Unsupervised spatio-temporal anomaly detection with missing data

ATD 2020 Challenge

Georgia Tech
Members: Yuexin Bian, Peibin Gong, Minghe Zhang
Faculty Adviser: Yao Xie

Contents

- Problem Statement
- Methods
- Results

- **Problem Statement**

Challenge:

try to create a model to find anomalous observations in sparsely sampled traffic flow observations.

Definition of anomaly:

For a given sensor, fit a linear trend model to its hourly flow over time. Subtract this trend from the flow, and re-add the original flow's mean. Denote this detrended flow as d .

For a given sensor s , hour h , and weekday w , the n_{th} observation of the detrended total flow d_{shw} is anomalous if,

$$\left|d_{shw}^{(n)} - \mu_{shw}\right| \geq 3\sigma_{shw}$$

- **Methods**

🌟 **Our Features:**

1. Use **tensors** to represent data (easy and fast for imputation)

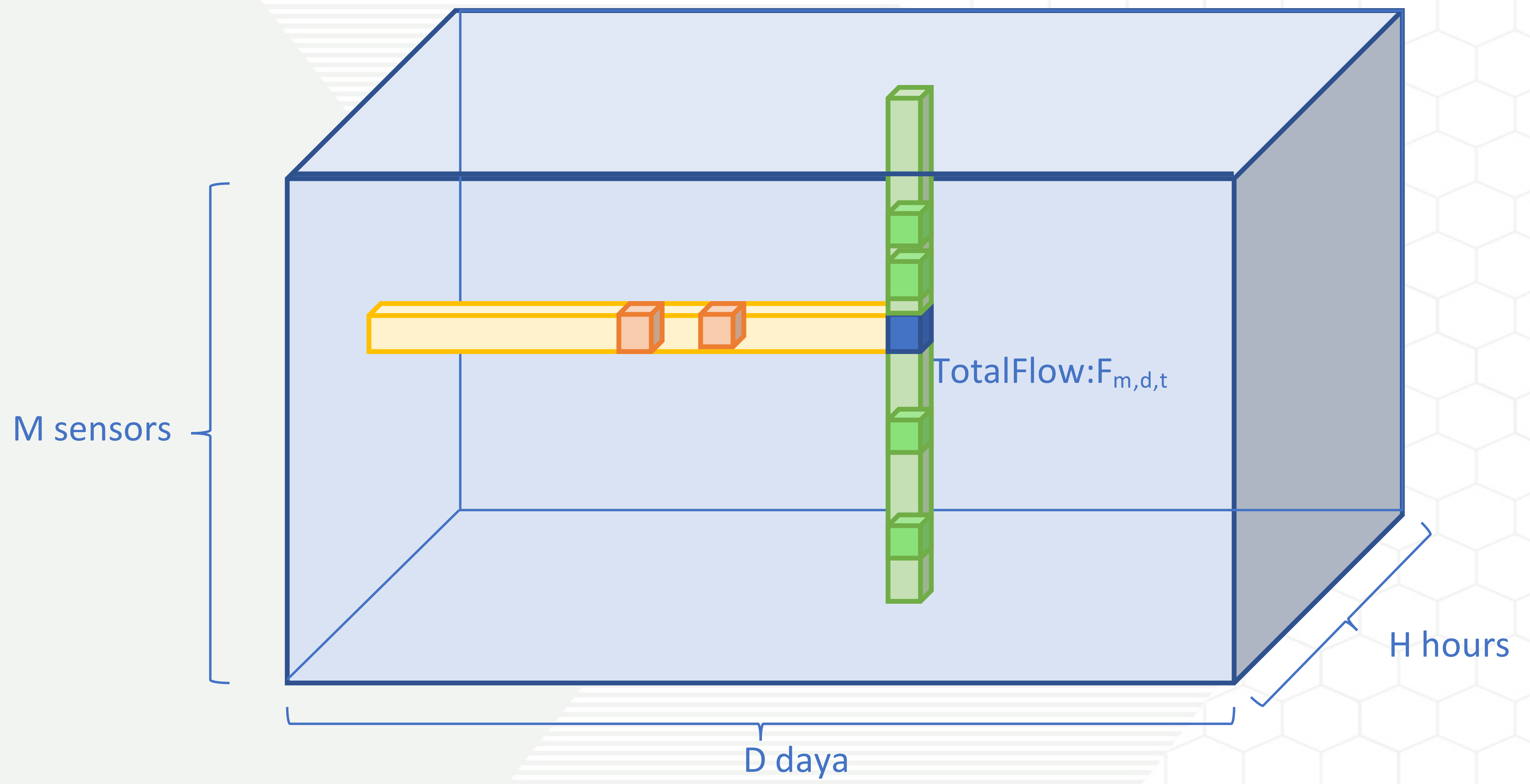
2. Use **imputation** to impute missing data

3. Treat data of **different sampling rates** combine algorithms that work best @different sampling rates

4. **Unsupervised** Method: no need for label information

5. **Iterative** method for better performance

- **Methods**



$$f_{m,d,t} = weight_{temporal} \times \sum_{pred=d\text{-}timerange}^{d} \frac{1}{date} f_{m,pred,t} + weight_{spatial} \times \sum_{j} a_{ij} f_{j,d,t}$$

date is the number of normal observations

we set $weight_{temporal}$ *and* $weight_{spatial}$ *accordding to different sampling rates*

⭐ *details in the following slides*

• First Step： Fit the data into tensor form

Georgia Tech

Fit the whole data into a **3D-tensor** ***F*** [shape = (M, D, H)]

Set $F_{m,d,t}$： measured traffic flow at monitoring sensor m on day d for hour t

The tensor is sparse：

For Observed data, we fill the true data in the corresponding place

For Unobserved data, we fill *nan* in the corresponding place

To begin with, the tensor F's nan is 92.38% *for City One*

- **Second Step：**

  Use mixed imputation methods to expand data amount

For Observed $F_{m,d,t}$：

We use it to impute other data points and then we use the imputed data to detect anomalies in observed data.
[We simply use the baseline detector group by (“ID”, “Weekday”, “Hour”)]

For different sampling rate sensors, we use different imputation methods and fine-tune different sets of parameters.

Second Step：

Use mixed imputation methods to expand data amount

For Observed $F_{m,d,t}$:

🌟 **Green marks** are parameters we fine-tune for sensors of different sampling rates

（1） temporal imputation:

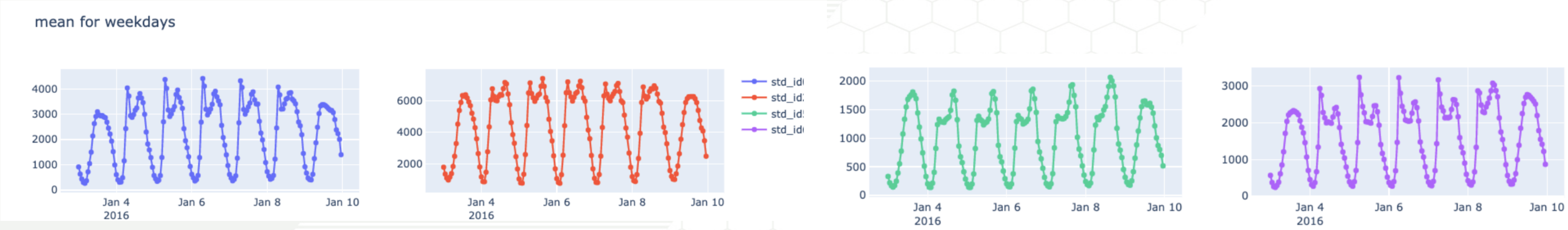
If d is weekday, then $imputeDate$ is weekdays in $(d, d + \textbf{time\_range})$

Else if d is weekend, then $imputeDate$ is weekends in $(d, d + \textbf{time\_range})$

Set:

$$F_{m,\, imputeDate,\, t} = F_{m,\, d,\, t}$$

Inspired by traffic flow's weekday and weekend pattern:



- **Second Step：**

Use mixed imputation methods to expand data amount

Georgia Tech

For Observed $F_{m,d,t}$:

🌟 **Green marks** are parameters we fine-tune for sensors of different sampling rates

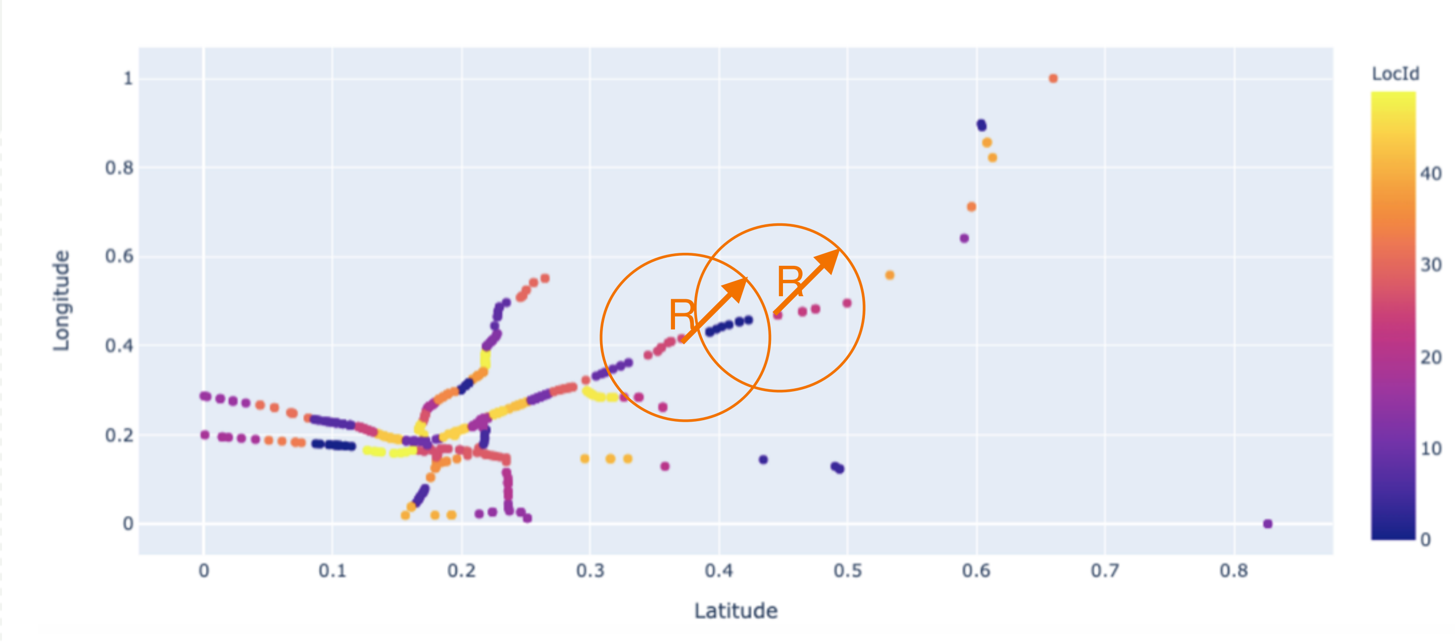
（2） spatial imputation:

Let $mean_{n,t}$ = mean(sensor=n, hour=t)

For any sensor n's location in circle(m.latitude, m.longitude, r) , meanwhile its $mean_{n,t}$ in range($mean_{m,t}$ ± **ratio**)

Set:

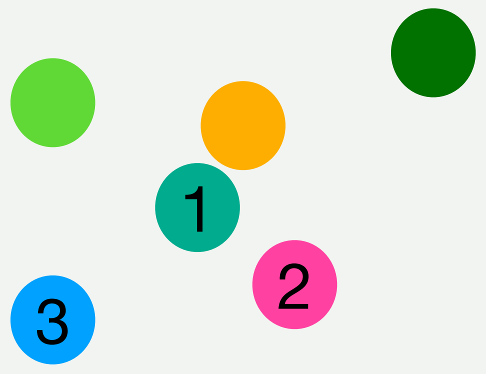
$$F_{n,d,t} = F_{m,d,t}$$



(2) spatial imputation:

Inspired by Network Hawkes process:

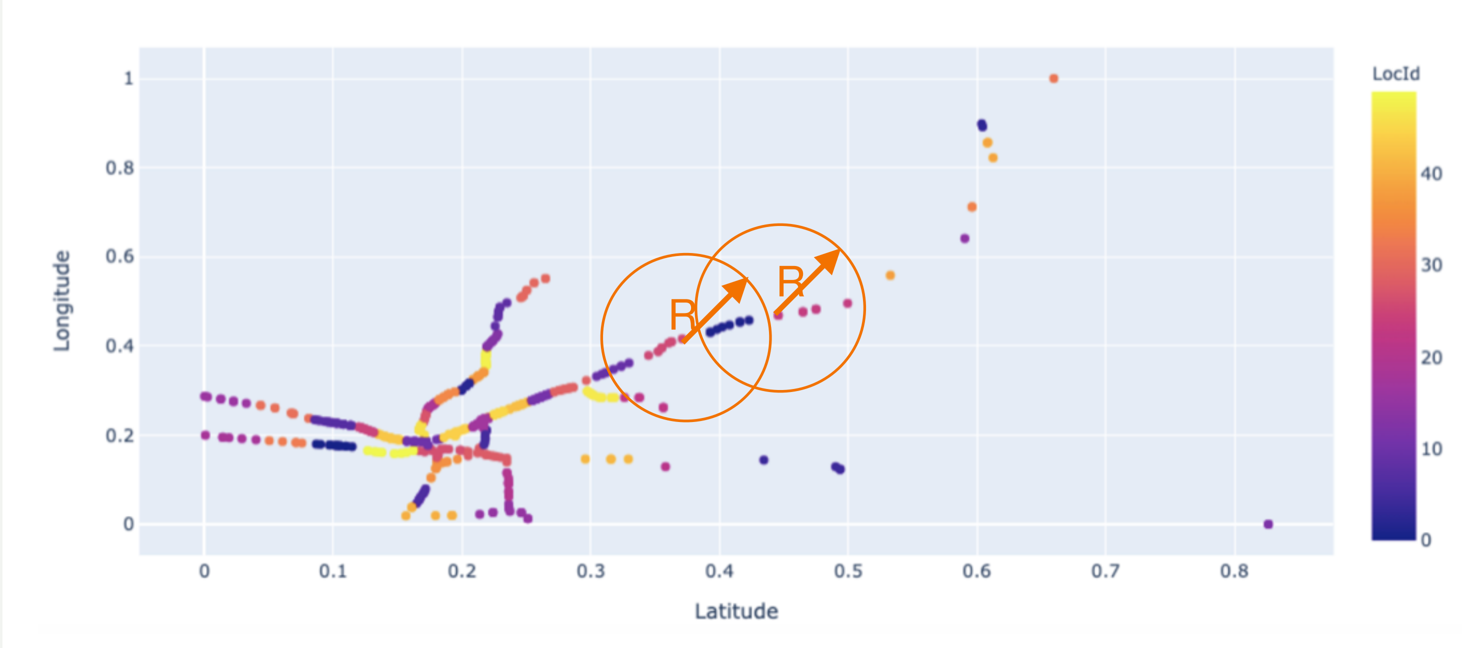
However consider the missing pattern, here we assume sensors near each other can indicate each other in a neighborhood



$$f_{i,d,t} = \varepsilon_{i,d,t} + \sum_{j} a_{ij} f_{j,d,t}$$

we simply set:

$$\begin{cases} a_{ij} = 0 & f_{j,d,t} \text{ is unobserved} \\ a_{ij} = \frac{1}{n} & f_{j,d,t} \text{ is observed} \end{cases}$$



m's neighborhood is defined as a circle(m.latitude, m.longitude, r)

n is the number of observed neighbors

• Second Step：

Use mixed imputation methods to expand data amount

For Observed $F_{m,d,t}$ and only high sampling rate（0.05/0.1/0.2） sensors

Additional spatial-temporal imputation： lag one hour from central sensor

Let $mean_{n,t}$ = mean(sensor=n, hour=t)

For any sensor n's location in circle(m.latitude, m.longitude, r) , meanwhile its $mean_{n,t+1}$ in range($mean_{m,t}$ ± **ratio**)

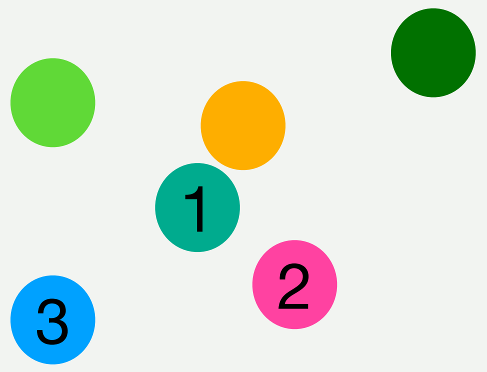
Set:

$$F_{n,d,t+1} = F_{m,d,t}$$

Additional spatial-temporal imputation : lag one hour from central sensor

Inspired by Network Hawkes process:

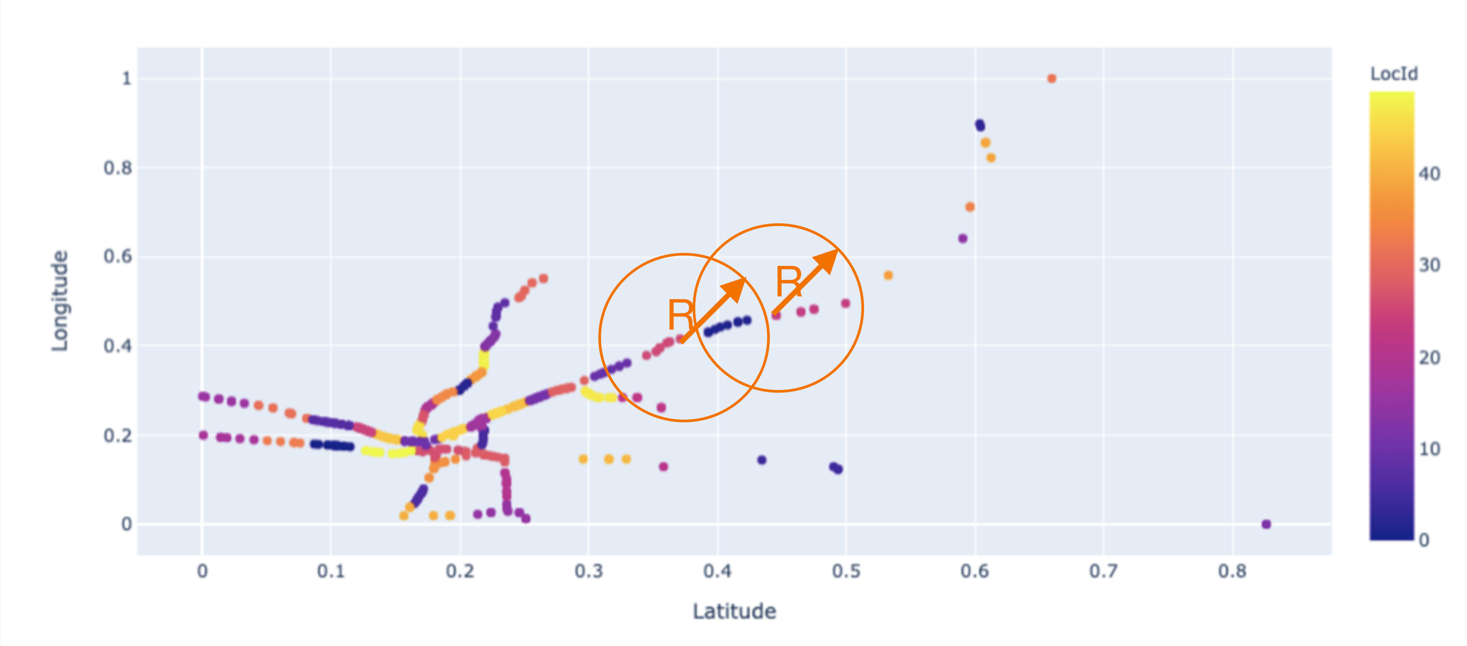
here we also assume sensors near each other can indicate each other, and it has lagging effect



$$f_{i,d,t+1} = \varepsilon_{i,d,t} + \sum_j a_{ij} f_{j,d,t}$$

we simply set:

$$\begin{cases} a_{ij} = 0 & f_{j,d,t} \textit{ is unobserved} \\ a_{ij} = \frac{1}{n} & f_{j,d,t} \textit{ is observed} \end{cases}$$



m's neighborhood is defined as a circle(m.latitude, m.longitude, r)

n is the number of observed neighbors

· Second Step：

Use mixed imputation methods to expand data amount

For Observed $F_{m,d,t}$ and only low sampling rate（0.01/0.02） sensors

Additional spatial imputation：

· Preparation：

Let $\text{mean}_{n,t}$ = mean(sensor=n, hour=t)

For sensor s, using [$\text{mean}_{s,0}$ $\text{mean}_{s,1}$ $\text{mean}_{s,2}$ ….. $\text{mean}_{s,23}$] to cluster sensors

[Here we use a kind of weighted k-means to cluster sensors and its parameter is: weight, classifiedNum]

· Imputation：

For any sensor s in the same label as m

Set：

$$F_{s,d,t} = F_{m,d,t}$$

- **Third Step：**

Do anomaly detection & impute without anomalies

We use a simple anomaly detection to filter out anomalies in the first stage.

Then we use filtered raw data to do a second imputation.

imputation → filter out anomalies → another imputation

- **Fourth Step：**

Use imputed data to detect anomaly

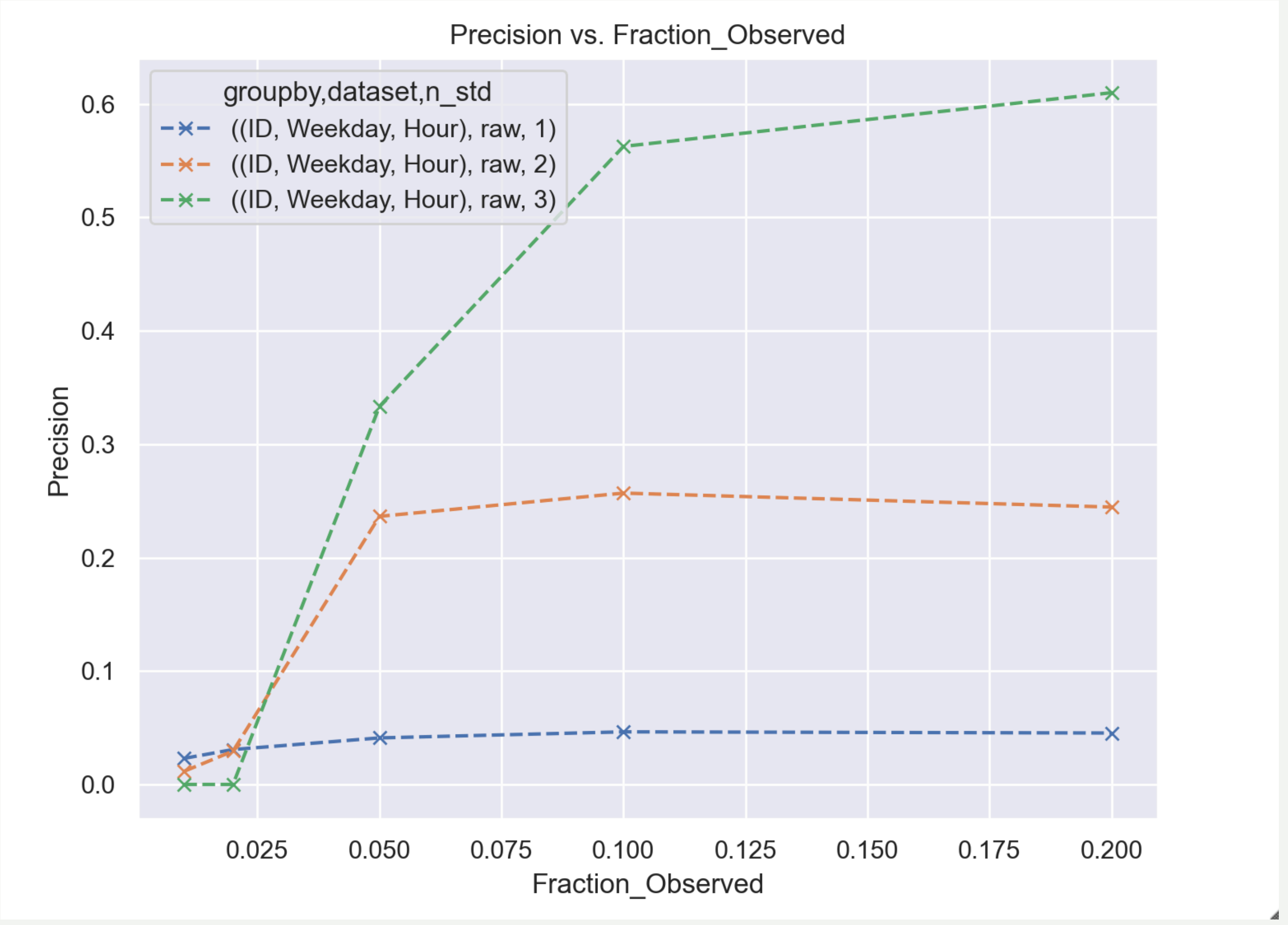
We simply use the baseline detector group by ("ID", "Weekday", "Hour")

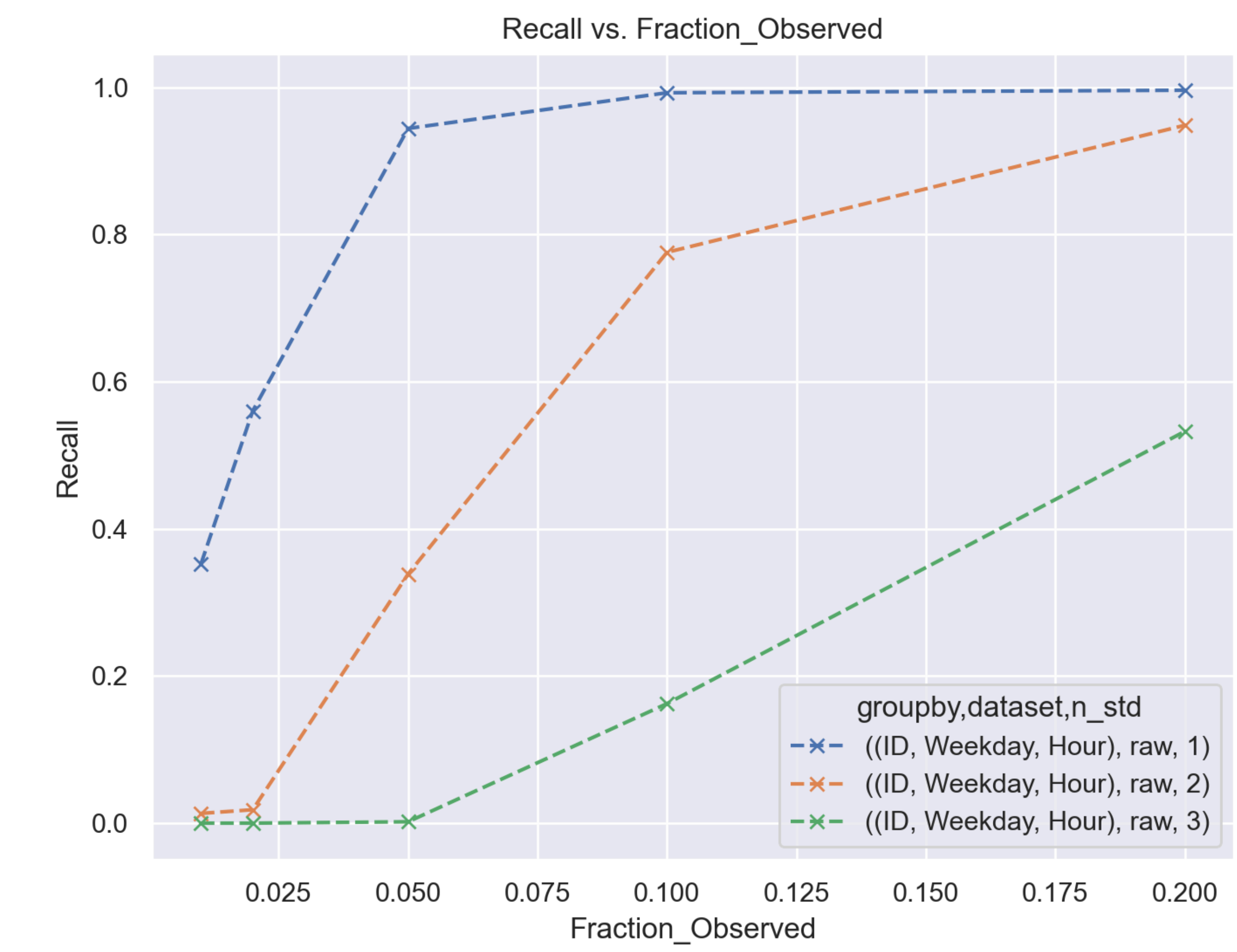
```
if t == 0.05:
    tmp = atd2020.detector.BaselineDetector(groupby=groupby).fit_predict(
        process_group, n_std=2.6
    )
elif t == 0.1:
    tmp = atd2020.detector.BaselineDetector(groupby=groupby).fit_predict(
        process_group, n_std=2.7
    )
else:
    tmp = atd2020.detector.BaselineDetector(groupby=groupby).fit_predict(
        process_group, n_std=3
    )
```

We set different stds for different sampling sensors

- **Fourth Step：**
  Use imputed data to detect anomaly

Different stds is inspired by baseline detector :





For a better F1 score, we fine-tune the std@different sampling rate

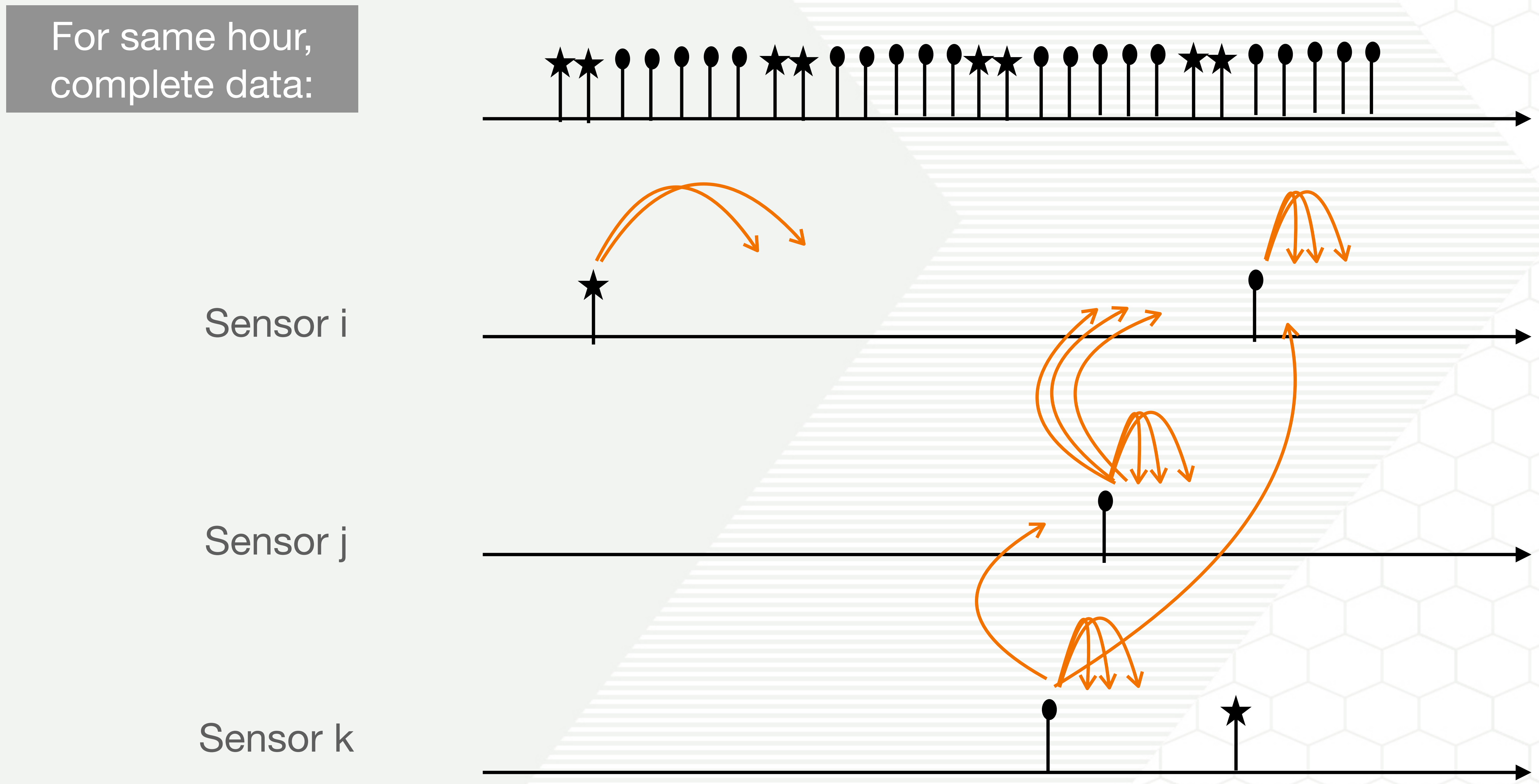
Small stds makes the results worse on *Precision* Score

Large stds makes the results worse on *Recall* Score

CREATING THE NEXT®

• Reasons why we use these imputation methods

- Incomplete data
- spatio-temporal pattern



Data has complex spatio-temporal connection

· Summary

· **Analyze the data challenge and find a mixed spatio-temporal imputation method to solve it.**

· **Introduce the inspiration and theory of the method, and talk about some details inside it.**

· **Perform the methods on *City One* data and get good results.**

· **Future Work:**
Train of the kmeans weights; Better method to estimate threshold;
Improved weights of neighbors