Baidu Cloud

PV monitoring based on linear regression

Personal profile

- Working in Baidu
- Focus on AIOps
  - Anomaly detection
  - Alert analysis
  - Troubleshooting

Baidu Cloud

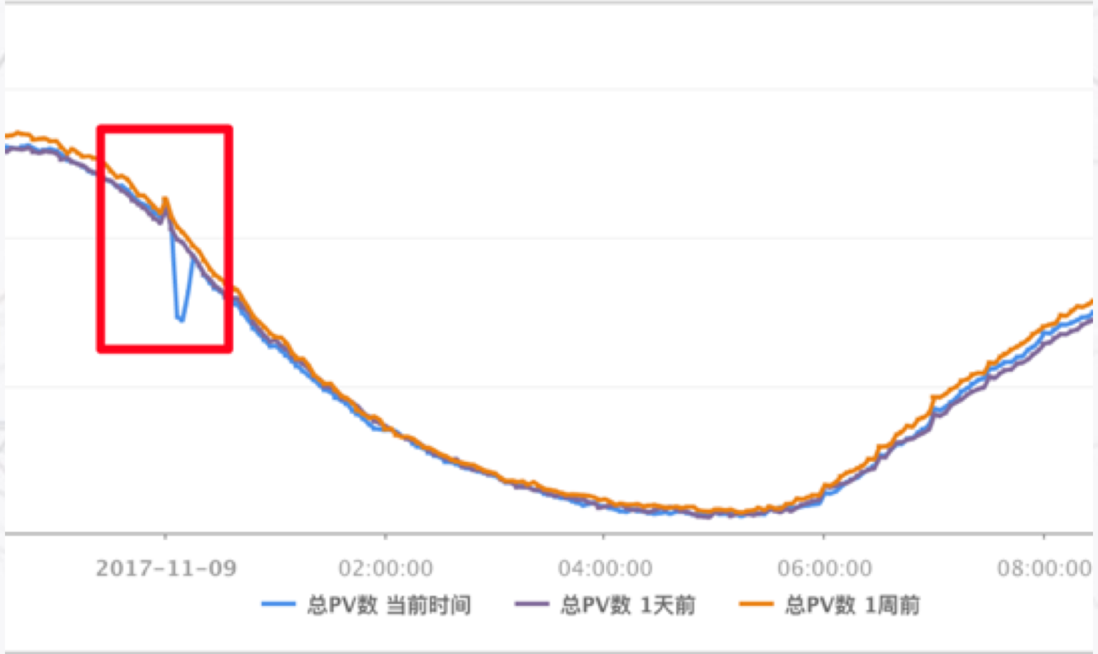
PV monitoring

- PV(Page View) is one of the golden signals of services
  - Extra-net failure
  - Module failure
  - Business logic error
  - Abnormal rise(crawlers 、 attack)

Baidu Cloud

The difficulties of PV monitoring

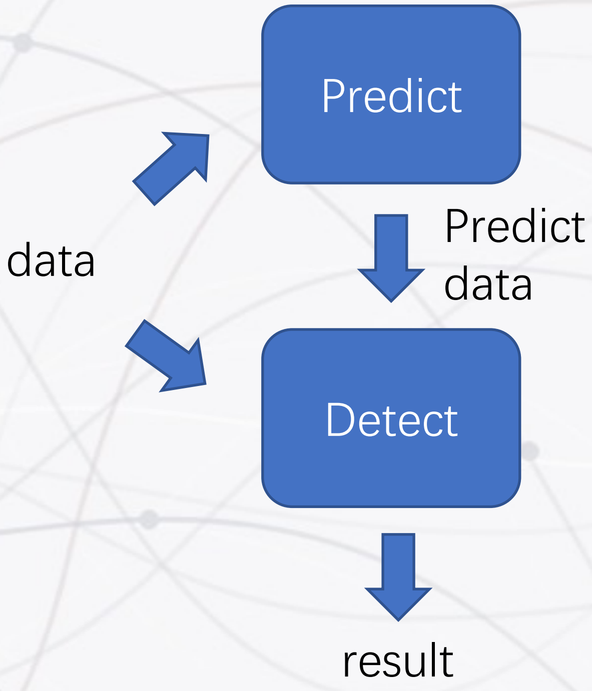
- Contextual
  - PV values
  - Local fluctuations
- Asymmetric concerns on rising and dropping
- Get rid of the influences of business logic changes



Baidu Cloud

Algorithm model

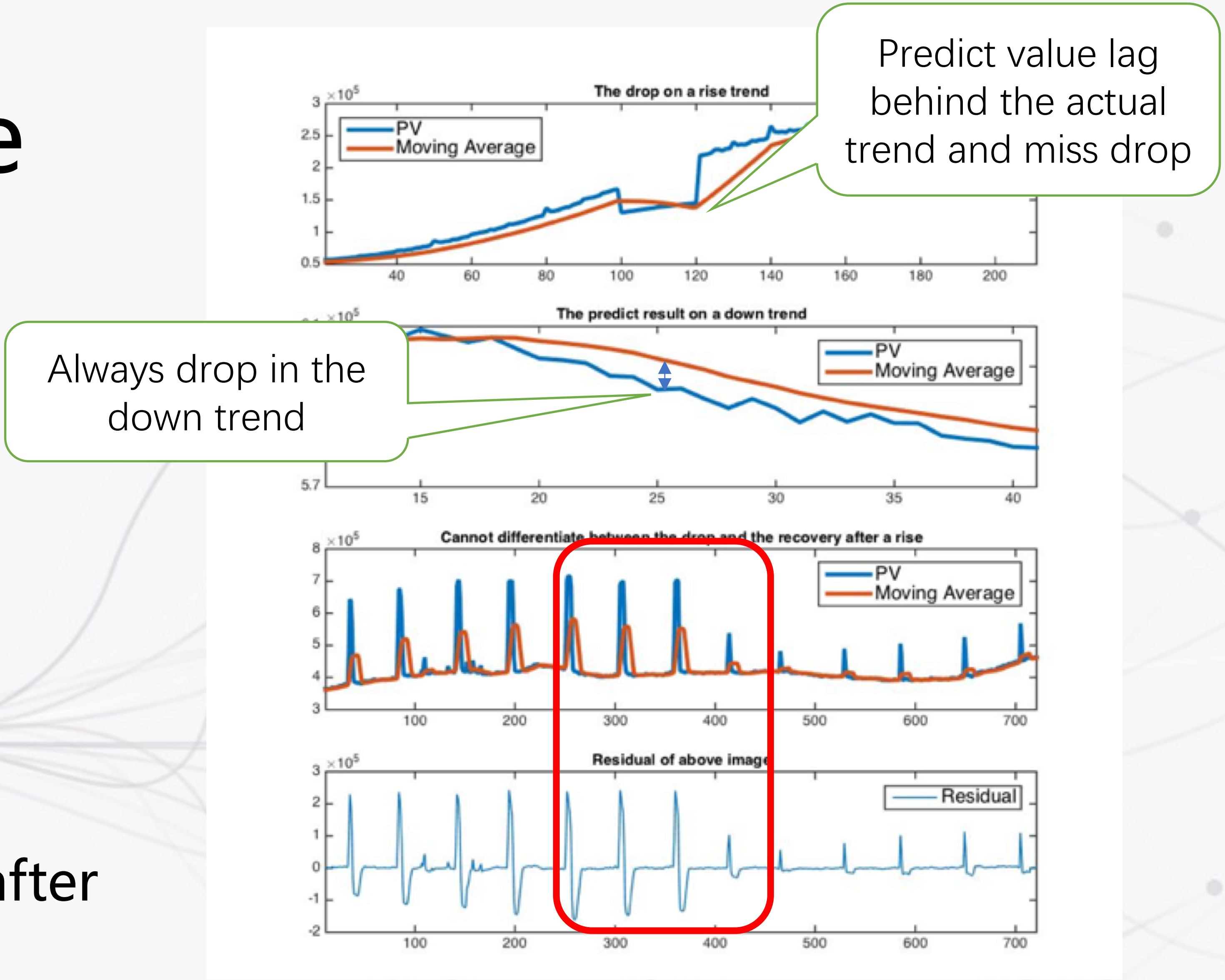
- Predict
  - Forecast contextual value
  - Differentiate rise and drop
  - Adaptive to level change
- Detect
  - Context-free threshold



Baidu Cloud

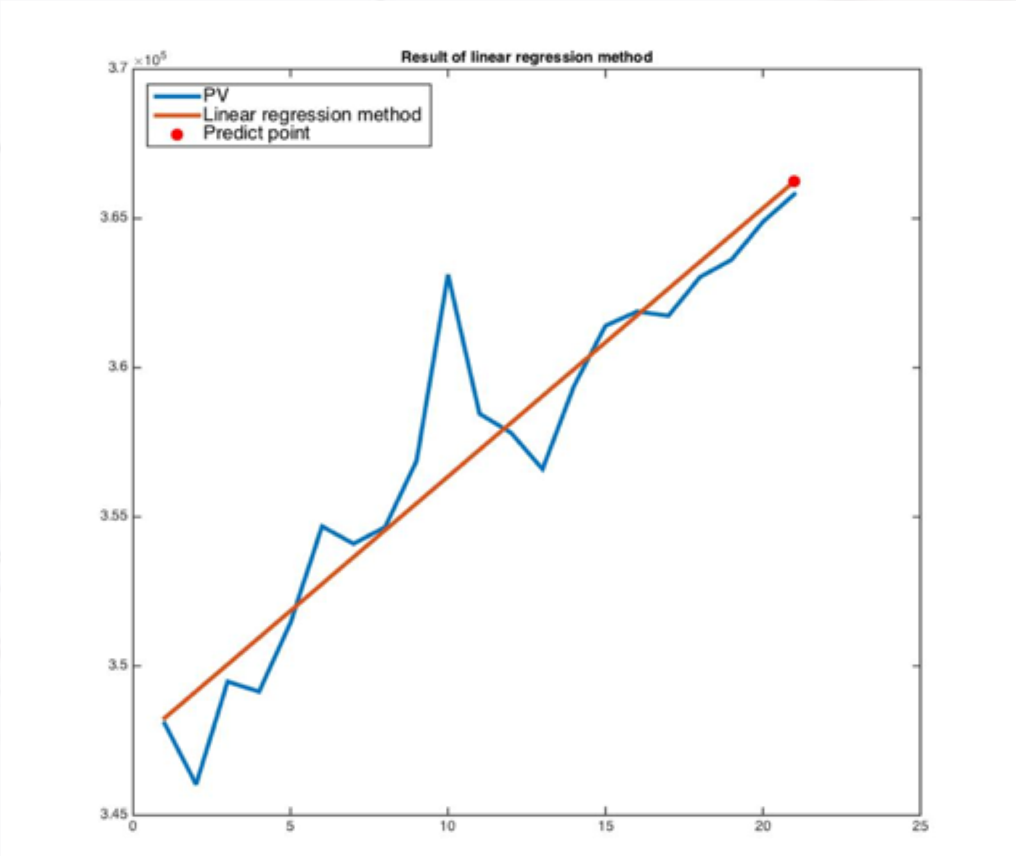
Moving average

- Time-series data
  - $\{y_t | t = 1 \ldots n\}$
- Prediction based on moving average
  - $\hat{y}_t = \frac{\sum_{\tau=t-w+1}^{t} y_\tau}{w}$
- Problems
  - Lag behind actual trend
  - Drop and the recovery after a rise



Linear regression

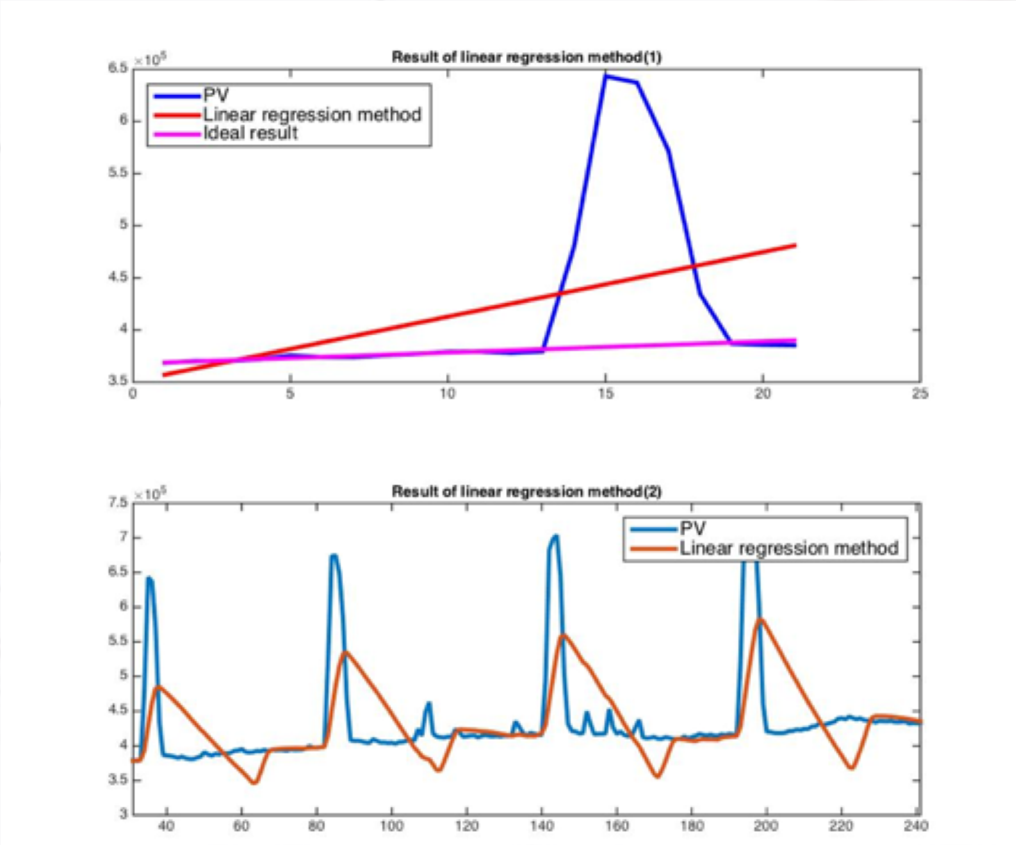
- Locally linear
  - $y_t \approx kt + b$
- Predict
  - $\hat{y}_t = kt + b$
- Calculate parameters
  - $k, b = \underset{k,b}{argmin}\, L$
- Least squares
  - $L_2 = \sum_{\tau=1}^{t} (y_\tau - \hat{y}_\tau)^2$



Baidu Cloud

Problem of linear regression

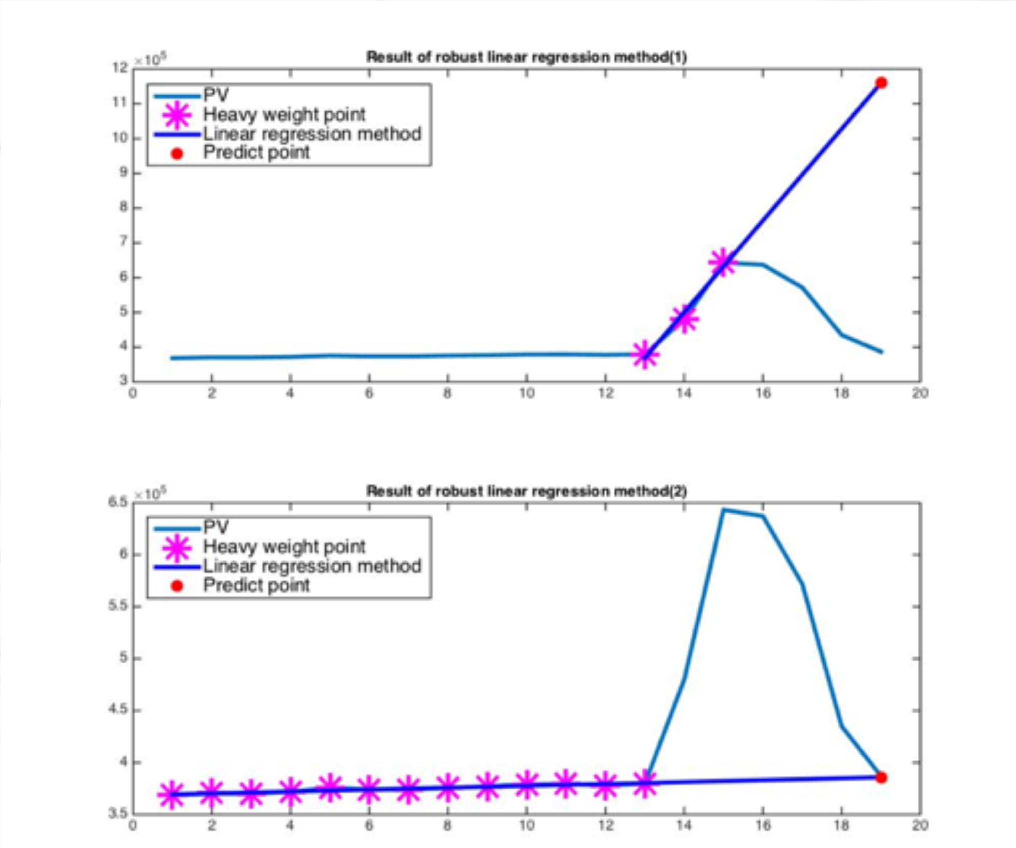
- Susceptible to abnormal values
  - The impact of abnormal values is much larger than normal values
  - Undesirable fluctuations



Baidu Cloud

Robust linear regression

- Least absolute deviations
  - $L_1 = \sum_{\tau=1}^{t} |y_\tau - \hat{y}_\tau|$
- Iteratively re-weighted least squares
  - $L' = \sum_{\tau=1}^{t} \omega_\tau (y_\tau - \hat{y}_\tau)^2$
  - $\omega_\tau = \frac{1}{|y_\tau - \hat{y}_\tau|}$
- Multiple solutions

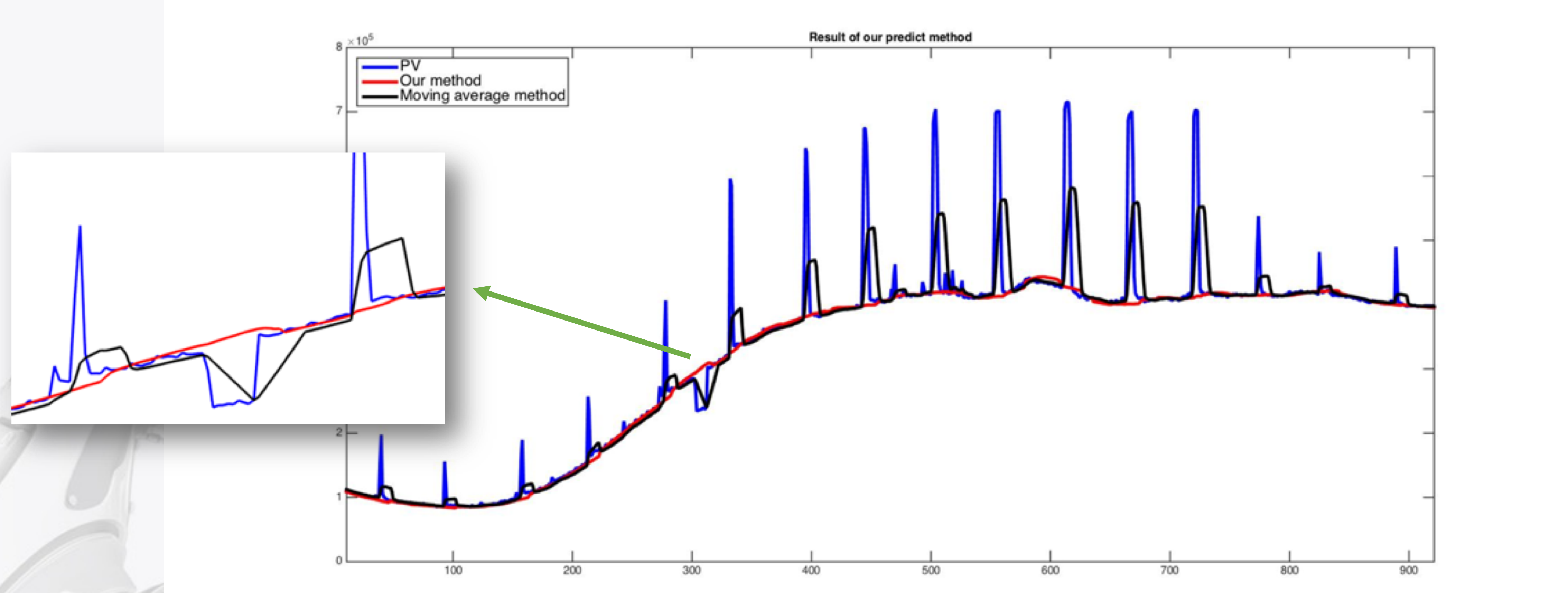


Baidu Cloud

Obtain ideal result

- k should be stable
- Initial weights
  - Initial weights from previous k
- Iteration converge to bad k
  - Checking $z\text{-}score = \frac{k-\mu}{\sigma}$
  - Use the best k in the window when z-score exceeds certain threshold

Prediction result



Baidu Cloud

Detection

- Detection based on absolute residual
  - $\epsilon_t = y_t - \hat{y}_t$
  - Different thresholds for different curves at different time
- Detection based on residual percentage
  - $r_t = \frac{\epsilon_t}{\hat{y}_t} \times 100\%$
  - Still need different thresholds at different time

Statistical hypothesis testing

- Probability Modeling
  - PV is number of page views in an interval: Poisson distribution

$$P(y_t;\lambda) = \frac{e^{-\lambda}\lambda^{y_t}}{y_t!}, \lambda = \hat{y}_t$$

  - Pick a threshold C such that $P(Y_t \le C) < p_1$
    - $P(Y_t \le C) = \sum_{v=0}^{C} P(v;\lambda)$
    - Y_t is a random variable for y_t
  - Emulate Poisson distribution with Normal distribution: $\mathcal{N}(y_t;\mu,\sigma^2)$
    - $\mu = \sigma^2 = \lambda = \hat{y}_t$
    - $y_t < C \Leftrightarrow y_t < \hat{y}_t - m\sigma$, for PV drop

$$z = \frac{y_t - \hat{y}_t}{\sqrt{\hat{y}_t}} < -m$$

Baidu Cloud

Why residual percentage doesn't work

$$Percentage\ model: \frac{y_t - \hat{y}_t}{\hat{y}_t} < -r$$

$$Poisson\ model: \frac{y_t - \hat{y}_t}{\sqrt{\hat{y}_t}} < -\mathrm{m}$$

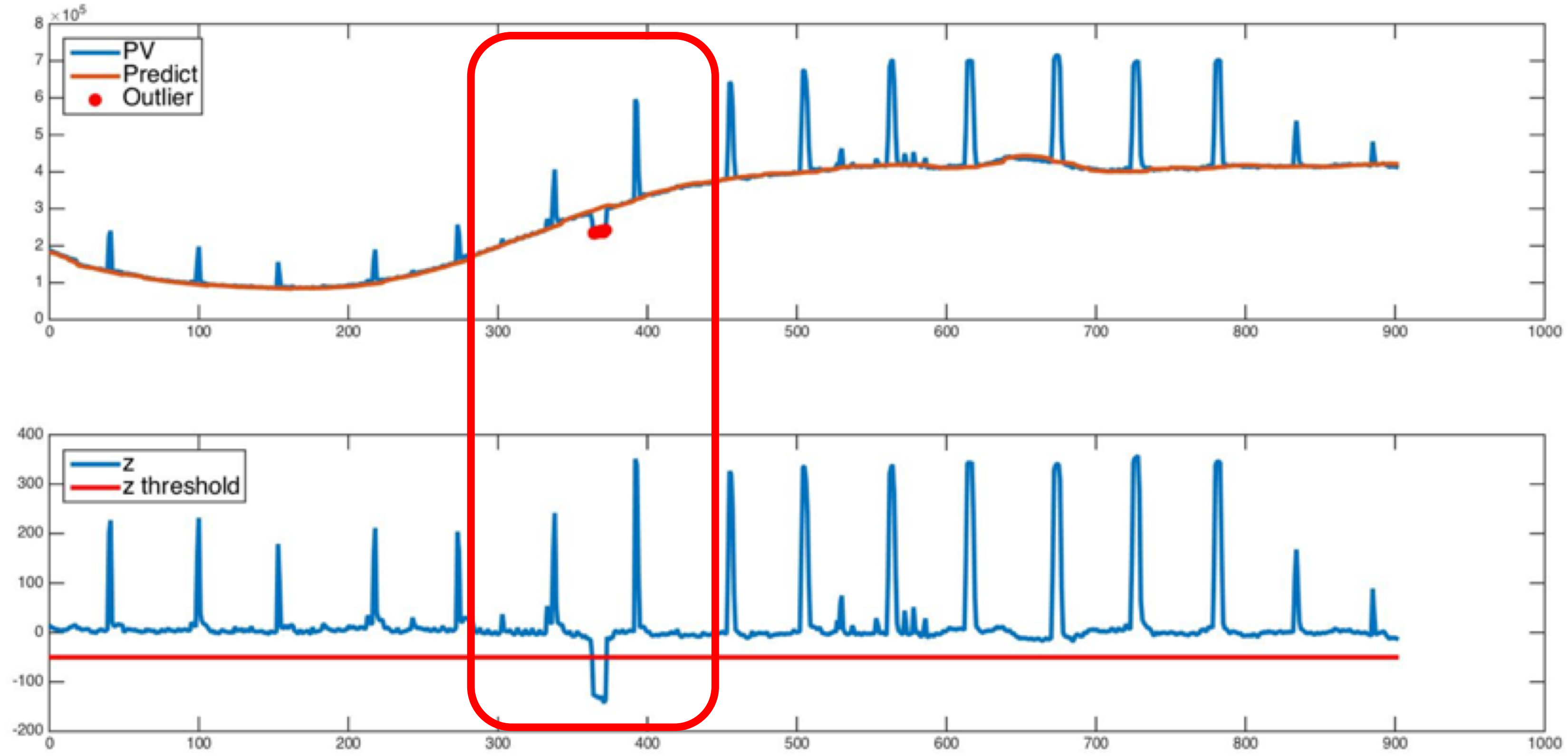
$$\Rightarrow \mathrm{m} = r\sqrt{\hat{y}_t}$$

$$m = r_{day} \times \sqrt{\hat{y}_{day}} = r_{night} \times \sqrt{\hat{y}_{night}}$$

$$\Rightarrow\ r_{night} = r_{day} \times \sqrt{\frac{\hat{y}_{day}}{\hat{y}_{night}}}$$

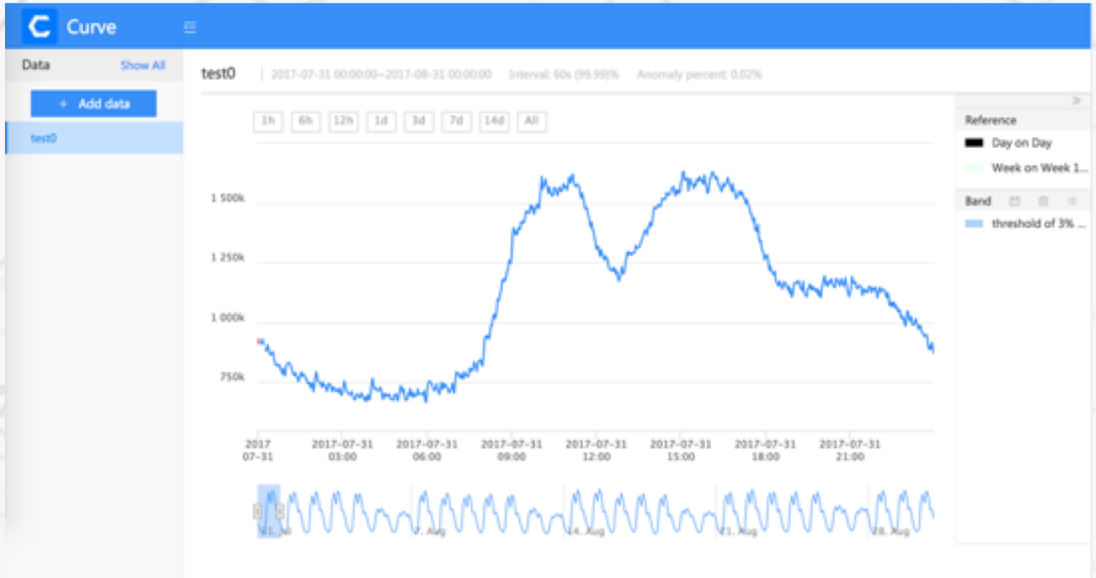
$$\Rightarrow\ r_{night} > r_{day}$$

Result



How to evaluate the results

- Label PV curve for evaluate by our labeling tool
  - http://curve.baidu.com
  - https://github.com/baidu/Curve
- Evaluate with precision and recall
  - Precision 80%+
  - Recall 95%+



Baidu Cloud

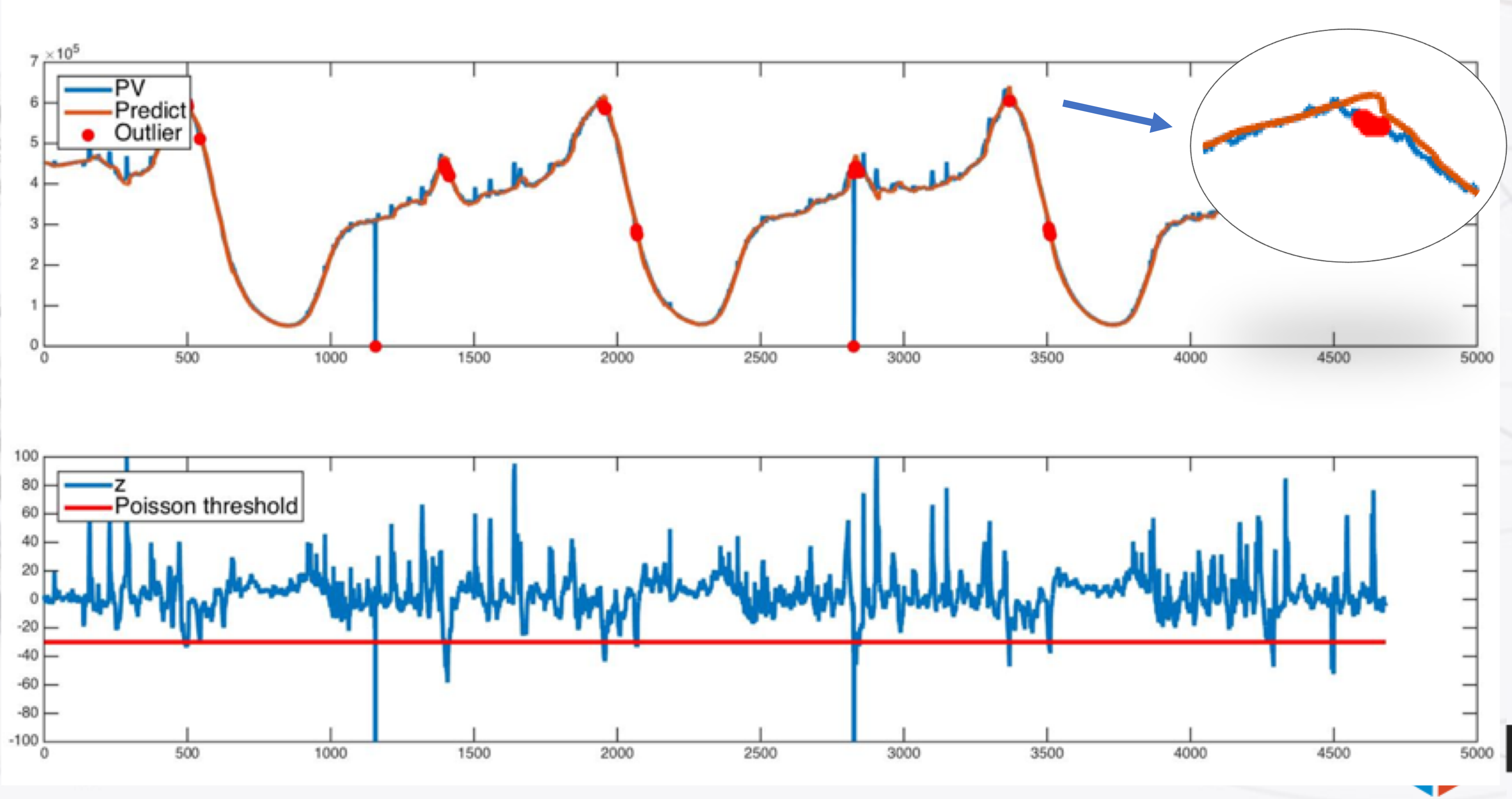
Summary

- Prediction based on robust linear regression
  - Least absolute deviations
  - Multiple solutions
  - Obtain ideal result
- Detection based on Poisson distribution
  - Probability Modeling

Baidu Cloud

Future work

- Linear regression fail to catch up on sharp turning points



Baidu Cloud

Thanks

Baidu Cloud