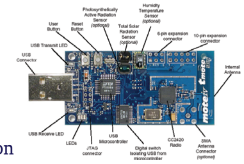
Data-Driven Processing In Sensor Networks

Jun Yang
Duke University
October 12, 2007

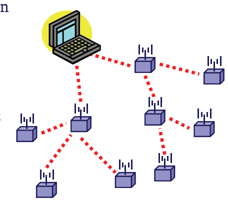
DUKE COMPUTER SCIENCE

What is a sensor network?

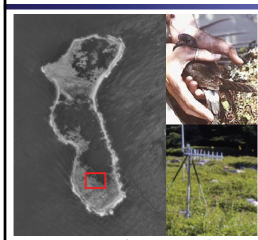
- Tiny, untethered nodes with severe resource constraints
  - Sensors, e.g., light, moisture, …
  - Tiny CPU and memory
  - Battery power
  - Limited-range radio communication
    - Usually dominates energy consumption



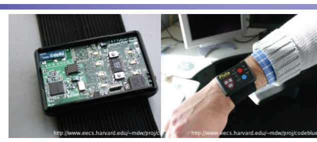
- Nodes form a multi-hop network rooted at a base station
  - Base station has plentiful resources and is typically tethered or at least solar-powered



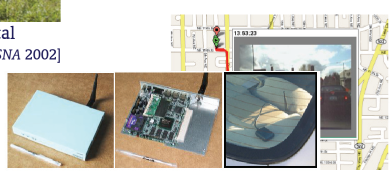
Sensor network applications



Environmental
[Mainwaring et al., *WSNA* 2002]

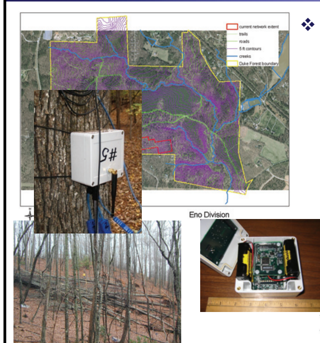


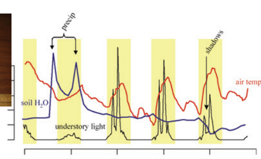
Medical
[Shnayder et al., Harvard Tech. Rep. 2005]



Urban
[Hull et al., *SenSys* 2006]

Duke Forest deployment





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

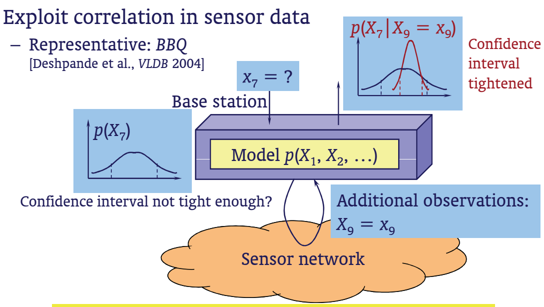
What do ecologists want?

- Collect all data (to within some precision)
  - Continuous "SELECT *": the most boring SQL query
- Fit stochastic models using data collected
  - Cannot be expressed as SQL queries

☞ *Sorry—this talk doesn't cover any of our favorite SQL queries (selection, join, aggregation…)*

Model-driven data collection: pull

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



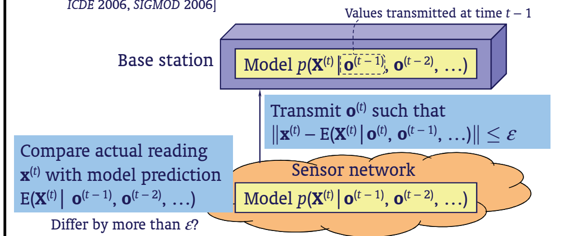
Answer correctness depends on model correctness
Risk missing the unexpected

Data-driven philosophy

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

Data-driven: push

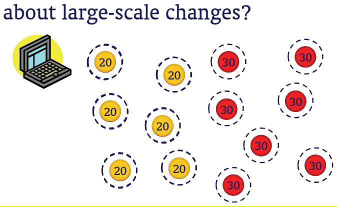
- Exploit correlation in data + put smarts in network
  - Representatives: *Ken* [Chu et al., *ICDE* 2006], *Conch* [Silberstein et al., *ICDE* 2006, *SIGMOD* 2006]



Regardless of model quality, base station knows $\mathbf{x}^{(t)}$ to within ε
Better model $\Rightarrow$ fewer transmissions

Temporal suppression example

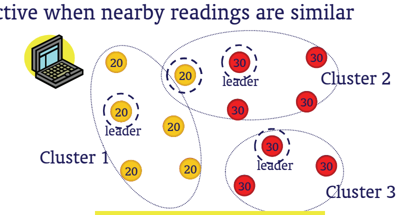
- Suppress transmission if
  $|$current reading – last transmitted reading$| \le \varepsilon$
  - Model: $X^{(t)} = x^{(t-1)}$
- Effective when readings change slowly
- What about large-scale changes?



Phenomenon is simple to describe, but all nodes transmit!

Spatial suppression example

- "Leader" nodes report for cluster
- Others suppress if
  $|$my reading – leader's reading$| \le \varepsilon$
  - Model: $X_{\text{me}} = x_{\text{leader}}$
- Effective when nearby readings are similar



Leaders always transmit!

Combining spatial and temporal

Spatiotemporal suppression condition = ?

- Temporal **AND** spatial?
  - I.e., suppress if both suppression conditions are met
  - Results in less suppression than either!
- Temporal **OR** spatial?
  - I.e., suppress if either suppression condition is met
  - Base station cannot decide whether to set suppressed value to the previous value (temporal) or to the nearby value (spatial)!

Outline

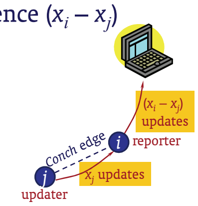
- How to combine temporal and spatial suppressions effectively
  - *Conch* [Silberstein et al., *SIGMOD* 2006]
- What to do about _______—the dirty little secret of suppression
  - *BaySail* [Silberstein et al., *VLDB* 2007]

Conch = constraint chaining

13

Temporally monitor spatial constraints (edges)

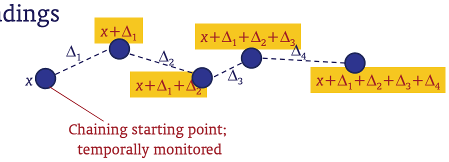
- x_i and x_j change in similar ways $\Rightarrow$ temporally monitor edge difference $(x_i - x_j)$
  - "Difference" can be generalized
- One node is reporter and the other updater
  - Reporter tracks $(x_i - x_j)$ and transmits it to base station if its value changes
  - Updater transmits its value updates to reporter
    - I.e., temporally monitor remote input to the spatial constraint



Recovering readings in *Conch*

14

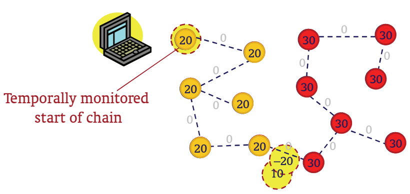
- Base station "chains" monitored edges to recover readings



- Discretize values to avoid error stacking
  - $[k\varepsilon, k\varepsilon+\varepsilon) \rightarrow k$
  - Monitor discretized values exactly
    - Discretization is the only source of error
    - No error introduced by suppression

Conch example

15



Only "border" edges transmit to base station
Combines advantages of both temporal and spatial suppression

Choosing what to monitor

16

- A spanning forest is necessary and sufficient to recover all readings
  - Each edge is a temporally monitored spatial constraint
  - Each tree root is temporally monitored
    - Start of chain

  (For better reliability, more edges can be monitored at extra cost)

- Some intuition
  - Choose edges between correlated nodes
  - Do not connect erratic nodes
    - Monitor them as singleton trees in the forest

Cost-based forest construction

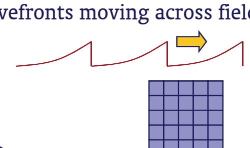
17

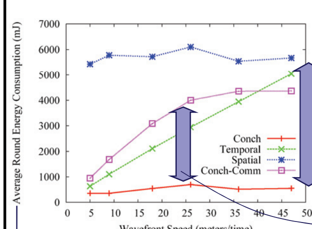
- Observe
  - In pilot phase, use any spanning forest to collect data
    - Even a poor spanning forest correctly collects all data
- Optimize
  - Use collected data to assign monitoring costs
    - # of rounds in which monitored value changes
  - Build a min-cost spanning forest (e.g., *Prim's*)
- Re-optimize as needed
  - When actual costs differ significantly from those used by optimization

Wavefront experiment

18

- Simulate periodic vertical wavefronts moving across field, where sensors are randomly placed at grid points





Conch beats both pure temporal and pure spatial

Communication tree is a poor choice for monitoring; optimization makes a huge difference

Based on accounting of bytes sent/received on Mica2 nodes

Conch discussion

19

- Key ideas in *Conch*
  - Temporally monitor spatial constraints
  - Monitor locally—with cheap two-node spatial models
  - Infer globally—through chaining
  - Push/suppress not only between nodes and base station, but also among nodes themselves
  - Observe and optimize
- Vision for ideal suppression
  - Number of reports $\propto$ description complexity of phenomenon

What's the catch?

Outline

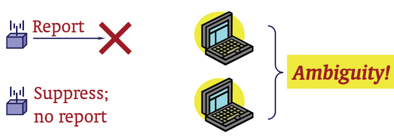
20

- How to combine temporal and spatial suppressions effectively
  - *Conch* [Silberstein et al., *SIGMOD* 2006]
- What to do about ***failures***—the dirty little secret of suppression
  - *BaySail* [Silberstein et al., *VLDB* 2007]

Failure and suppression

21

- 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

22

- Avoid missing data: ACK/Retransmit
  - Often supported by the communication layer
  - Still no guaranteed delivery $\rightarrow$ 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

23

- Model $p(\mathbf{X} \mid \Theta)$ 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 $p(\mathbf{X}_{mis}, \Theta \mid \mathbf{x}_{obs})$
  - Joint distribution instead of point estimates
  - Quantifies uncertainty in model; model can be improved

$\Leftarrow$ data-driven philosophy

- Problem: non-reports are treated as generically missing
  - But most of them are "engineered"
  - Non-report $\neq$ no information!

How do we incorporate knowledge of suppression scheme?

BaySail

24

Bayesian Analysis of Suppression and Failure

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

Redundancy strikes back

25

At app level, piggyback redundancy on each report

- ***Counter***: number of reports to base station thus far — ***Good systems idea!***
- ***Timestamps***: last r timesteps when node reported — ***Not that cute…***
- ***Timestamps+Direction Bits***: in addition to the last r reporting timesteps, bits indicating whether each report is caused by (actual – predicted $> \varepsilon$) or (predicted – actual $> \varepsilon$) — ***Why on earth?!***

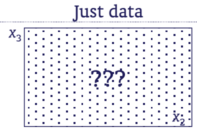
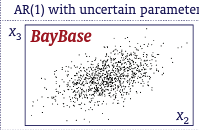
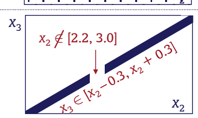
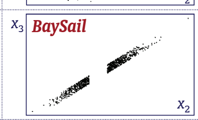
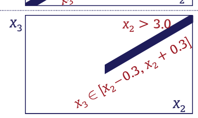
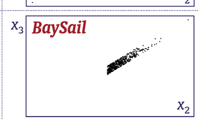
Suppression-aware inference

26

- Redundancy + knowledge of suppression scheme ⇒ hard constraints on $\mathbf{X}_{\text{mis}}$
  - Temporal suppression with $\varepsilon = 0.3$, 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*** (r=1)
    - (2.5, failed, suppressed, 2.7)
    - $|x_2 - 2.5| > 0.3$; $|x_3 - x_2| \le 0.3$; $|2.7 - x_2| > 0.3$
  - With ***Timestamps+Direction Bits*** (r=1)
    - (2.5, failed & under-predicted, suppressed, 2.7 & over-predicted)
    - $x_2 - 2.5 > 0.3$; $-0.3 \le x_3 - x_2 \le 0.3$; $x_2 - 2.7 > 0.3$
  - With ***Counter***
    - One suppression and one failure in x_2 and x_3; not sure which
    - A very hairy constraint!
- Posterior: $p(\mathbf{X}_{\text{mis}}, \Theta \,|\, \mathbf{x}_{\text{obs}})$, with $\mathbf{X}_{\text{mis}}$ subject to constraints

Benefit of modeling/redundancy

27

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

Redundancy design considerations

28

- Benefit: how much uncertainty it helps to remove
  - *Counter* can cover long periods, but helps very little in bounding particular values
- Energy cost
  - *Counter* < *Timestamps* < *Timestamps+Direction Bits*
- Complexity of in-network implementation
  - Coding app-level redundancy in TinyOS was much easier than finding the right parameters to tune for *ACK/Retransmit*!
- Cost of out-of-network inference
  - May be significant even with powerful base stations!

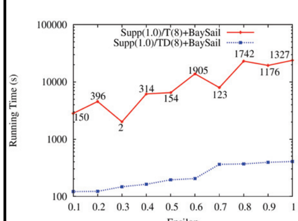
Inference

29

- Arbitrary distributions & constraints: difficult in general
  - 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
  - $\text{Cov}(Y_{s,t}, Y_{s',t'}) = \sigma^2 (\phi^{|t-t'|}/(1-\phi^2)) \exp(-\tau \|s - s'\|)$
  - $\phi \in (0, 1)$ controls how fast moisture escapes soil
  - τ controls the strength of the spatial correlation over distance
- Given $\mathbf{y}_{\text{obs}}$, find $p(\mathbf{Y}_{\text{mis}}, \phi, \sigma^2, \tau \,|\, \mathbf{y}_{\text{obs}})$ subject to constraints
- Gibbs sampling
  - Markovian ⇒ okay to sample each cluster of missing values in turn
  - Gaussian + linear constraints ⇒ efficient sampling methods

Inference cost

30



Timestamps translate to "$|\ldots| > \varepsilon$" constraints (disjunction); difficult to work with; naïve technique generates lots of rejected samples

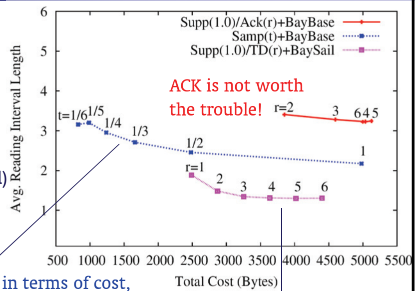
Timestamps+Direction Bits translate to a set of linear constraints; use [Rodriguez-Yam, Davis, Scharf 2004] and there are no rejections

>100× speed-up!
Major reason for adding the direction bits!

Energy cost vs. inference quality

31

- 30% message failure rate
- Roughly 60% suppression
- Cost: bytes transmitted (including any message overhead)
- Quality: size of 80% high-density region



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

BaySail discussion

32

- ❖ Suppression vs. redundancy
  - Goal of suppression was to remove redundancy
  - Now we are adding redundancy back—why?
  - Without suppression, we have to rely on naturally occurring redundancy ↔ want to control where redundancy is needed, and how much
- ❖ Many interesting future directions
  - Dynamic, local adjustments to ε and degree of redundancy
  - In-network resolution of suppression/failure
  - Failure modeling
  - Provenance: is publishing received/interpolated values enough?

Concluding remarks

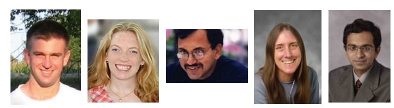
33

All models are wrong, but some models are useful
— George Box

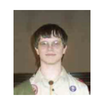
- ❖ Data-driven approach
  - Use model to optimize, not to substitute for real data → suppression
  - Quantify uncertainty in models; use data to learn/refine → Bayesian
- ☞ ***Conch***: suppression by chaining simple spatiotemporal models
- ☞ ***BaySail***: suppression-aware inference with app-level redundancy to cope with failure (suppression's dirty little secret)
- ❖ This model-based stuff is not just for statisticians!
  - Cost-based optimization
  - Interplay between system design and statistical inference
  - Representing and querying data with uncertainty

Acknowledgement

34



Adam Silberstein, Rebecca Braynard, Pankaj Agarwal, Carla Ellis, Kamesh Munagala (computer science)



Greg Filpus (undergrad)

Jim Clark (ecology)

Alan Gelfand, Gavino Puggioni (statistics)

Paul Flikkema (EE, NAU)

National Science Foundation

35

Thanks!

Duke Database Research Group

http://www.cs.duke.edu/dbgroup/

Related work

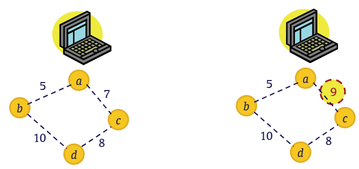
36

- ❖ Sensor data acquisition/collection
  - *BBQ* [Deshpande et al. *VLDB* 2004], *Snapshot* [Kotidis *ICDE* 2005], *Ken* [Chu et al. *ICDE* 2006], *PRESTO* [Li et al. *NSDI* 2006], contour map [Xue et al. *SIGMOD* 2006] , …
- ❖ Sensor data cleaning
  - *ESP* [Jeffery et al. *Pervasive* 2006], *SMURF* [Jeffery et al. *VLDB* 2006], *StreamClean* [Khoussainova et al. *MobiDE* 2006], …
- ❖ Uncertainty in databases
  - *MYSTIQ* [Dalvi & Suciu VLDB 2004], *Trio*/*ULDB* [Benjelloun et al. VLDB 2006], *MauveDB* [Deshpande & Madden SIGMOD 2006], *factors* [Sen & Deshpande ICDE 2007], …

Conch redundancy

37

- Monitor more edges/nodes!



$d = a + 5 + 10$ and
$d = a + 9 + 8$ cannot both be true!
A failure occurred—but where?

Conch recovery

38

- Constraints
  - True node and edge values $\mathbf{x}_t$ must be consistent
  - Received values $\mathbf{x}_t^{\text{obs}}$ are true
  - Non-reported values stay same (as time $t - 1$) or reports failed
- Maximum likelihood approach: roughly speaking, find $\mathbf{x}_t$ that maximizes $p(\text{receiving } \mathbf{x}_t^{\text{obs}} \,|\, \mathbf{x}_t, \mathbf{x}_{t-1})$
  - Assume independent failures with known probabilities
  - Assume known change probabilities
  - Can formulate as MIP

Maximize:

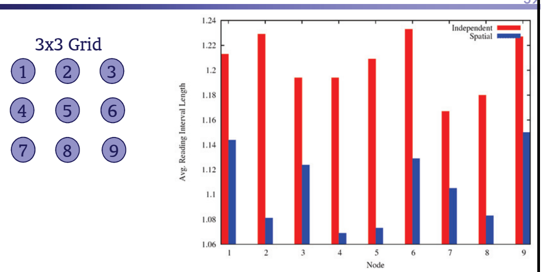
$$\sum_{u_i \in N_m - N_r} x_i \log \frac{c_i f_i}{1 - c_i} + \sum_{e_{ij} \in E_m - E_r} y_{ij} \log \frac{c_{ij} f_{ij}}{1 - c_{ij}},$$

Subject to:

$$\forall u_i \in N_m - N_r: \ (x_i)(\max) \geq |v_i - v_i^{old}| \tag{5}$$
$$\forall e_{ij} \in E_m - E_r: \ (y_{ij})(\max) \geq |d_{ij} - d_{ij}^{old}| \tag{6}$$
$$\forall e_{ij} \in E_m: \ v_i + d_{ij} = v_j \tag{7}$$
$$\forall u_i \in N_r: \ v_i = v_i^r \tag{8}$$
$$\forall e_{ij} \in E_r: \ d_{ij} = d_{ij}^r \tag{9}$$

BaySail: infer with spatial correlation

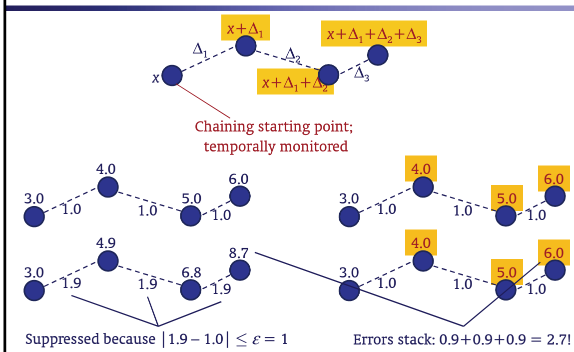
39



- Spatial correlation definitely helps!
  - Can you tell which nodes got less help?

Error stacking

40



Discretization

41