Bilinear Logistic Regression for Factored Diagnosis Problems

Sumit Basu[1], John Dunagan[1,2], Kevin Duh[1,3], and Kiran-Kumar Munuswamy-Reddy[1,4]

[1]Microsoft Research
[2]Microsoft
[3]NTT Labs
[4]Harvard University

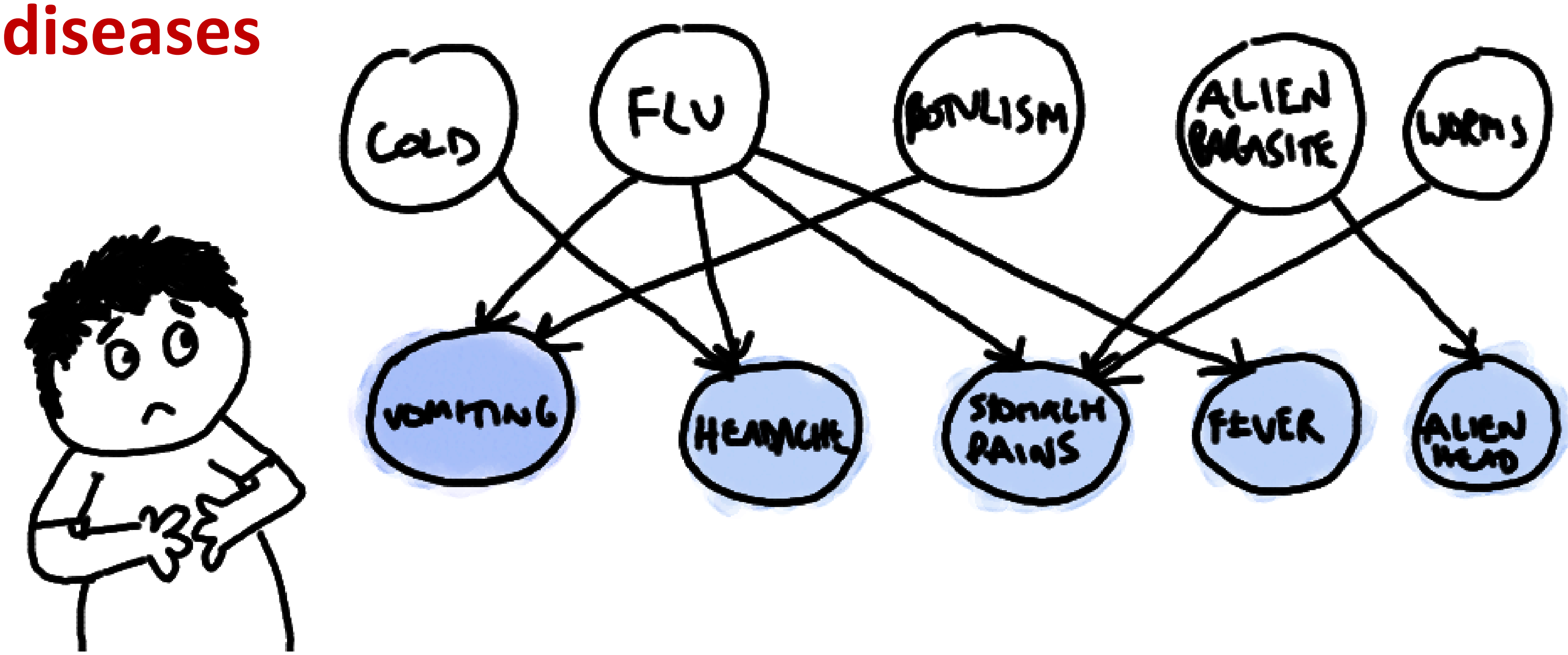
Note: if you use content from these slides in your presentations/papers, please attribute it to: S. Basu, J. Dunagan, K. Duh, and K-K. Munuswamy-Reddy. "Bilinear Logistic Regression for Factored Diagnosis Problems." In *Proceedings of SLAML 2011*. Cascais, Portugal. October, 2011.

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**

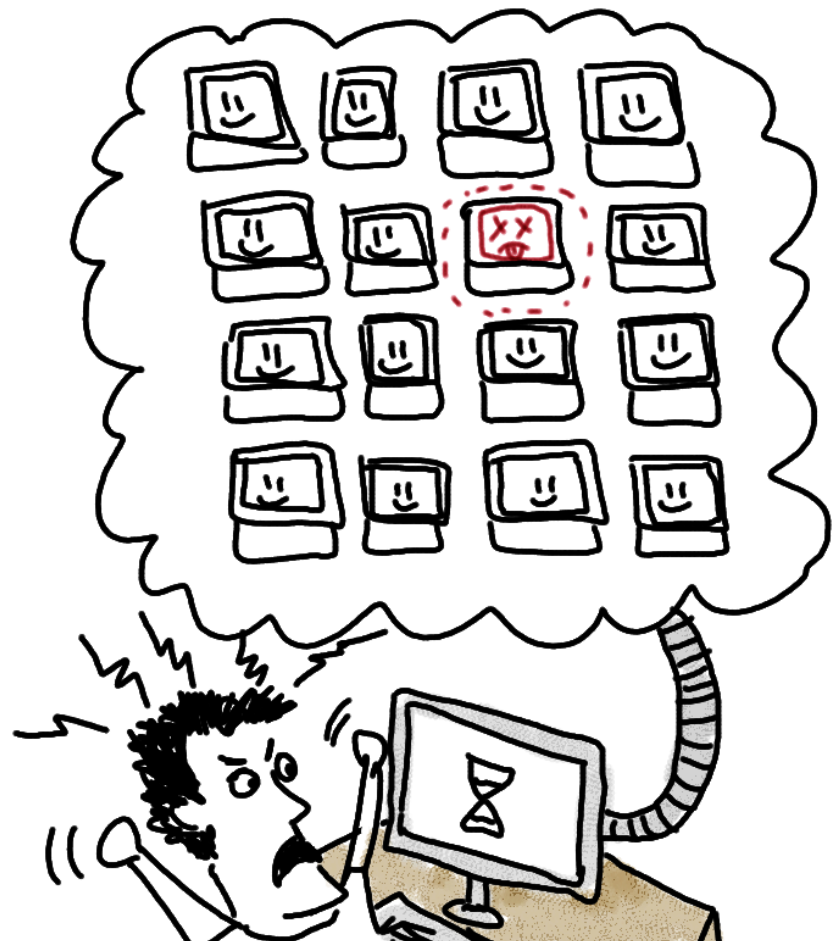
Forms of Diagnosis Problems

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



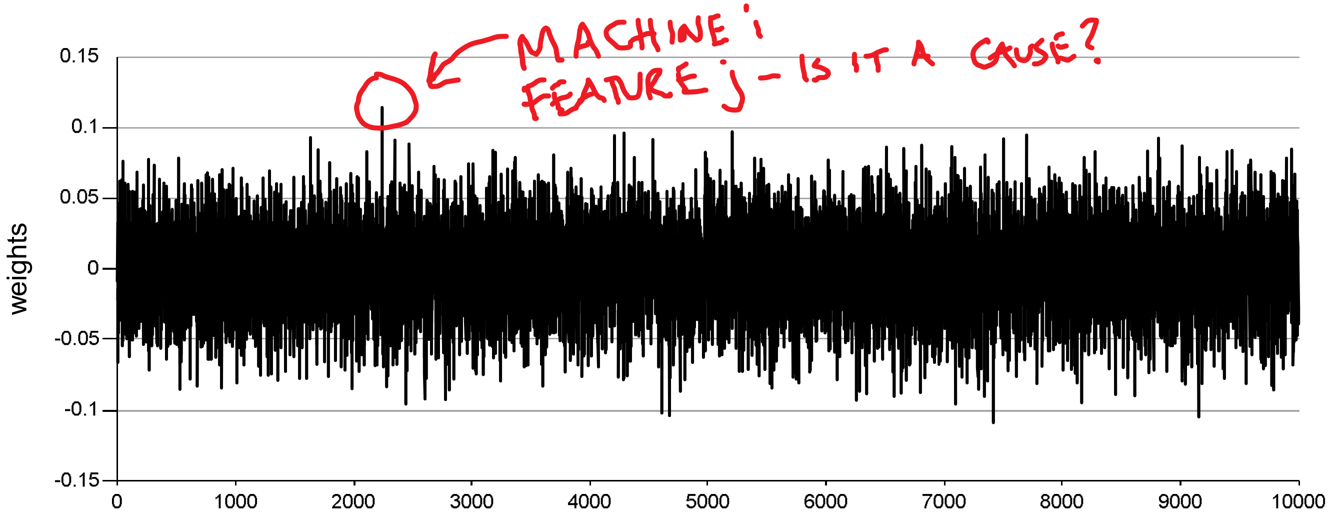
Forms of Diagnosis Problems 2

- "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*K parameters w_i (10,000’s)
  - Only hundreds of labels
  - Use L1 regularization for sparsity?



An Alternative Approach: Factorize!

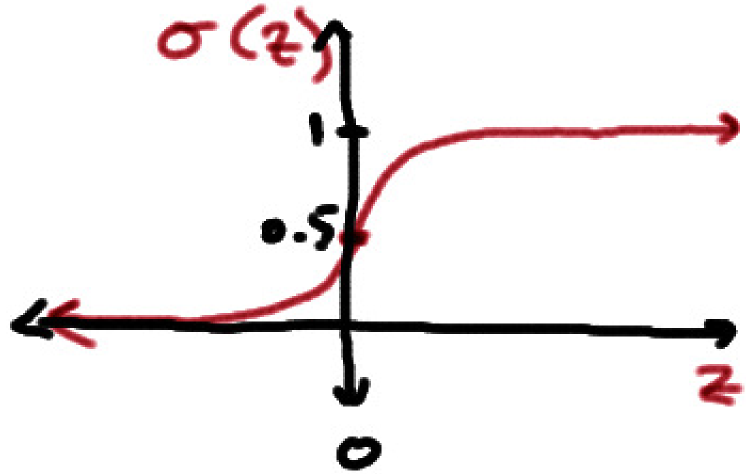
- Leverage factored nature of the problem
  - Parameterize J*K parameters as the product of J entity weights α_j and K feature weights β_k
  - Only J+K parameters!
  - So: $w_{jK+k} = \alpha_j \beta_k$
  - (more intuition coming soon…)

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

$z = w^T x + b = 0$



$p(y=1|x) = \sigma(z)$

Ordinary Logistic Regression

- Probability Model

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

- Likelihood

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

- 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

$$
\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{JK} \end{bmatrix}
\rightarrow
\begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_K \\ w_{K+1} & \cdots & & \cdots & w_{2K} \\ \vdots & & \ddots & & \vdots \\ w_{(J-1)K+1} & \cdots & & \cdots & w_{JK} \end{bmatrix}
\approx
\begin{bmatrix} \alpha_0\beta_0 & \alpha_0\beta_1 & \cdots & \alpha_0\beta_K \\ \vdots & & \ddots & \vdots \\ \alpha_J\beta_0 & \cdots & \cdots & \alpha_J\beta_K \end{bmatrix}
$$

$$
= \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_J \end{bmatrix} \begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_K \end{bmatrix} = \alpha\beta^T
$$

Bilinear Logistic Regression

- Probability Model

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

- Likelihood

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

- Negative Log Likelihood

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

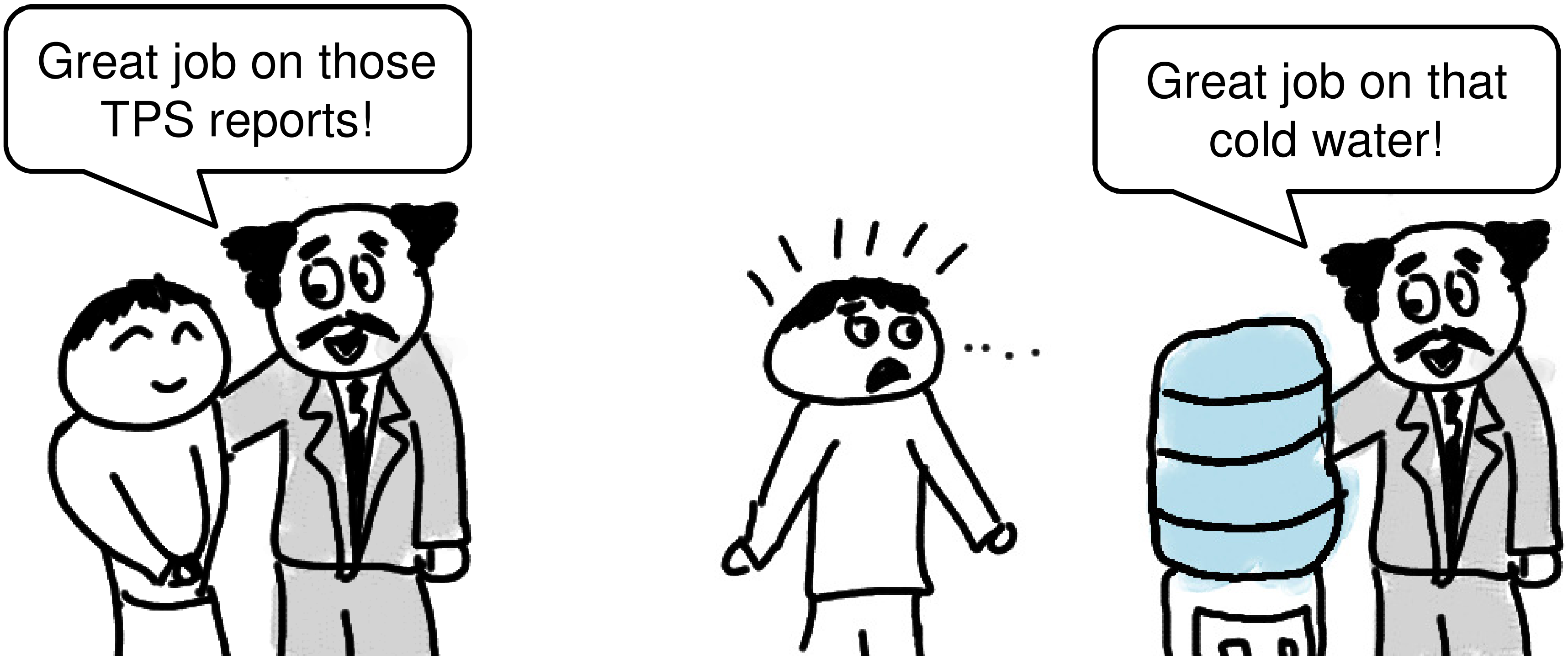
- Enforce Positive α_j for interpretability

$$\alpha_j = \gamma_j^2$$

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

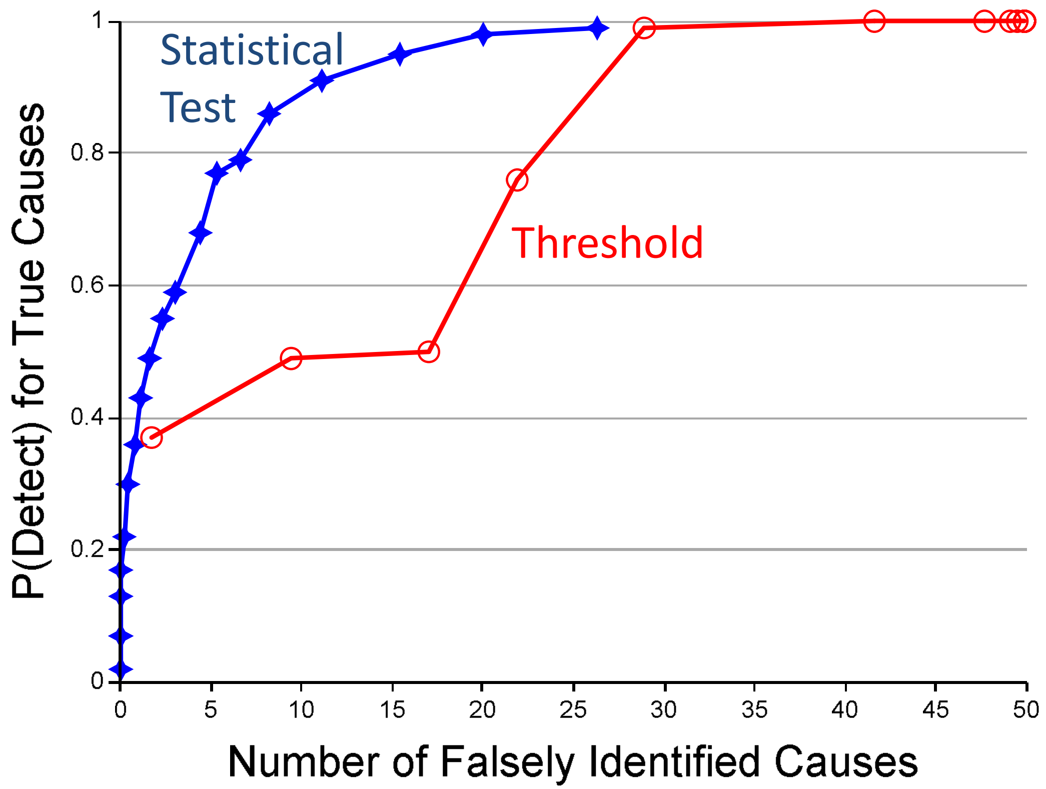
Common Principle for Both Questions: the "Does my boss like me?" Problem



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**

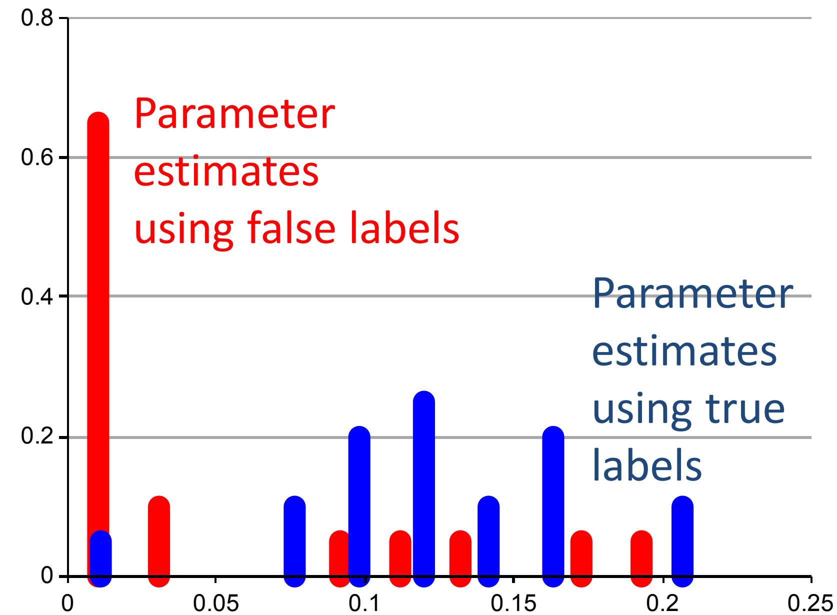
Question 1: When are Parameters Significant?

- 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):

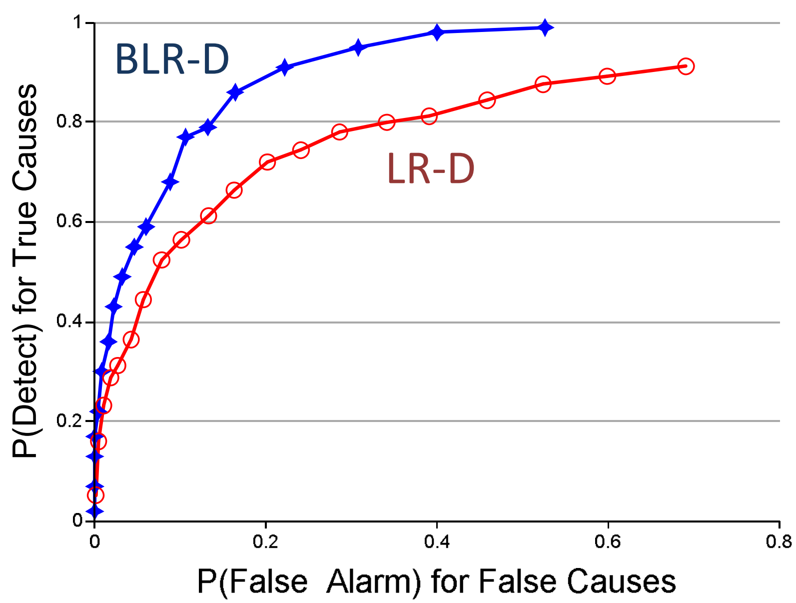
$$FDR(a) = E\left[\frac{F(a)}{S(a)}\right] \cong \frac{E[F(a)]}{E[S(a)]} \cong \frac{\sum_{q=1}^{Q} \frac{N(D^q, a)}{Q}}{N(D, a)}$$

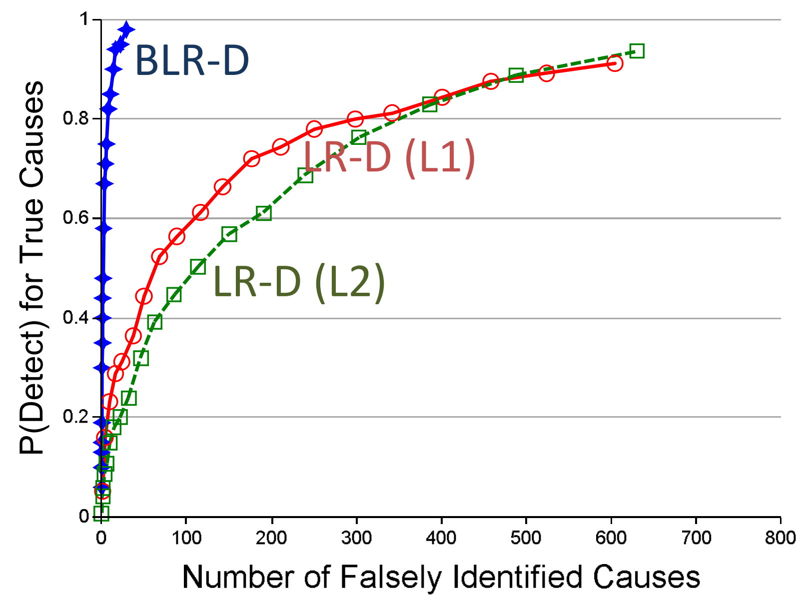
The Overall Procedure: BLR-D

- 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

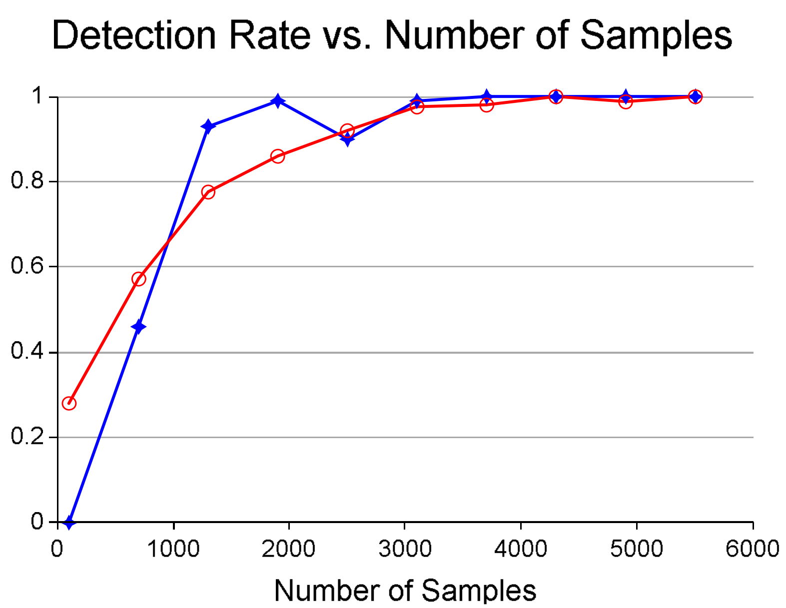
- Synthetic Model of Datacenter
  - J machines (base: 30)
  - Each has K normally-distributed features (base: 30), some of which are fault-causing (5)
  - Some machines are fault-prone (base: 5)
  - When a fault-prone machine has a fault-causing feature exceed a probability threshold, a system fault (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

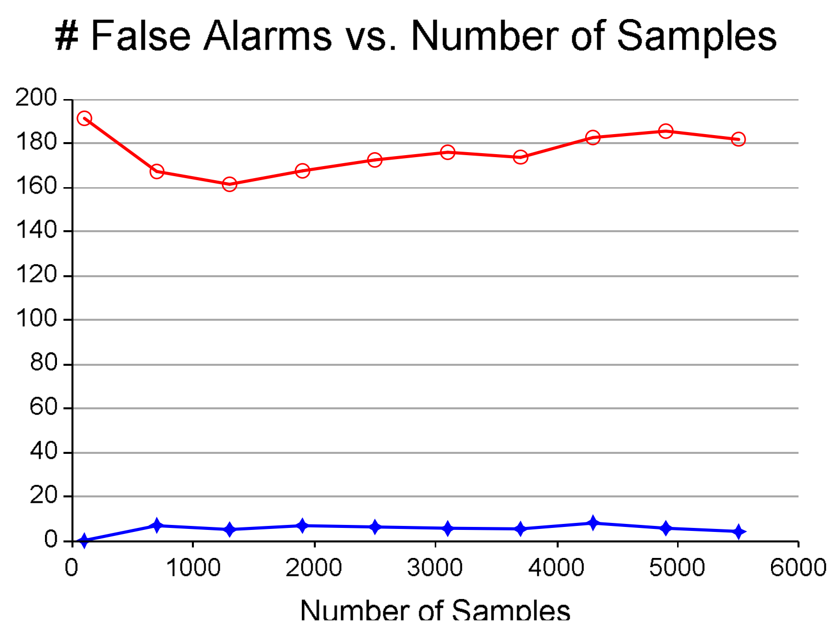
Experimental Variations

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

Experiment 1a

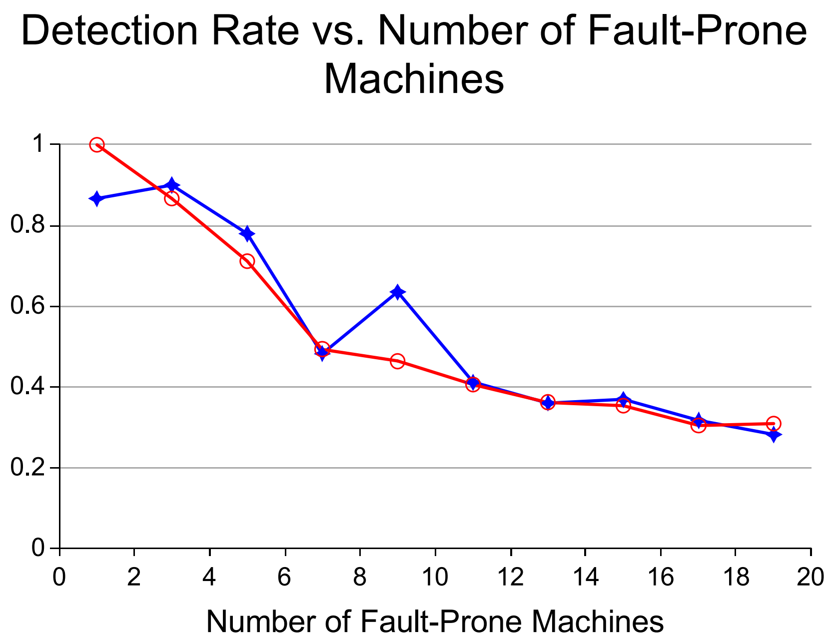
- Performance vs. Number of Samples

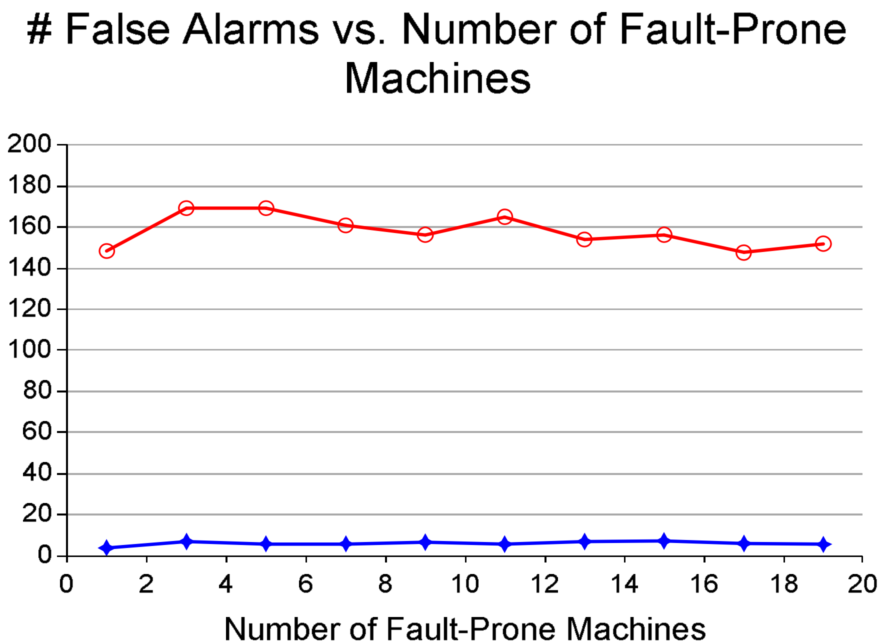




Experiment 1b

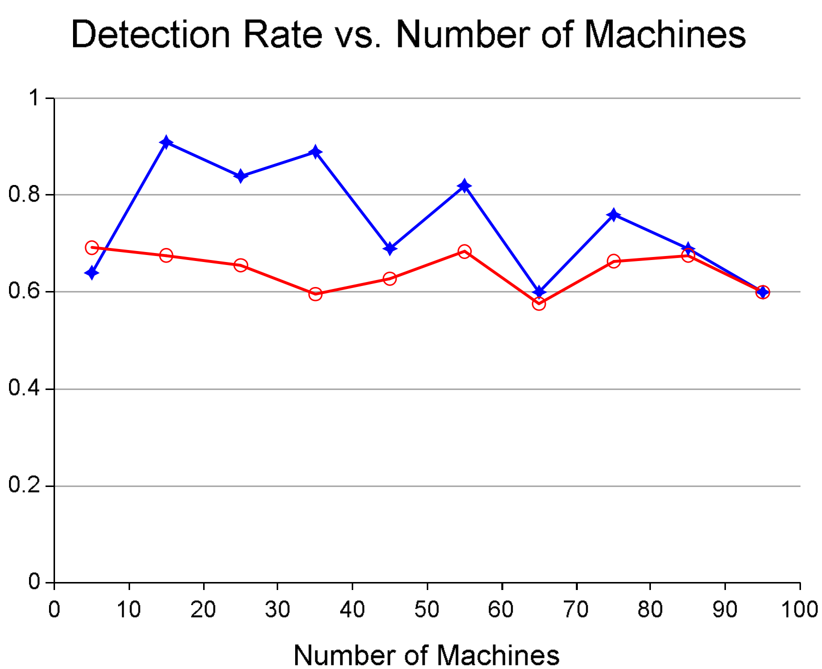
- Performance vs. Fraction of Faulty Machines

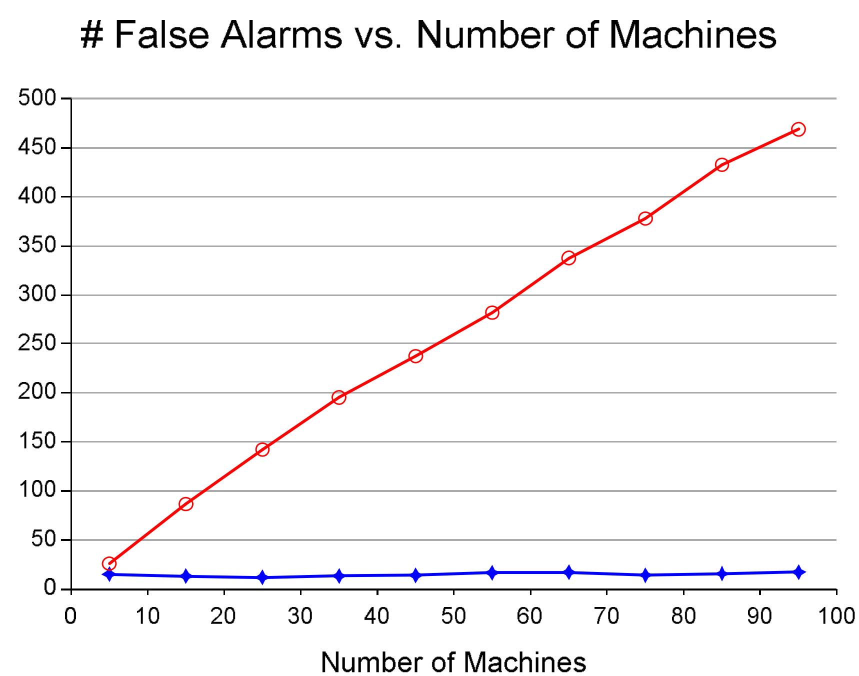




Experiment 1c

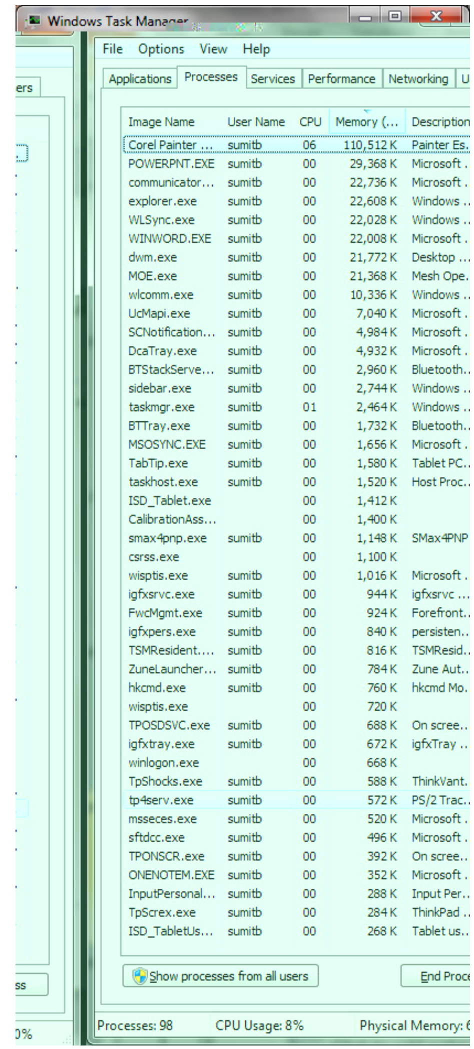
- Performance vs. Number of Machines





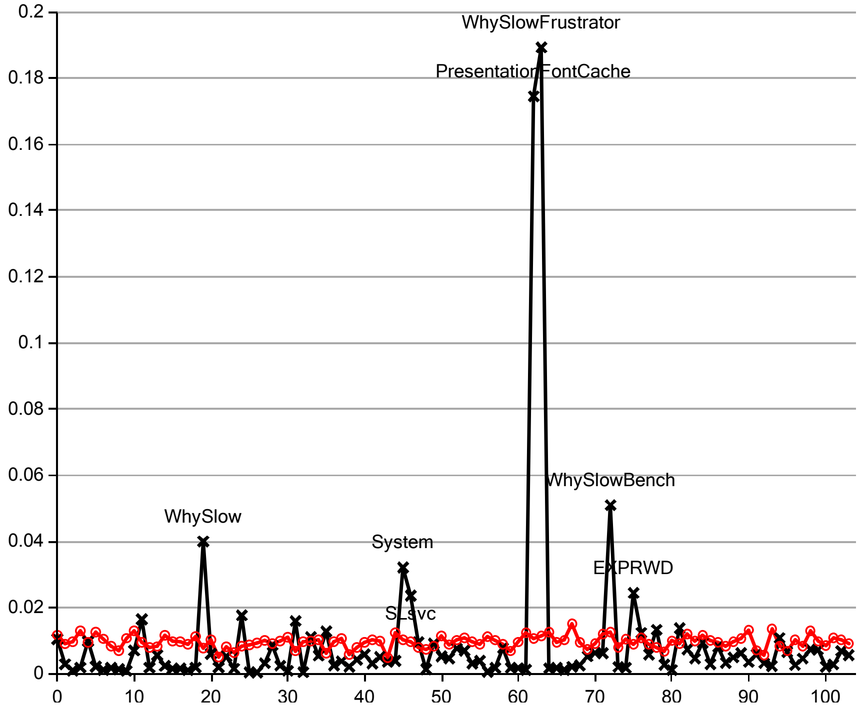
Experiment 2: Processes on a Machine

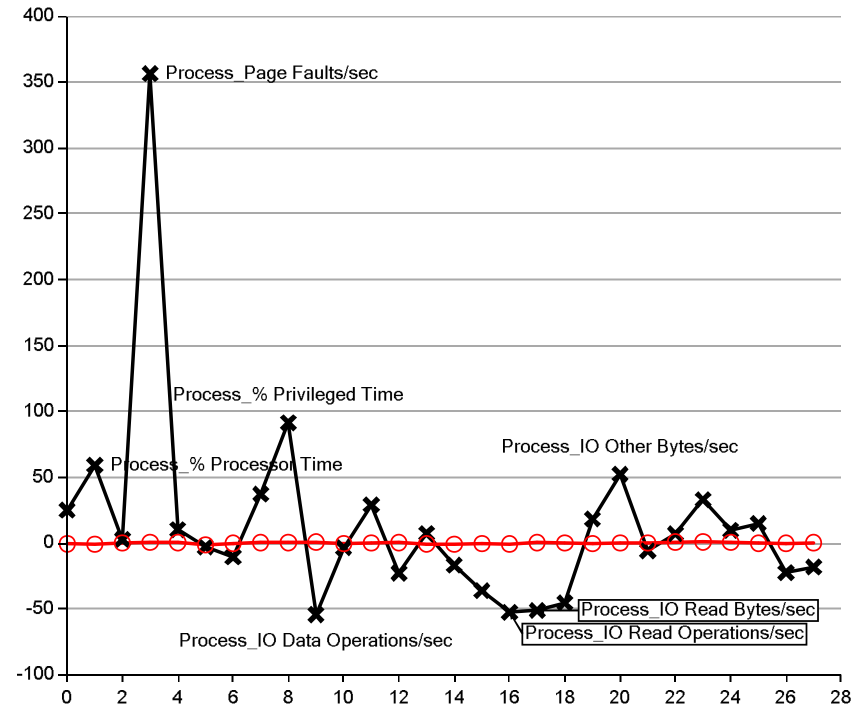
- Typical Windows PC has 100+ processes running at all times
- Subject to occasional, unexplained hangs
- Which process is responsible?
- Our Experiment
  - Record all performance counters for all processes
  - User UI for lableling hangs
  - “WhySlowFrustrator” process that chews up memory, causing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



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 $\alpha_0\beta_0^T + \alpha_1\beta_1^T + \cdots$?
  - 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!**