Anomaly Detection on Golden Signals

Yu Chen

IOP@Baidu

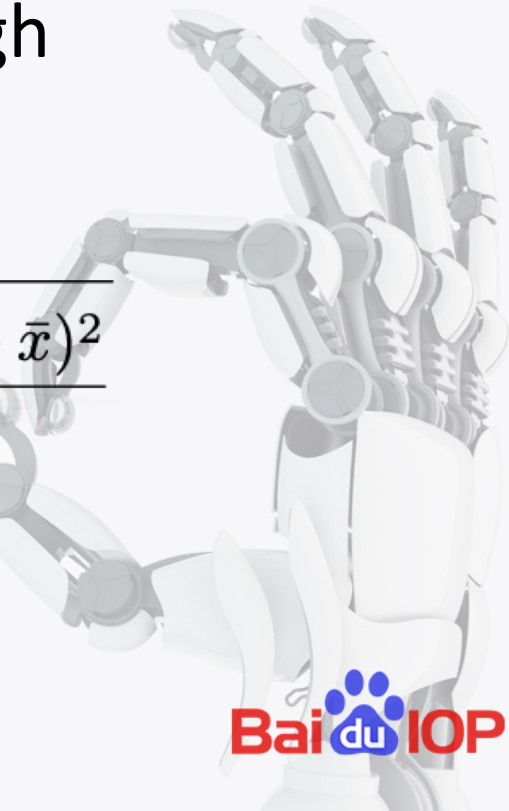
Monitoring the healthiness of services

- Golden signals
  - Latency, traffic, errors, saturation
- Challenges
  - Abnormal data are rare (<1%)
  - Hard to label
- Method
  - Statistical analysis + machine learning
  - >90% precision & recall

Baidu IOP

Latency: usual approach

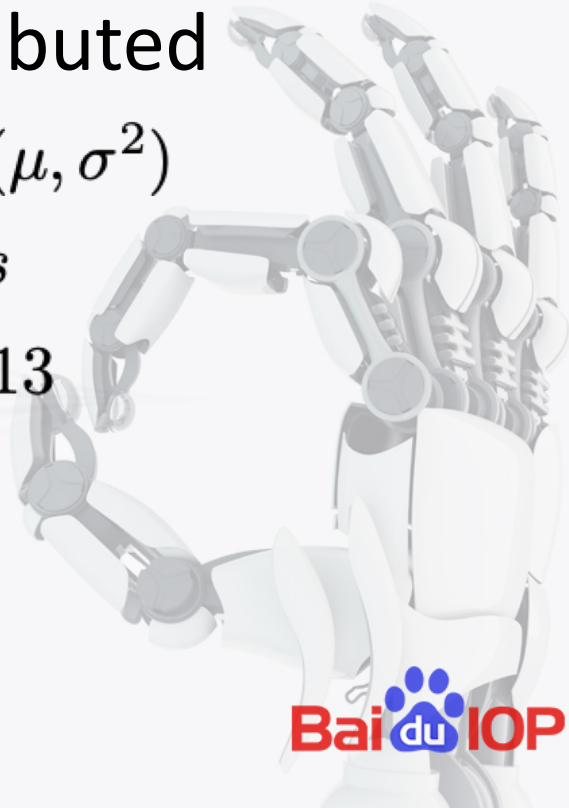
- Detect whether the latency is too high
  - History data $\{x_i\}, i = 1, 2, \ldots, n$
  - Mean $\bar{x} = \dfrac{x_1 + x_2 + \cdots + x_n}{n}$
  - Standard deviation $s = \sqrt{\dfrac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$
  - Alert when $x_j > \bar{x} + 3s$
    - x_j is an observation of latency



Baidu IOP

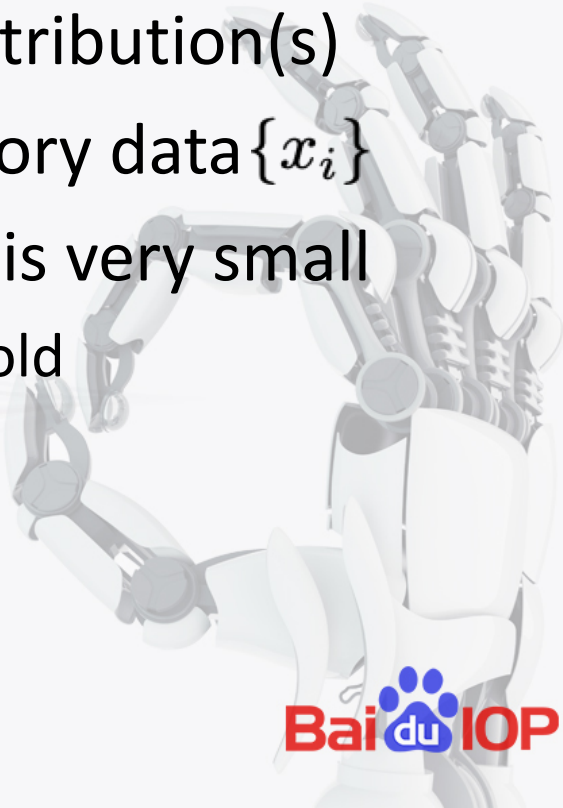
Translation to statistics

- Assume latency is normally distributed
  - Normal (Gaussian) Distribution $\mathcal{N}(\mu, \sigma^2)$
  - Parameter estimation $\mu = \bar{x}, \sigma = s$
  - Probability $P\{x_j > \mu + 3\sigma\} = 0.0013$



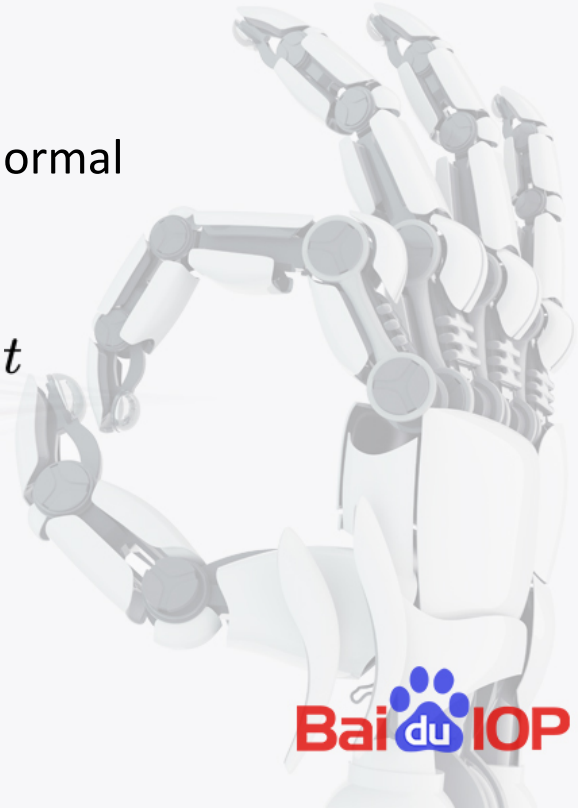
The statistical approach

- The signal x follows some random distribution(s)
- Estimate the distribution(s) using history data $\{x_i\}$
- Pick a threshold t such that $P\{x > t\}$ is very small
  - $p = P\{x > t\}$ is the probability threshold
  - t is the signal threshold



Why using probability?

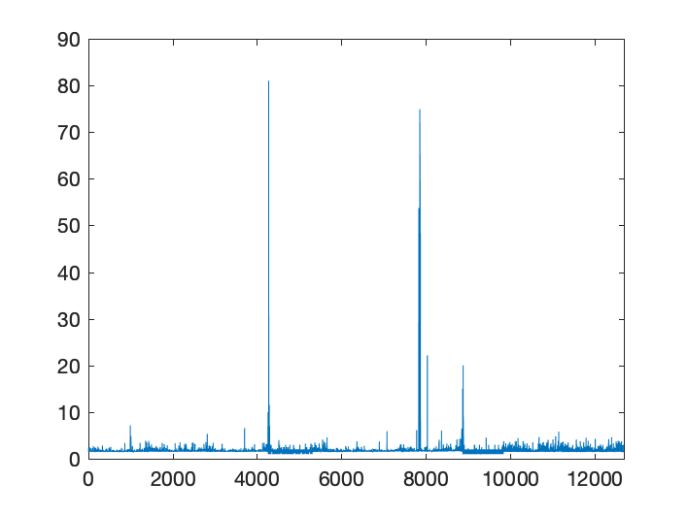
- p refers to the probability of false alarms
  - Close to engineers' feeling
    - Anomalies are rare, rare values are usually abnormal
  - Remains constant
  - Rebuild the distribution model periodically
    - Automatic adjustment of the signal threshold t
- How to choose p ?
  - 1 point per minute
  - 1 false alarm per day
  - 1440 points per day → $p \approx 0.0007$

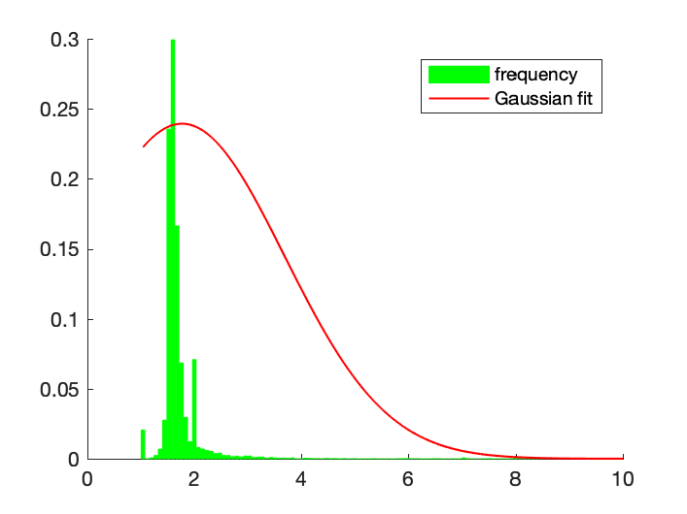


Baidu IOP

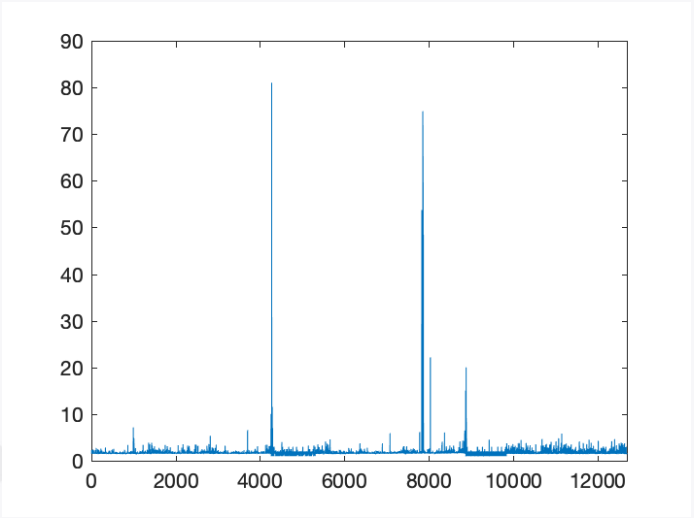
Building the distribution model

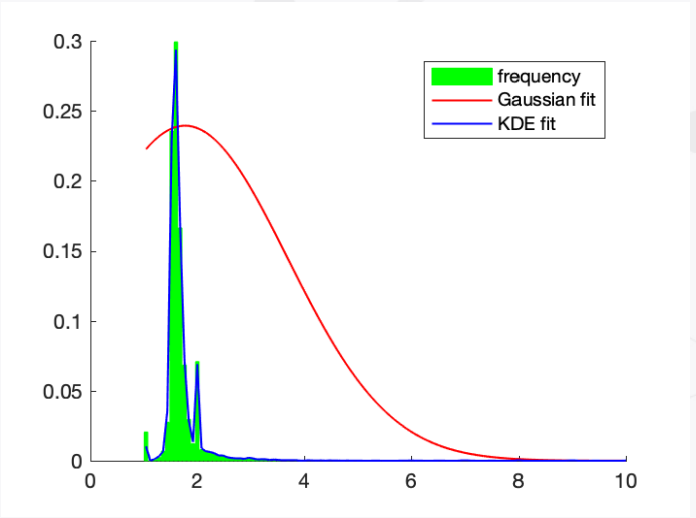
- "Assume latency is normally distributed"
  - Correct or not?





Kernel density estimation





$$\text{History} = \{x_1, x_2, \cdots, x_n\}$$

$$f(x) = \frac{1}{n}\sum_i K(x; x_i)$$

$$K(x; x_i) = \mathcal{N}(\mu = x_i, \sigma^2)$$

$$\sigma \approx 1.06 s n^{-\frac{1}{5}}$$

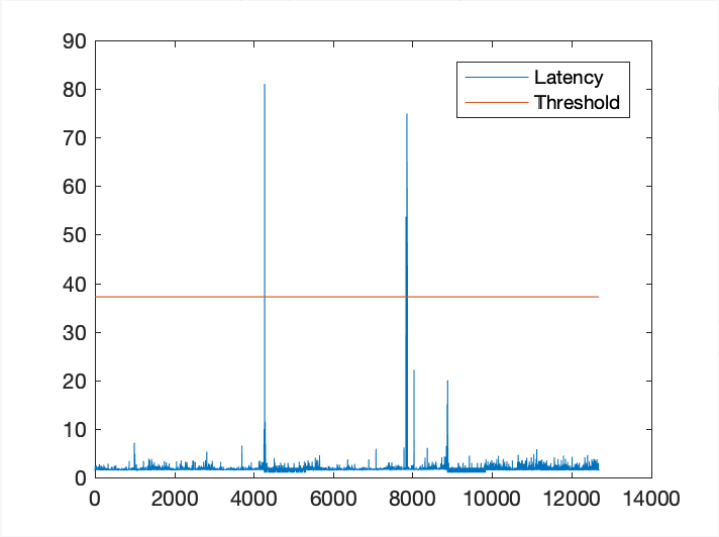
Baidu IOP

Abnormal values in history

- Signal threshold t
  - Probability threshold

    $p = 0.001$
  - Find t such that

$$P\{x > t\} = \int_{t}^{+\infty} f(x)dx = p$$



Abnormal values enlarge the threshold

Baidu IOP

Removing abnormal values

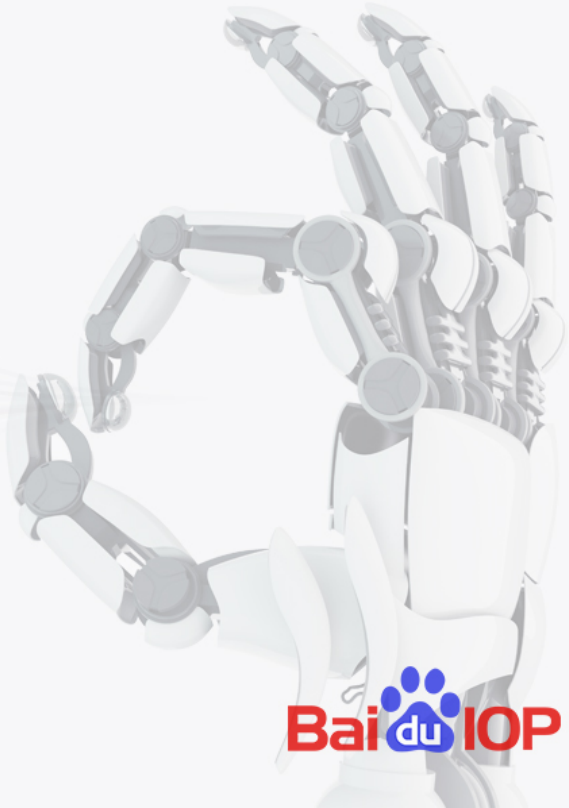
- Linear discrimination analysis (LDA)
  - Sort $x_1 \leq x_2 \leq \cdots \leq x_n$
  - Split into 2 clusters at every possible position

    $\{x_1, x_2, \ldots, x_i\} \quad \{x_{i+1}, x_{i+2}, \ldots, x_n\}$
  - Compute scatter within cluster

$$\bar{x}_l = \frac{x_1 + x_2 + \cdots + x_i}{i}$$

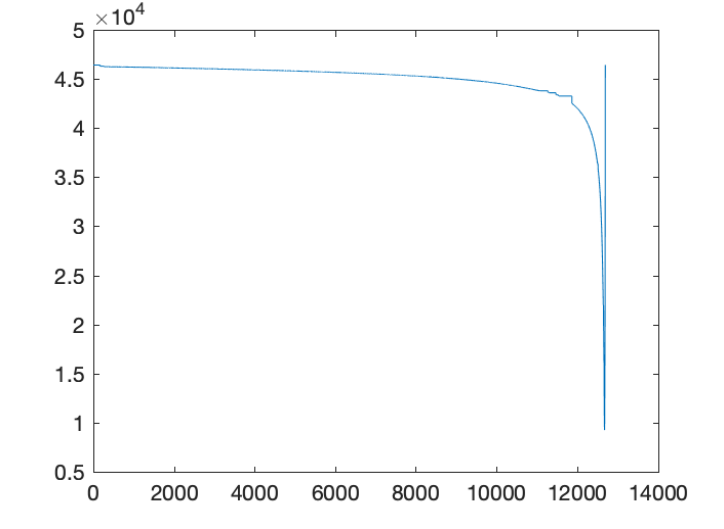
$$\bar{x}_r = \frac{x_{i+1} + x_{i+2} + \cdots + x_n}{n - i}$$

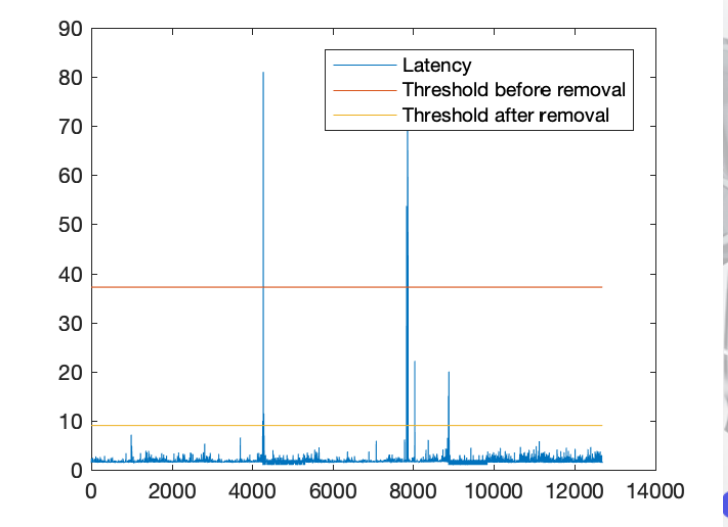
$$s_w = \sum_{k=1}^{i} (x_k - \bar{x}_l)^2 + \sum_{k=i+1}^{n} (x_k - \bar{x}_r)^2$$



Removing abnormal values (cont.)

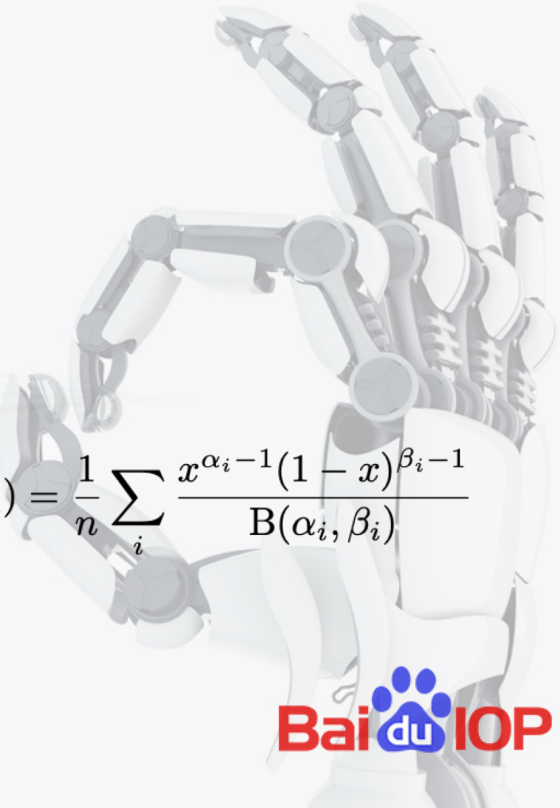
- Search for downward spike in the position-scatter plot
  - Remove values at the right side of the spike





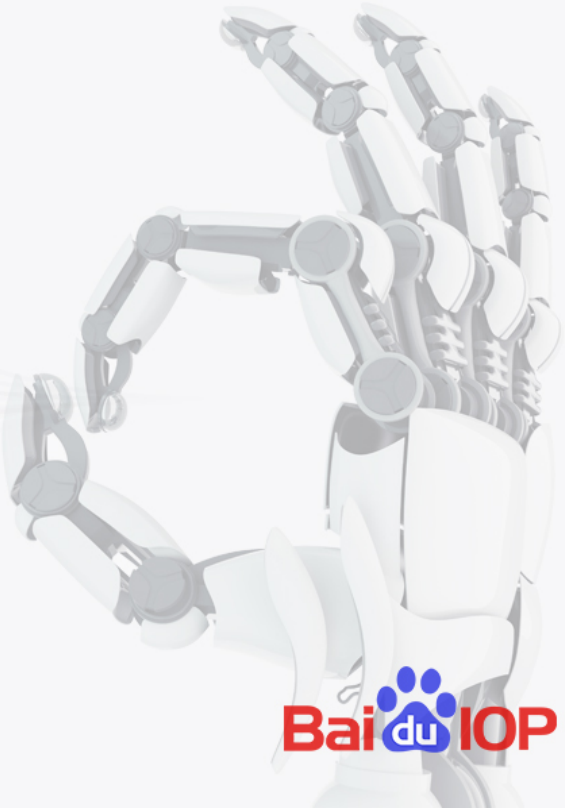
Saturation

- Utilization of resources
  - CPU, Memory, Disk, Network
  - In the form of ratio/percentage
- Alert for overload
- Alert for unexpected resource consumption
  - Similar to latency alerting: KDE
  - Gaussian kernels fail to ensure thresholds in $[0, 1]$
  - Beta distribution as kernels $f(x) = \frac{1}{n}\sum \mathrm{Beta}(x; \alpha_i, \beta_i) = \frac{1}{n}\sum_i \frac{x^{\alpha_i - 1}(1-x)^{\beta_i - 1}}{\mathrm{B}(\alpha_i, \beta_i)}$
  - $x_i = \frac{\alpha_i - 1}{\alpha_i + \beta_i - 2} \quad \alpha_i = cx_i + 1 \quad \beta_i = c(1 - x_i) + 1$
  - Empirically $c \in [100, 200]$

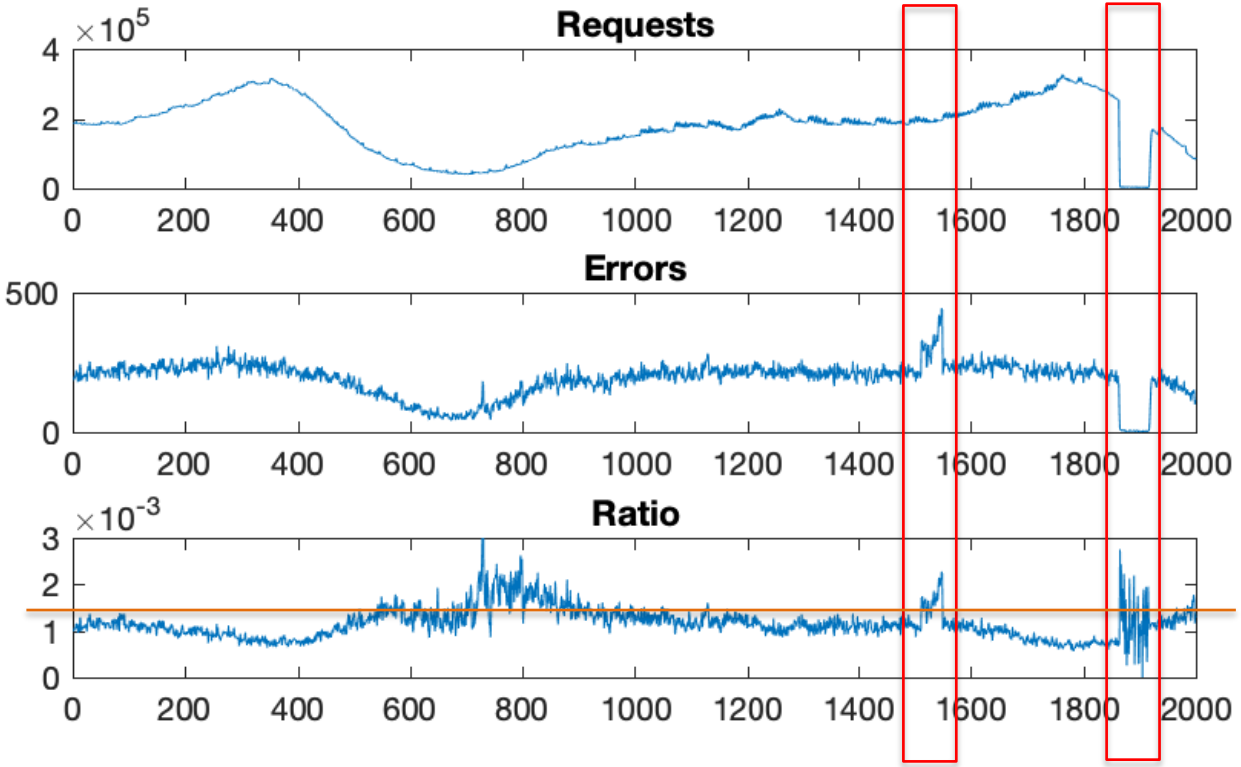


Errors

- The question
  - x requests, y errors, is y too large?
- Approaches
  - Thresholding on absolute value
    - Larger x leads to larger y
    - Different thresholds for different signals
  - Thresholding on ratio $r = y/x$
    - Variance of r is larger with smaller x



Problems with ratio



Solution: binomial distribution

- y follows binomial distribution with parameters x and p_0
  - Coin flipping: each request would have a chance of p_0 to end as an error

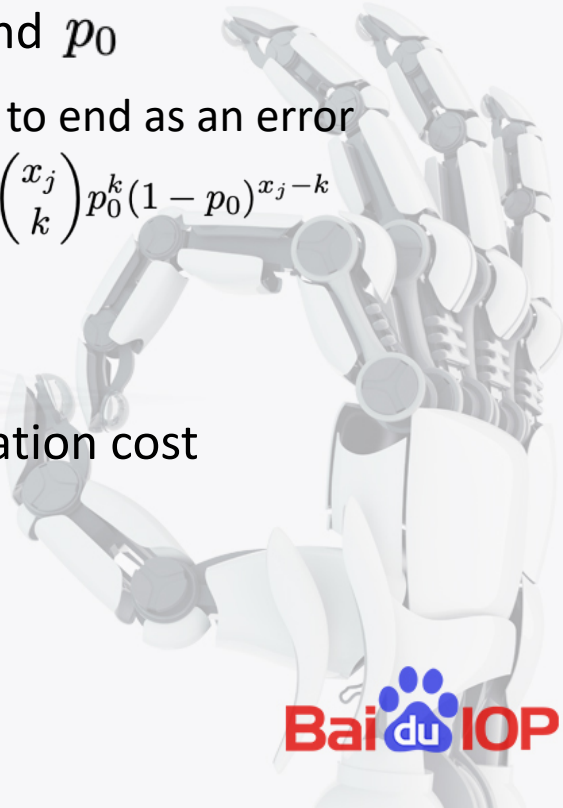
$$P\{y; x, p_0\} = \binom{x}{y} p_0^y (1-p_0)^{x-y} \qquad P\{y > y_j; x_j, p_0\} = \sum_{k=y_j+1}^{x_j} \binom{x_j}{k} p_0^k (1-p_0)^{x_j-k}$$

  - Estimate p_0 with history data

$$p_0 = \frac{\sum_i y_i}{\sum_i x_i}$$

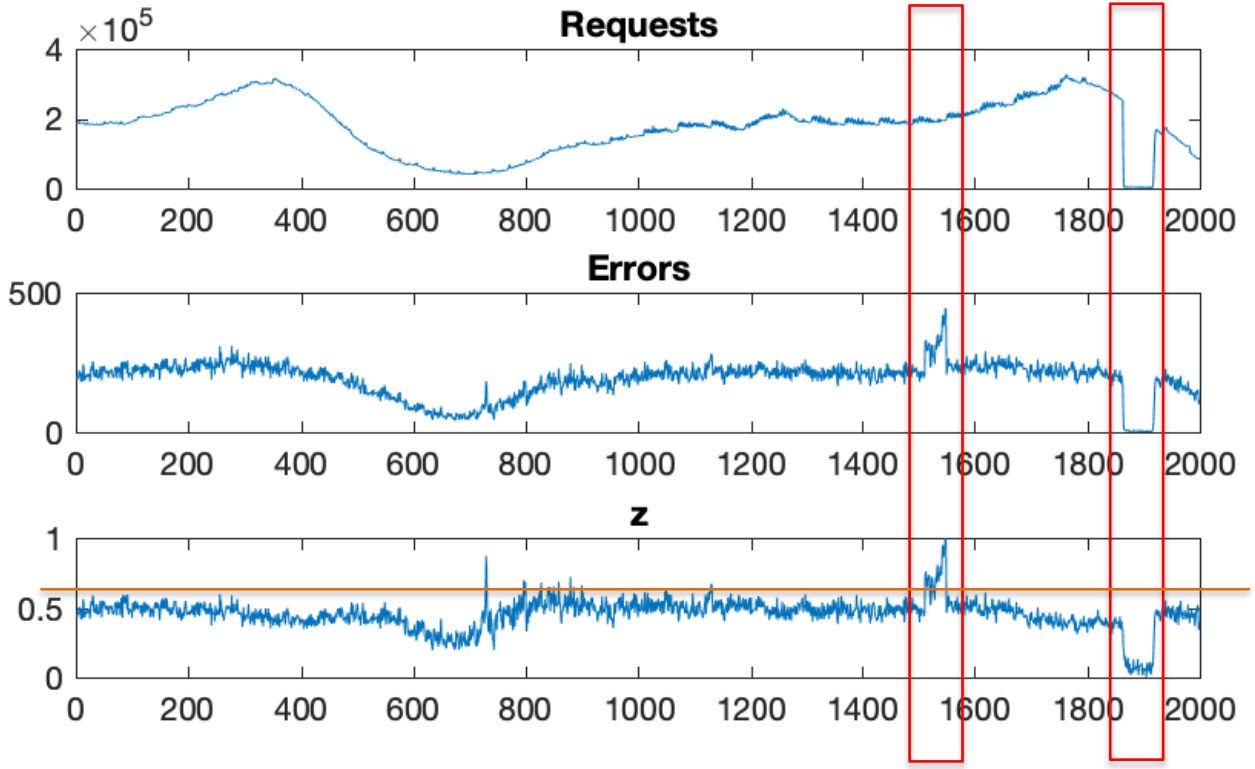
- Estimate by Gaussian distribution to reduce computation cost
  - Proportion z-test

$$z = \frac{\frac{y_j}{x_j} - p_0}{p_0(1-p_0)} \sqrt{x_j}$$



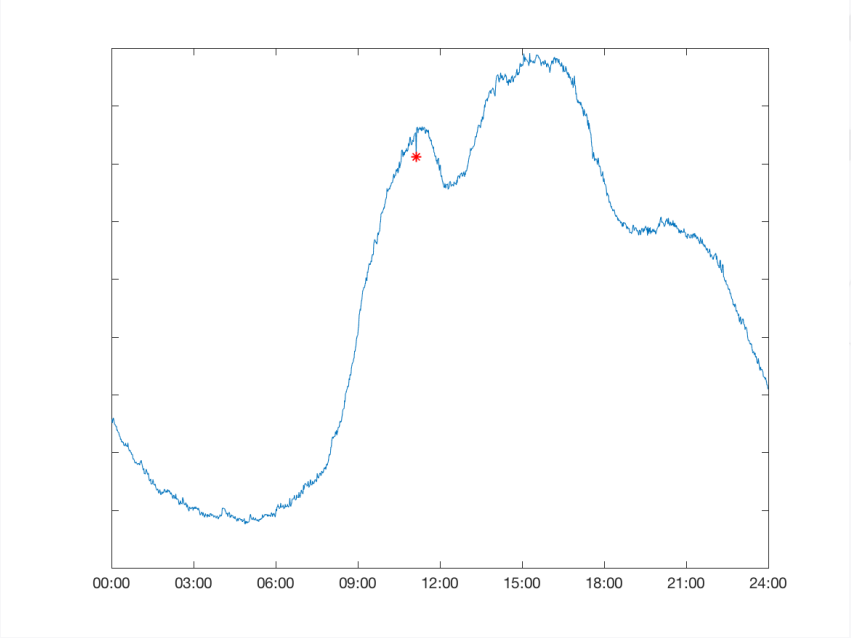
Baidu IOP

Result of proportion z-test



Traffic

- # of requests per interval
- Need to predict expected values
  - Daily fluctuation
  - Drop is more severe than rise
- Basic idea
  - Predict
  - Detect based on prediction and observation



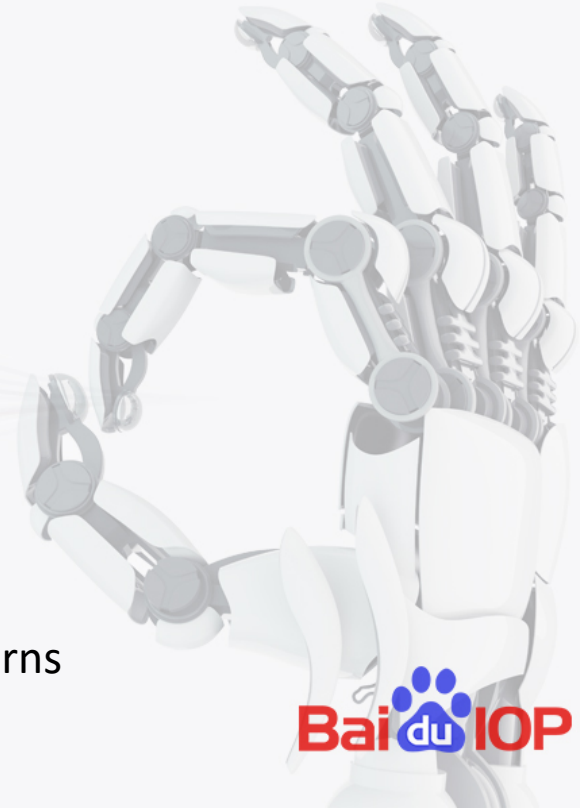
Baidu IOP

Prediction

- Moving average
  - Same weight in window
- Exponential smoothing
  - Larger weight on new data
- Robust linear regression
  - Linear assumption on short segment
- Variational auto encoding (VAE)
  - Non-linear assumption + long segment
- Holt-Winters
  - Linear trend + periodic pattern
- Periodic pattern mining
  - Multiple daily patterns

Smoothing: smooth baseline + noises

Pattern: Periodic patterns



Baidu IOP

Robust linear regression

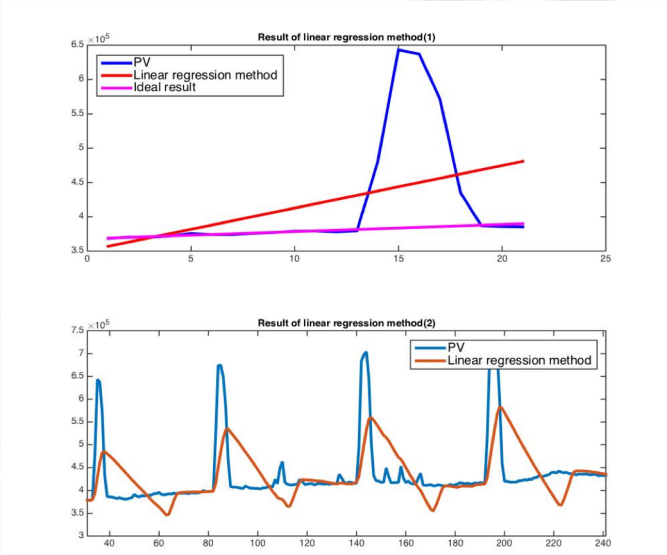
- Assumptions
  - Observed traffic $y_t = \hat{y}_t + \epsilon_t$
  - Smooth baseline $\hat{y}_t = k \times t + b$
    - Linear in short segments
- Linear regression

$$k^*, b^* = \mathrm{argmin}_{k,b} \sum_t (k \times t + b - y_t)^2$$

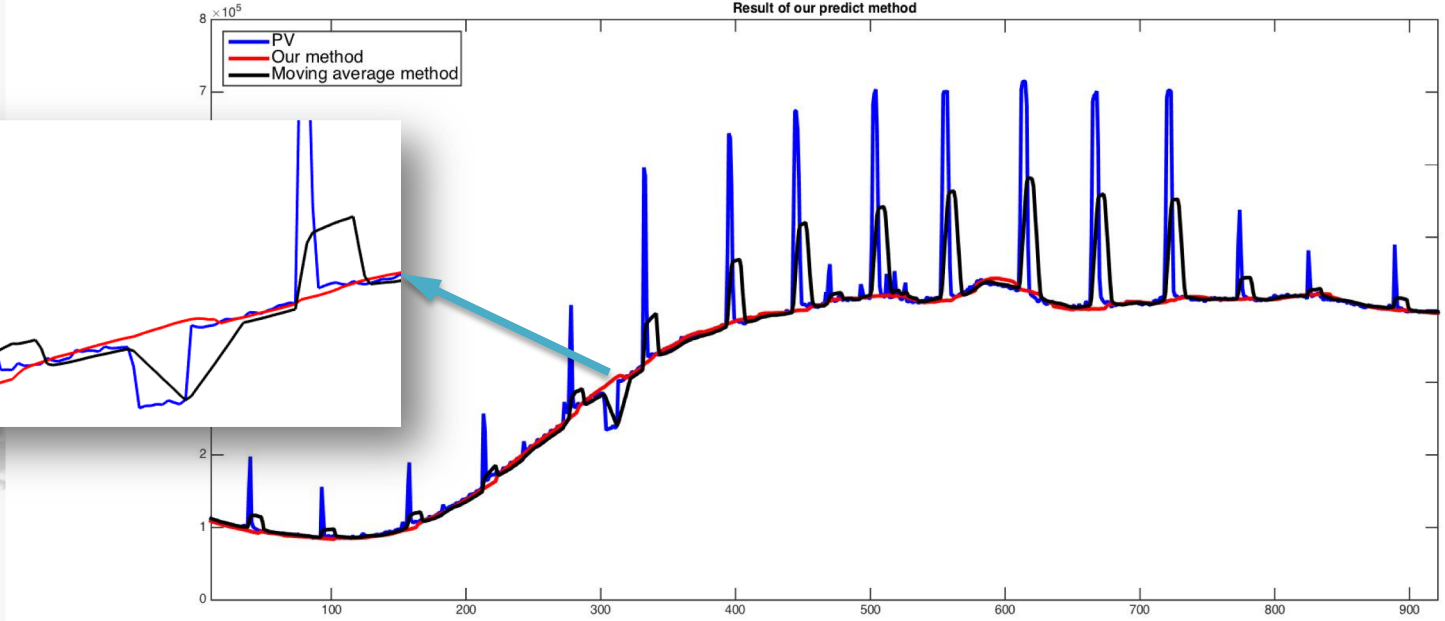
  - Susceptible to abnormal points
- Robust linear regression

$$k^*, b^* = \mathrm{argmin}_{k,b} \sum_t |k \times t + b - y_t|$$

  - Resistant to abnormal points

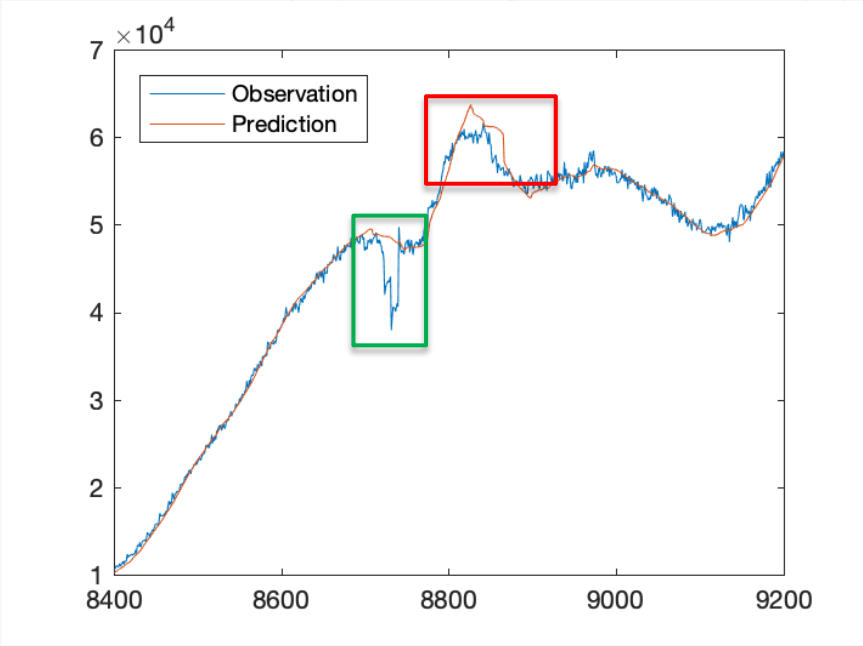


Robust linear regression (result)



The slow catch-up effect

- Resistant to abnormal points means slow catch up
  - on abnormal points
  - on sharp turns
- Non-linear regressions?

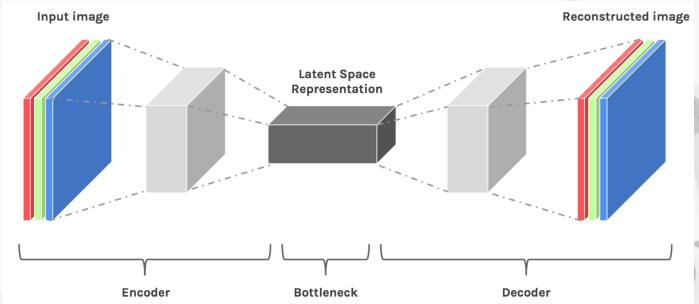


Baidu IOP

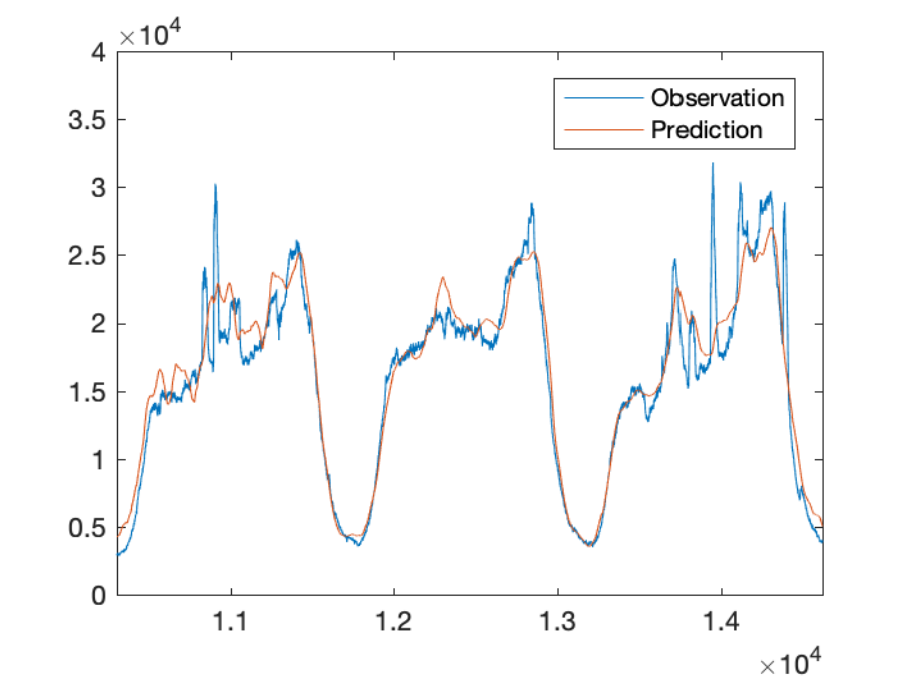
Variational auto-encoder

- Neural network
  - Input layer: signal segments
    - E.g. 2000 points
  - Hidden layer: “features” of input signal segments
    - E.g. 1000 vertices
  - Output layer: reconstructed “signal segments”
    - Mean and variance
- Prediction value = mean of the output layer
- Width of the hidden layer
  - Large → encode more sharp turns
  - Small → produce smoother output

- Paper
  - “Donut”, Xu et al. WWW 2018

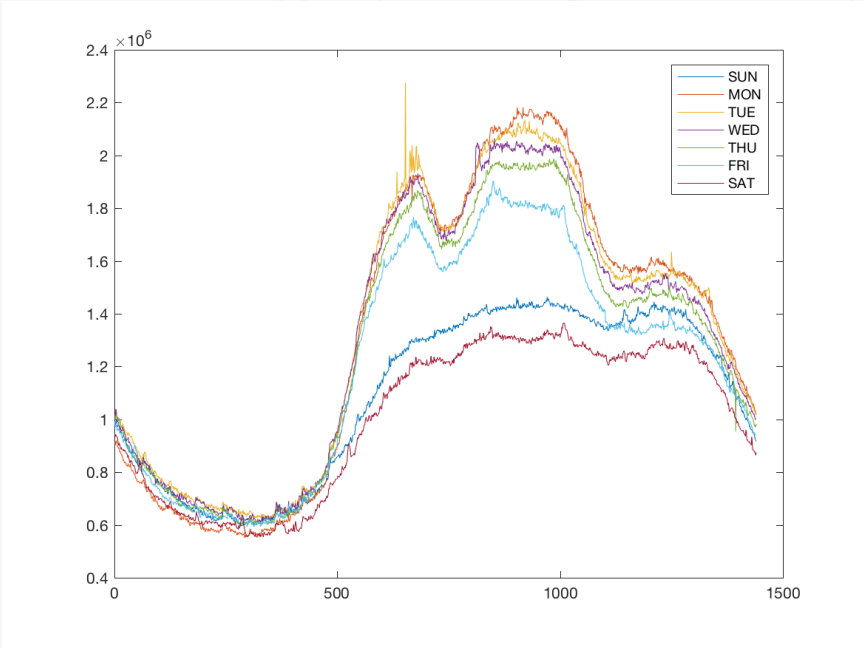


Variational auto-encoder



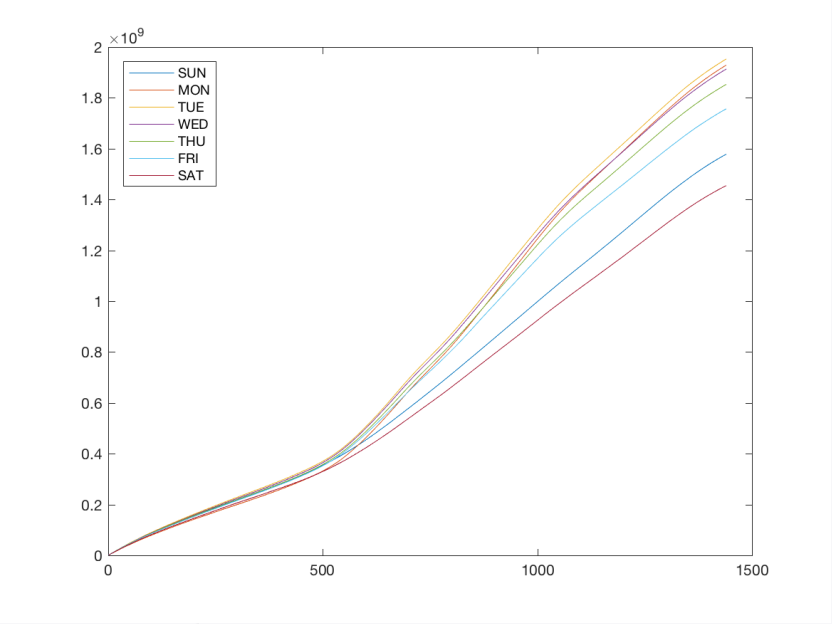
Periodic pattern mining

- Small and long drops
- Multiple patterns
  - Workdays / weekends / holidays
- Level drift
- Idea
  - Shape extraction
  - Level expansion

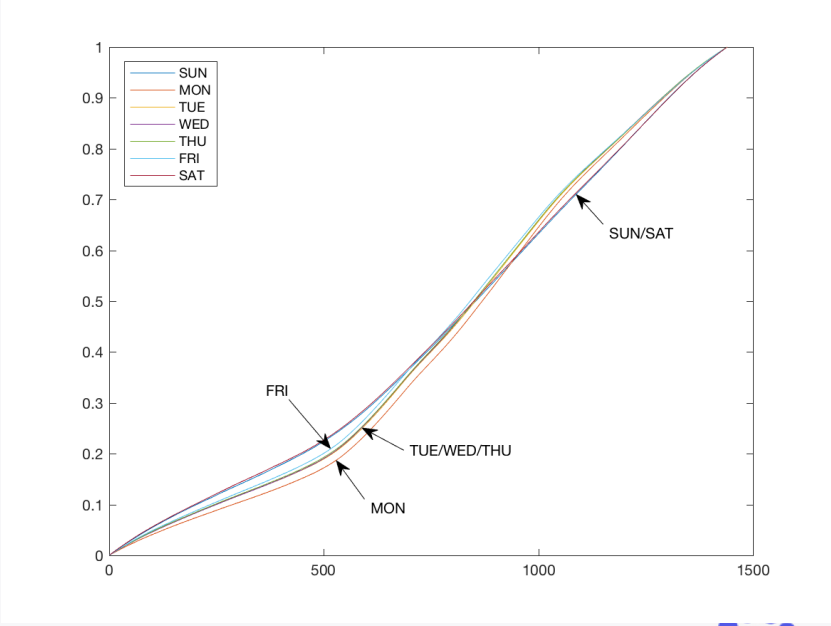


Shape extraction

- Cumulative sum



- Normalize

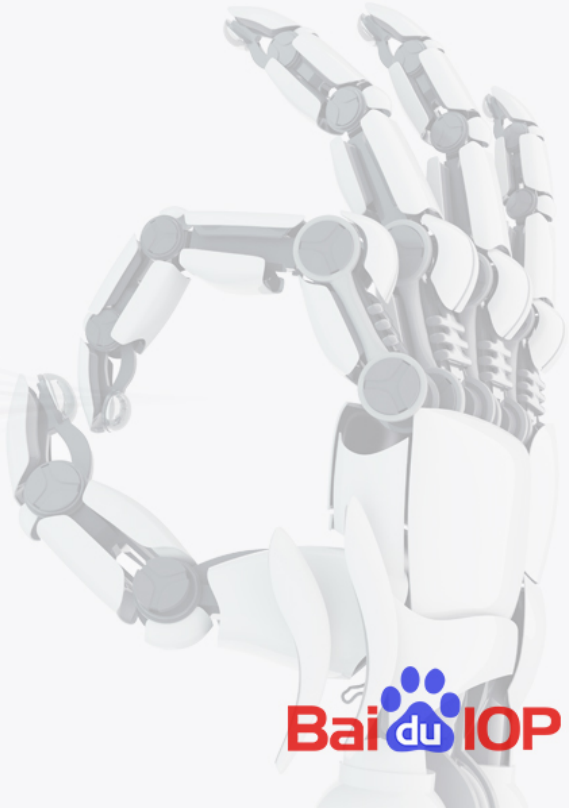


Shape extraction: the math

- Multinoulli distribution
  - T bins per day
    - Sampling interval = 1min → $T = 1440$
  - N requests per day
    - Put N requests to T bins based on a random distribution
  - The normalized curve is the CDF of the distribution
- Shape similarity = distribution similarity
  - Measurement: Kolmogorov-Smirnov test

$$D = \sup_x |F_1(x) - F_2(x)|$$

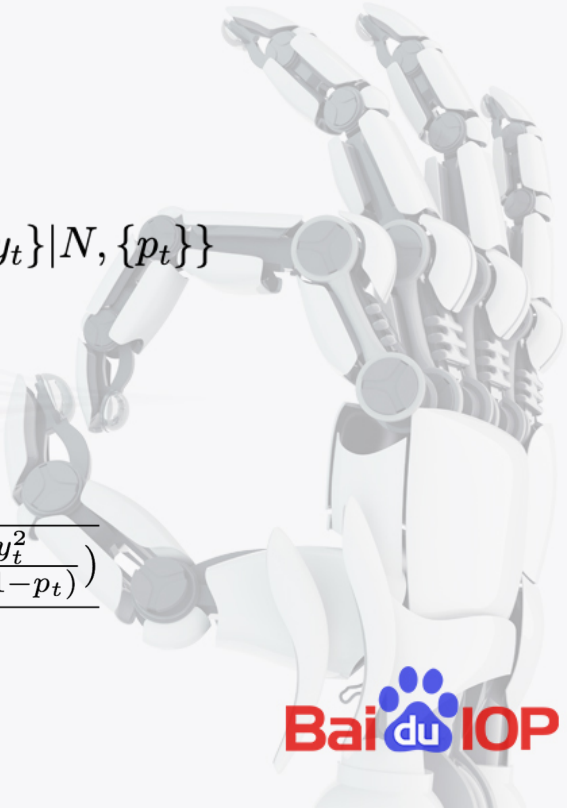
- Clustering: DBSCAN
  - DBSCAN: remove outlier days
  - Hierarchical clustering



Level expansion

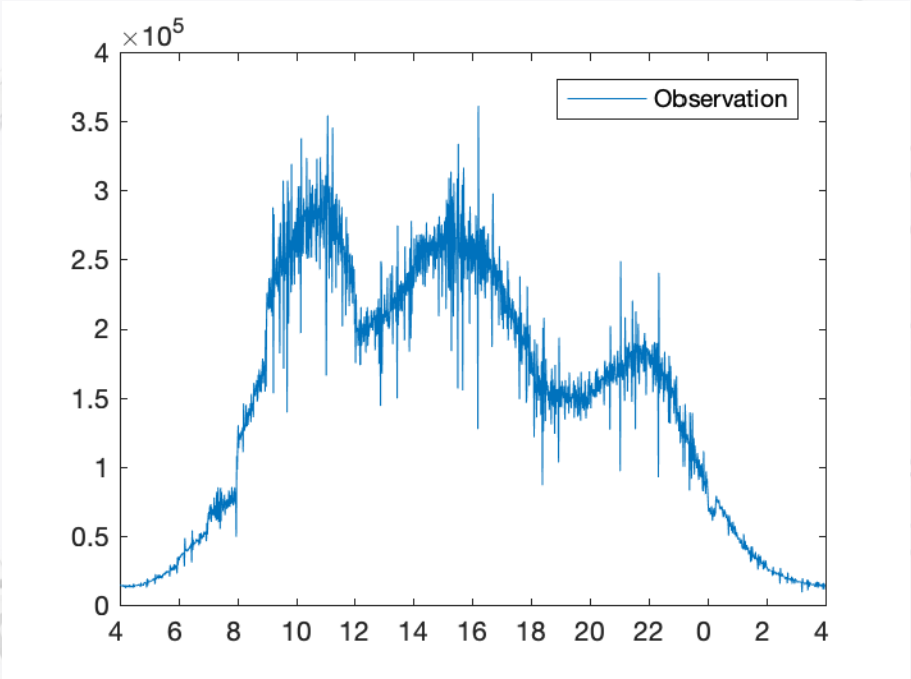
- Given a prefix of observations $\{y_t\}$
- And the shape $\{p_t\}$
  - Average of a cluster
- Find N to maximize the likelihood $N^* = \underset{N}{\operatorname{argmax}} P\{\{y_t\}|N, \{p_t\}\}$
- Multinomial distribution
  - Susceptible to abnormal values $N^* = \frac{\sum_t y_t}{\sum_t p_t}$
- T independent binomial distributions

$$N^* = \frac{-|\{y_t\}| + \sqrt{|\{y_t\}|^2 + 4(\sum_t \frac{p_t}{1-p_t})(\sum_t \frac{y_t^2}{p_t(1-p_t)})}}{2\sum_t \frac{p_t}{1-p_t}}$$

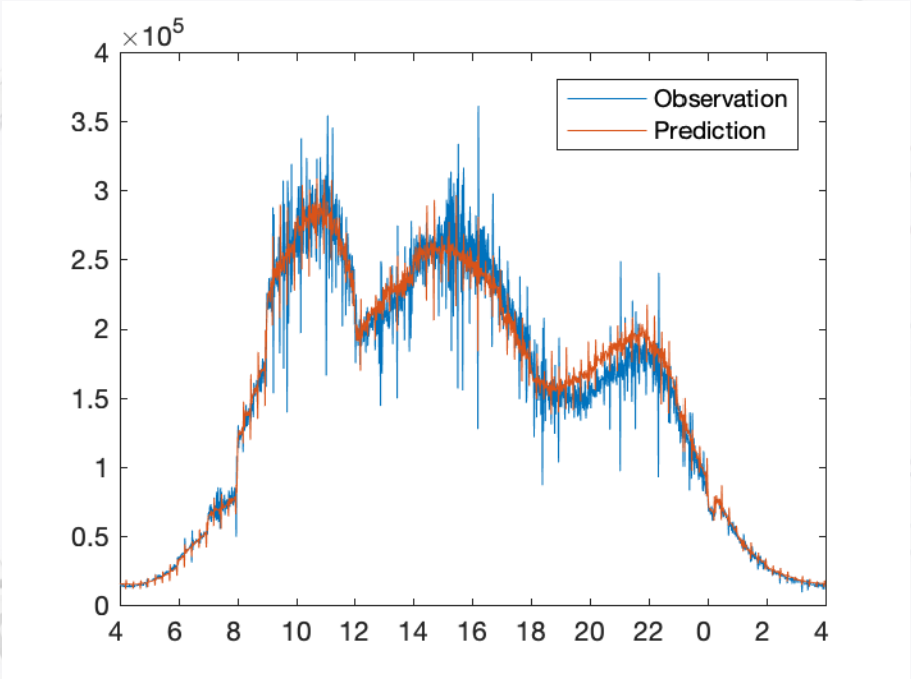


Baidu IOP

Periodic prediction



Periodic prediction



Anomaly detection for traffic

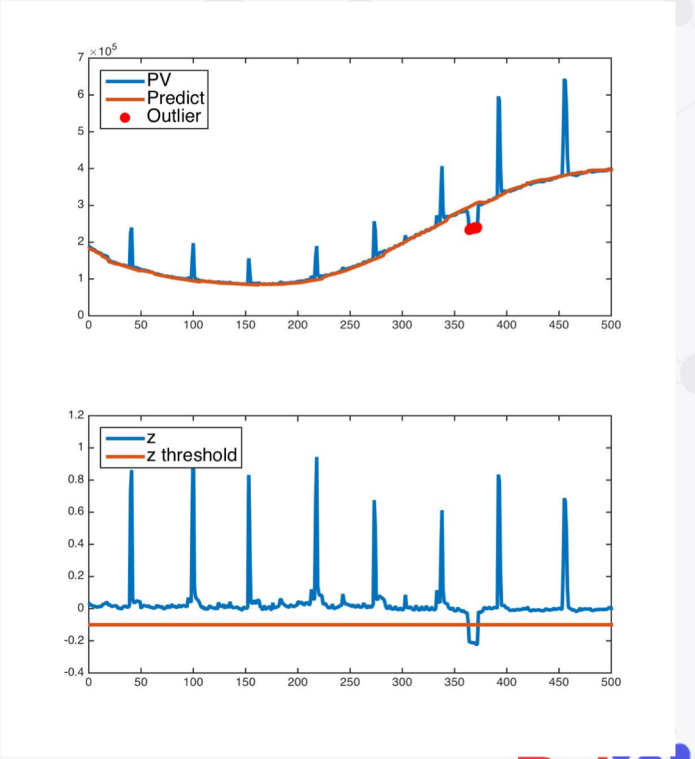
- Observation y_t
- Prediction $\hat{y}_t$
- Poisson distribution

$$\text{Poisson}(k;\lambda) = \frac{\lambda^k}{k!}e^{-\lambda}$$

$$\lambda = \hat{y}_t \qquad k = y_t$$

- Estimated by Gaussian distribution

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



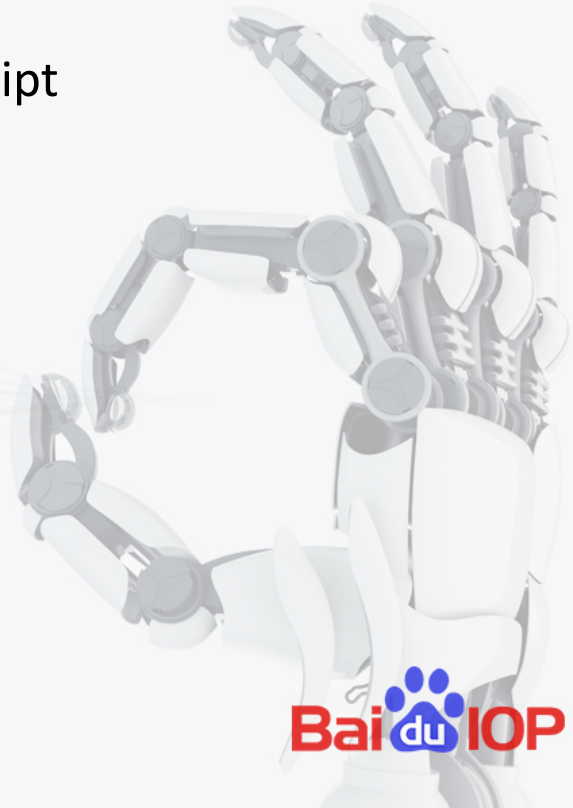
Are requests really independent?

- No, they are dependent!
  - One click generates multiple requests by JavaScript
  - Threshold of z depends on the signal
- How to get around
  - Observed traffic $\{y_t\}$ and true traffic $\{x_t\}$
  - Assume $y_t = c \times x_t$
  - Then $z_t^y = \frac{y_t - \hat{y}_t}{\sqrt{y_t}} = \frac{cx_t - c\hat{x}_t}{\sqrt{cx_t}} = \sqrt{c} z_t^x$
  - $z_t^x \sim \mathcal{N}(0,1) \longrightarrow z_t^y \sim \mathcal{N}(0,c)$
  - Median absolute deviation to estimate robustly

$$\tilde{z} = \text{median}(\{z_t\})$$

$$\text{MAD}(\{z_t\}) = \text{median}(\{|z_t - \tilde{z}|\})$$

$$\text{MAD}(\{z_t^y\}) = \text{median}(\{z_t^x\})\sqrt{c} = 0.6745\sqrt{c}$$



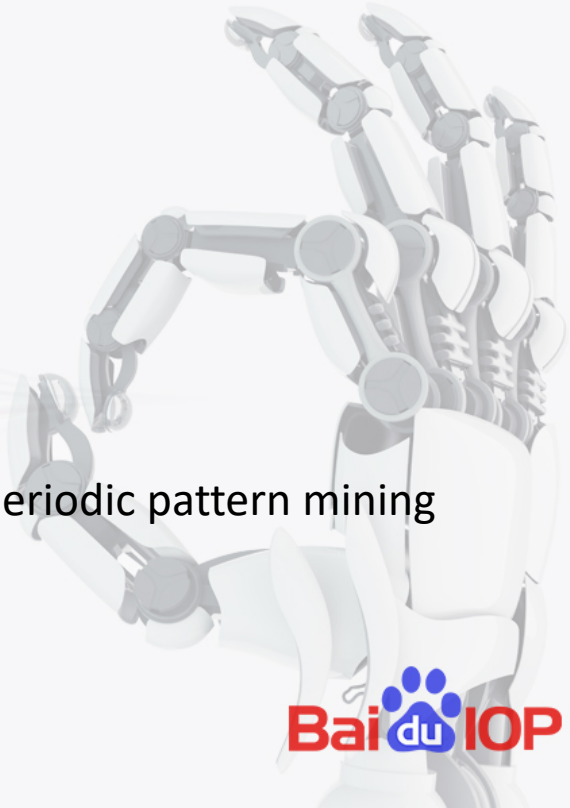
Baidu IOP

Applications

Scenario	Precision	Recall	Signal type
Search engine traffic	100%	100%	Traffic
Feed traffic	100%	100%	Traffic
Feed click ratio	100%	100%	Errors
Advertisement income	95%	95%	Traffic
External network connectivity monitoring	94%	96.3%	Errors + Traffic
Datacenter network connectivity monitoring	92.3%	99.2%	Errors

Conclusion

- Statistical analysis + Unsupervised learning
- Latency
  - KDE, automatic removal of abnormal values in history
- Saturation
  - KDE, Beta kernel
- Errors
  - Binomial distribution, proportion z-test
- Traffic
  - Prediction algorithms: robust linear regression, VAE, periodic pattern mining
  - Poisson distribution, MAD



Baidu IOP

THANK YOU !

BAIDU@IOP出品