Toward automated quality control for hydro-meteorological weather station data

Tom Dietterich
tgd@cs.orst.edu
@tdietterich

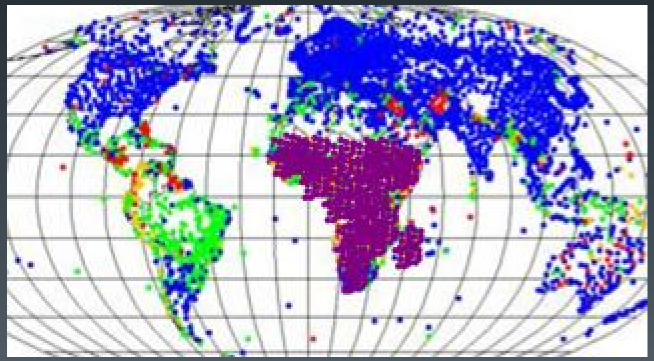
Irené Tematelewo

Outline

- TAHMO Project
- Sensor Network Quality Control
  - Rule-based methods
  - Probabilistic methods
  - SENSOR-DX approach
- Neighbor Regression for Precipitation
  - Improved anomaly detection

TAHMO: Motivation

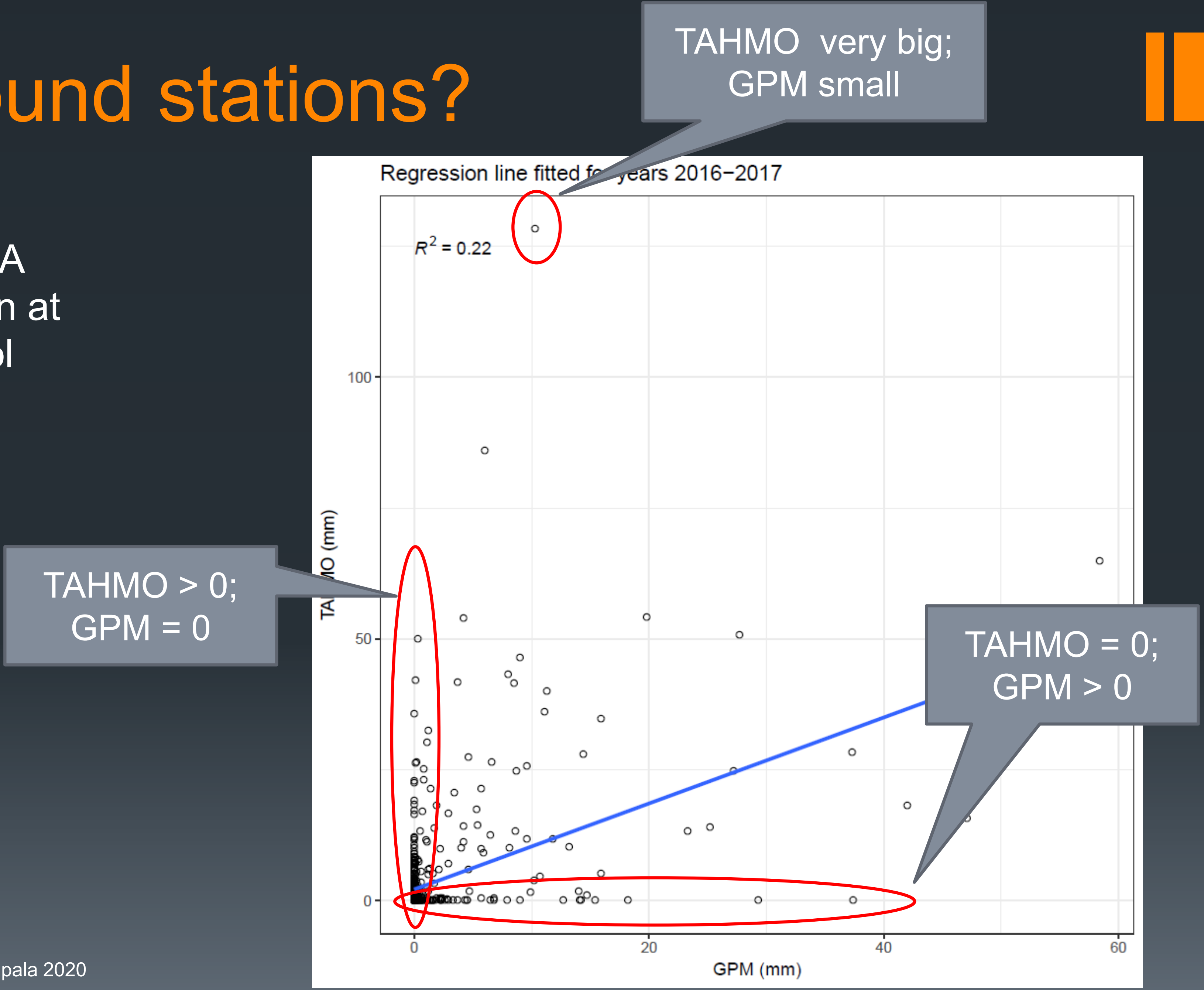
- Africa is very poorly sensed
  - Only a few weather stations reliably report data to WMO (blue points in map)
  - Poor sensing ➔No crop insurance ➔Low agricultural productivity



- TAHMO Goal:
  - Make Africa the best-sensed continent & improve agriculture
  - Self-sustaining non-profit company

Do we need ground stations?

- Scatterplot of precipitation estimate from satellite (NASA GPM) versus TAHMO station at South Tetu Girls High School

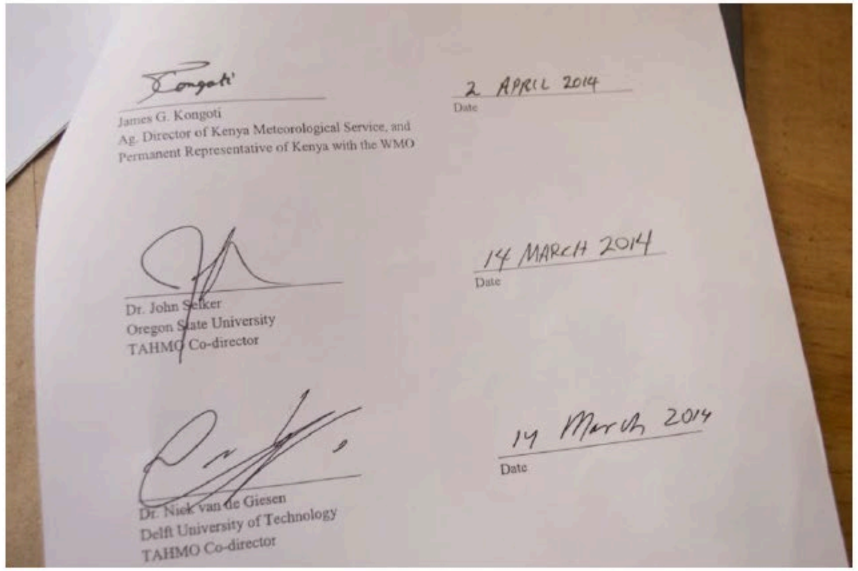


Business Plan

- Negotiate Memoranda of Understanding (MOUs) with each country in Sub-Saharan Africa
- Raise funds (gifts and grants) to develop and deploy weather stations
- Operating funds provided by selling the data
  - Free access for
    - The meteorological agency in each country
    - Education
    - Research
- Eager to collaborate with startups to create new businesses based on weather data

Memoranda of Understanding (MoUs)

MoU's	Close to complete
Kenya	Rwanda
Ghana	Ivory Coast
Malawi	Cameroon
Benin	Zambia
Togo	Senegal
Mali	
Burkina Faso	
Uganda	
Ethiopia	
Tanzania	
Nigeria	
South Africa	



Finances

- Deployment cost
  - 20,000 stations x $2000 per station = $40M
- Operating cost
  - $600/stations/year = $12M
- Weather data market
  - Estimate $40,000M/year

- Status: >500 stations deployed
  - Funding from USAID, UN, EU, IBM
  - School2School program

Technology

- Weather Stations
- Automated Quality Control

Generation 1 Weather Station

- cables
- 3 moving parts
- 5 components

Generation 3 station

- No moving parts
- No cables
- Two components

Generation 3 Features

- Solar power
- 6-month reserve battery
- GSM/GPRS radio
- GPS & Compass
- Temperature (3 ways)
- Relative Humidity
- Accelerometer
- Sonic wind
- Drip-count rain
- Shortwave solar radiation
- Barometer
- Lightning detector
- 5 open sensor ports: soil moisture etc.

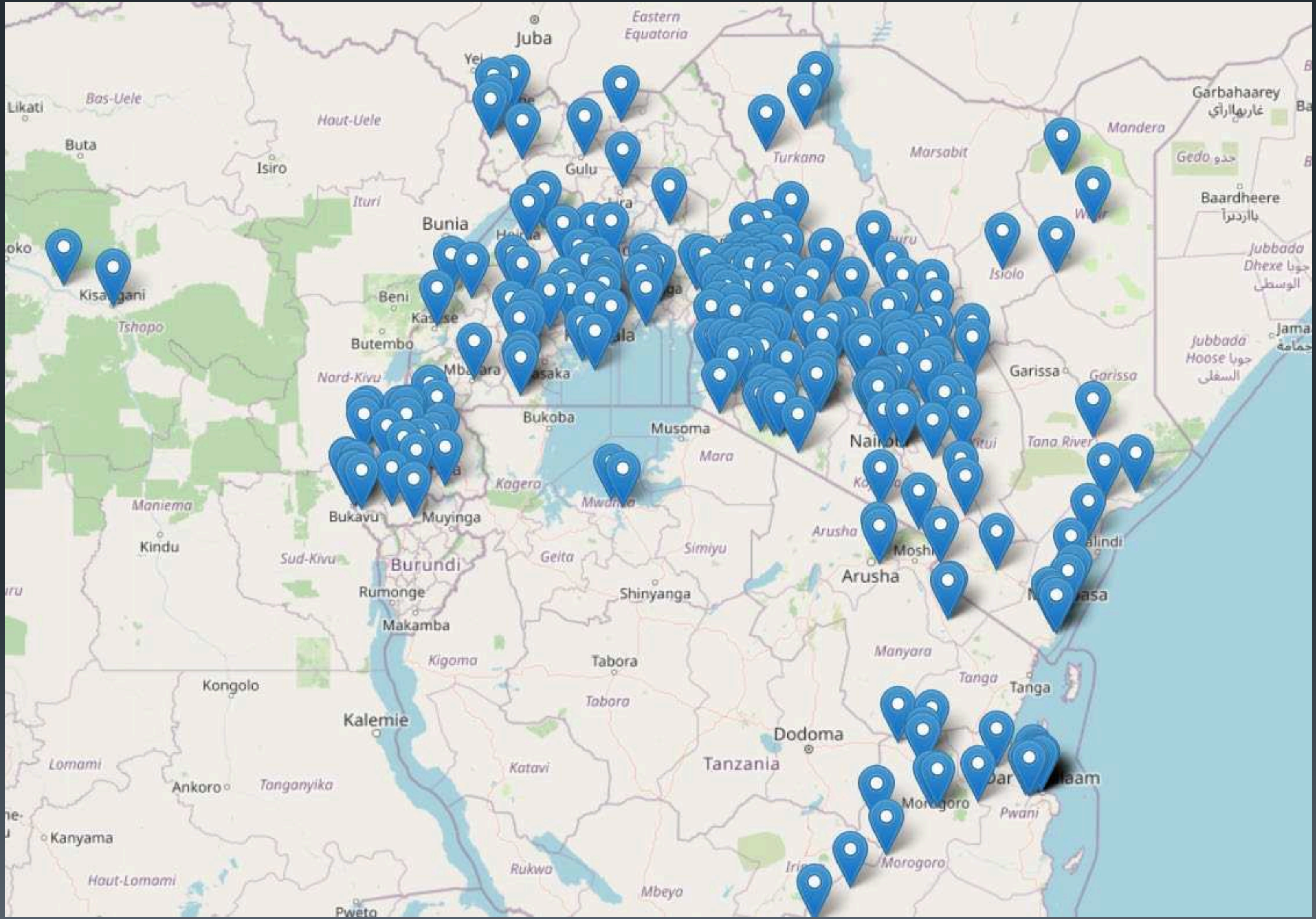
Station Placement and Security

- General strategy: Place stations at schools
  - Teacher monitors the station and clean it regularly
  - Use the station as an educational resource
    - TAHMO provides educational materials and lesson plans
    - Students can download data and analyze it

- School2School Program
  - Schools in US and Canada can purchase two stations
    - One for their school
    - One for a school in Africa
    - Students learn about their partner school starting with the weather

Current Status

Uganda and Kenya (Lake Victoria Region)



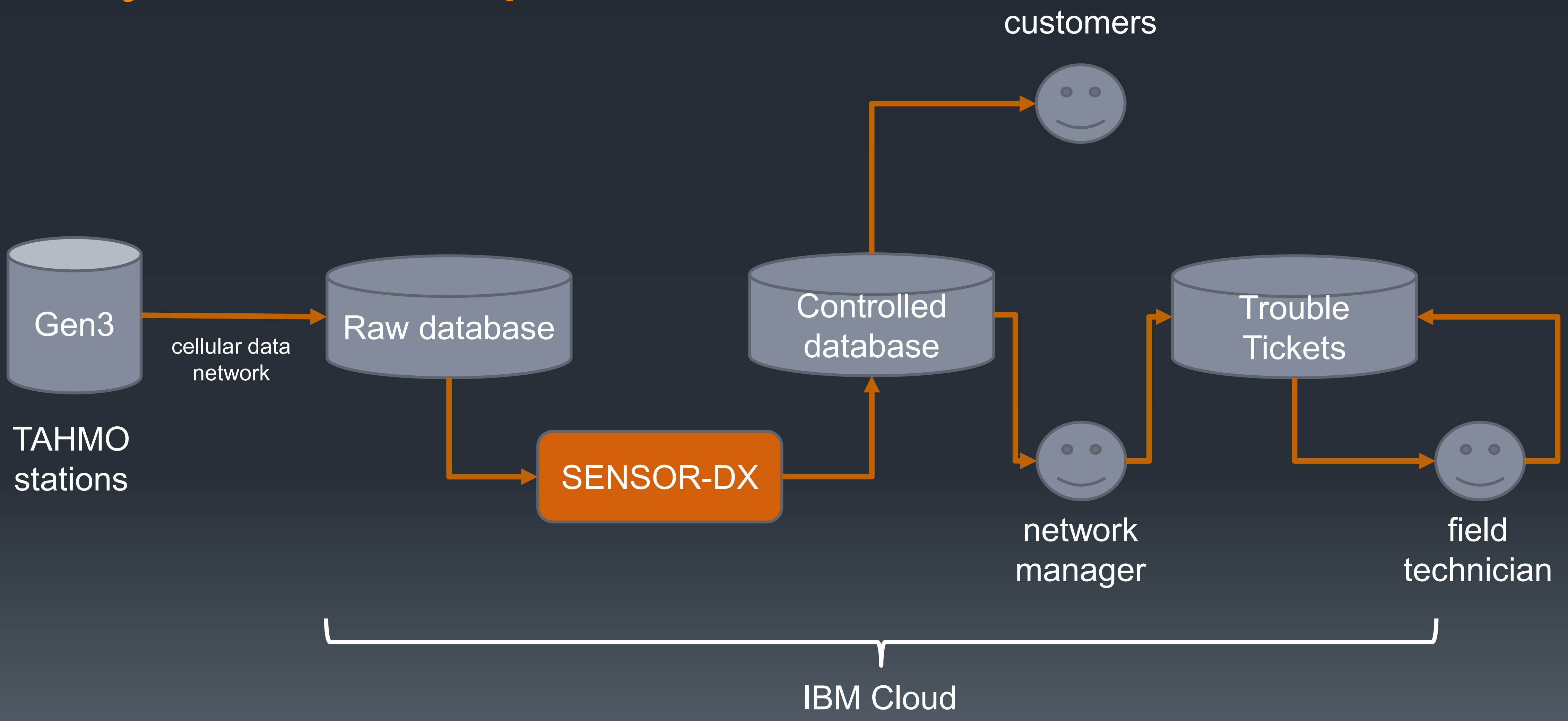
Quality Control

- Weather Sensors Fail
  - Solar radiation sensor gets dirty
  - Wind sensors (anemometers) get dirty or blocked
  - Rain gauge becomes obstructed
  - Novel failures occur often
- Battery Failure
  - Poor cellular telephone connectivity

Ant Infestation

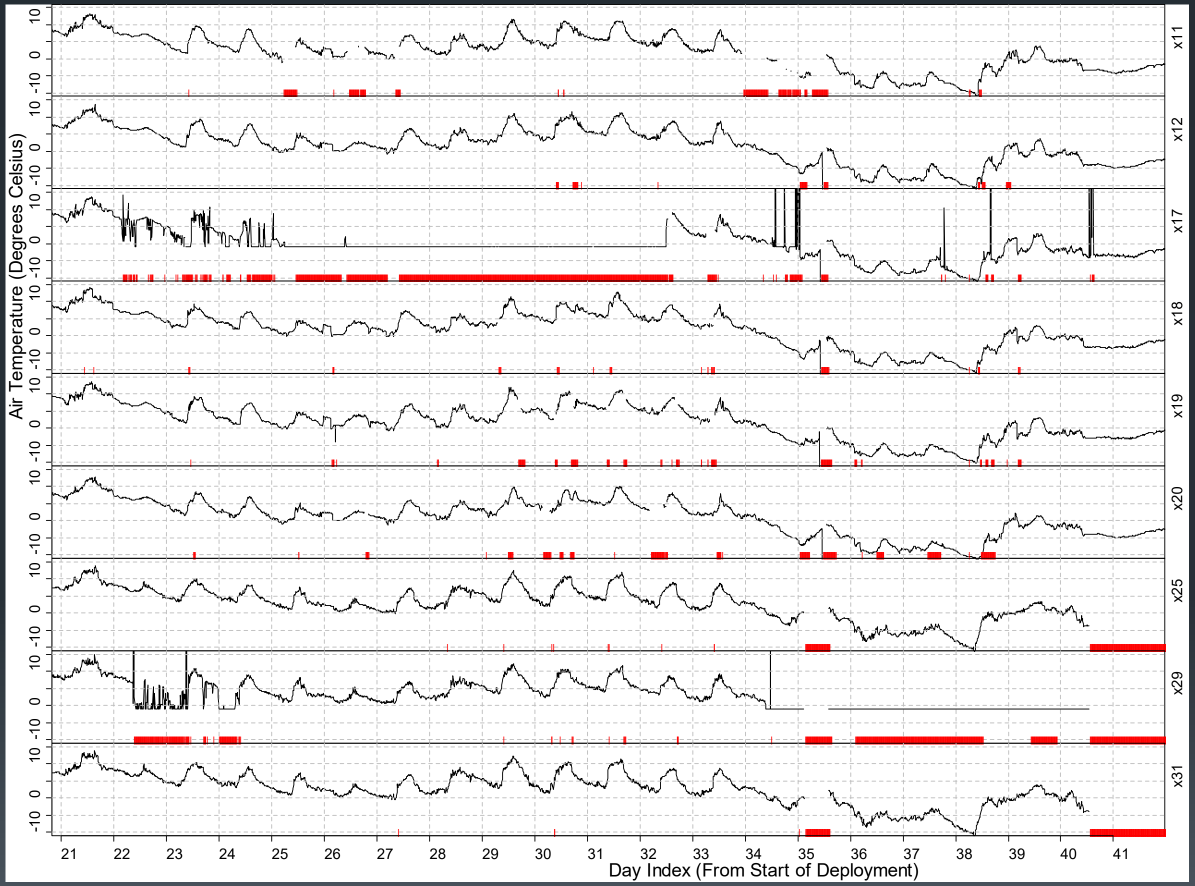
Wasps in the Anemometer

Quality Control Pipeline



Data Quality Control

- Goal: Identify all sensor values that correspond to malfunctioning sensors



Existing Approaches to Quality Control

- Manual Inspection (used at H J Andrews LTER)
- Complex Quality Control (OK Mesonet)
- Probabilistic Quality Control (Rawinsonde Network)

- All of these require large amounts of expert time
- TAHMO is much larger than these networks
- TAHMO will be larger than the networks used by the US National Weather Service
- We need a fully-automated QC method

Existing Methods 1: Complex Quality Control

- Rule-based approach that raises an alarm if a rule is violated
  - Step test: $x_{t+1} - x_t < \theta_1$
  - Flatline test: # of consecutive steps where $x_{t+1} = x_t$ must be $< \theta_2$
  - Buddy test: $|x_t - y_t| < \theta_3$ for two identical sensors x and y
  - etc.

Complex Quality Control

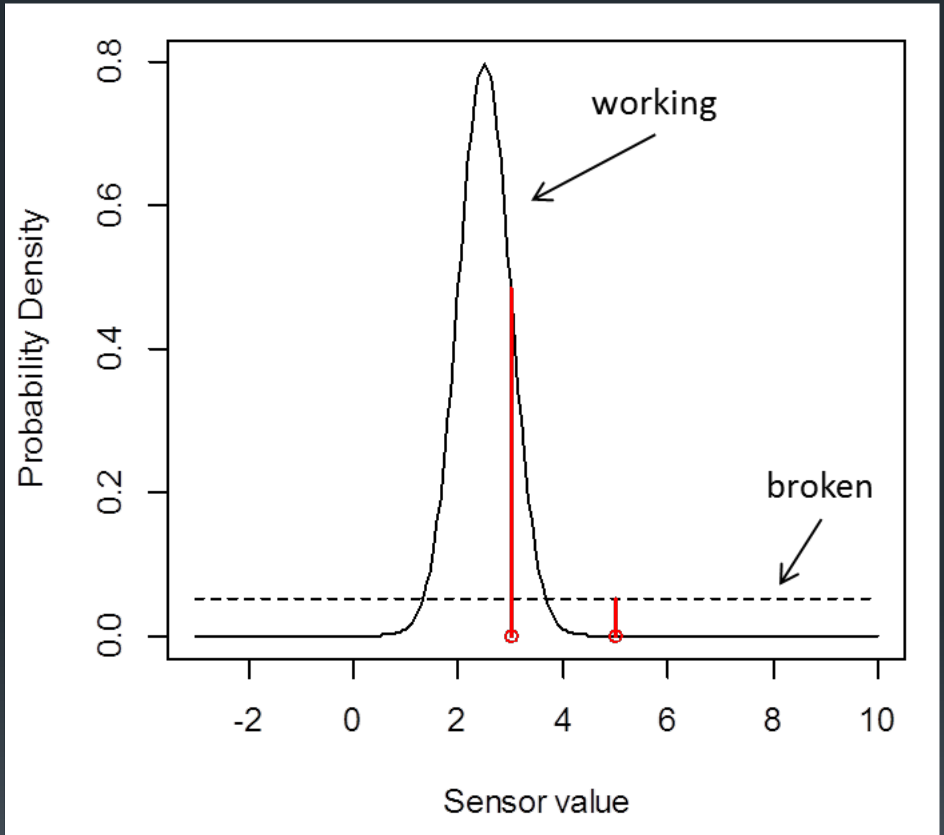
- Problems:
  - No unifying principles
  - Considers each variable separately
  - Hard to maintain
- Advantages:
  - Practical
  - Easily extended by adding new rules
  - Does not require a model of the signals

Probabilistic Quality Control

- Define s_t to be the state of the sensor at time t

  $s_t \in \{0,1\}$ where 0 = OK and 1 = Broken
- $P(x_t | s_t = 0)$ is the “normal” probability density for the sensor
- $P(x_t | s_t = 1)$ is the “broken” probability density for the sensor
- $P(s_t)$ is the prior over sensor states
- Query:

$$P(s_t | x_t) = \frac{P(s_t)P(x_t | s_t)}{P(x_t)}$$



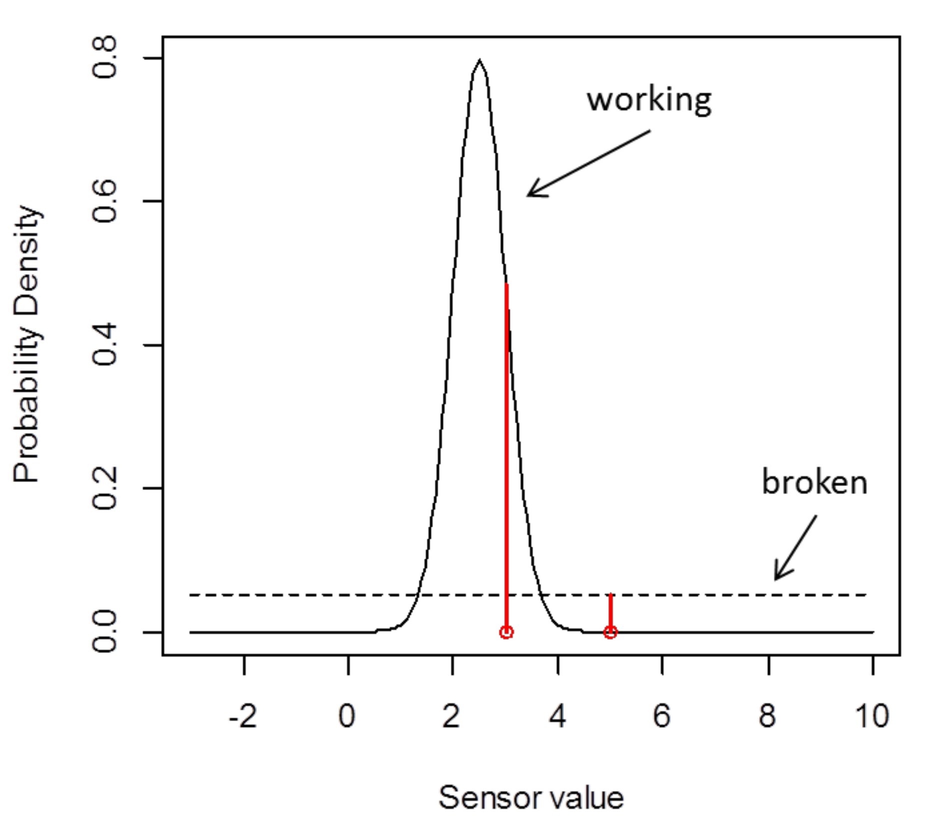
Challenge: Modeling the Broken distribution

- Modeling $P(x|s = 0)$
  - Lots of data; virtually all data points are from this case
  - However, the distribution may still be complex

- Modeling $P(x|s = 1)$ is very difficult
  - Bad sensor values are rare, so little data
  - Sensors break in novel ways, so hard to predict the sensor readings

Hack: "Junk Bucket" Distribution

- Assume $P(x_t|s_t = 1)$ is the uniform distribution
- This is equivalent to setting a threshold on $P(x_t|s_t = 0)$
- Hard to do this well
- Hard to model multiple sensors

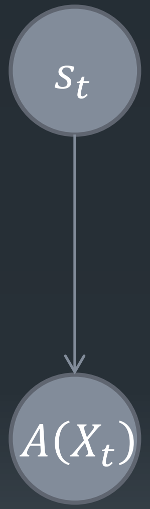


Our Idea: Apply Anomaly Detection Methods

- Suppose we could assign an anomaly score $A(x_t)$ to each observation x_t
  - Scores near 0 are “normal”
  - Scores > 0.5 are “anomalous”
- Learn a probabilistic model of the anomaly scores instead of the raw signals

$$P(A(x_t)|s_t)$$

Basic Configuration



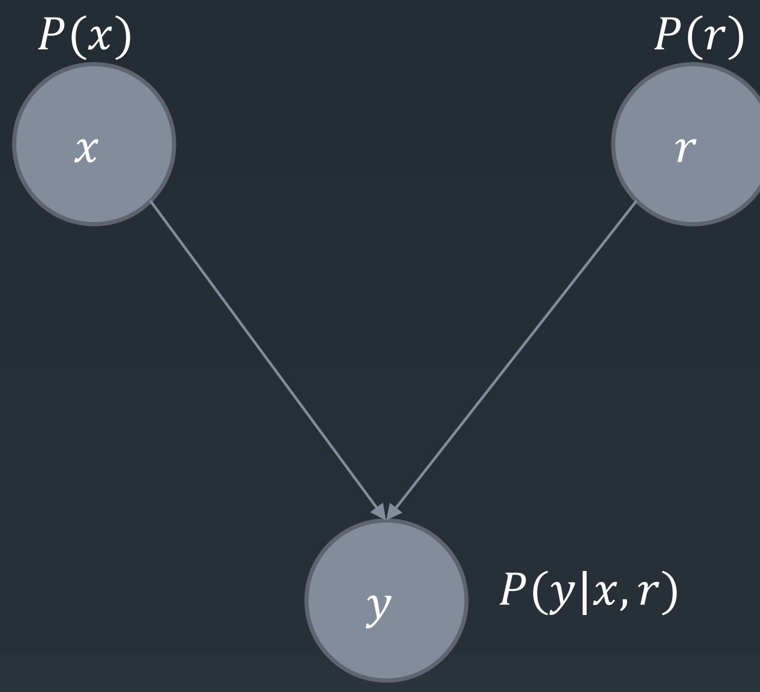
Observe X_t

Compute $A(X_t)$

Compute $\arg\max_{s_t} P(s_t)P(A(X_t)|s_t)$

Probabilistic Graphical Models

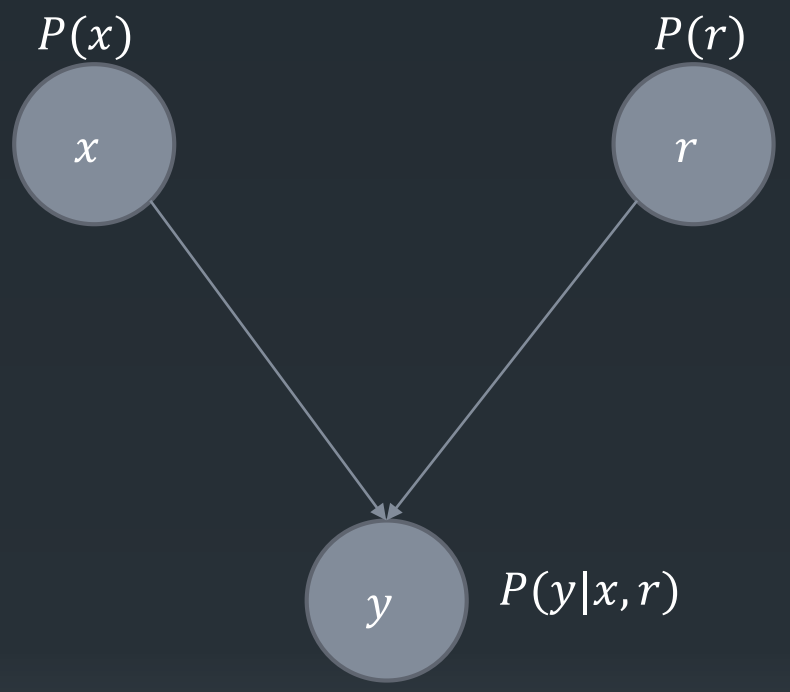
- Graph
  - Each node is a random variable
  - Each edge denotes a probabilistic dependence
  - If a node x has no incoming edges, then its distribution is $P(x)$
  - If a node y has incoming edges from x, r, then its distribution is $P(y|x,r)$
- Joint probability distribution is the product of the distributions in each node



$$P(r,x,y) = P(x)P(r)P(y|x,r) \;\; \forall x,y,r$$

Queries

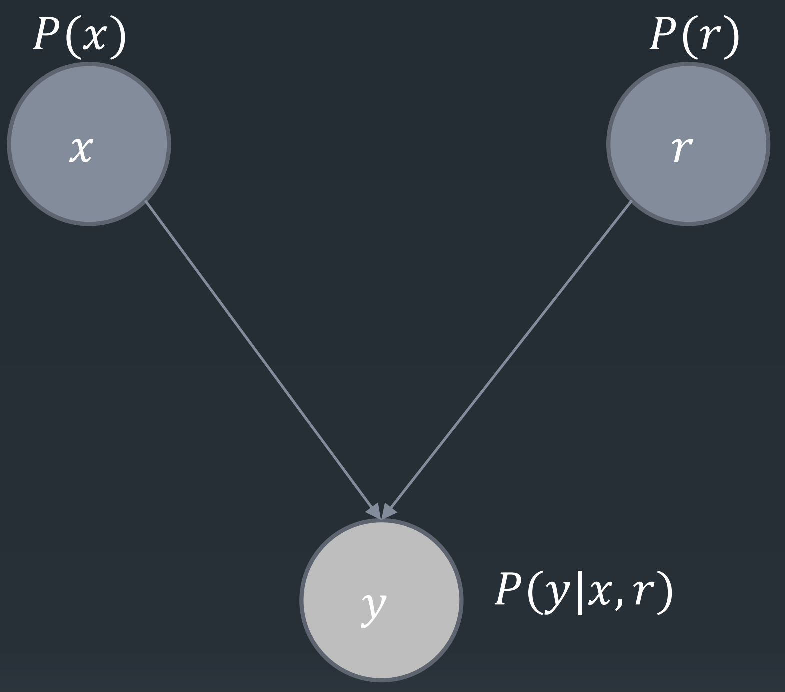
- Observe some variables
- Compute the probability of one or more remaining variables
  - $P(x|y) = \frac{P(x,y)}{P(y)}$
- Inference
  - $P(x,y) = \sum_r P(r)P(x)P(y|x,r)$
  - $P(y) = \sum_r \sum_x P(r)P(x)P(y|x,r)$
  - $P(y|x) = \frac{\sum_r P(r)P(x)P(y|x,r)}{\sum_r \sum_x P(r)P(x)P(y|x,r)}$
  - Simplify algebraically



$$P(r,x,y) = P(x)P(r)P(y|x,r)\ \forall x,y,r$$

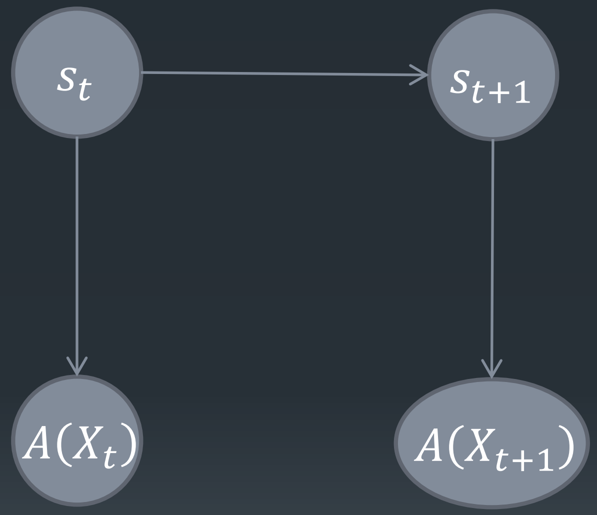
MAP Query

- MAP query
  - $x^* = \arg\max_x P(x|y=0)$
  - Shaded nodes are “observed”



$$P(r,x,y) = P(x)P(r)P(y|x,r) \ \forall x,y,r$$

Cool Things We Can Do: Model Persistence of Sensor State



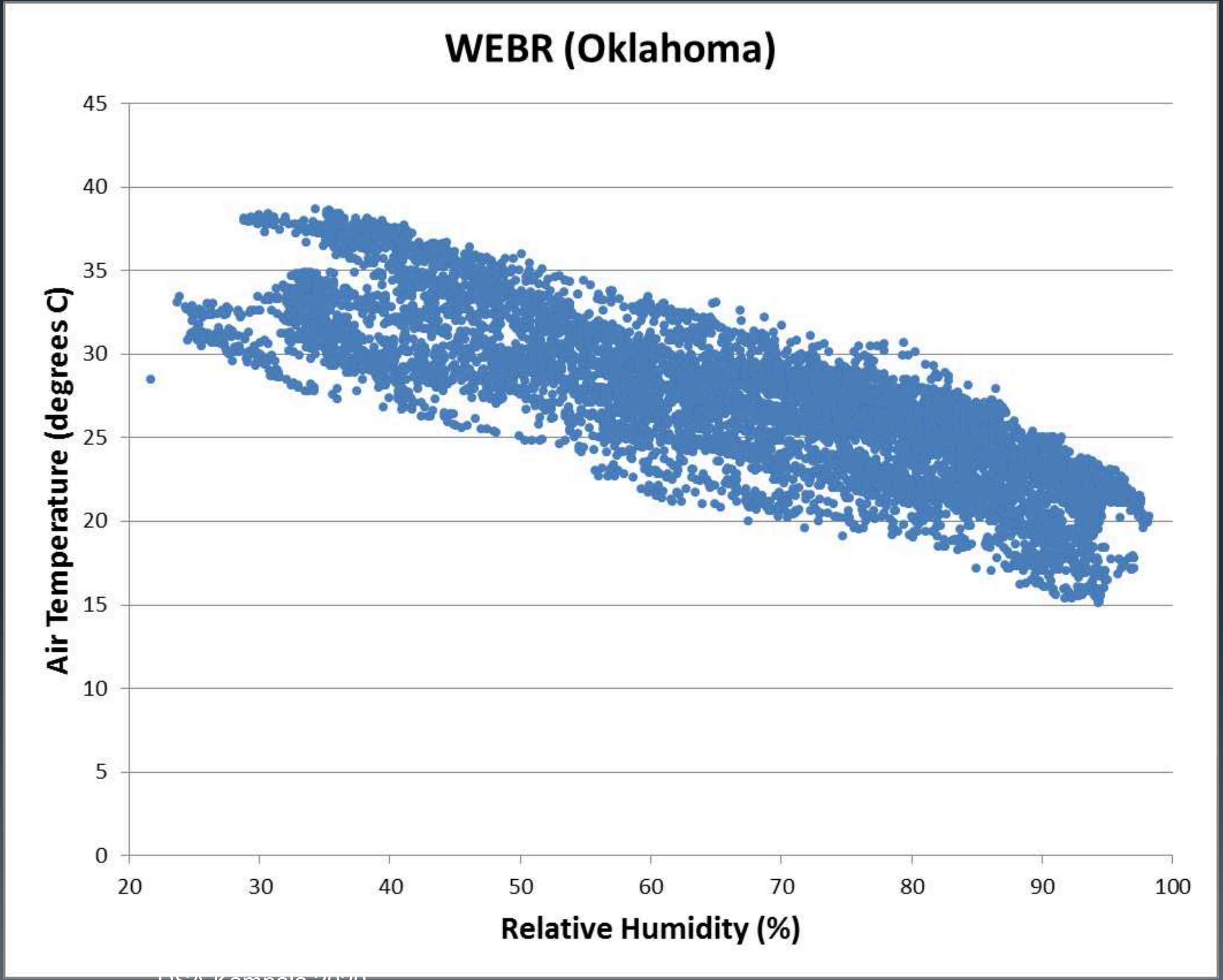
$P(s_{t+1}|s_t)$ encodes persistence of sensor state

- Sensors that are working usually continue working
- Sensors that are broken usually stay broken (until cleaned/repaired)

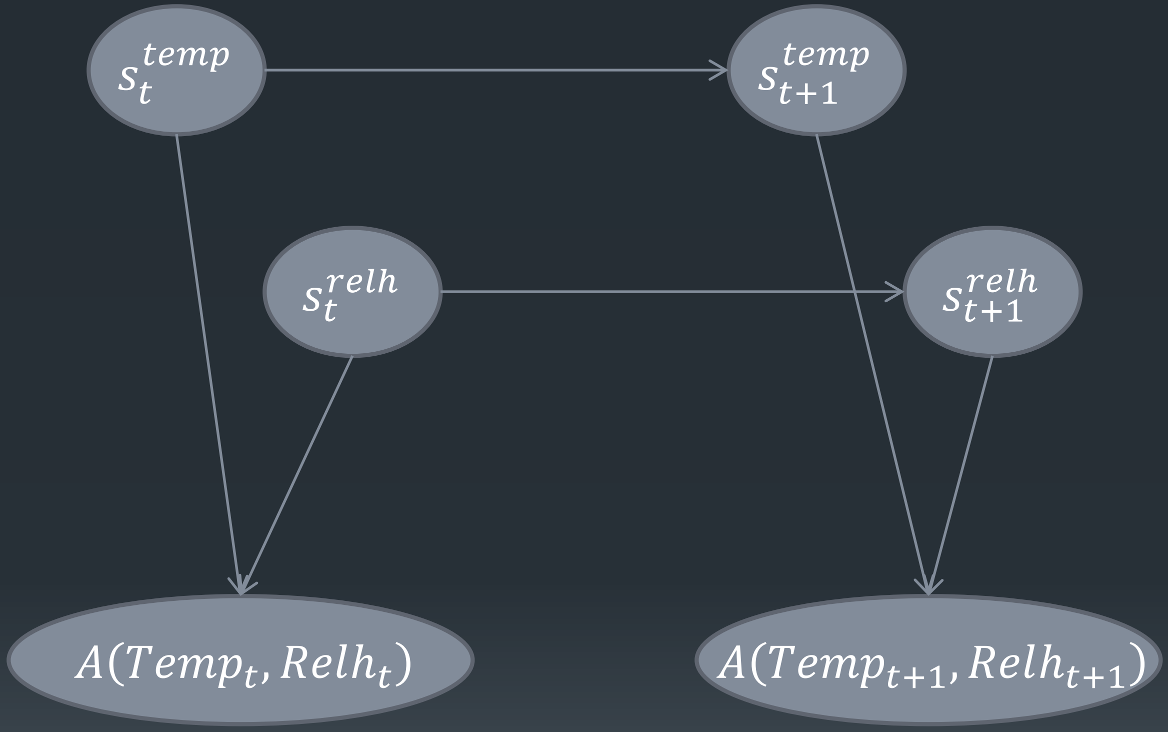
Cool Things We Can Do #2: Model the Joint Distribution of Sensors

Example: Temperature and Relative Humidity are strongly (negatively) correlated

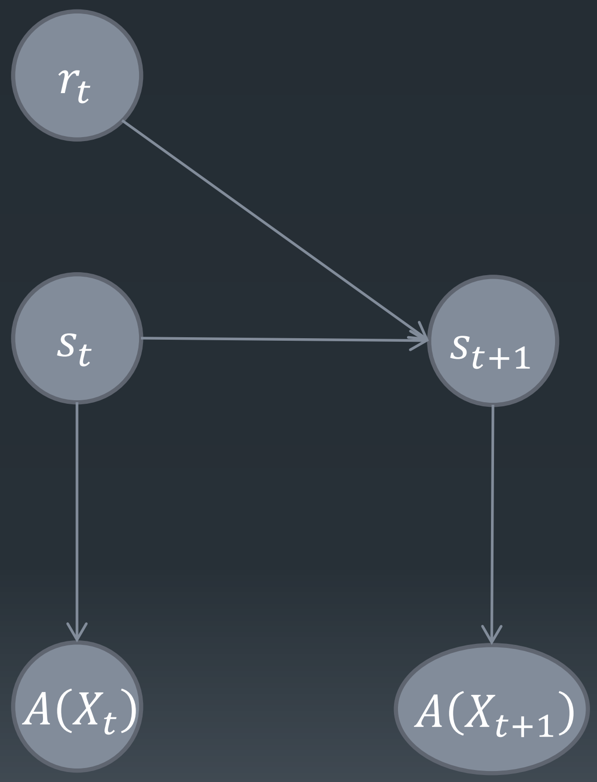
July 2009



Joint Anomaly Detection

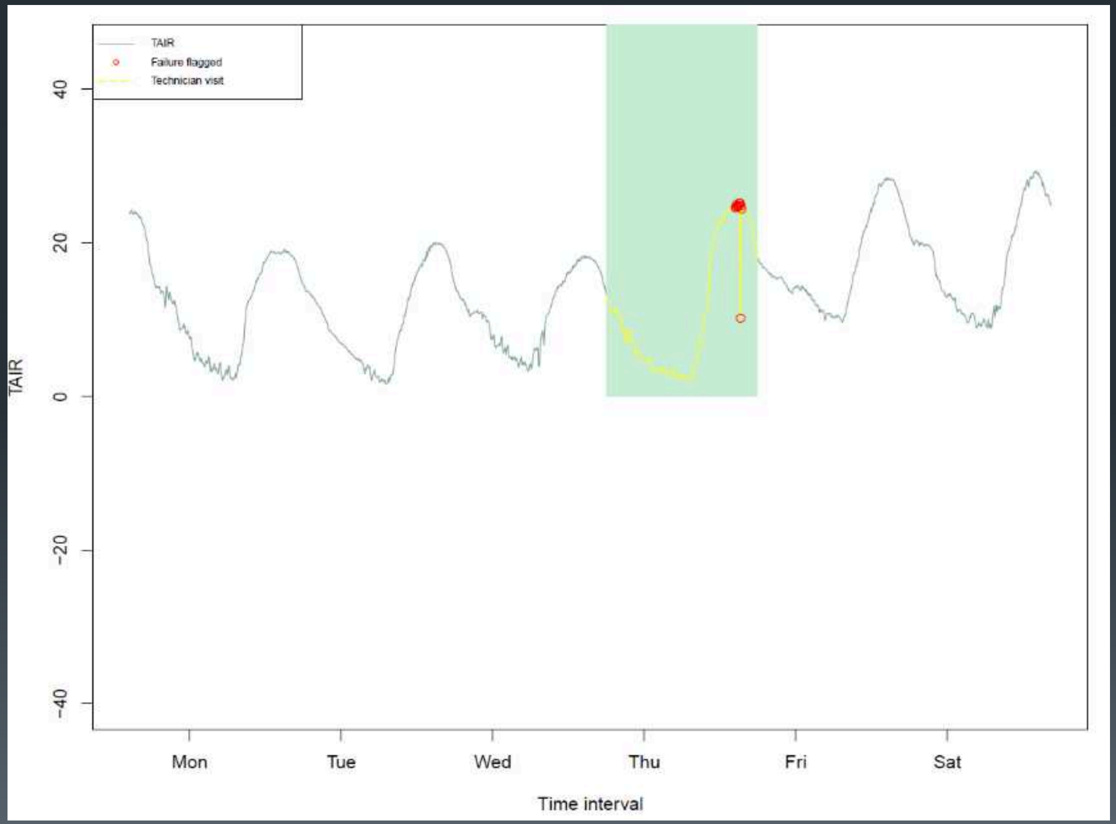


Cool Things We Can Do #3: Incorporate Technician Visits



Let $r(t) = 1$ if technician visited station at time t

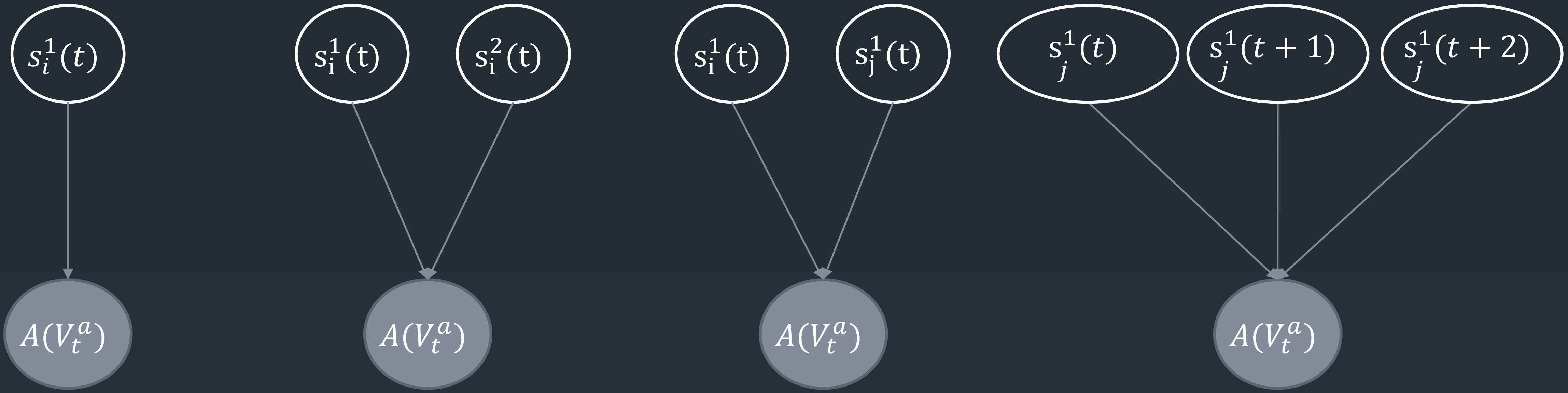
Technician can repair – or break – sensors



SENSOR-DX: Multiple View Approach

- Define many “views” of the data
- Compute anomaly scores in each view
- Perform probabilistic inference to determine the most likely state of each sensor at each time step

Four View Types



Single variable & a single station

Single variable across multiple stations

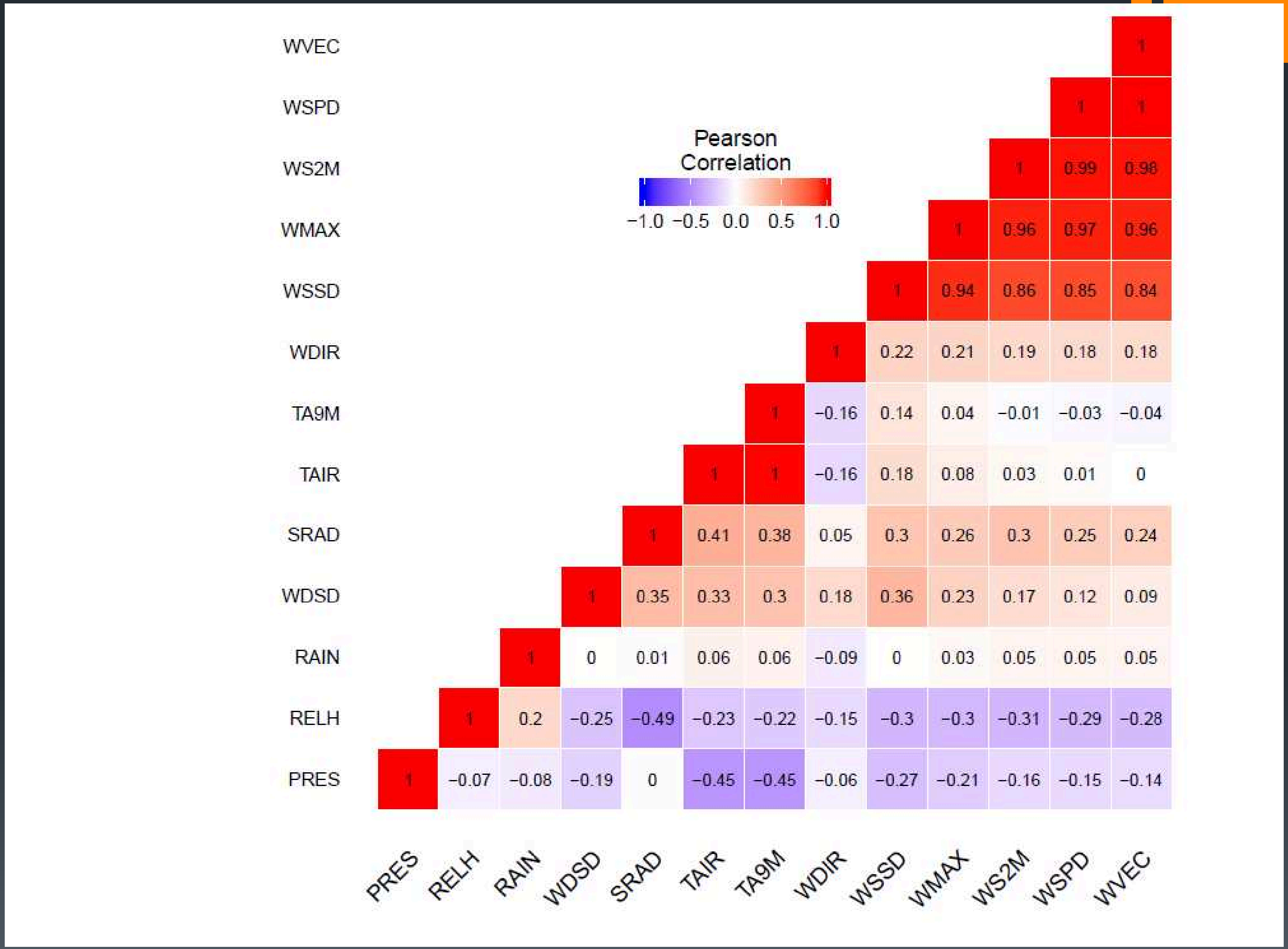
Multiple variables over single station

Single variable over multiple time points

Designing good views on a single weather station

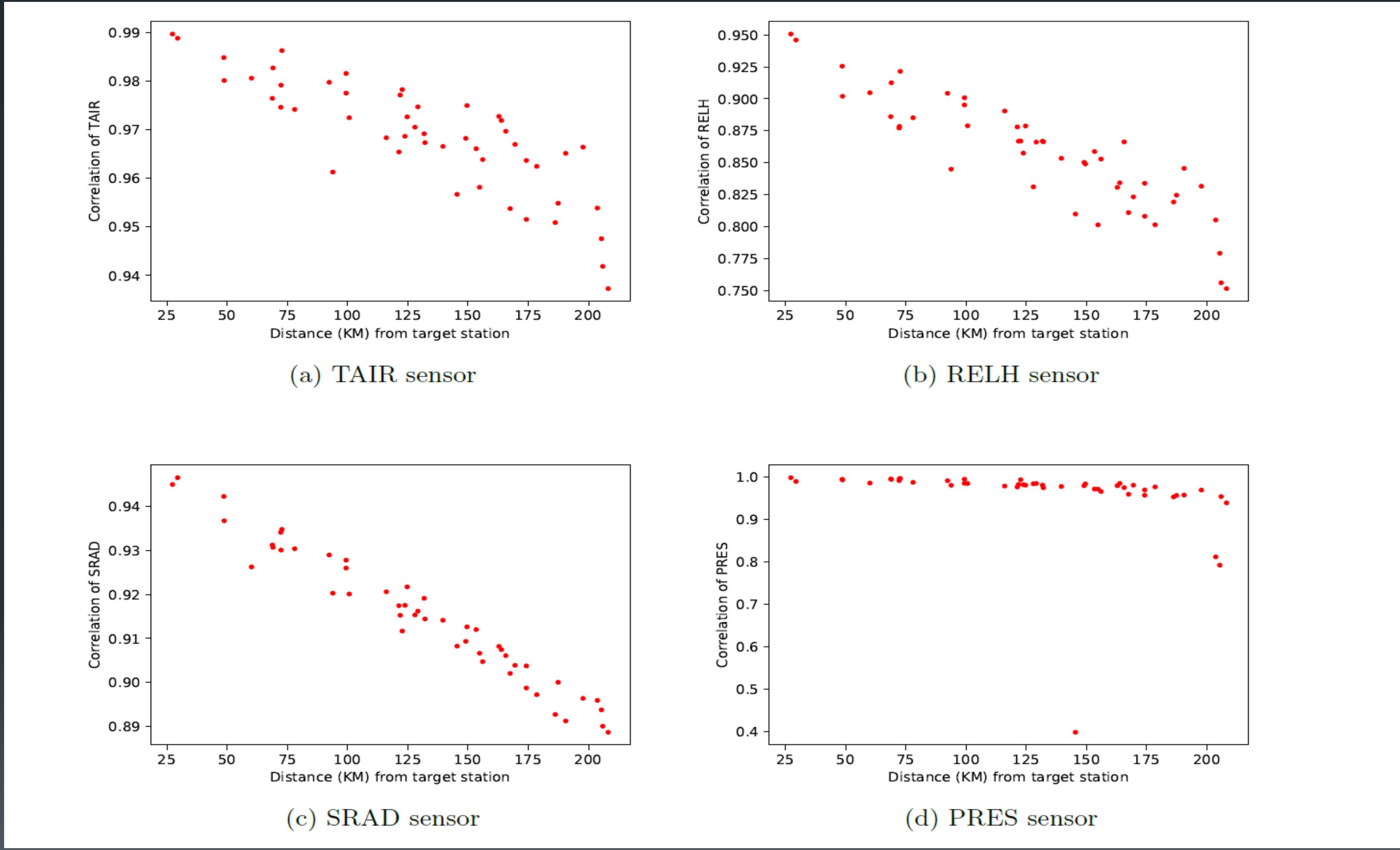
- TAIR: Air temperature
- RELH: Relative humidity
- SRAD: Solar radiation
- PRES: Pressure

- WVEC: Wind Speed (vector average)
- WSPD: Wind Speed
- WS2M: Wind Speed @ 2m
- WMAX: Max wind speed
- WSSD: Stdev wind speed
- WDIR: Wind Direction
- TA9M: Air temperature @9m
- WDSD: Stdev wind direction



Sensor variable correlations

Designing good views across multiple weather stations



Correlation of sensor readings across space

Joint Probability Distribution

Consider a single station at time t

Let i index the sensors at the station

Let j index the views and $v^j(t)$ be the view tuples involving time t

$$P(S(t)|A(v), r(t))$$
$$= \prod_i \underbrace{P(s_t^i|s_{t-1}^i, r_t)P(s_{t-1}^i)}_{\substack{\text{Spontaneous state changes} \\ \text{State changes caused by repair visits}}} \prod_j \underbrace{P\left(A\left(v^j(t)\right)\middle|parents(v^j(t))\right)}_{\substack{\text{Extent to which the sensor states} \\ \text{explain the observed anomaly scores}}}$$

Anomaly Detection

- Collect data for 2019
  - Divide the year into blocks of 20 days
    - Jan 1 → Jan 20; Jan 21→ Feb 10; Feb 11→ Mar 2; etc.
  - Compute features from the observations in each hour
    - mean, variance, max, min, median
  - Fit an Isolation Forest to the data points for each view in each block
- Scoring 2020
  - Use the isolation forest from the corresponding 20-day period

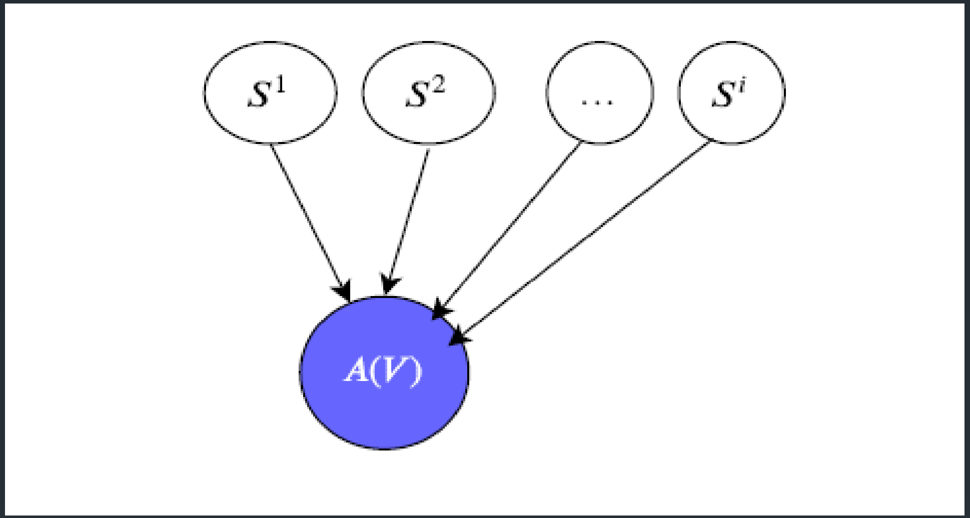
Fitting the Conditional Probability Model

- $P(A(v)|\, s^1, \dots, s^N)$
  - There are 2^N configurations!

- Reducing the number of parent configurations
  - Let $nbs(s^1, \dots, s^N)$ = "number of broken sensors"
  - Model the anomaly score as a function of the number of broken sensors

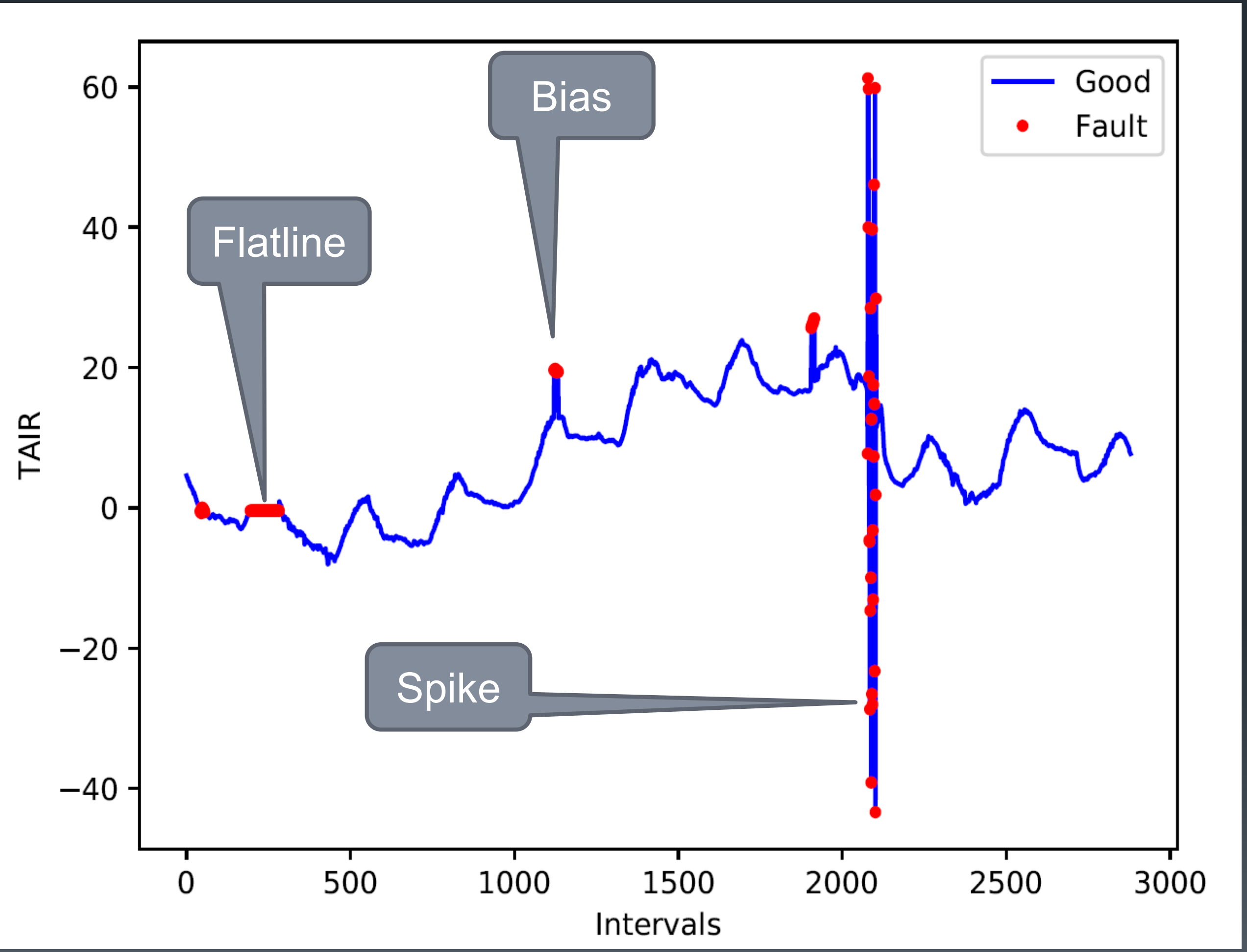
$$P(A(v)|\, s^1, \dots, s^N) \approx P(A(v)|nbs(s^1, \dots, s^N) \; = i)$$

  - Only $N + 1$ configurations!

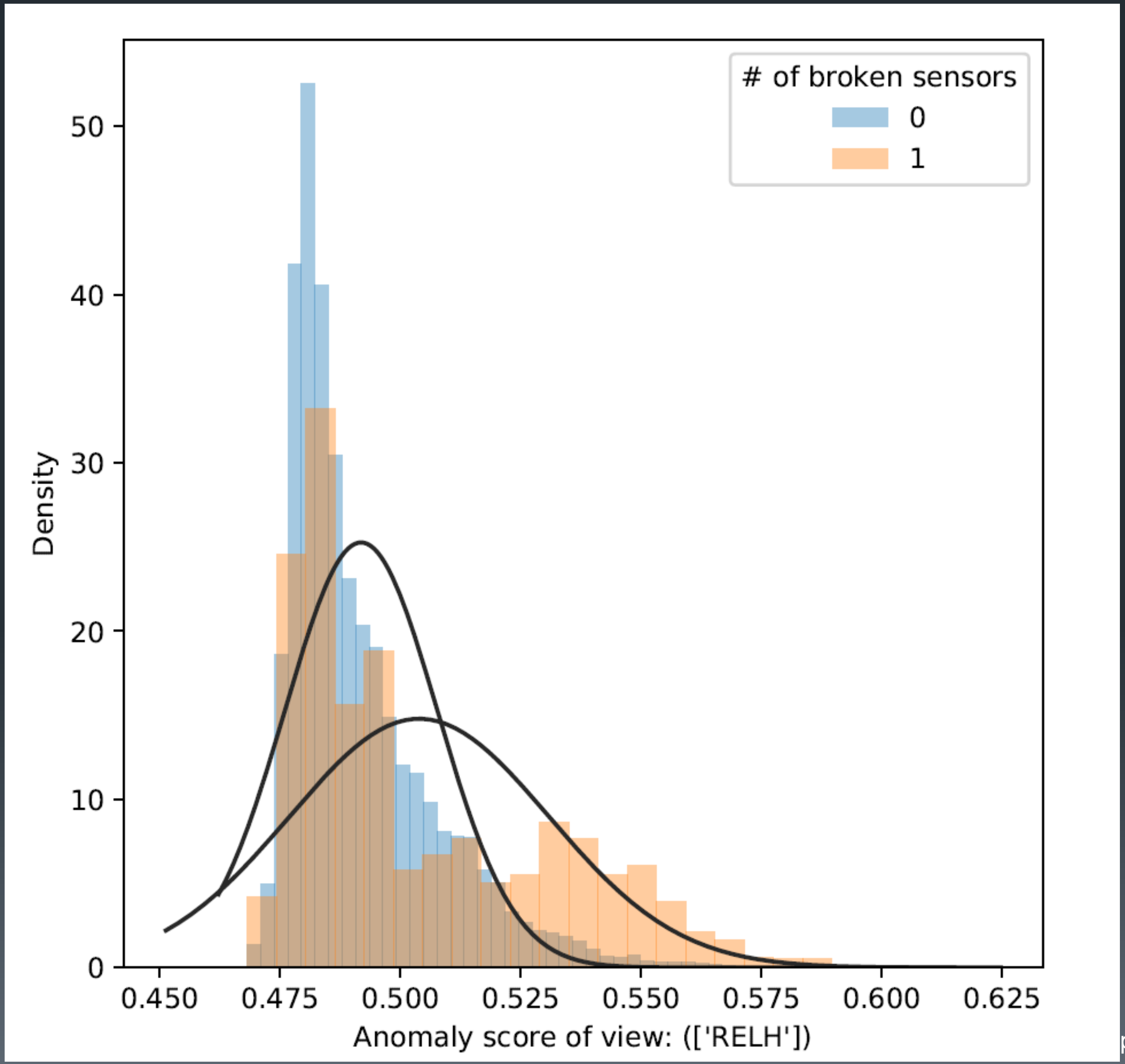


Generating Training Data for Broken Sensors

- To fit $P(A(v)|nbs)$, we need training data for broken sensors
- There is not enough real data
- Engineering solution:
  - Insert simulated faults into the data
  - Compute anomaly scores
  - Fit Gaussian distribution to the scores

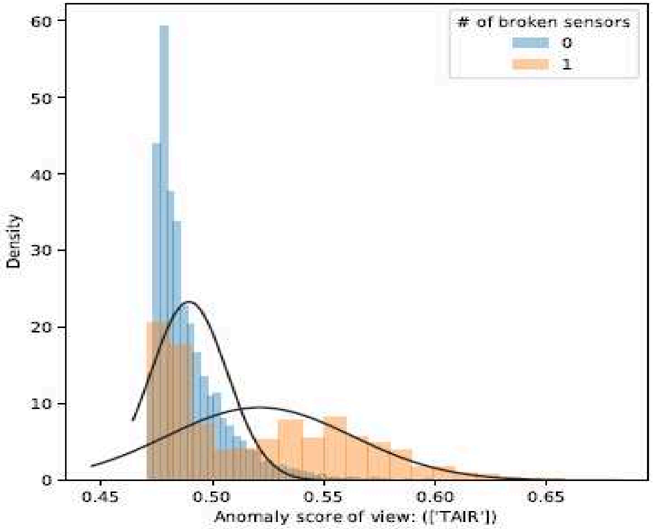


Examples of Fitted $P(A(v)|nbs(s^1,\ldots,s^N))$

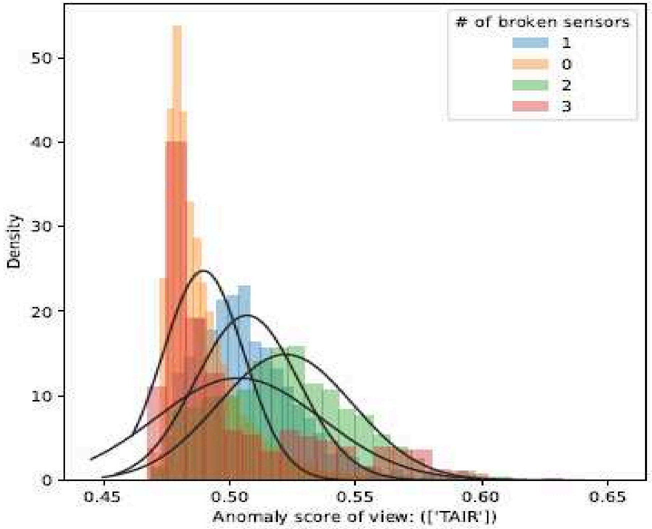


Relative Humidity

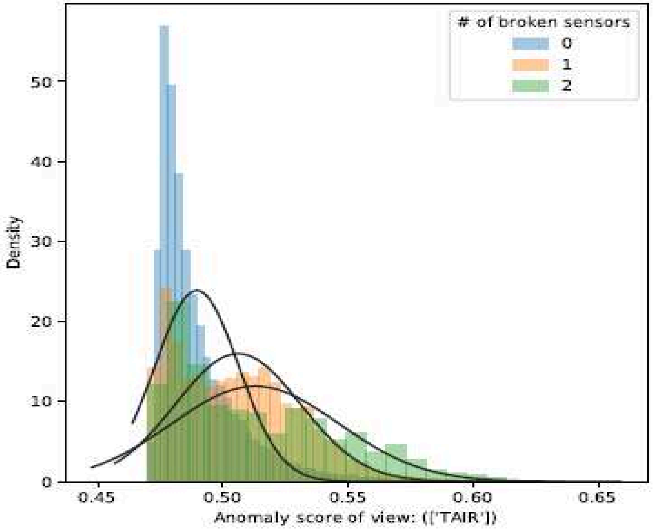
May be able to improve performance by fitting a non-Gaussian distribution



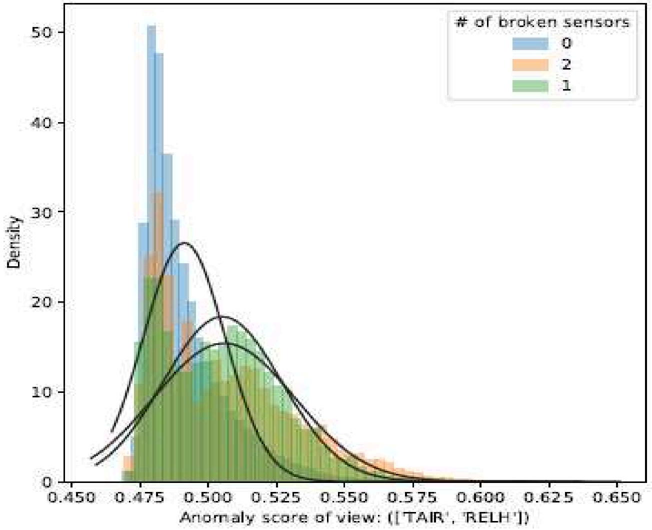
(a) Single sensor and single station view, $|S| = 1$



(b) Single variable over temporal scale view, $|S| = 1$



(c) Multi-station sensor view, $|S| = 2$



(d) Multi-sensor single station view, with $|S| = 2$

Run Time Quality Control

- Assemble incoming data into view tuples
- Compute anomaly score for each view tuple
- Perform probabilistic inference to determine which sensor states best explain the observed anomaly scores:

$$\arg\max_{S} P(S|A(V))$$

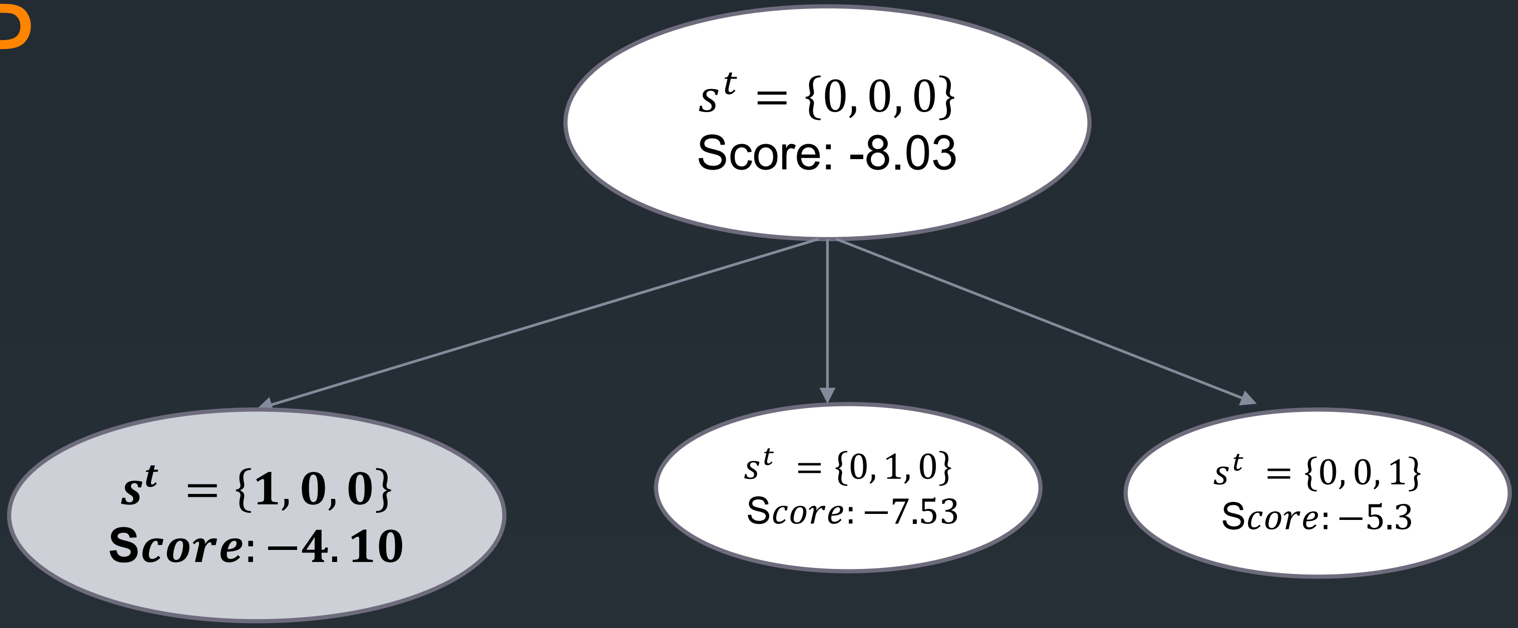
Inferring the Sensor States

- Ideal MAP inference

$$S^* = \arg\max_S P(S_{1:T}^s = S \mid A(V_{1:T}^v))$$

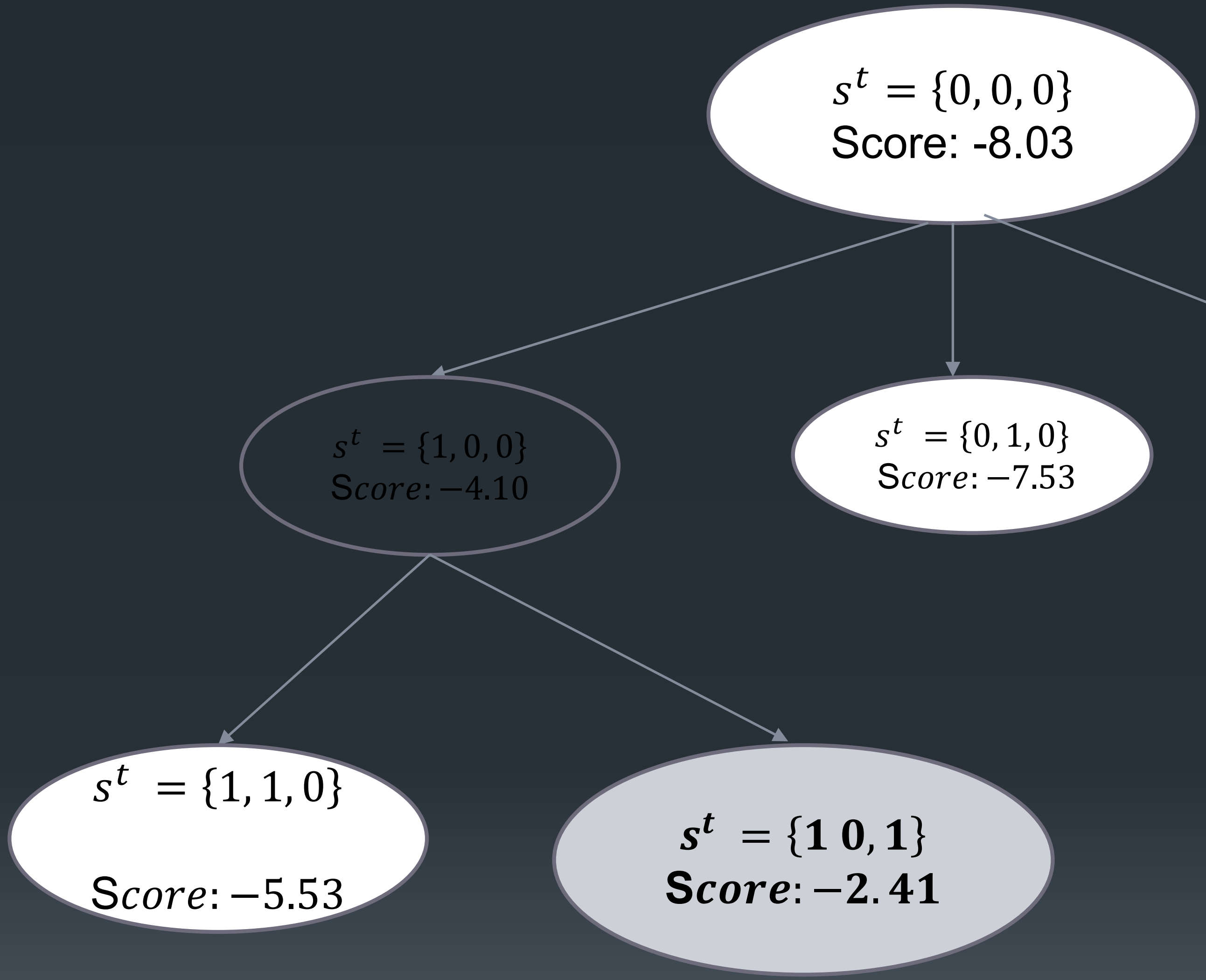
- Exact inference is intractable: N sensors and T timesteps requires scoring 2^{NT} configurations
- To overcome this, we introduce two approximations
  - SearchMAP [Dereszynski 2012] for computing the MAP assignment
  - Filter-and-Commit (FAC) for incremental MAP inference

SearchMAP

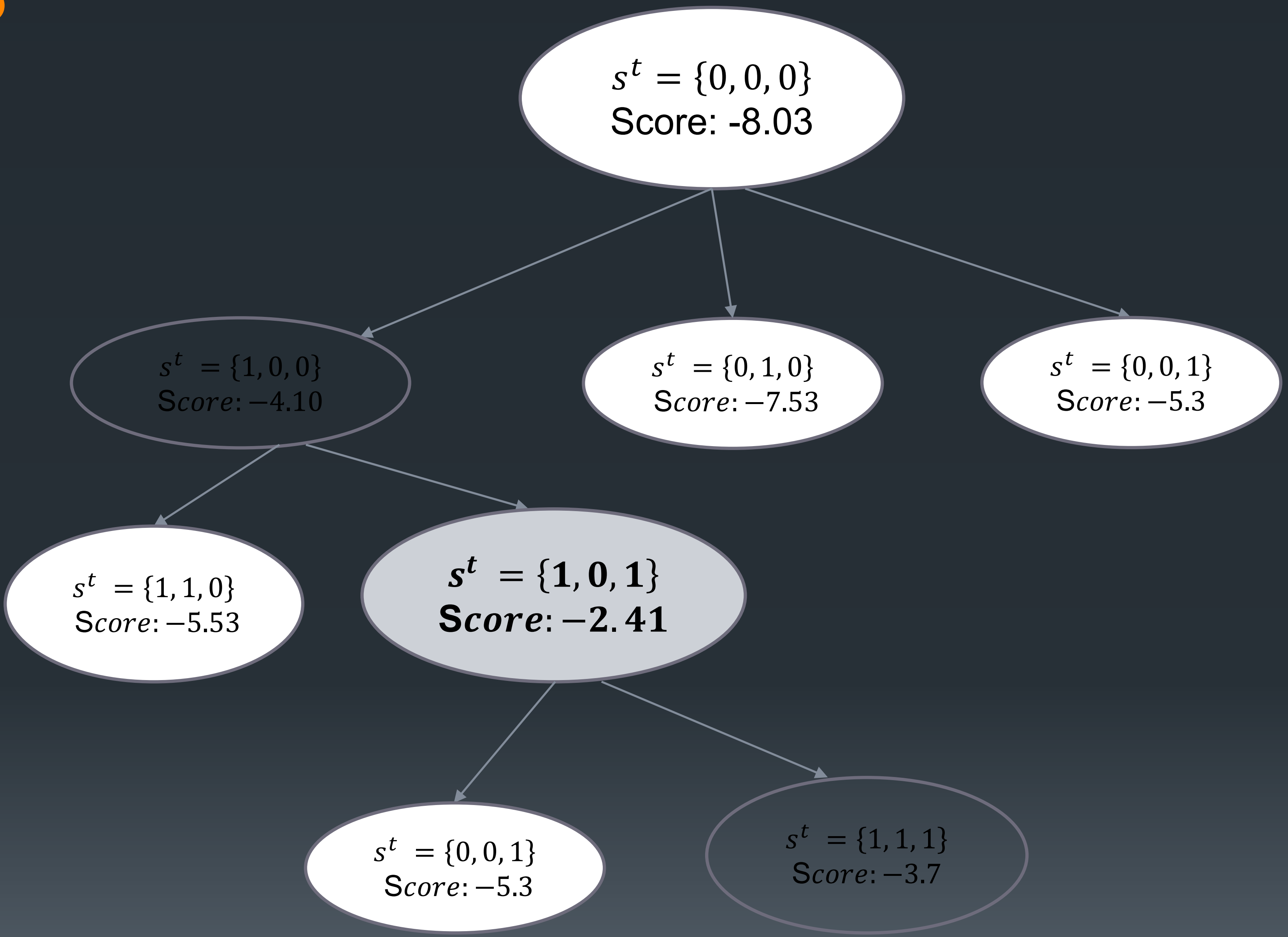


Greedy algorithm
Flip sensor states until no single flip increases the likelihood

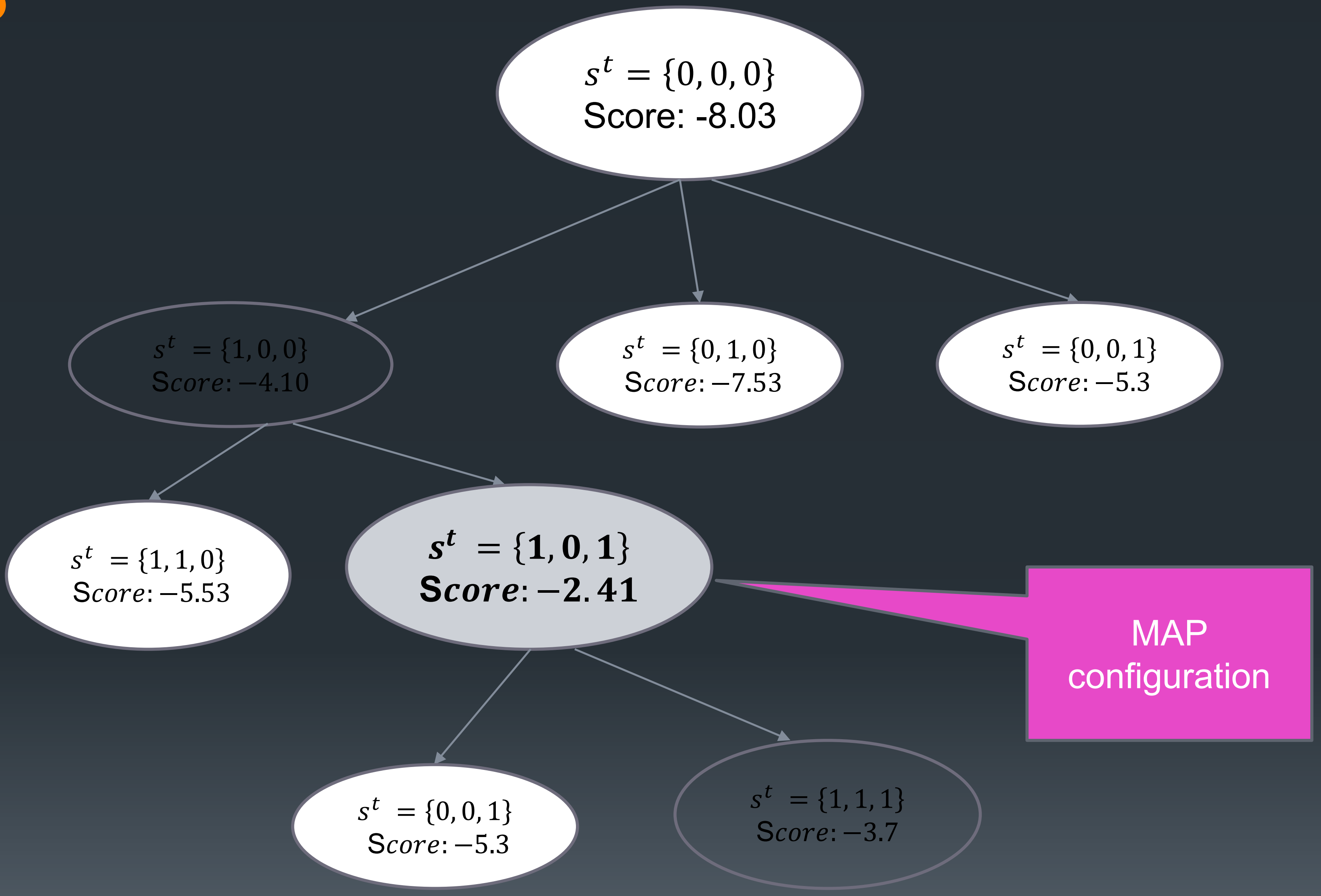
SearchMAP


$s^t = \{0, 0, 0\}$
Score: -8.03
$s^t = \{1, 0, 0\}$
$Score: -4.10$
$s^t = \{0, 1, 0\}$
$Score: -7.53$
$s^t = \{0, 0, 1\}$
$Score: -5.3$
$s^t = \{1, 1, 0\}$
$Score: -5.53$
$\boldsymbol{s^t = \{1\ 0, 1\}}$
$\boldsymbol{Score: -2.41}$

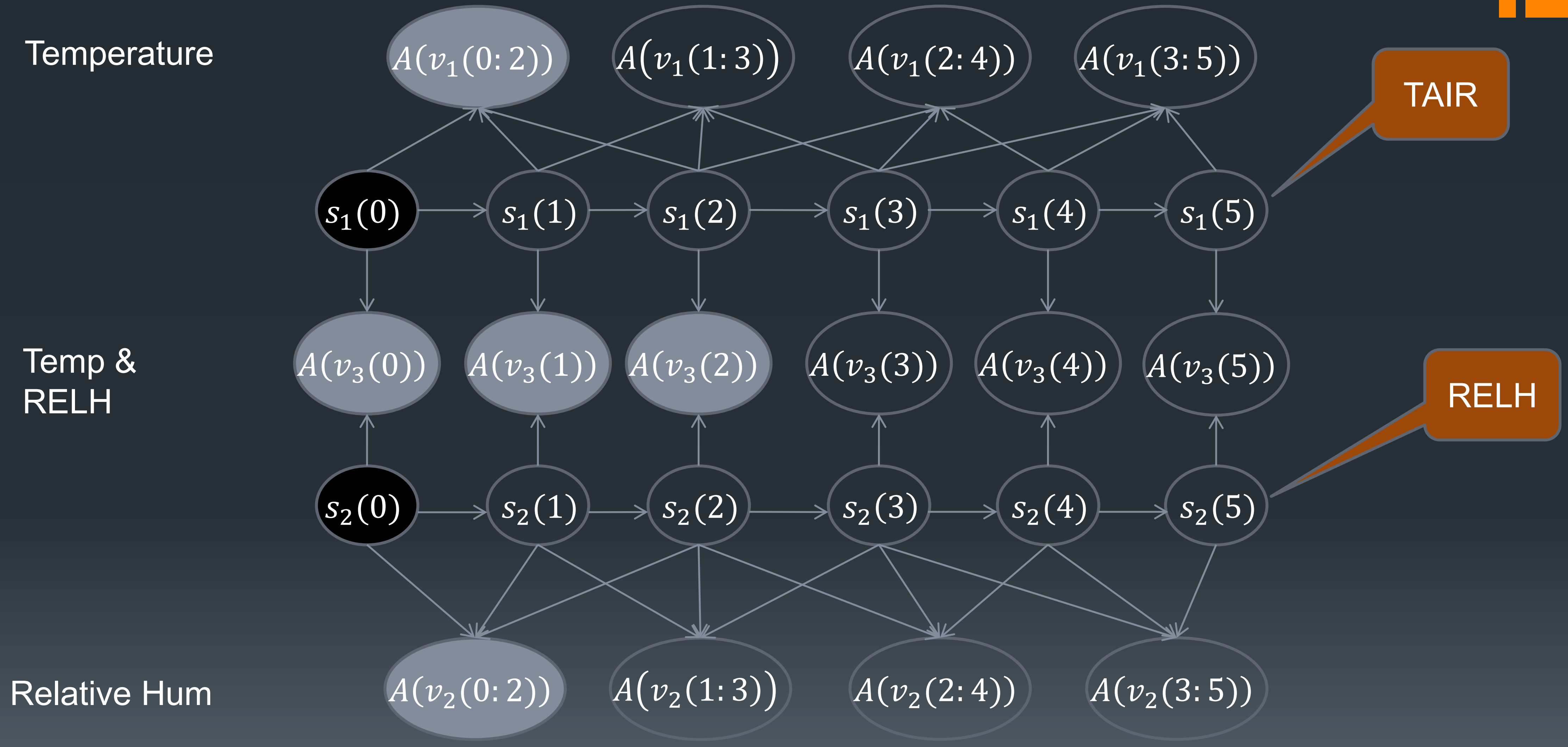
SearchMAP


$s^t = \{0, 0, 0\}$
Score: -8.03
$s^t = \{1, 0, 0\}$
$Score: -4.10$
$s^t = \{0, 1, 0\}$
$Score: -7.53$
$s^t = \{0, 0, 1\}$
$Score: -5.3$
$s^t = \{1, 1, 0\}$
$Score: -5.53$
$\boldsymbol{s^t = \{1, 0, 1\}}$
$\boldsymbol{Score: -2.41}$
$s^t = \{0, 0, 1\}$
$Score: -5.3$
$s^t = \{1, 1, 1\}$
$Score: -3.7$

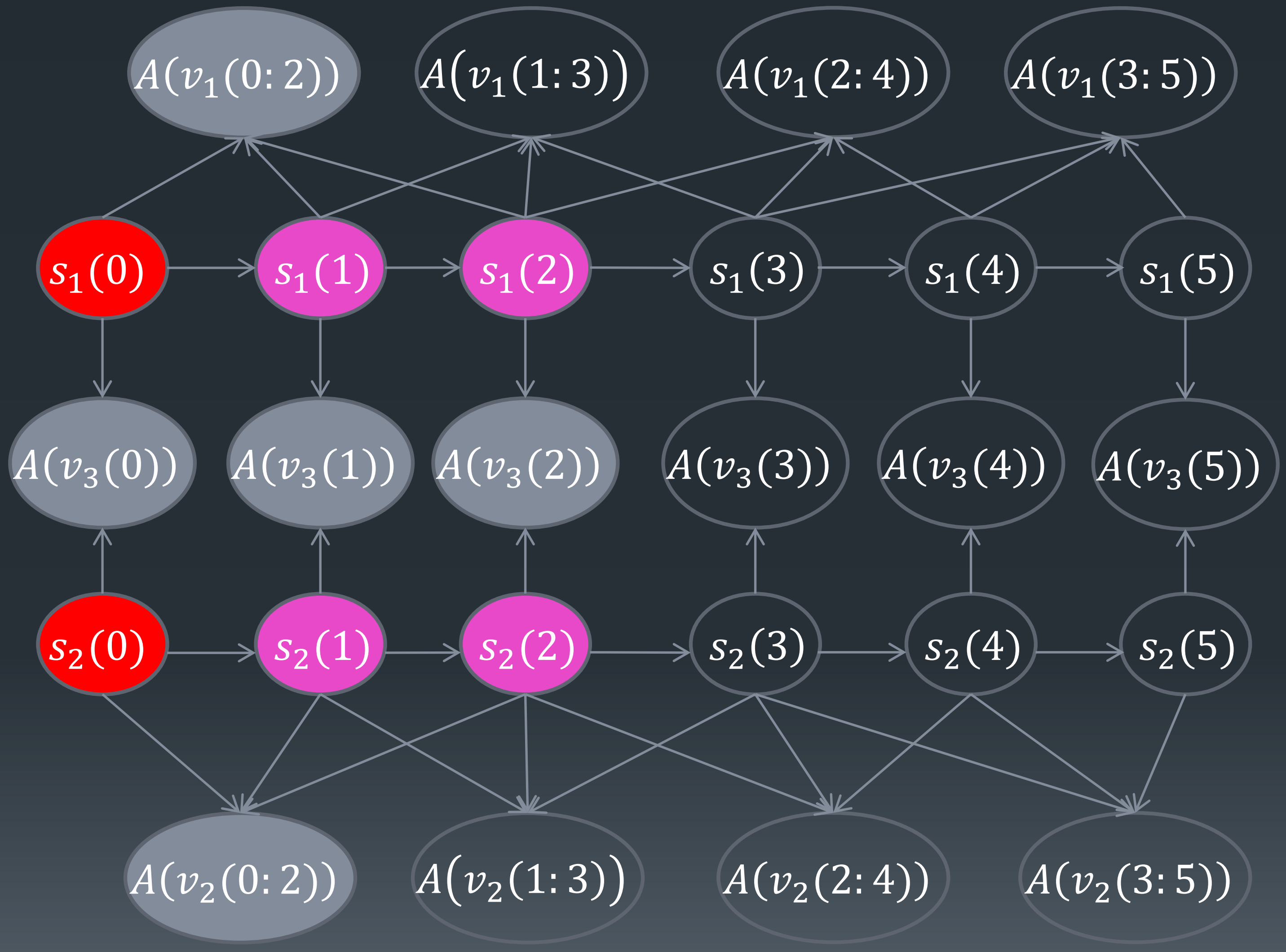
SearchMAP


$s^t = \{0,0,0\}$
Score: -8.03
$s^t = \{1,0,0\}$
$Score: -4.10$
$s^t = \{0,1,0\}$
$Score: -7.53$
$s^t = \{0,0,1\}$
$Score: -5.3$
$s^t = \{1,1,0\}$
$Score: -5.53$
$\boldsymbol{s^t = \{1,0,1\}}$
$\boldsymbol{Score: -2.41}$
MAP configuration
$s^t = \{0,0,1\}$
$Score: -5.3$
$s^t = \{1,1,1\}$
$Score: -3.7$

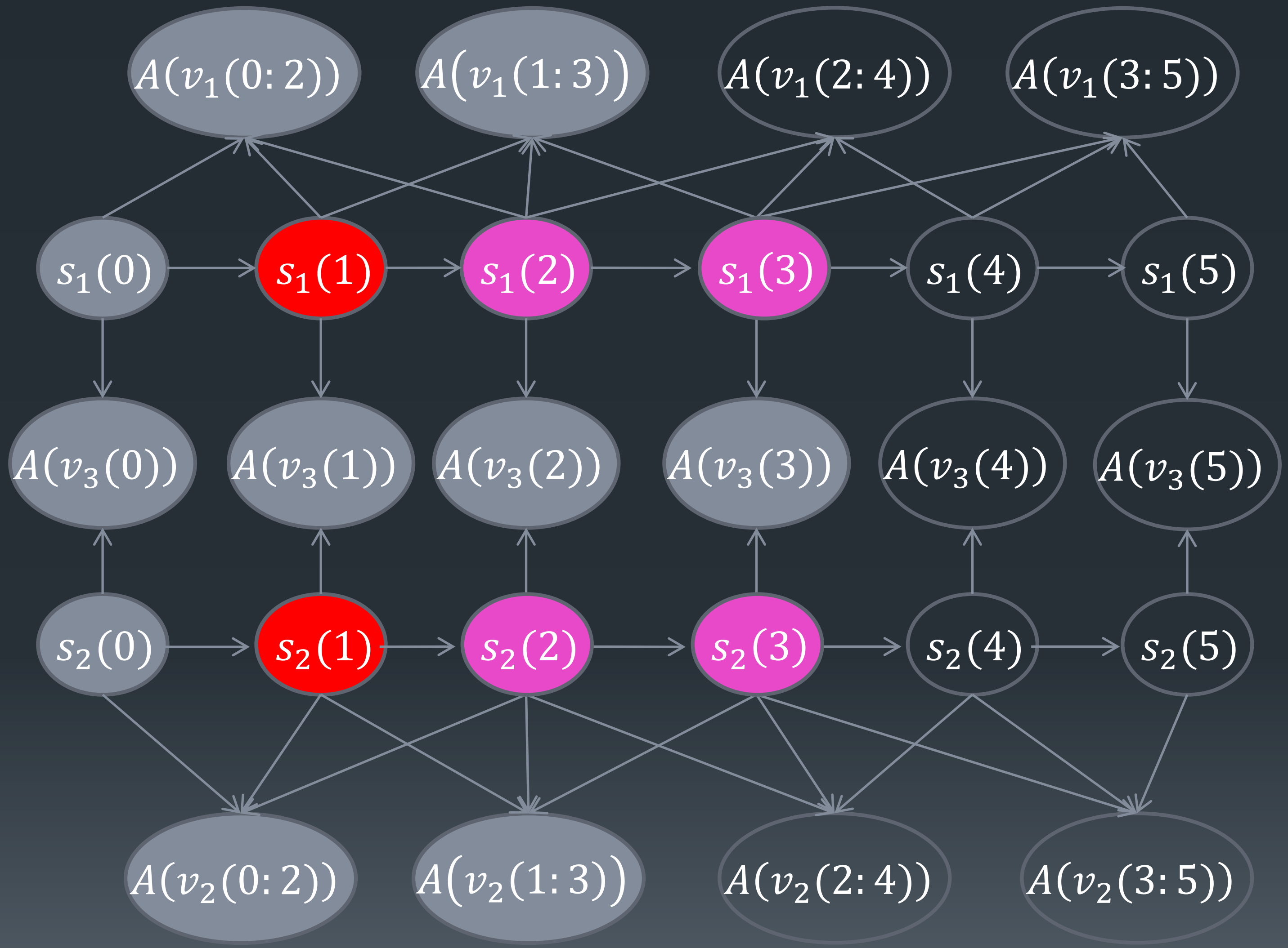
Filter-and-Commit (FAC)



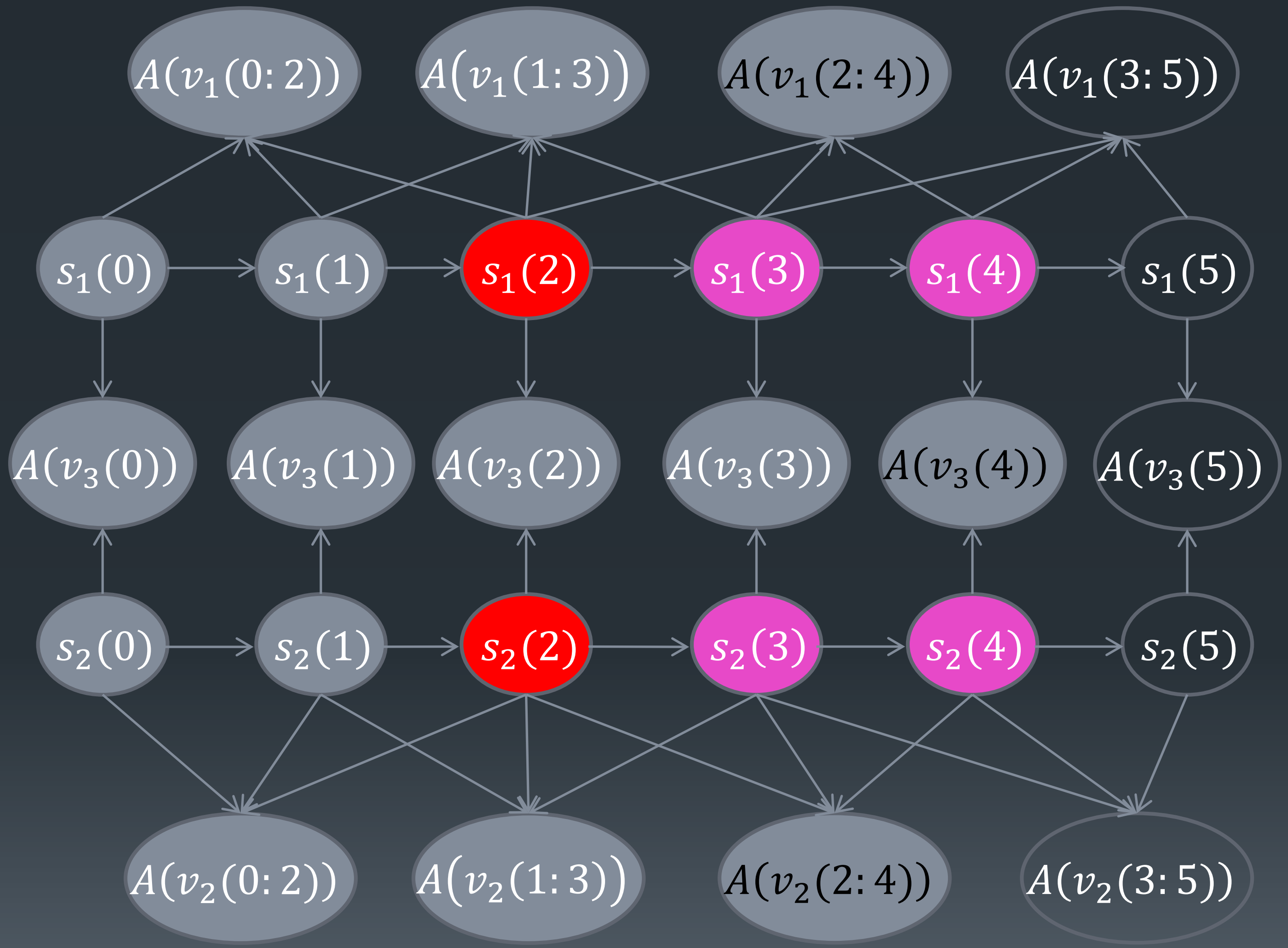
Observation time: 2 Focus time: 0 Commit time: 0



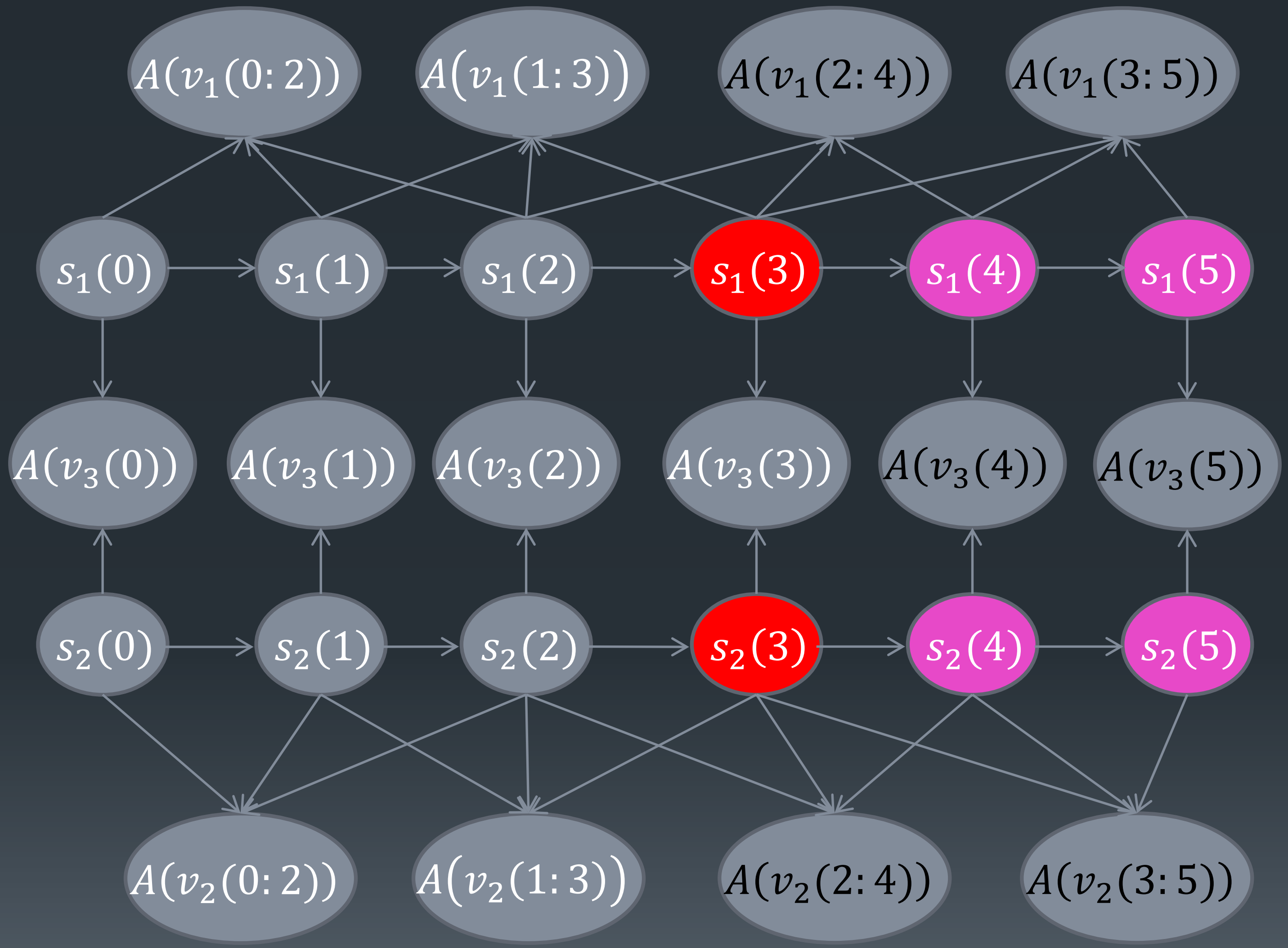
Observation time: 3 Focus time: 1 Commit time: 1



Observation time: 4 Focus Time: 2 Commit time: 2



Observation time: 5 Focus Time: 3 Commit time: 3



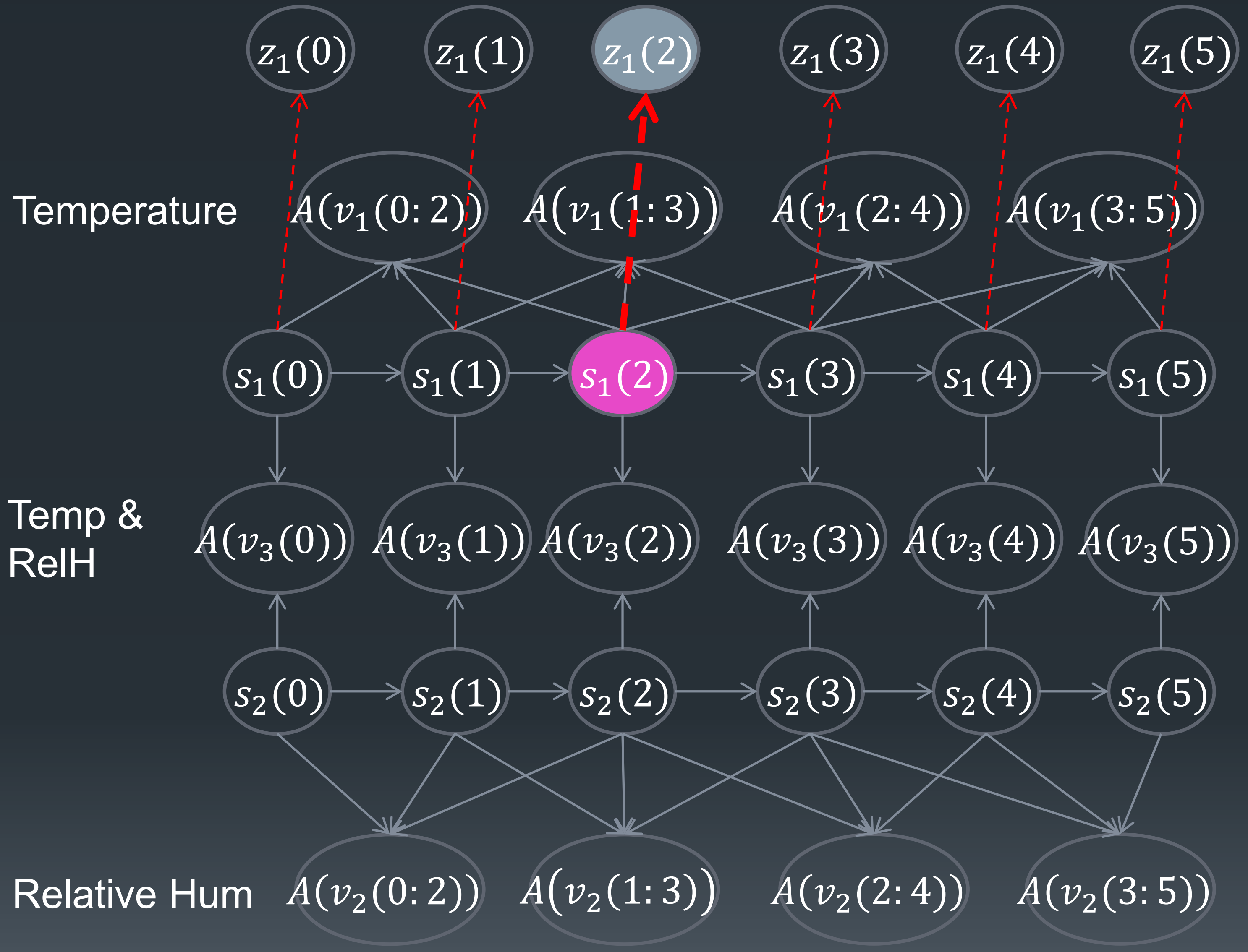
Controlling False Alarms vs. Missed Alarms

Introduce two parameters:

- $P(z_1(t) = 0|s_1 = ok) = \pi_{ok}$
- $P(z_1(t) = 0|s_1 = broken) = \pi_{broken}$

The difference determines the relative penalty/bonus for assigning $s_1 = ok$ vs $s_1 = broken$

Example $s_1(2)$:
$z_1(2) = 0$ is always "observed"



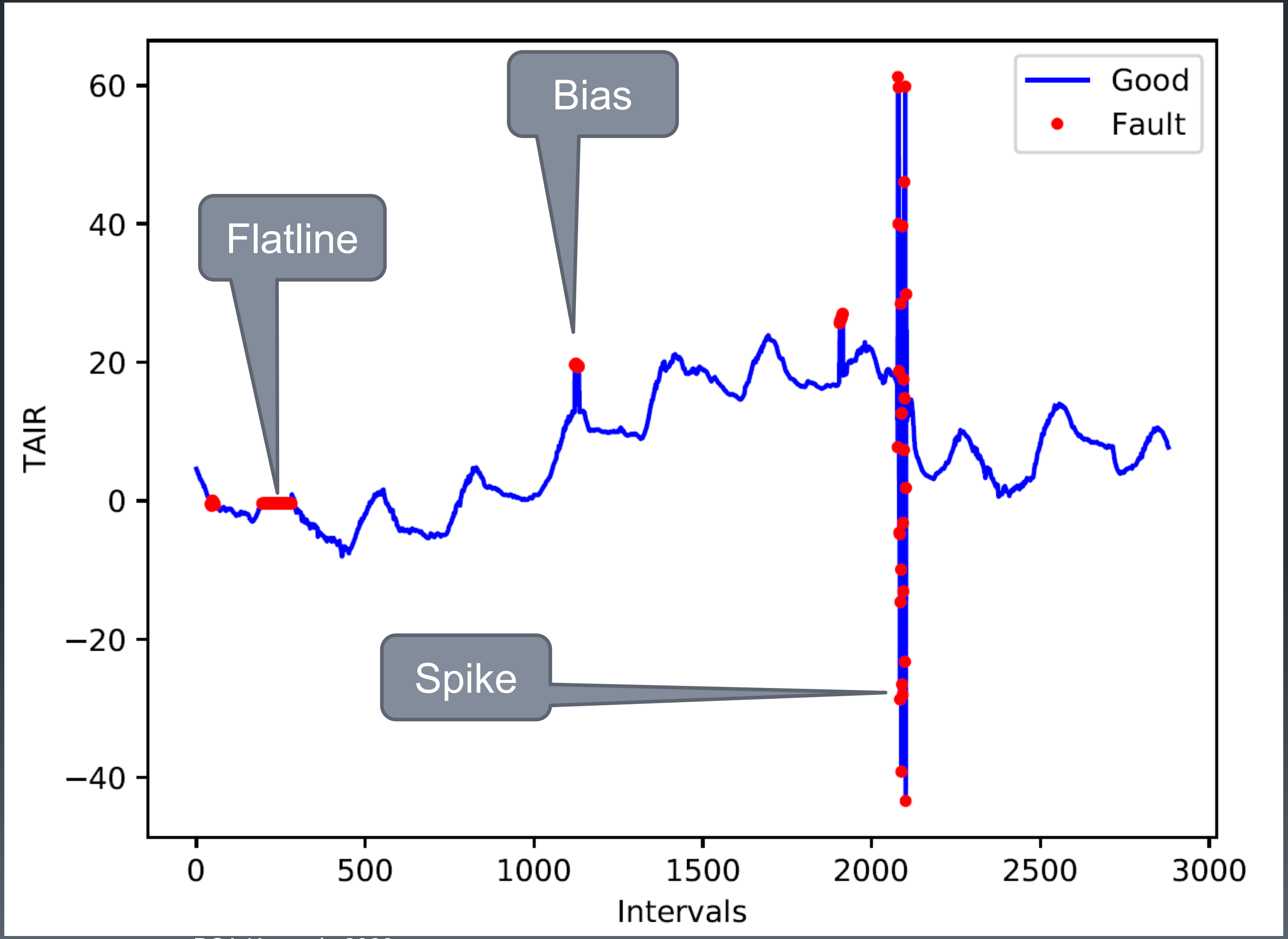
Experimental Evaluation: Experiment Design

- Data: Oklahoma Mesonet
  - 4 stations:
    - OKCE, OKCN, OKCW, NRMN
  - 2 years
  - 5 minute reporting interval
  - Hourly sensor state
  - Sensors:
    - TAIR, RELH, SRAD, PRES
- Baseline:
  - Single sensor view based detection
- Metrics:
  - Precision and recall

View type	State/period	Total #views
Single sensor view	1	16
Same sensor two station view	2	24
Two sensor single station view	2	24
Single sensor three hour view	3	14
Total views per block		80

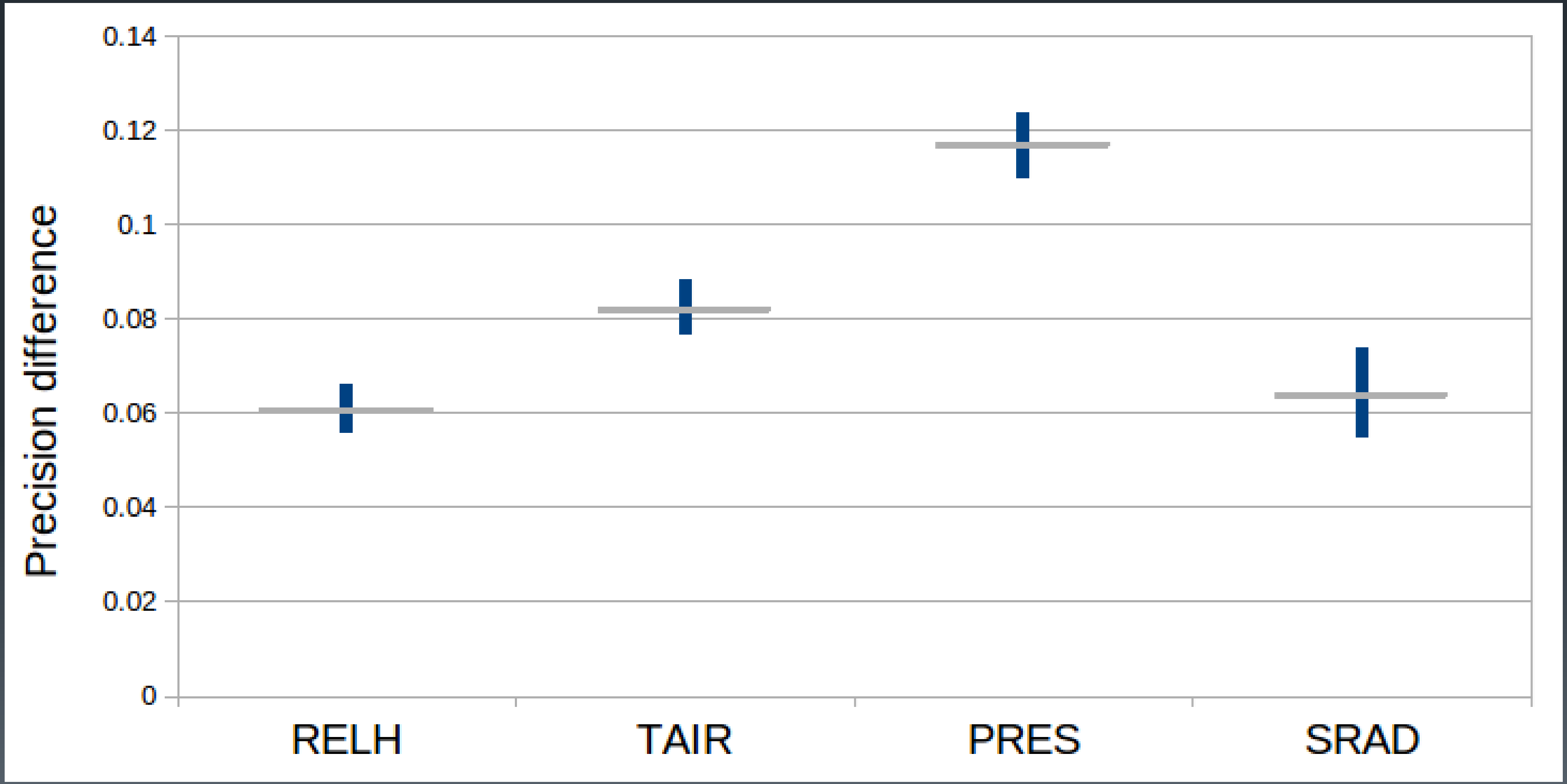
Synthetic Fault Insertion

- Fault types:
  - Flatline
  - Spike
  - Bias
- Fault proportion:
  - $[\frac{1}{2}, \frac{1}{3}, \frac{1}{6}]$



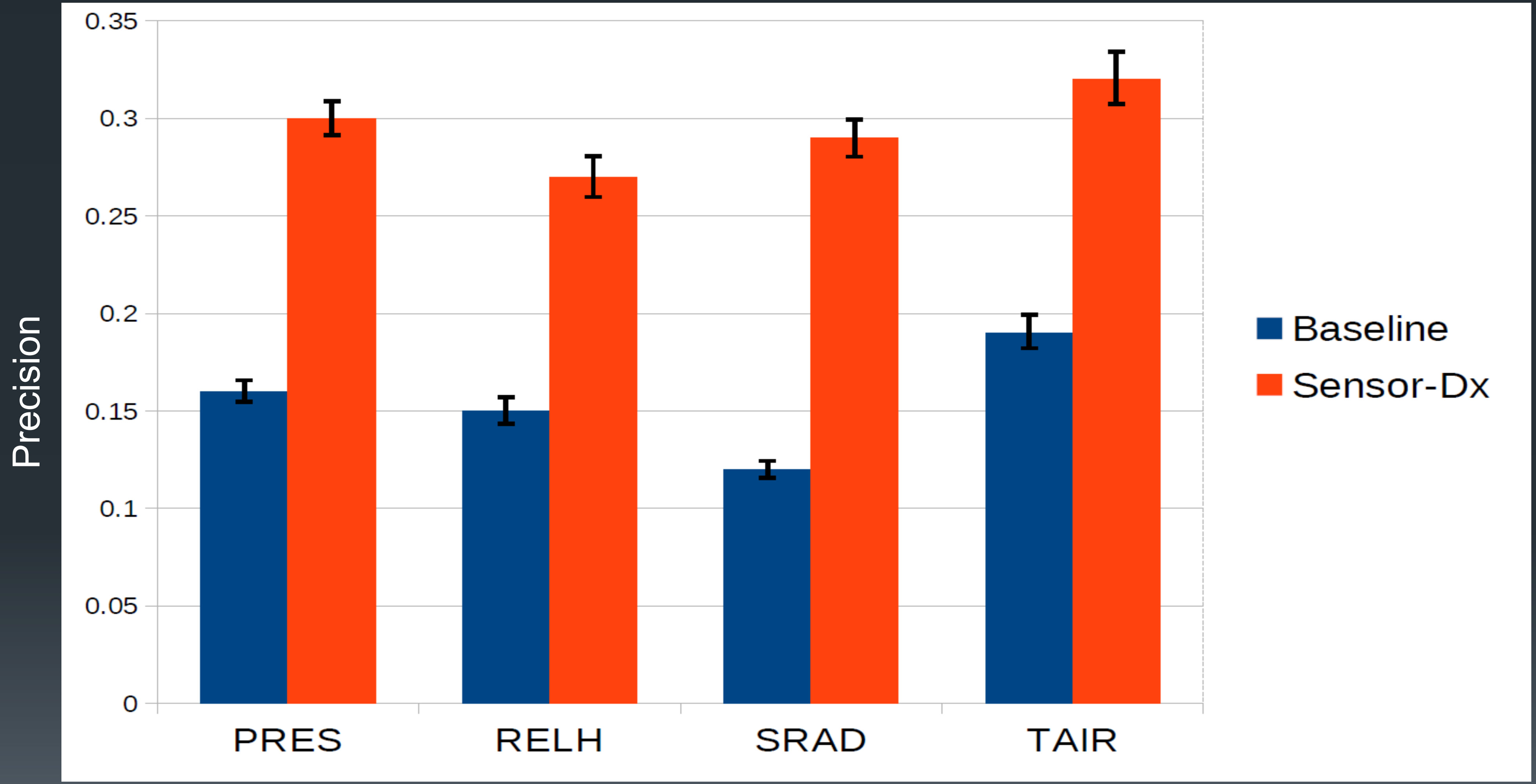
Result: Sensor-DX improves precision

Difference in precision of multi-view method versus single-view baseline



95% two-sided paired differences bootstrap confidence intervals

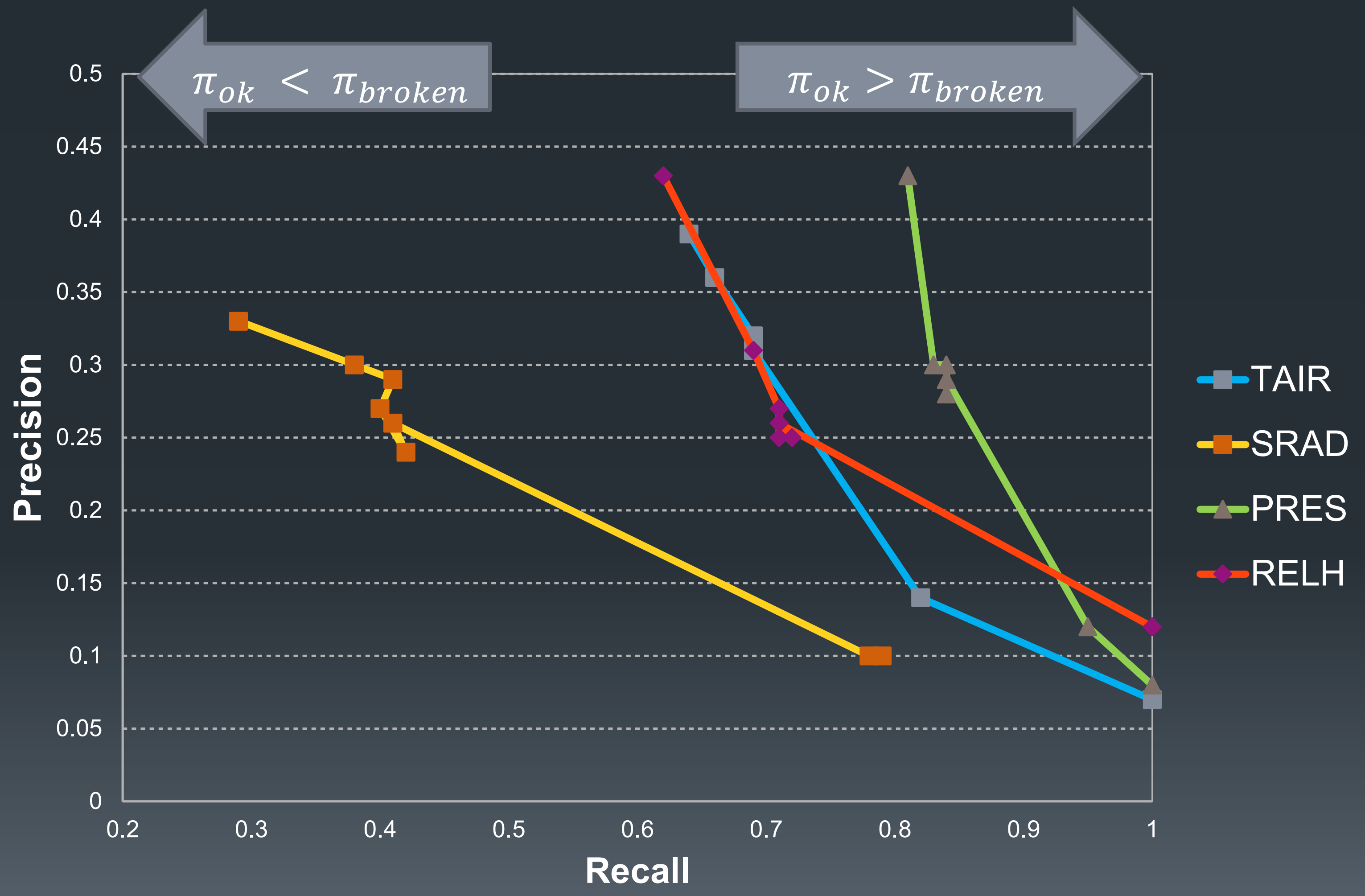
Precision at Matching Recall Level



95% confidence intervals

Sensor-DX improves precision, but the false alarm rate will still be quite high

Precision-recall of π_{ok} & π_{broken} tradeoff



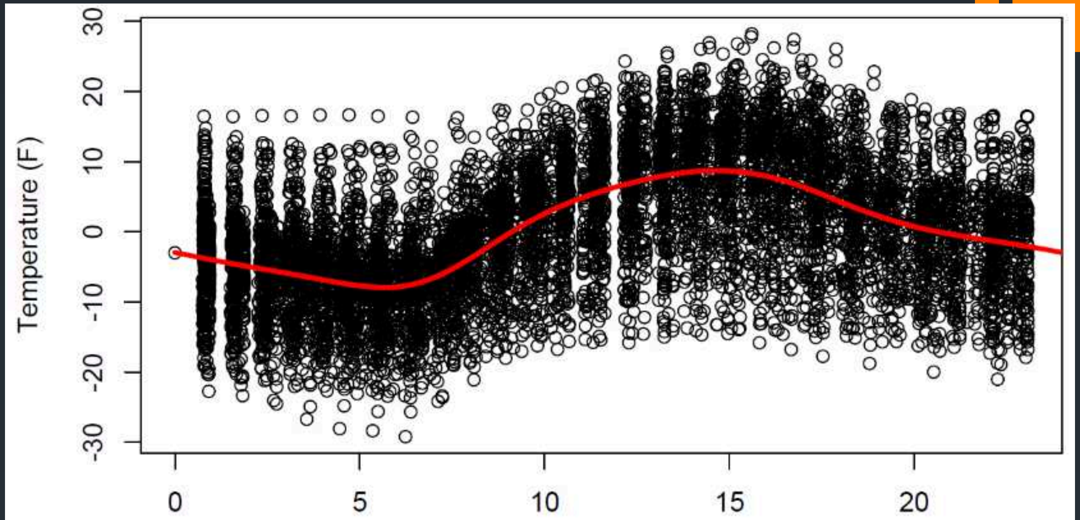
- PRES (atmospheric pressure) is best
- SRAD (solar radiation) is much worse than the others
  - We believe that by incorporating theoretical max SRAD we can greatly improve this in future work

Next Steps

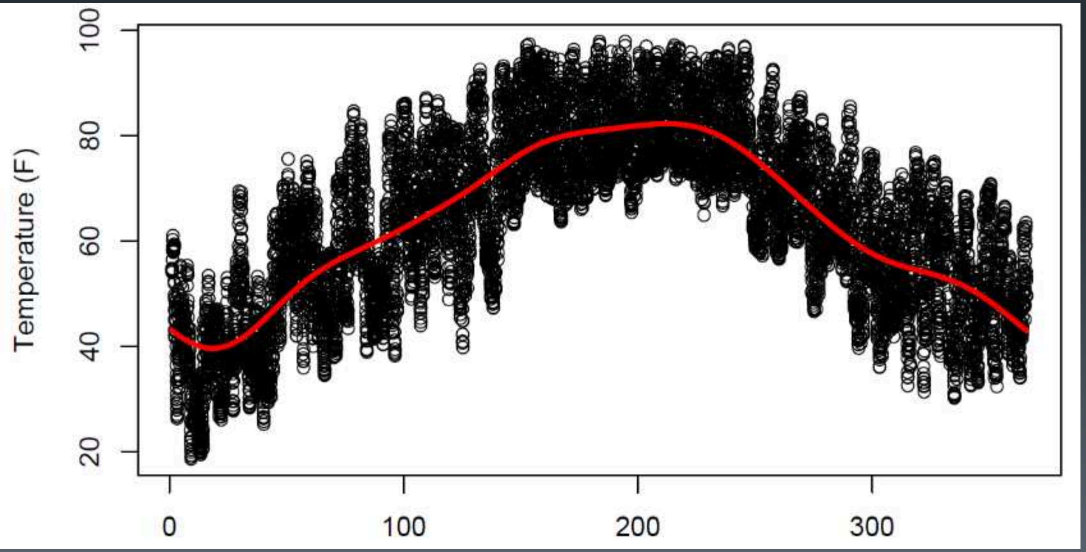
- Improved probability model for $P(A(v)|nbs)$
- Improved anomaly detection models based on Neighbor Regression

Dealing with Non-Stationarity

- Weather data is non-stationary
  - 24-hour cycle (“diel”)
  - 365-day cycle (“annual”)
  - storm system: irregular 2-5 days
- Three approaches:
  - Model and remove the cycles
  - Blocking
  - Use neighboring stations that experience the same cycles



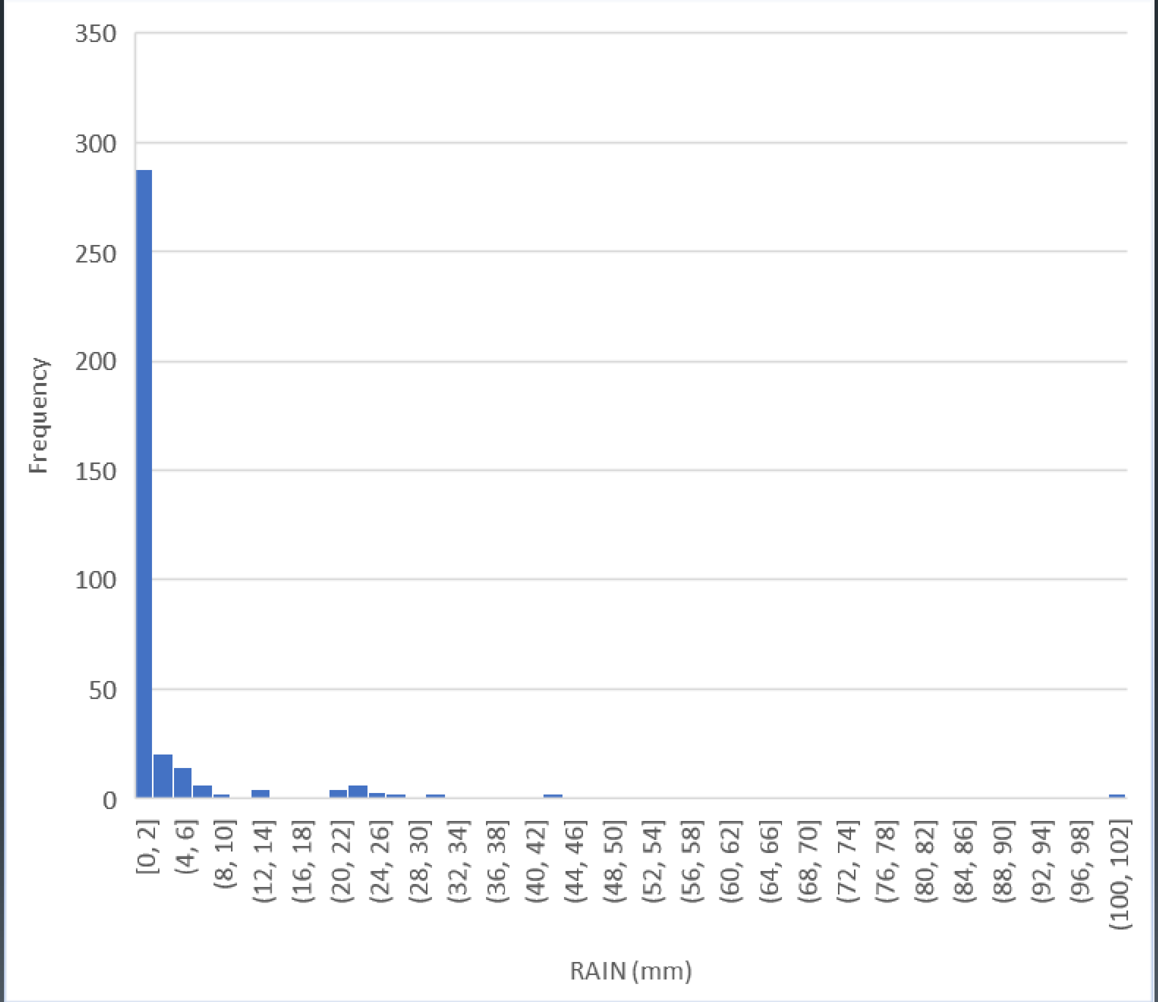
hour of day



day of year

Neighbor Regression for Precipitation

- Precipitation is most important variable:
  - Sub-Saharan 95%, Latin America 90% & 65% of South East Asia relies on rainfed Agriculture [Wani et al., 2009]
- Anomaly detection for precipitation is very difficult
  - Rainfall is zero on most days
  - Rainfall can be large
  - Very non-Gaussian

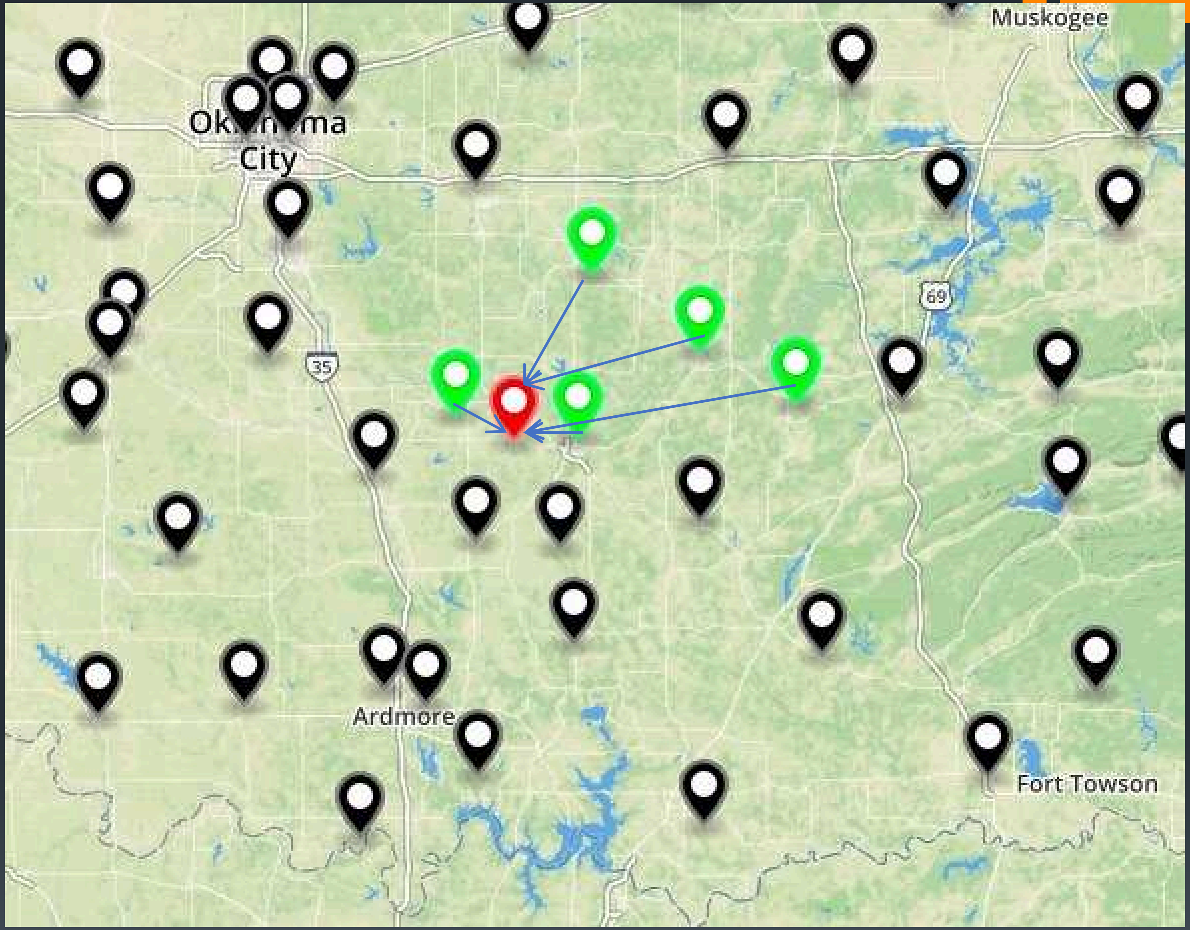


Station ADAX from Oklahoma Mesonet

Problem setting

Notation

- Let $s_1, s_2, \ldots, s_n$ a network of weather stations
- Let $R(s,t)$ rainfall measured at station s at time t
- $r_{\eta(s)}(t)$ denote vector of rainfall at time t for k neighboring stations



Goal:

- Detect rain gauge blockage at station s

Approach:

- Define a set $\eta(s)$ of k stations similar to s
- Fit a model f to predict $R(s,t)$ given $r_{\eta(s)}(t)$
- Compare prediction to observation
  - $\rho = y - \hat{y}$ "residual"

Single station unconditional mixture model

$$P(R(s,t)=r) = (1-p_1)\delta_0(r) + p_1 N(\log(r)\,;\mu,\sigma^2)$$

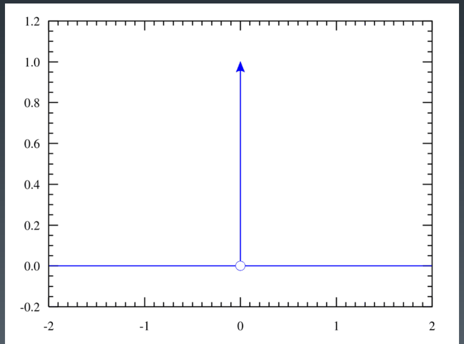
$P(R(s,t)=r\,)$

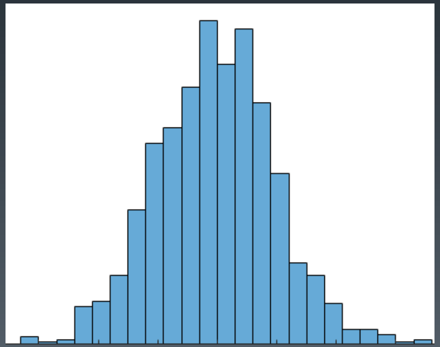
$r = 0$ → $(1-p_1)\delta_0$

$r > 0$ → $p_1\mathcal{N}(\log(r)\,,\mu,\sigma^2)$

$p_1 =$ probability of rainy day $R(s,t) > 0$

δ_0





$\mathcal{N}(\log(r)\,,\mu,\sigma^2)$

Condition on the Neighboring Stations η

- $r_{\eta(t)}$: observations from neighboring stations at time t

  - $P(R(s,t) = r \mid r_\eta(t)) =$

$$\begin{cases} \left(1 - p_1\left(r_\eta(t); \alpha\right)\right) \delta_0 & r = 0 \\ p_1\left(r_\eta(t); \alpha\right) N\left(\log(r)\,; \beta_0 + \beta_1^T \log\left(r_\eta(t) + \epsilon\right), \sigma^2\right) & r > 0 \end{cases}$$

- where:
  - $p_1\left(r_\eta(t); \alpha\right)$: logistic regression model with weight vector α
  - $\beta_0, \beta_1^T, \sigma^2$: Are parameters of the log-norm regression with covariates of $\log\left(r_\eta(t) + \epsilon\right)$
  - ϵ: small constant added to avoid log of zero

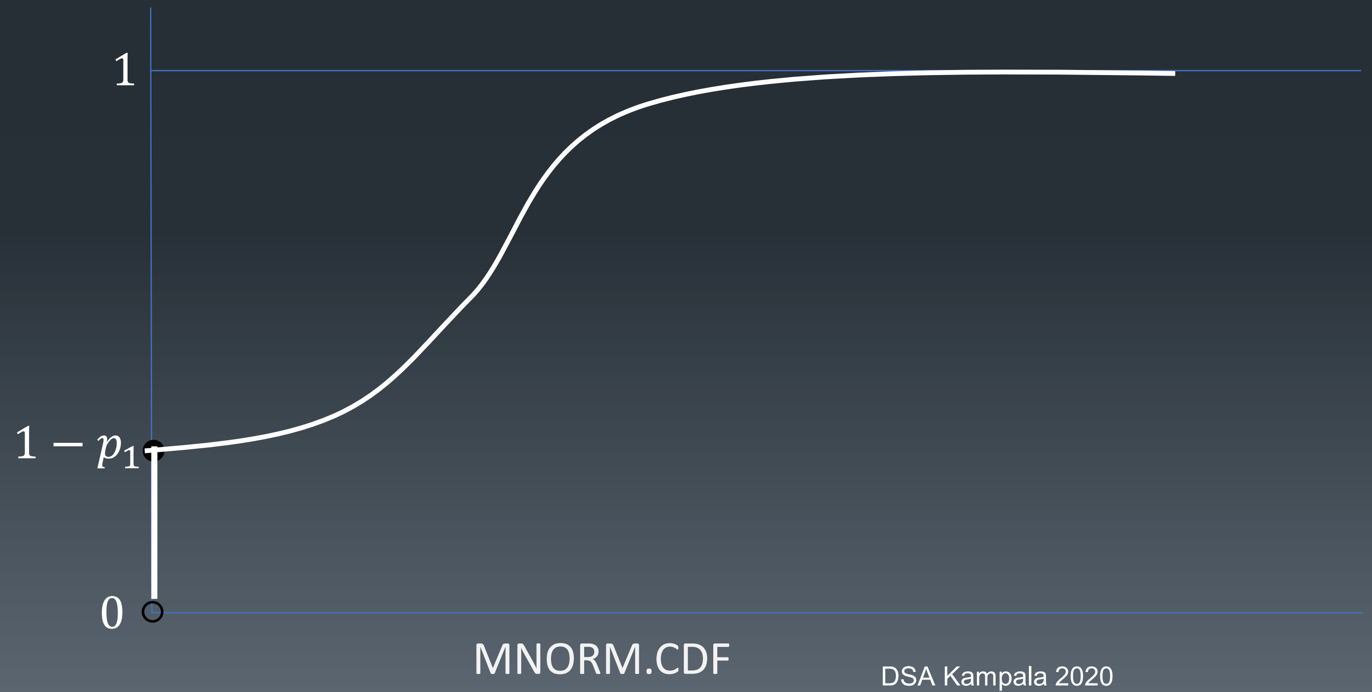
Estimating parameters

- Two-stage procedure [Min & Agresti, 2002]
- To estimate α, fit the logistic regression to $y = 1$ if $r > 0$ else $y = 0$
  - $P\left(y = 1 \middle| r_\eta(t); \alpha\right) = \frac{1}{1+e^{-(\alpha_0+\alpha^\top r_\eta(t))}}$
- To estimate parameters of lognorm β_0, β_1^T and σ^2
  - we restrict to case of $R(s,t) = r(s) > 0$ and plug $\hat{p}_1(s,t) = P\left(y = 1 \middle| r_\eta(t); \alpha\right)$

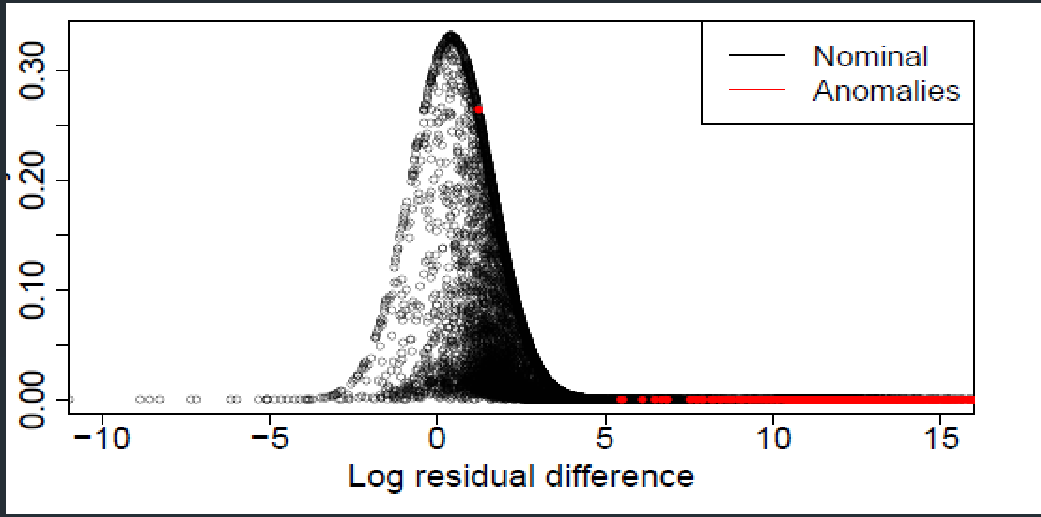
$$l(\beta) = \sum_t \hat{p}_1(s,t) \left[\underbrace{\log(r(s) + \epsilon) - \sum_{s' \in \eta(s)} \beta_{s'} \log\left(r(s')\right) - \beta_0}_{\text{Residual}} \right]^2$$

Two ways of computing anomaly score
Method 1: score using p-value of mixture model

- $MNORM.CDF(y) = -\log[\min\{F(\rho(y)), 1 - F(\rho(y))\}]$
  - ρ residual of neighbor regression model
  - $F(p) = (1 - p_1) + p_1\Phi(\rho, 0, \sigma^2)$



Method 2: Scoring based on NLL



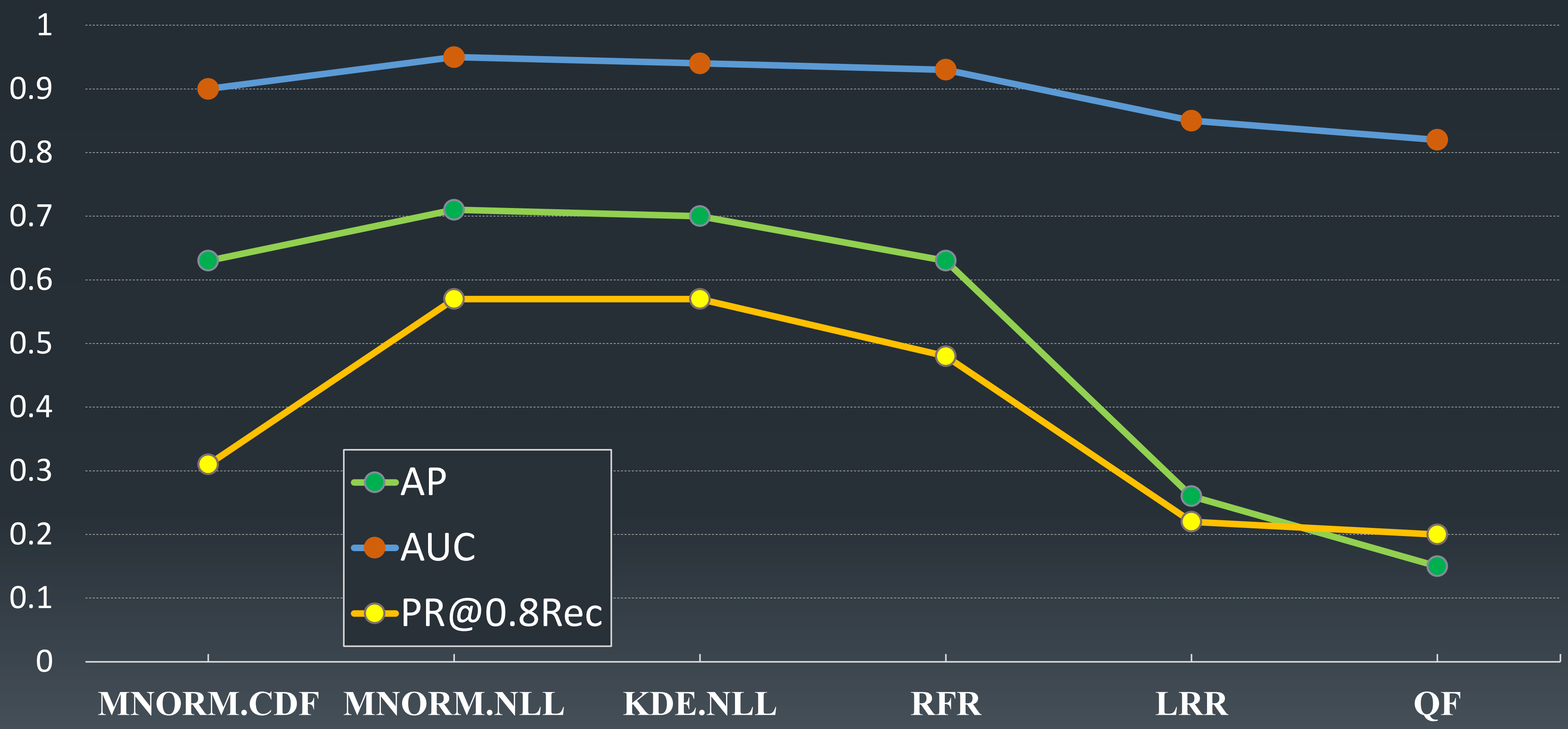
$$P\left(R(s,t) = r \middle| r_{\eta(t)}\right) = \begin{cases} \min\{(1-p_1)\delta_0\,, p_1 f(\rho,\beta\,|x)\} & y = 0 \\ p_1 f(\rho,\beta\,|x) & y > 0 \end{cases}$$

- $MNORM.NLL(r) = -\log P(R(s,t) = r|r_\eta(t))$
  - where $f(\rho,\beta\,|x)$ residual fitted to probability distribution

Experimental Study

- Data:
  - 2 year of Oklahoma mesonet data
  - Synthetic faults inserted to simulate rain gauge blockage
- Research questions:
  - RQ1: What is the best way of scoring anomaly?
  - RQ2: Which model is best?
- Metrics:
  - Prec@80: precision at 80% recall (detect 80% of blocked gauges)
  - Average precision
  - AUC

Comparison of scoring functions on 3 metrics



MNORM.NLL is the best

Status and Next Steps

- Precipitation model has been deployed on the TAHMO network

- Neighbor regression models for the other sensors
  - solar radiation
  - temperature
  - temperature and relative humidity (joint)
  - atmospheric pressure
  - wind speed and direction (joint)

Summary

- TAHMO is creating a weather station network of unprecedented size
  - QC must be automated as much as possible
- Existing QC Methods
  - Rule-based (ad hoc)
  - Probabilistic (requires modeling the sensor values when the sensor is broken)
- SENSOR-DX Approach
  - Define multiple views
  - Fit an anomaly detector to each view
  - Probabilistic QC by modeling the anomaly scores of broken sensors
  - Diagnostic reasoning to infer which sensors are broken
  - Out-performs baseline methods substantially

Summary (2): Neighbor Regression

- Predict sensor readings at station s from a nearby stations $\eta(s)$
- For Precipitation, we learn a mixture model
  - Logistic regression to predict the probability that $R(s,t) > 0$: p_1
  - Log-linear regression to predict the amount of precipitation $R(s,t)$ based on the amount at the neighbors
  - Anomaly score computed using log likelihood of the prediction error (residual)