URL in paper)
- Goal: Identify fault-prone machines and fault-causing features
- Baseline: LR-D (with L1 regularization)
 - Use same statistical tests as BLR-D

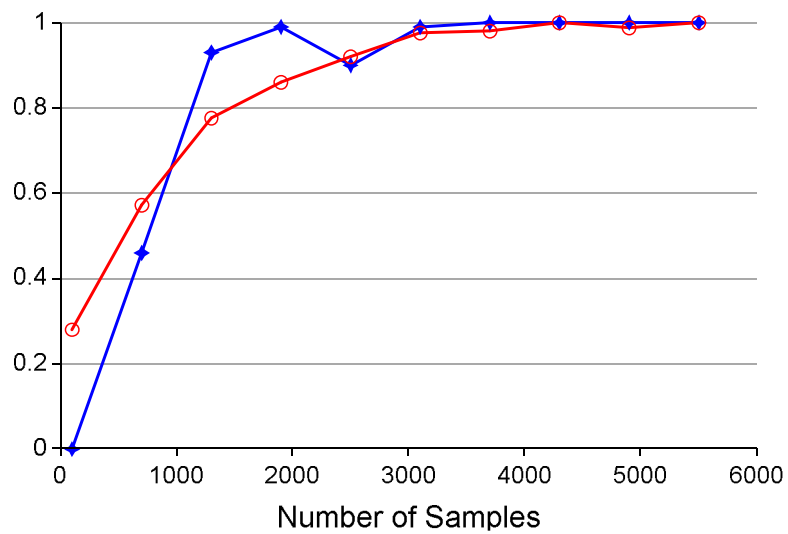
Experimental Variations

- Number of Data Samples/Frames
- Number of Machines in Datacenter
- Fraction of Fault-Prone Machines

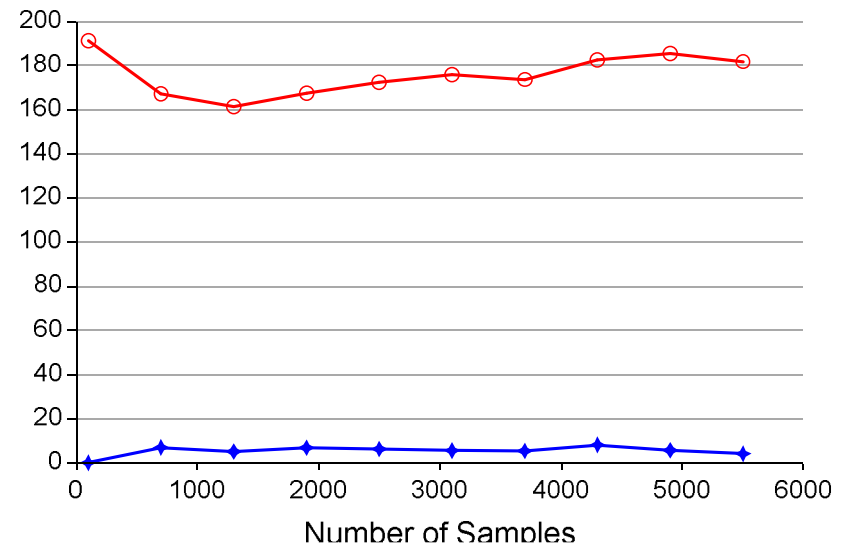
Experiment 1a

- Performance vs. Number of Samples

Detection Rate vs. Number of Samples



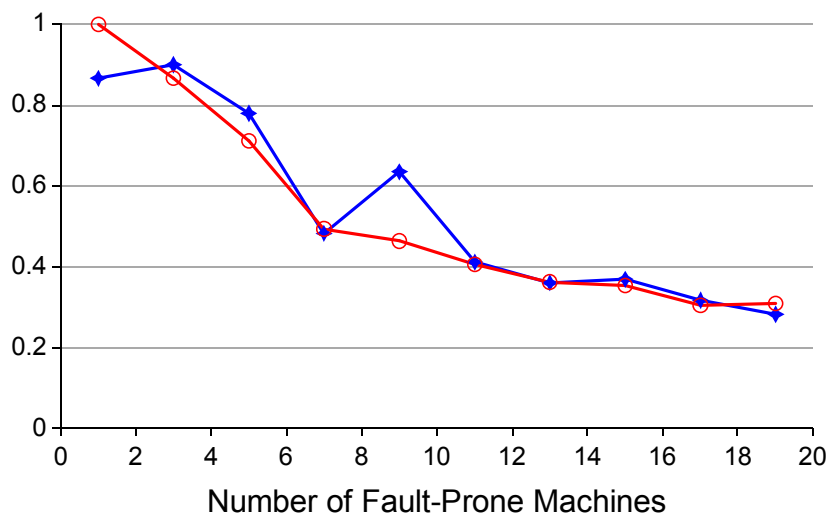
False Alarms vs. Number of Samples



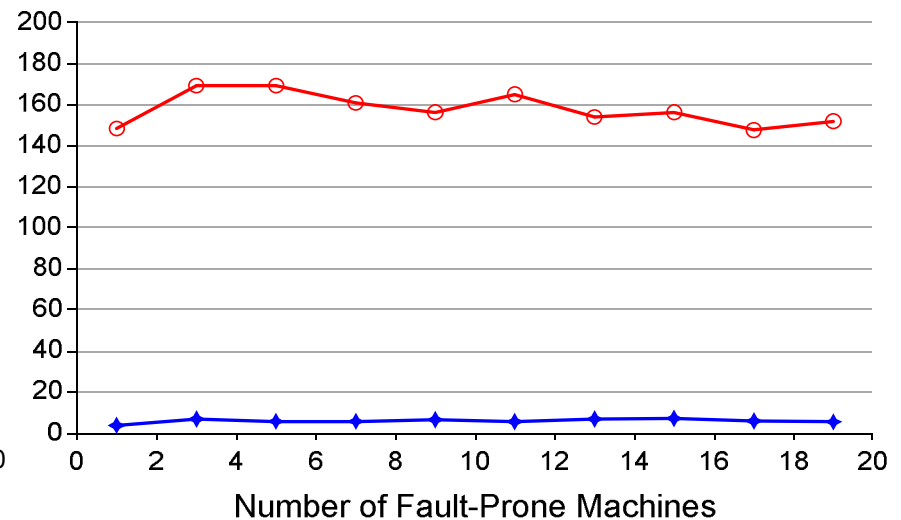
Experiment 1b

- Performance vs. Fraction of Faulty Machines

Detection Rate vs. Number of Fault-Prone Machines



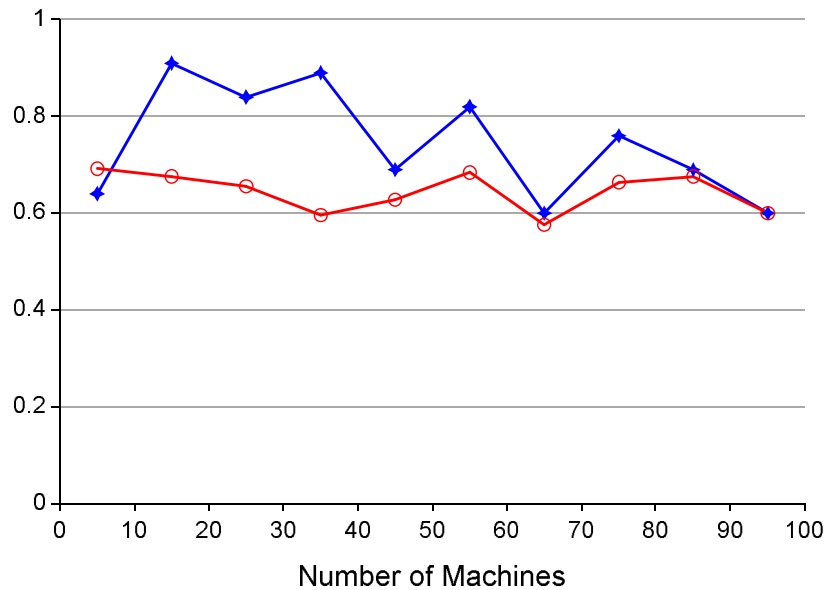
False Alarms vs. Number of Fault-Prone Machines



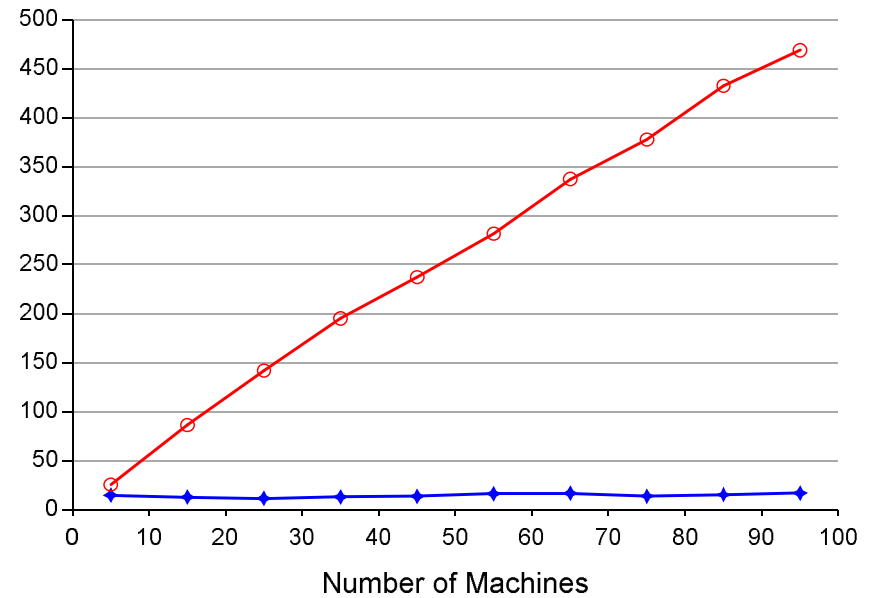
Experiment 1c

- Performance vs. Number of Machines

Detection Rate vs. Number of Machines



False Alarms vs. Number of Machines



Experiment 2: Processes on a Machine

- Typical Windows PC has 100+ processes running at all times
- Subject to occasional, unexplained hangs
- Which process is responsible?
- Our Experiment
 - Record all performance counters for all processes
 - User UI for labeling hangs
 - “WhySlowFrustrator” process that chews up memory, causing a hang
 - One month of data, 2912 features per timestep (once per minute)
 - 63 labels (many false negatives)

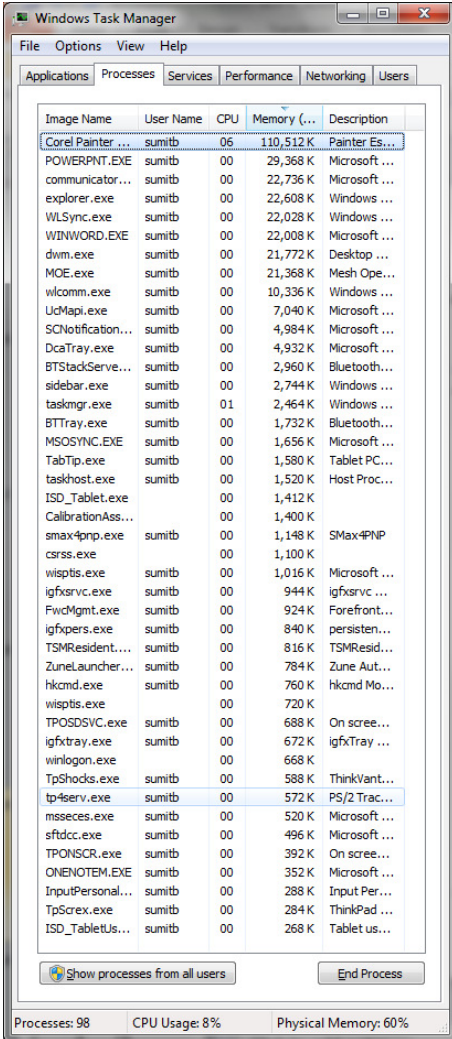
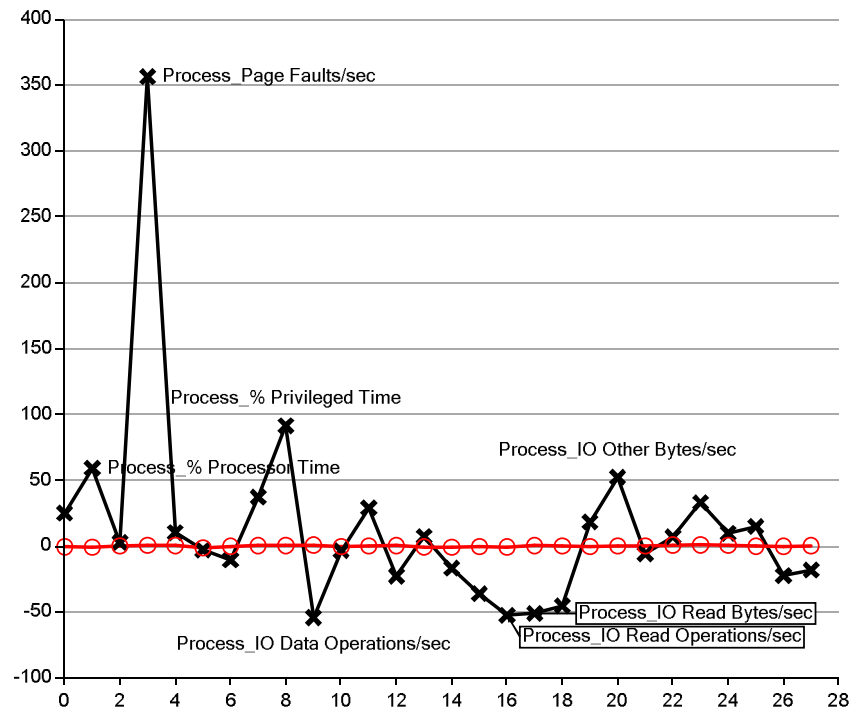
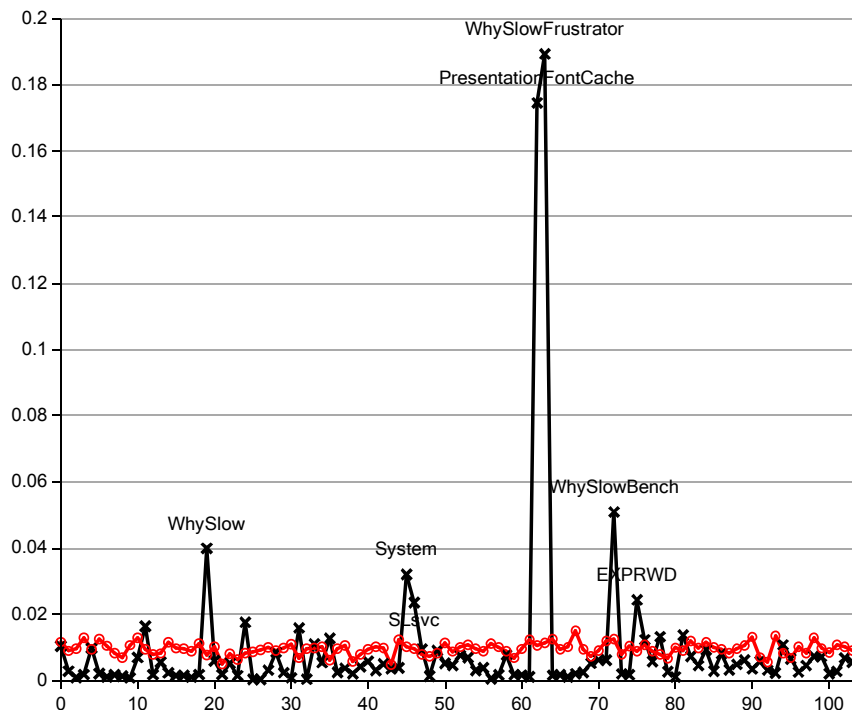


Image Name	User Name	CPU	Memory (...)	Description
Corel Painter ...	sumitb	06	110,512 K	Painter Es...
POWERPNT.EXE	sumitb	00	29,368 K	Microsoft ...
communicator...	sumitb	00	22,736 K	Microsoft ...
explorer.exe	sumitb	00	22,608 K	Windows ...
WLSync.exe	sumitb	00	22,028 K	Windows ...
WINWORD.EXE	sumitb	00	22,008 K	Microsoft ...
dwm.exe	sumitb	00	21,772 K	Desktop ...
MOE.exe	sumitb	00	21,368 K	Mesh Ope...
wlcomm.exe	sumitb	00	10,336 K	Windows ...
UcMapi.exe	sumitb	00	7,040 K	Microsoft ...
SCNotification...	sumitb	00	4,984 K	Microsoft ...
DcaTray.exe	sumitb	00	4,932 K	Microsoft ...
BTStackServe...	sumitb	00	2,960 K	Bluetooth...
sidebar.exe	sumitb	00	2,744 K	Windows ...
taskmgr.exe	sumitb	01	2,464 K	Windows ...
BTTray.exe	sumitb	00	1,732 K	Bluetooth...
MSOSYNC.EXE	sumitb	00	1,656 K	Microsoft ...
TabTip.exe	sumitb	00	1,580 K	Tablet PC...
taskhost.exe	sumitb	00	1,520 K	Host Proc...
ISD_Tablet.exe	sumitb	00	1,412 K	
CalibrationAss...	sumitb	00	1,400 K	
smax4np.exe	sumitb	00	1,148 K	SMax4NP
csrss.exe	sumitb	00	1,100 K	
wispts.exe	sumitb	00	1,016 K	Microsoft ...
igfxsvc.exe	sumitb	00	944 K	igfxsvc ...
FwcMgmt.exe	sumitb	00	924 K	Forefront...
igfxpers.exe	sumitb	00	840 K	persisten...
TSMResident...	sumitb	00	816 K	TSMResid...
ZuneLancher...	sumitb	00	784 K	Zune Aut...
hkcmd.exe	sumitb	00	760 K	hkcmd Mo...
wispts.exe	sumitb	00	720 K	
TPOSDSVC.exe	sumitb	00	688 K	On scree...
igfxtray.exe	sumitb	00	672 K	igfxTray ...
winlogon.exe	sumitb	00	668 K	
TpShocks.exe	sumitb	00	588 K	ThinkVant...
tp4serv.exe	sumitb	00	572 K	PS/2 Trac...
mssecsv.exe	sumitb	00	520 K	Microsoft ...
sftdccc.exe	sumitb	00	496 K	Microsoft ...
TPONSCR.exe	sumitb	00	392 K	On scree...
ONENOTEM.EXE	sumitb	00	352 K	Microsoft ...
InputPersonal...	sumitb	00	288 K	Input Per...
TpScrex.exe	sumitb	00	284 K	ThinkPad ...
ISD_TabletUs...	sumitb	00	268 K	Tablet us...

Processes: 98 CPU Usage: 8% Physical Memory: 60%

Experiment 2: Processes on a Machine

- Results
 - Adjusted Mann-Whitney threshold to achieve 0 FDR
 - 2 processes were “significant”: WhySlowFrustrator and PresentationFontCache; no features were “significant”



Extensions: Multiple Modes

- Analogy to SVD
- $\alpha\beta^T$ is a rank 1 approximation to the w (in matrix form)...
- So why not $\alpha_0\beta_0^T + \alpha_1\beta_1^T + \dots$?
 - Handle *multiple modes* of failure
 - J+K additional parameters per term
 - But... identifiability issues become a problem

Take-Home Messages

- Is your problem factorable?
 - **Factor it!**
- Which parameters are important?
 - **Test them statistically, not with a threshold!**
- Wondering how valid your “causes” are?
 - **Use FDR!**