Understanding and Controlling Instrumented Physical Systems: Modeling is Complex, but Optimization is Easy

Bradley Bennett, Alberto Cerpa, Jessica Feng, Farinaz Koushanfar, Richard Park, Thomas Schoellhammer, Jennifer L. Wong, and Miodrag Potkonjak
Group for Stochastic Algorithmics http://www.cs.ucla.edu/~miodrag

Introduction: Data Integrity and Computational Sensing

We address two of the *canonical* problems in sensor networks:
•Data integrity
•Computational sensing.

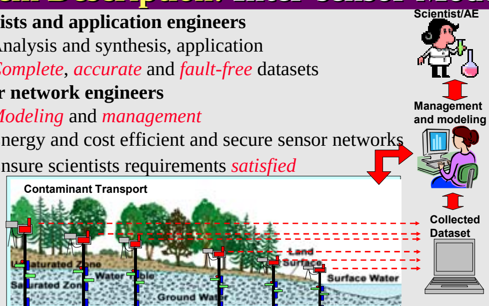
Due to the *large scale* and *distributed* nature of sensor networks, their *heterogeneous node structure*, *cost and power constraints*, operation in *unpredictable* and *unconditioned* (and *often harsh*) environmental surroundings and *inherent unreliability of sensors* sensor networks often collect data with *errors*, *faults* and *missing samples*.

We have developed a generic approach for both tasks (i.e. data integrity and computational sensing) that has three phases: *(i) statistical modeling*, *(ii) prediction*, and *(iii) fusion and analysis*.

Key conceptual novelties include new techniques for capturing of physical *hidden covariates*, *mapping of statistical problem into combinatorial domain*, prediction using *constraint manipulation*, and introduction of *class-based networks* for analysis of computational complexity of associated optimization problems. We demonstrate that commonly used modeling techniques are inherently inaccurate and that majority of the optimization problems are surprisingly easy to solve.

Problem Description: Inter-sensor Modeling and Prediction

- **Scientists and application engineers**
  - Analysis and synthesis, application
  - *Complete*, *accurate* and *fault-free* datasets
- **Sensor network engineers**
  - *Modeling* and *management*
  - Energy and cost efficient and secure sensor networks
  - Ensure scientists requirements *satisfied*



Three phase approach

- **Phase-1: Modeling**
  - *Statistical data-driven* models
  - Unique features of computational sensor modeling
- **Phase-2: Prediction**
  - Constraint manipulation
- **Phase-3: Fusion and analysis**
  - Formulate data integrity as an *optimization problem*
  - Objective function: Minimize the *discrepancies* between the sensor readings and the models
  - Constraints: Model constraints and user's specified constraints

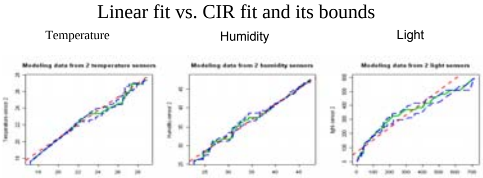
Experimental Results: Modeling, Prediction, and Optimization

Inter-sensor Modeling

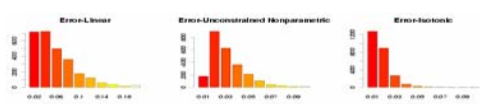
- **Data from sensors X and Y, find the model $\hat{y}=f(x)$**
- **Hidden covariate problem captured by *isotonicity constraint*:**
  - For model f: $x_1<x_2 \Rightarrow f(x_1)\leq f(x_2)$
- **Univariate CIR (*Combinatorial Isotonic Regression*):**
  - Given data (x_i, y_i, ω_i), i=1,…,K
  - Given an error measure ε_p and $x_1<x_2<x_3<\ldots<x_K$
  - ε_p isotonic regression is set $(x_i, \hat{y}_i)$, i=1,…,K s.t.
  - Objective function: min $\varepsilon_p(x_i, \hat{y}_i, \omega_i)$
  - Constraints: $\hat{y}_1\leq\hat{y}_2\leq\hat{y}_3\leq\ldots\leq\hat{y}_K$

Modeling

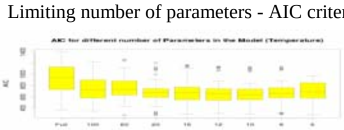
Linear fit vs. CIR fit and its bounds



Prediction error over all nodes



Limiting number of parameters - AIC criteria

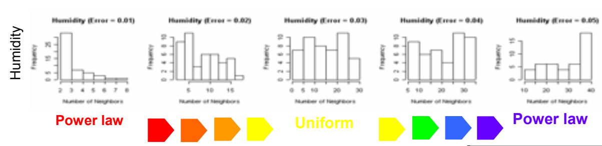


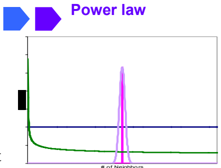
Combinatorial Domain-based Statistical Modeling

- **Flexibility of the combinatorial modeling**
  - ***Nodes:*** Different error norms, outlier elimination, robust regression, etc.
  - ***Edges:*** Maximum/minimum slope, adding and removing ordering constraints, etc.
  - ***Paths:*** Unimodular, convex, locally monotonic, number of break points, etc.
  - *Optimization-friendly*
- **Density estimation, consistency-based techniques**

Sleeping Coordination

Mean Error Bound	Prediction Method: Linear	Nonpar	CIR	Lim CIR	Average CPLEX Time (s)
1%	1	1	2	1	0.07
2%	3	3	6	5	0.31
3%	3	4	8	6	1.16
4%	4	4	10	9	1.16
5%	6	8	13	12	0.98
6%	7	12	24	17	3.22





- **Constant degree networks**
- **Random graph networks**
  - Suitable for probabilistic analysis, difficult for optimization
- **Small world networks**
  - Models different phenomena: e.g. social networks, Internet
  - Less suitable for probabilistic analysis
  - Better for optimization that random graphs, but can be also difficult
- **Class-based networks**
  - Explain a group of phenomena: e.g. genome functional interaction, WADN
  - Less suitable for probabilistic analysis
  - Optimization-friendly