

UCLA

Posters

Title

Bayesian selection of non faulty sensors (SYS 6)

Permalink

<https://escholarship.org/uc/item/4v58r0kp>

Authors

Kevin Ni
Greg Pottie

Publication Date

2006

Bayesian Selection of non-Faulty Sensors

Kevin Ni and Greg Pottie

Introduction: From a set of sensors, we need to determine which ones are faulty

Motivation

- **Sensor failures must be identified**
Faulty sensors must be identified in the field online so that the issue can be resolved. If sensor faults are not identified, then there will be no meaningful data collected for extended periods of time.
- **Reliable sensors may be trusted more**
By identifying a set of non-faulty sensors, we can determine expectations of proper behavior for future computations.

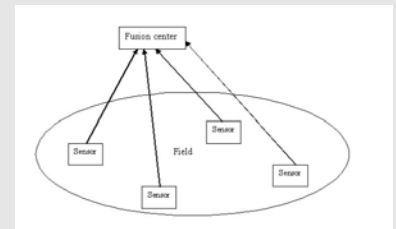
Issues

- **Requires defining what is a “good” sensor**
 - We define acceptable sensor behavior relative to the group of other nodes
 - If a sensor does not move with the group, we identify an issue
- **We need to identify what is the expected behavior of sensors**
In order to determine what is faulty, we need an idea of how a sensor should behave. We need to identify what kind of data we’re expecting from each sensor

Problem Description:

Problem setup:

- Sensors are recording temperatures over a field and forwarding their data to a fusion center
- We assume can allow for the field to be smoothly varying, i.e., no jumps in temperatures spatially
- We assume a Gaussian noise model across the field
- We first identify a trusted set, then use this to determine whether other sensors are faulty or non-faulty



Proposed Solution: Use M.A.P. to find a trusted subset and then determine faulty sensors

Problem formulation:

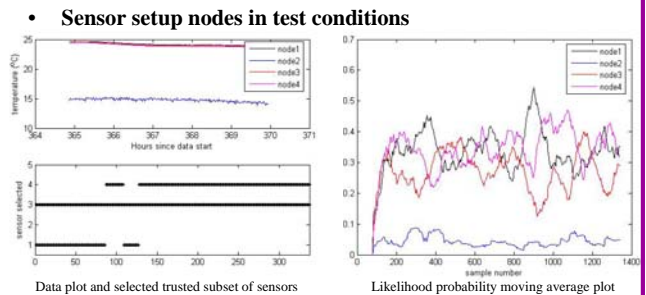
- **Enumerate the allowable subsets of sensors**
 - Subsets of sensors are required to have at minimum two sensors. Since we only test among a relatively small number of sensors, this is tractable.
 - Represented by $\vec{\phi}$, where the components $\phi_i = \{0, 1\}$
- **Use MAP to select the Bayesian optimal sensor subset**

$$\hat{\vec{\phi}} = \arg \max_{\text{all } \vec{\phi}} (P(\vec{\phi} | \vec{D}, \xi))$$
 We select subset with the maximum a-posteriori probability. This is the trusted subset.

Implementation:

- **The main issue is to find the posterior probability $P(\vec{\phi} | \vec{D}, \xi)$**
 - Involved with finding this is finding the prior probability $P(\vec{\phi} | \xi)$
 - We also must determine the likelihood value $f(\vec{D} | \vec{\phi}, \xi)$
- **We can estimate the covariance matrix in order to find the likelihood value at each time instant**
- **Prior probabilities can be based upon outside information, however we assume equal priors at the start and priors are updated with previous posterior probabilities.**
- **To estimate covariance matrix, we fit a linear model for the previous N data points for each sensor. Then we have the expected values with which we can estimate the covariance matrix.**
- **In order to account for offsets in the data, we subtract out the y-intercepts of the linear model from the data, effectively grounding the sensor data. This is in essence looking at the gradient of the data and not the actual data when determining the posterior**
- **Once we have the trusted sensor subset, we can determine the likelihood probabilities.**
- **Taking a moving average of the likelihood probabilities we can implement a threshold test and determine faulty or non faulty sensors.**

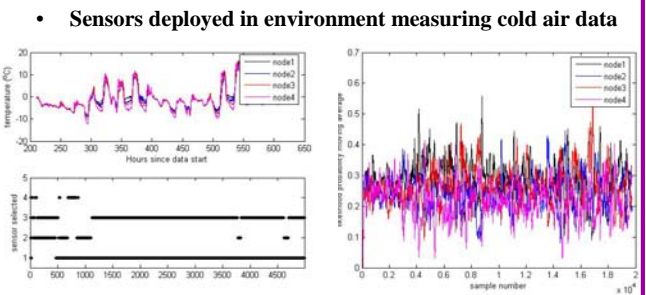
Results:



Data plot and selected trusted subset of sensors

Likelihood probability moving average plot

Threshold test correctly identifies node 2 as being faulty relative to the trusted subset



Data plot and selected trusted subset of sensors

Likelihood probability moving average plot

Threshold test generally works, and mostly identifies that all nodes are not faulty relative to the trusted subset. Every so often, a node is indicated as faulty

Conclusions:

- We can identify a trusted subset of sensors using M.A.P
- We require a better metric in determining faulty sensor nodes