A Case Study of Handling Missing Sensor Data

Data Cleaning & Integration

CompSci 590.01 Spring 2017

DUKE
COMPUTER SCIENCE

Some contents were based on:
My own slides for talks given at various places

A. Silberstein et al. “Suppression and failures in sensor data: a Bayesian approach.” *VLDB*, 2007

- Not a clean-and-then-learn approach
- But shows the end-to-end process from data collection to analysis

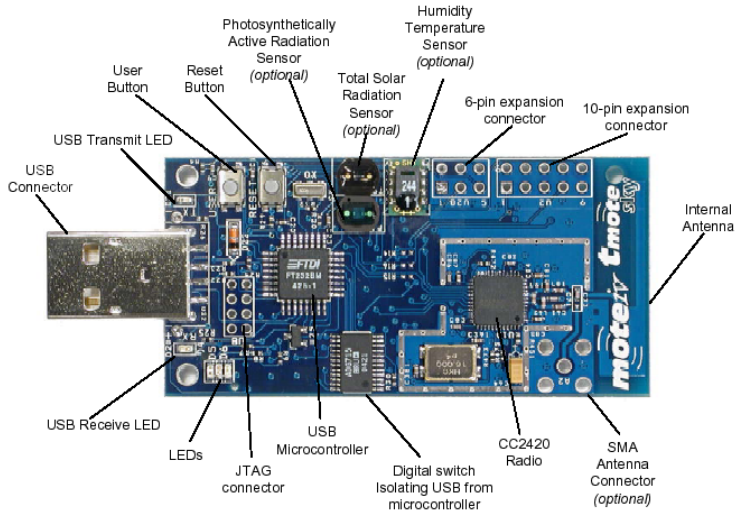
Biggest take-away points?

(For Jun:)

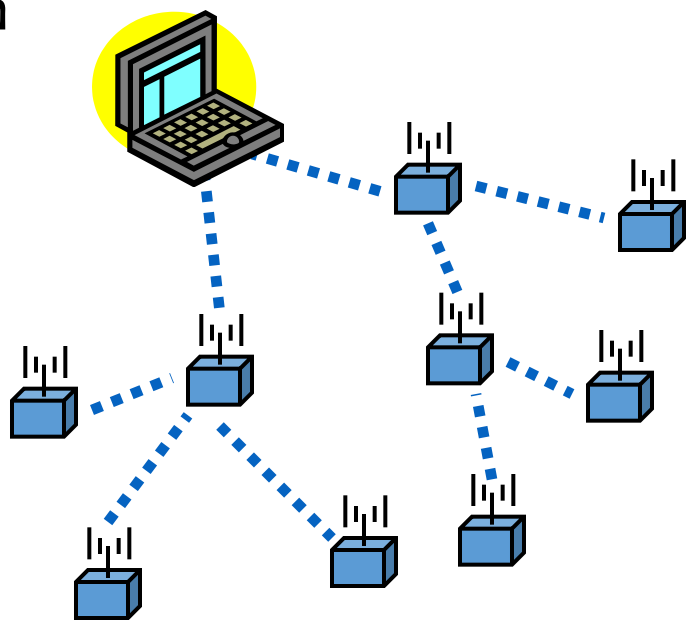
- Interpretation of dirty (missing) data depends on your data collection process
- Tweaking data collection allows interesting trade-offs
- Expose uncertainty
- Model-driven vs. data-driven data collection

Sensor network

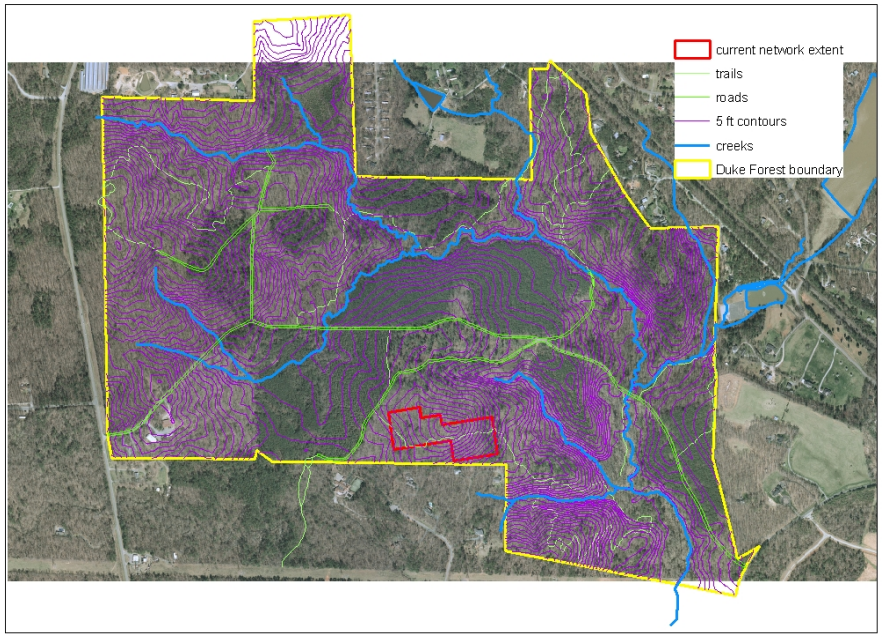
- Small, untethered nodes with severe resource constraints
  - Sensors, e.g., light, moisture, …
  - Tiny CPU and memory
  - Battery power
  - Limited-range radio communication
    - Often dominates energy consumption



- Nodes form a multi-hop network rooted at a base station
  - Base station has plentiful resources and is typically tethered or at least solar-powered

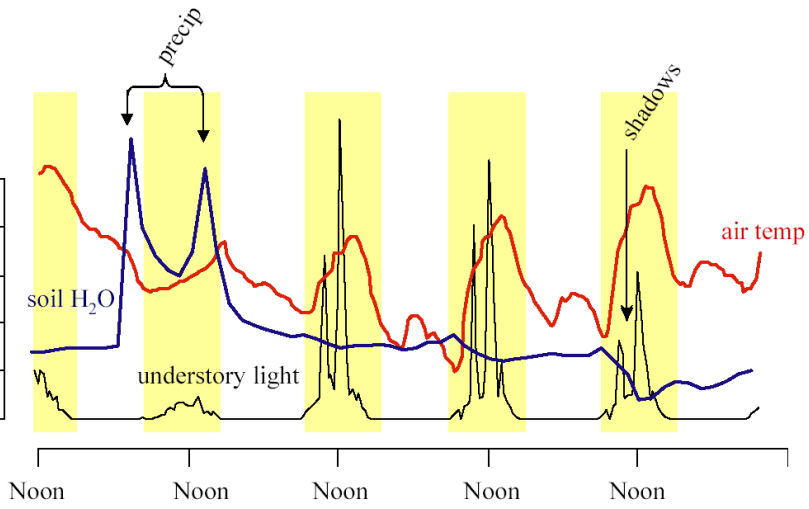


Duke Forest deployment



- Use wireless sensor networks to study how environment affects tree growth in Duke forest
  - Collaboration with Jim Clark (ecology), Alan Gelfand (stats) et al. since 2006

Eno Division



So, what do ecologists want?

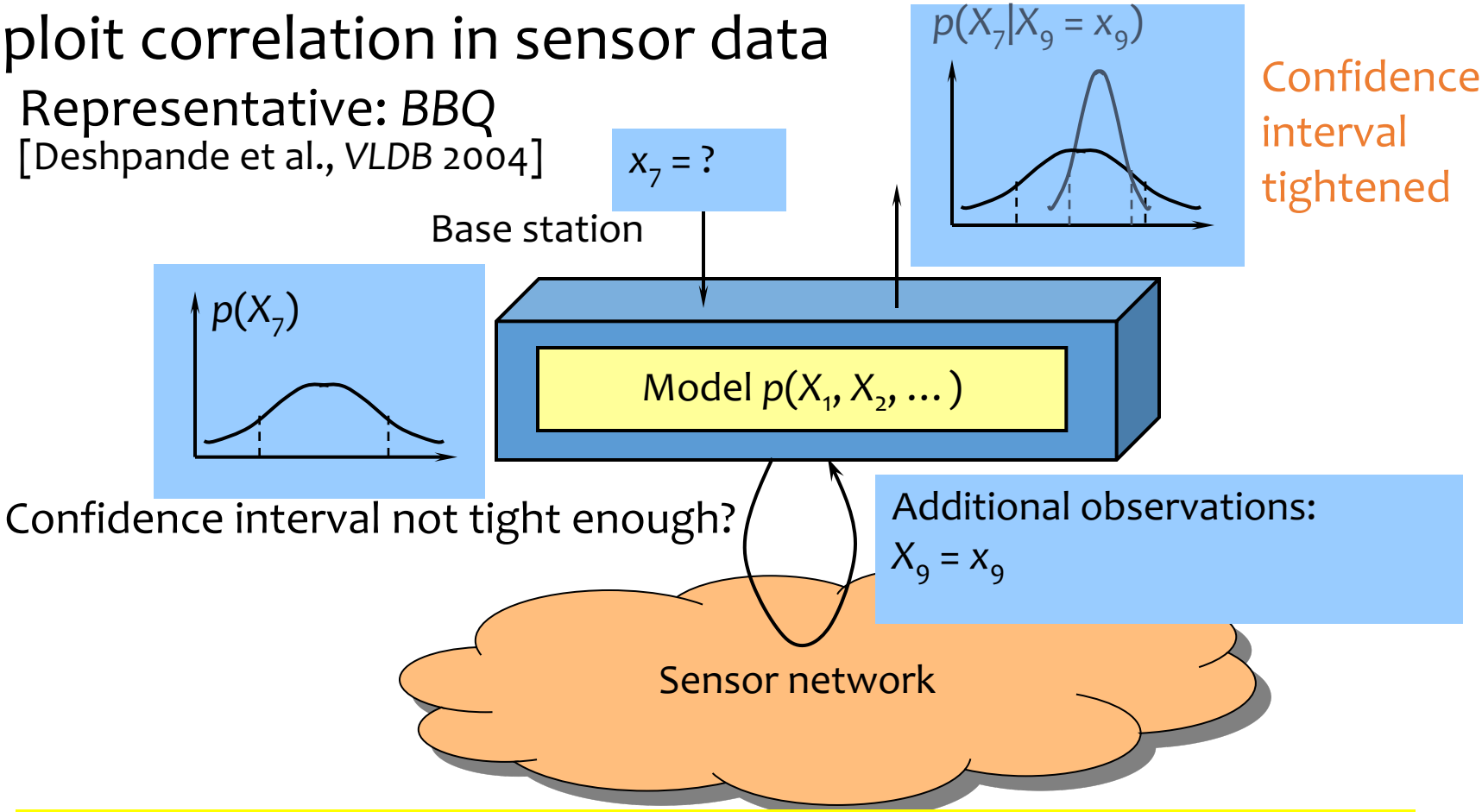
- Collect all data (to within some precision)
  - Probably the most boring database query
- Fit stochastic models using data collected
  - Cannot be expressed as database queries

☞*Very different from how database researchers would think about "querying data"*

- E.g., SQL, selection, join, aggregation…

Model-driven data collection: pull

- Exploit correlation in sensor data
  - Representative: *BBQ*
    [Deshpande et al., *VLDB* 2004]



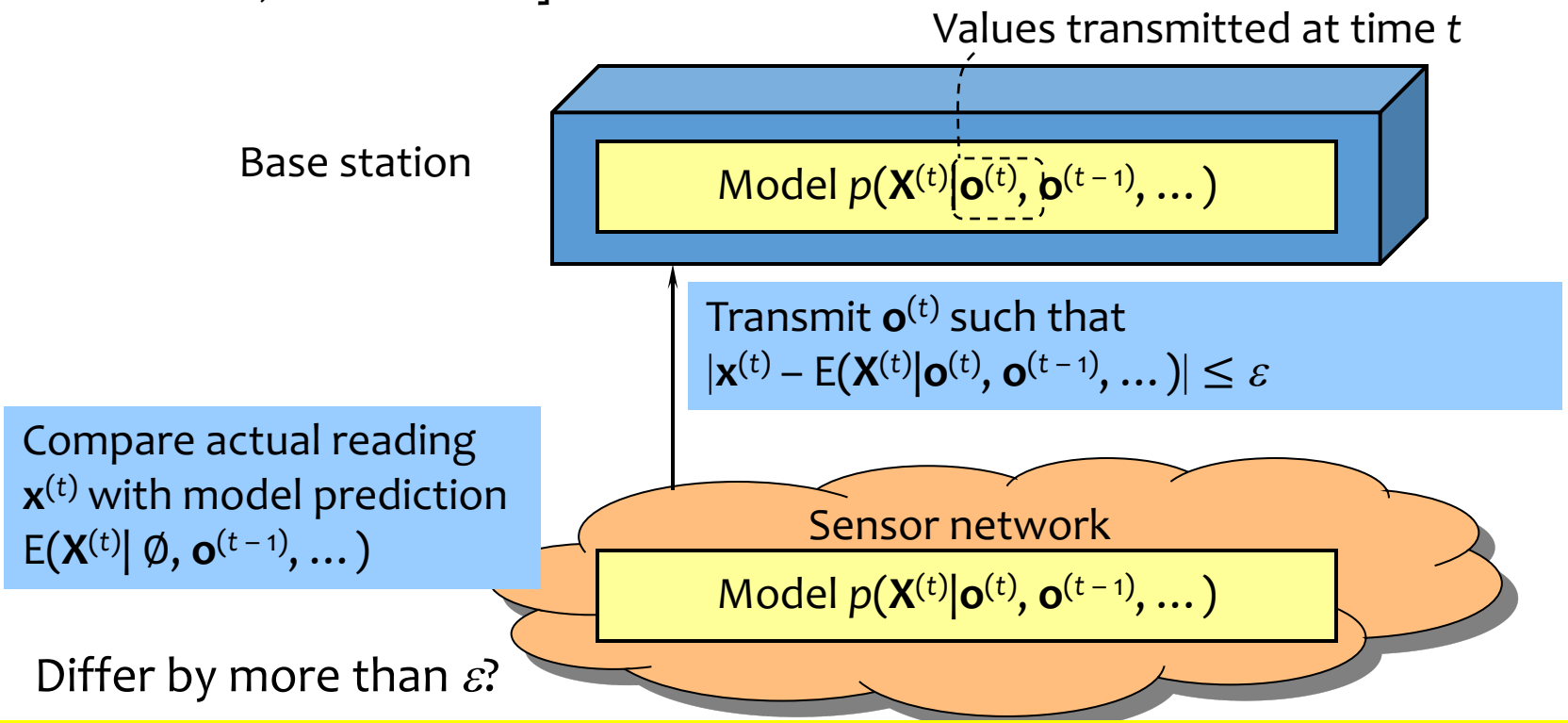
Answer correctness depends on model correctness
Risk missing the unexpected

Data-driven philosophy

- Models don't substitute for actual readings
  - Particularly when we are still *learning* about the physical process being monitored
  - Correctness of data collection should not rely on correctness of models
- Models can still be used to *optimize* collection

Data-driven: push

- Exploit correlation in data + put smarts in network
  - Representatives: *Ken* [Chu et al., *ICDE* 2006], *Conch* [Silberstein et al., *ICDE* 2006, *SIGMOD* 2006]



Regardless of model quality, base station knows $x^{(t)}$ to within ε
Better model → more "suppression" → fewer transmissions

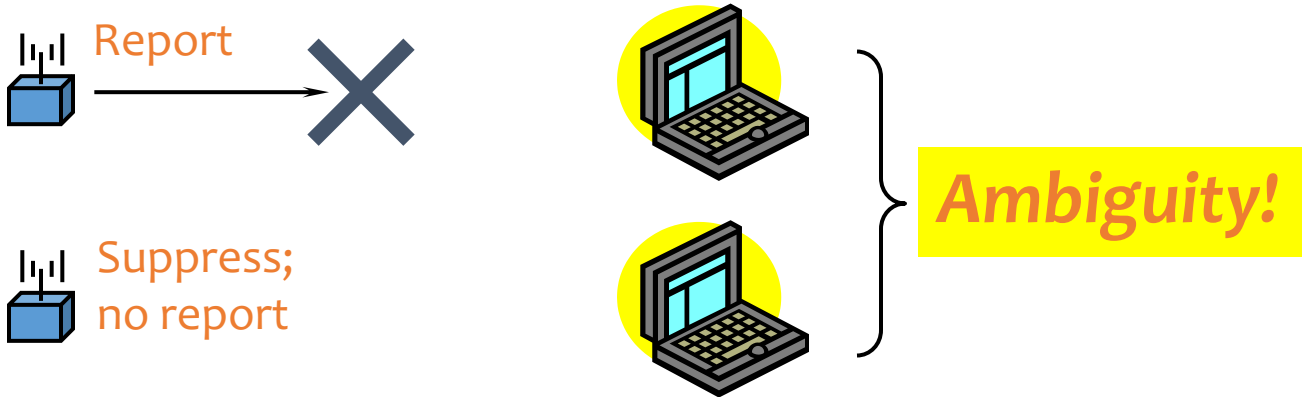
Simple temporal suppression

- Suppress transmission if
  |current reading – last transmitted reading| $\leq \varepsilon$
  - Model: $X^{(t)} = X^{(t-1)}$
- Effective when readings change slowly

- More sophisticated schemes are possible
  - Use more complex models
  - Suppress spatially
  - Send "deltas" rather than values

What's the catch?

- Message failure common in sensor networks
  - Interference, obstacles, congestion, etc.



- Is a non-report due to suppression or failure?
  - Without additional information/assumption, base station has to treat every non-report as plain "missing"—no accuracy bounds!

A few previous approaches

- Avoid missing data: ACK/Retransmit
  - Often supported by the communication layer
  - Still no guaranteed delivery—does not help with resolving ambiguity
- Deal with missing data
  - Interpolation
    - Point estimates are often wrong or misleading
    - Uncertainty is lost—important in subsequent analysis/action
  - Use a model to predict missing data
    - Can provide distributions instead of point estimates
    - But we have to trust the model!

BayBase: basic Bayesian approach

- Model $p(\mathbf{X}|\Theta)$ with parameters Θ
  - Do not assume Θ is known
  - Any prior knowledge can be captured by $p(\Theta)$
- $\mathbf{x}_{obs}$: data received by base station
- Calculate posterior $p(\mathbf{X}_{mis}, \Theta|\mathbf{x}_{obs})$
  - Joint distribution instead of point estimates
  - Quantifies uncertainty; model can be improved
    ← ***data-driven philosophy***

☞ Problem: non-reports are treated as generically missing

- But most of them are "engineered"
- Non-report ≠ no information!

How do we incorporate knowledge of suppression scheme?

BaySail

Bayesian Analysis of Suppression and Failure

- Bayesian, data-driven
- Add back some redundancy
- Infer with redundancy + knowledge of suppression scheme

Some intuition

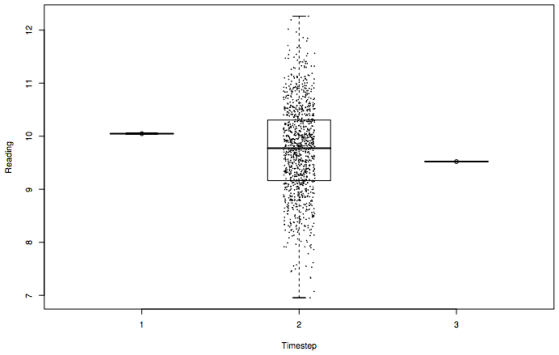


Figure 1: Missing value.

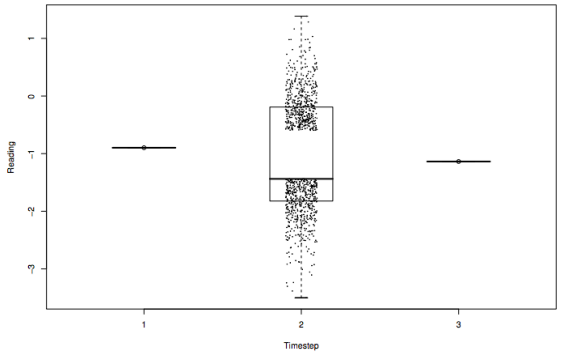


Figure 2: Missing value is failure.

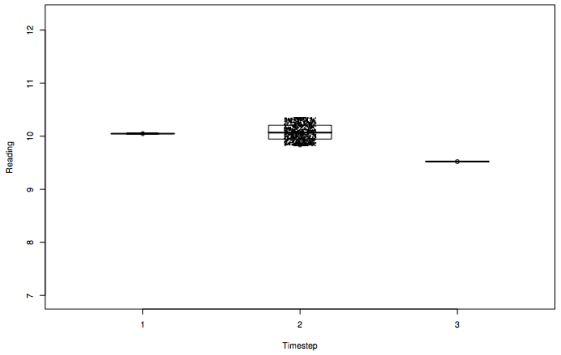


Figure 3: Missing value is suppression.

Redundancy strikes back

At app level, piggyback redundancy on each report

- ***Counter***: number of reports to base station thus far

A good CS trick, but…

- ***Timestamps+Direction Bits***:
  last *r* timesteps when node reported +
  bits indicating whether each report is caused by
  (actual – predicted > ε) or
  (predicted – actual > ε)

Why?!

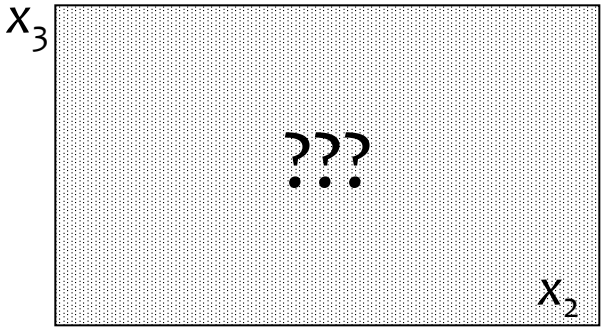
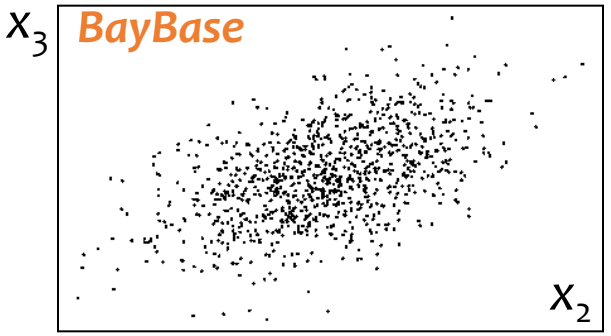
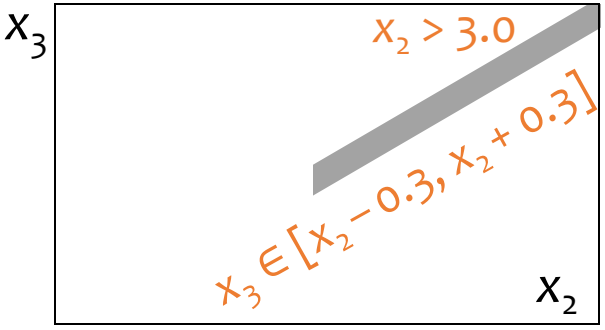
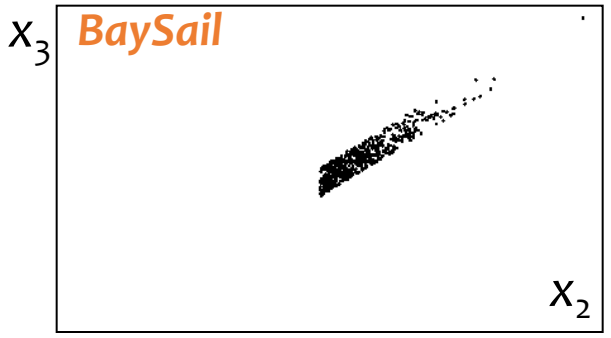
Suppression-aware inference

- Redundancy + knowledge of suppression scheme → hard constraints on $\mathbf{X}_{mis}$

  - Temporal suppression: $\varepsilon = 0.3$, prediction = last reported
  - Actual: (x_1,x_2,x_3,x_4) = (2.5/sent, 3.5/sent, 3.7/suppressed, 2.7/sent)
  - Base station receives: (2.5, nothing, nothing, 2.7)
  - With ***Timestamps+Direction Bits*** (r=1)
    - (2.5, failed & under-predicted, suppressed, 2.7 & over-predicted)
    - $x_2 - 2.5 > 0.3$; $-0.3 \leq x_3 - x_2 \leq 0.3$; $x_2 - 2.7 > 0.3$
  - With ***Counter***
    - One suppression and one failure in x_2 and x_3; not sure which
    - Hairy constraints!

Posterior: $p(\mathbf{X}_{mis}, \Theta|\mathbf{x}_{obs})$, with $\mathbf{X}_{mis}$ subject to constraints

Benefit of modeling/redundancy

	Just data	Bayesian, model-based AR(1) with uncertain parameter
No knowledge of suppression		
Knowledge of suppression & ***Timestamps+ Direction Bits***		

Inference

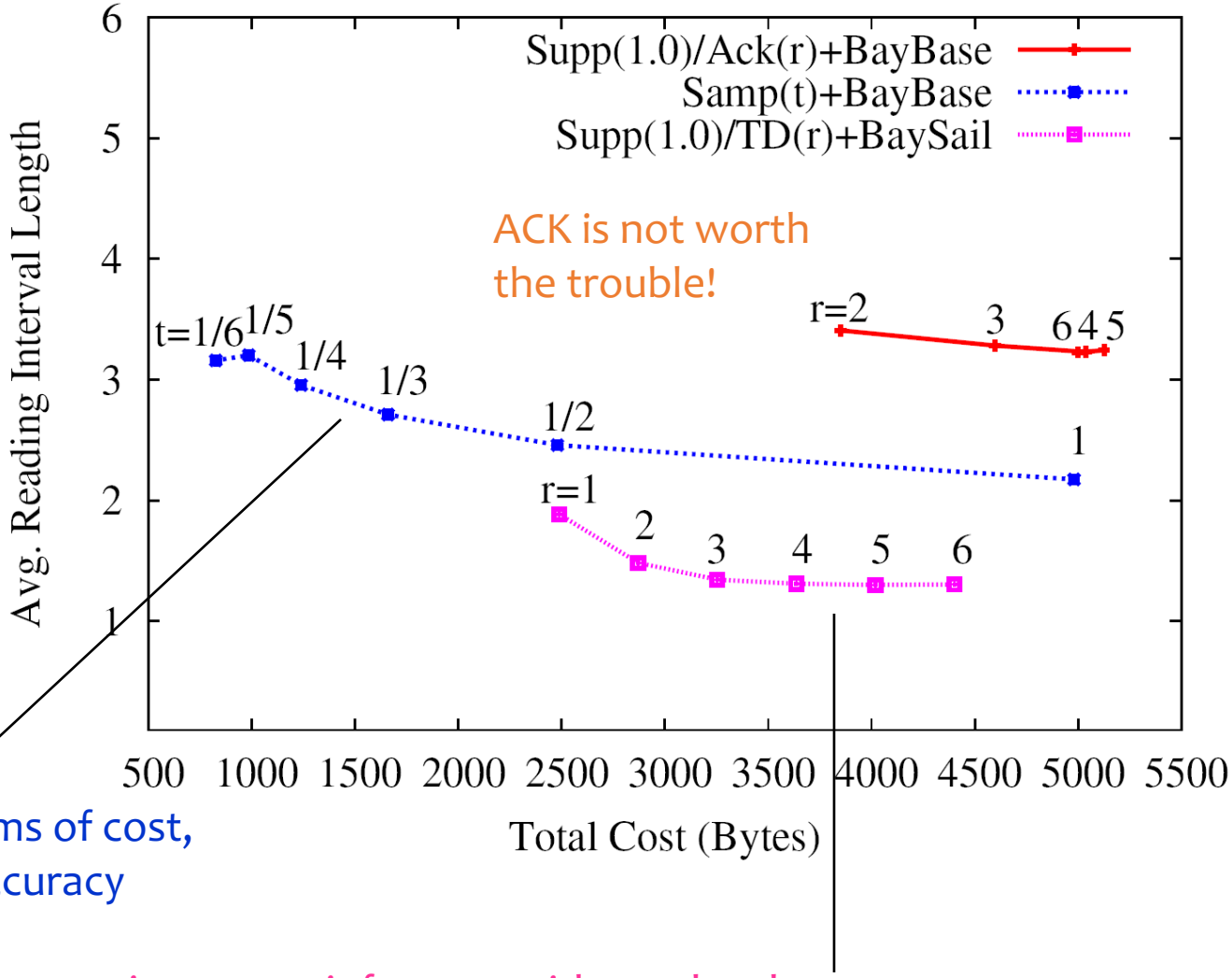
- Arbitrary distributions & constraints are difficult
  - Monte Carlo methods generally needed
  - Various optimizations apply under different conditions

- A simplified soil moisture model: $y_{s,t} = c_t + \phi\, y_{s,t-1} + e_{s,t}$
  - c_t is a series of known precipitation amounts
  - $\mathrm{Cov}(Y_{s,t}, Y_{s',t'}) = \sigma^2 \left(\phi^{|t-t'|}/(1-\phi^2)\right) \exp(-\tau|s-s'|)$
  - $\phi \in (0, 1)$ controls how fast moisture escapes soil
  - τ controls the strength of the spatial correlation over distance
- Given $\mathbf{y}_{obs}$, find $p(\mathbf{Y}_{mis}, \phi, \sigma^2, \tau \mid \mathbf{y}_{obs})$ subject to constraints
- Gibbs sampling
  - Markovian → sample each cluster of missing values in turn
  - Gaussian + linear constraints → efficient sampling method
    - Timestamps + direction bits give us linear constraints!

Cost vs. uncertainty in recovered data

- Cost: bytes tx'd (incl. message overhead)
- Quality: size of 80% high-density region
- 30% message failure
- ~60% suppression

Why this measure?

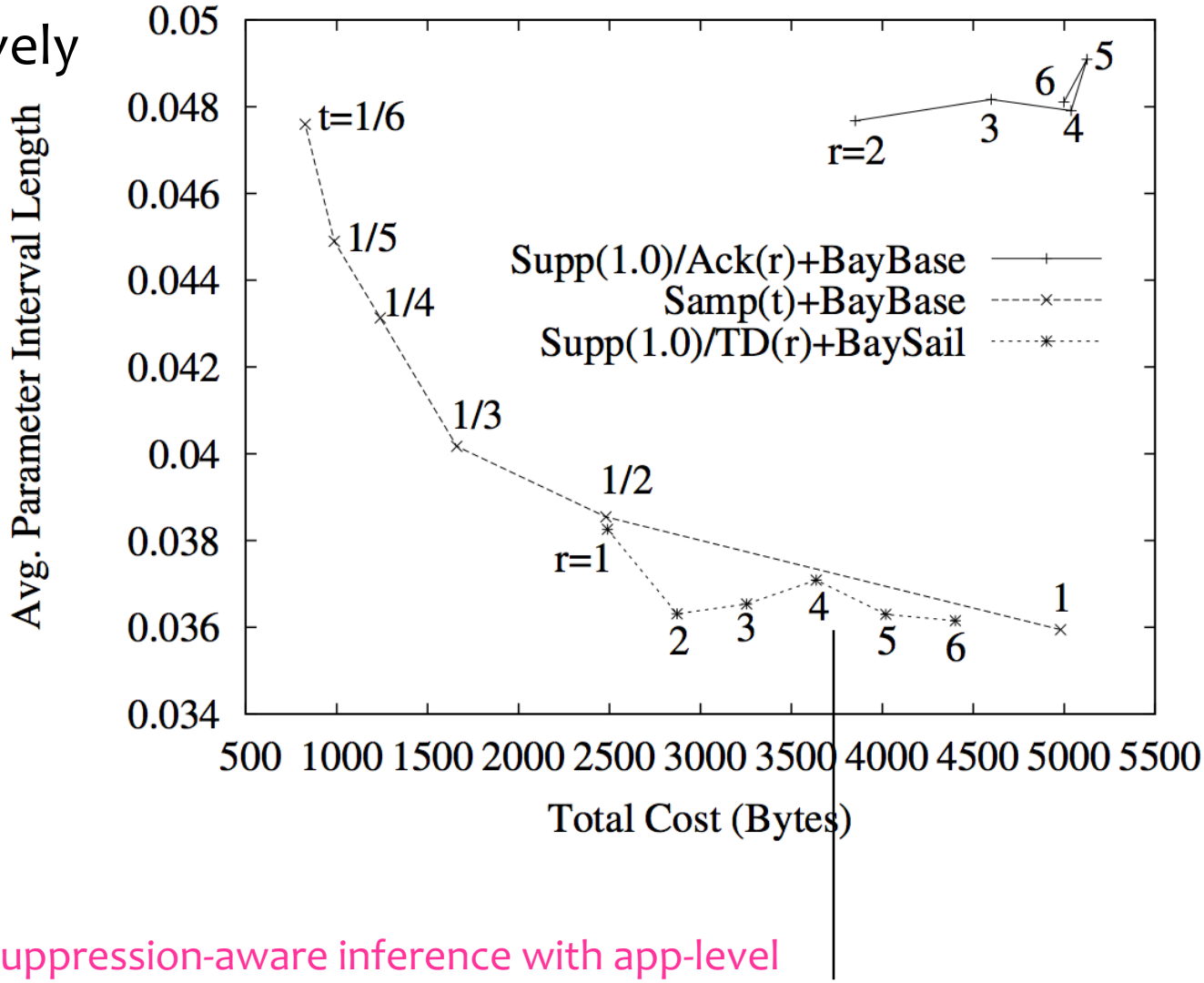


Sampling is okay in terms of cost, but has trouble with accuracy

Suppression-aware inference with app-level redundancy is our best hope to get higher accuracy

Cost vs. uncertainty in model parameter

For ϕ, in this relatively simple model

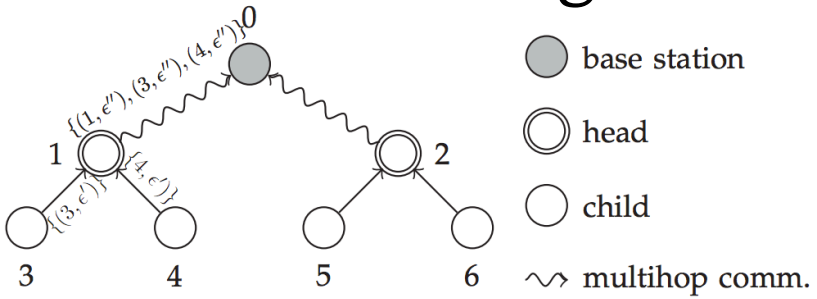


Suppression-aware inference with app-level redundancy is still best, but only with a small margin

Beyond *BaySail*

Zhang, Lum, Yang. *TKDE* 2012.

- Cascaded (spatiotemporal) suppression + convolution coding



Two-tier cascaded suppression.

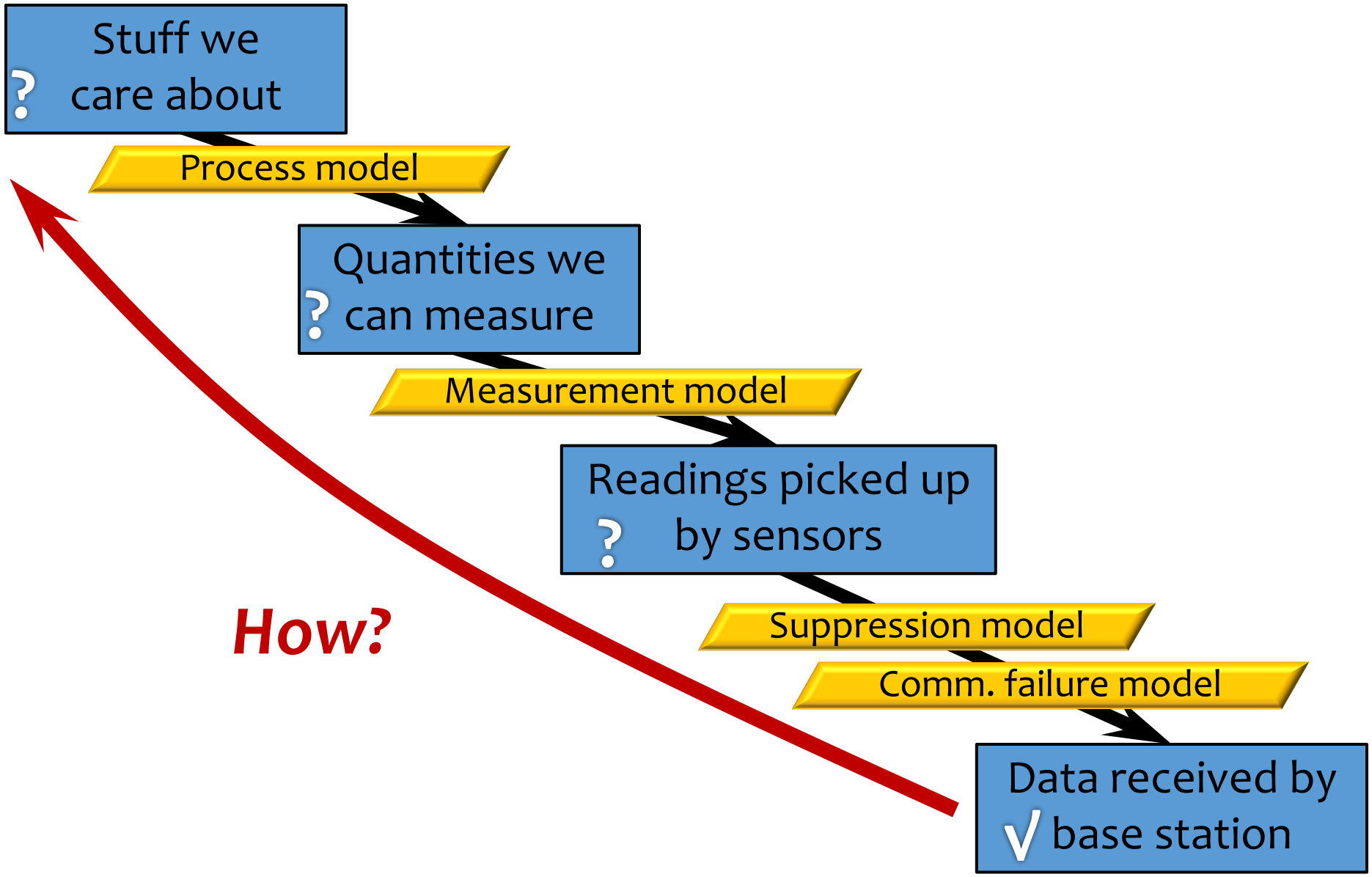
Cluster members report to their head using simple temporal suppression; cluster heads report a subset of values to root using MVN for suppression

Need to handle the messy case when suppression is based on wrong/outdated information about other nodes

A possible outcome of cascaded suppression with transmission failures. Wrong values/bounds are boxed.

Time	Node 3's reading	$3 \to 1$	Node 1's prediction	$1 \to 0$	Bound at base
1	$x^{(1)}$	$\surd$	$x^{(1)}$	$\surd$	$x^{(1)} \pm \epsilon'$
2	$x^{(2)}$	$\perp$	$x^{(1)}$	$\perp$	$x^{(1)} \pm (\epsilon' + \epsilon'')$
3	$x^{(3)}$	$\times$	$\boxed{x^{(1)}}$	$\perp$	$\boxed{x^{(1)} \pm (\epsilon' + \epsilon'')}$
4	$x^{(4)}$	$\perp$	$\boxed{x^{(1)}}$	$\perp$	$\boxed{x^{(1)} \pm (\epsilon' + \epsilon'')}$
5	$x^{(5)}$	$\surd$	$x^{(5)}$	$\surd$	$x^{(5)} \pm \epsilon'$

Model, model, everywhere



Summary

- Must consider how missing (dirty) data was generated in the first place
  - But things get really hairy if you want to fully account for the data collection process
  - Is the extra complexity worth it?
- To what extent data collection should be tailored to analysis?

A note on reviews (3-3-2-2)

- At least three important things that the paper says
- At least three interesting take-away points that you learned from the paper
  - They can be related to the paper's fundamental contribution, or just little things like a non-obvious pitfall, an uncanny insight, or a neat trick
- At least two things you didn't like about the paper
- At least two directions in which you can improve the paper or extend the work

Due the morning of the lecture on Piazza
No requirement on length—a good review can be as brief as 400 words

Presentation logistics

- You will present 1-2 papers, in groups of 2-3
- How it works
  - Read paper in advance; meet, discuss, make draft slides
  - Meet with me the day before the lecture
  - Read other reviews the morning of the lecture
    - You don't need to submit reviews for papers you present
  - Present, and lead discussion
    - Think about questions to ask!
  - Send me slides to be posted on the course website
- A tentative reading list/schedule will be up by this weekend on Google Sheet
  - Use Google Sheet to "sign up" for at least three papers you are willing to lead discuss