SAIT AIRC Invited Seminar II - Industrial AI & Application in Manufacturing

Sunghee Yun

Co-founder / VP - AI Technology & Product Strategy

Erudio Bio, Inc.

Machine Learning algorithms for TS data

TS data

- definition of times-series:

$$x : T \to \mathbf{R}^n \text{ where } T = \{\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots\} \subseteq \mathbf{R}$$

- example: material measurements: when $n = 4$

$$x_t = \begin{bmatrix} \text{thickness}(t) \\ \text{temperature}(t) \\ \text{pressure}(t) \\ \text{feature\_size}(t) \end{bmatrix}$$

- for (semi-)supervised learning, we assume two time series

$$x : T \to \mathbf{R}^n \text{ and } y : T \to \mathbf{R}^m$$

Time index

- time index does not have to be *time* index

- more general defintion

$$x : T \to \mathbf{R}^n \text{ where } T = \{\ldots, s_{-2}, s_{-1}, s_0, s_1, s_2, \ldots\}$$

  where $\cdots < s_{-1} < s_0 < s_1 < \cdots$ defines *an* ordering ($e.g.$, total ordering)

- for example, x_s and $y(s)$ can represent the features and target values for a processed material ($e.g.$, wafer in semiconductor manufacturing), s, where they are not measured at the same time

- (throughout this talk, though, we will use time-index)

Supervised learning for TS

- canonical problem:

$$\begin{array}{rl} \text{(stochastically) predict} & y_{t_k} \\ \text{given} & x_{t_k}, x_{t_{k-1}}, \ldots, y_{t_{k-1}}, y_{t_{k-2}}, \ldots \end{array}$$

- various methods exist - depend assumptions on data
  - $e.g.$, if assume joint probability distribution, optimal solutions exist, $e.g.$, LSE sense

- however, will *not* make such assumptions

Problem formulation

- canonical problem formulation:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K} w_{K-k}\, l(y_{t_k}, \hat{y}_{t_k}) \\ \text{subject to} & \hat{y}_{t_k} = g_k(x_{t_k}, x_{t_{k-1}}, \ldots, y_{t_{k-1}}, y_{t_{k-2}}, \ldots) \end{array}$$

  where

  - $g_1, g_2, \ldots : \mathcal{D} \to \mathbf{R}^m$ - optimization variables
  - $\mathcal{D} = \mathbf{R}^n \times \mathbf{R}^n \times \cdots \mathbf{R}^m \cup \{\text{null}\} \times \mathbf{R}^m \cup \{\text{null}\} \times \cdots$ - domain of g_k
  - $l : \mathbf{R}^m \times \mathbf{R}^m \to \mathbf{R}_+$ - loss function
  - w_i - (decreasing) weight on loss

- no label is given for some k, *i.e.*, $y(t_k) = \text{null}$

ML solution candidates

- ignore temporal dependency - $\hat{y}_{t_k} = g(x_{t_k})$
  - supervised learing such as DL ($e.g.$, MLP), decision trees
  - classiscal statistical learning such as lasso, ridge regression, partial least squares
  - boosting algorithms such at `XGBoost`
- consider temporal dependency - sequential MLs
  - RNN-base: LSTM, GRUs
  - attention mechanism, $e.g.$, classical attention-type, Transformer-type, $etc.$

Credibility intervals for TS value prediction

- prediction of uncertainty of prediction

- every point prediction is wrong!

  - $\mathbf{P}(\hat{y}_t = y_t) = 0$

- reliability of prediction matters
  - *none* literature deals with this (properly)

- critical for our customers, $e.g.$, *downstream applications*
  - if used for APC, need to know when it should be used
  - sometimes, *more crucial than algorithm accuracy*

Find credibility intervals

- multiple criteria
  - probability of true value falling into an interval: for fixed $a > 0$

$$\mathbf{P}(|Y_k - \hat{Y}_k| < a) = \mathbf{P}(Y_k \in (\hat{Y}_k - a, \hat{Y}_k + a))$$

  - predictive distribution size: find $a > 0$ such that

$$\mathbf{P}(|Y_k - \hat{Y}_k| < a) = 90\%,\ e.g.$$

  - distribution of Y_k: find PDF of Y_k
- out solution - Bayesian inference
  - given initial distribution or prior, p
  - update p with new data using Bayesian inference

Bayesian approach for credibility intervals

- assume conditional distribution ith predictor parameterized by $\theta_{i,k} \in \Theta$

$$p_{i,k}(y(t_k)|x_{t_k}, x_{t_{k-1}}, \ldots, y(t_{k-1}), y(t_{k-2}), \ldots) = p_{i,k}(y(t_k); x_{t_k}, \theta_{i,k})$$

  - depends on prior & current input, $i.e.$, $\theta_{i,k}$ & x_{t_k}

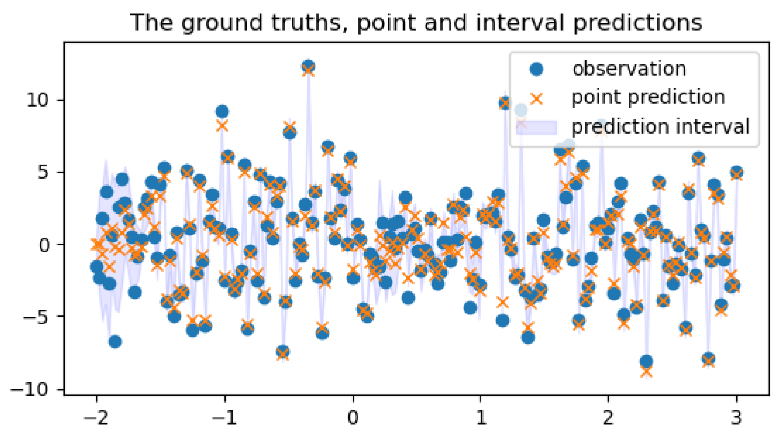
- update $\theta_{i,k+1}$ from $\theta_{i,k}$ after observing true $y(t_k)$ using Bayesian rule

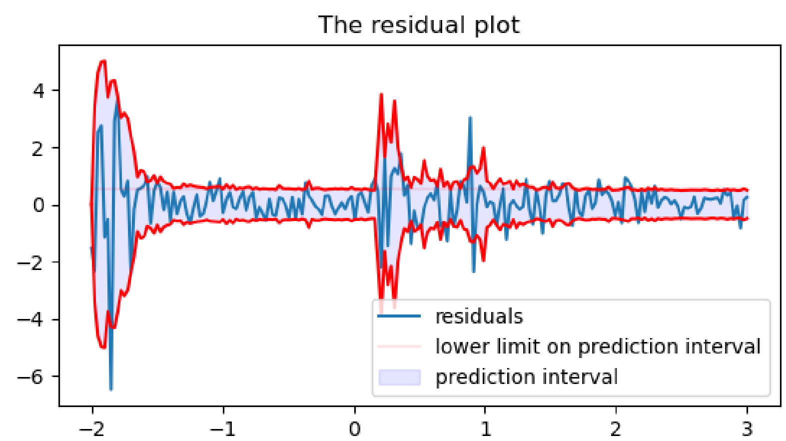
$$p(w; \theta_{i,k+1}) := p(w|y(t_k); x_{t_k}, \theta_{i,k}) = \frac{p(y(t_k)|w, x_{t_k})p(w; \theta_{i,k})}{\int p(y(t_k)|w, x_{t_k})p(w; \theta_{i,k})dw}$$

- *if* $p(\cdot; \theta)$ is conjugate prior, can update $\theta_{i,k}$ very efficiently in online manner *within fraction of milliseconds*

Real application

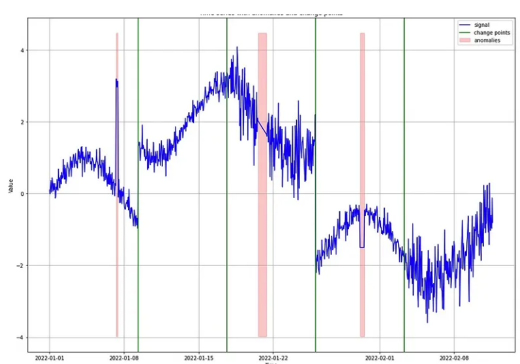
- observe
  - initially predictor *not sure* about its prediction
  - after a while, the *credibility interval (CI)* converges
  - when shift happens, CI increases (as it should be)
- this information *crucial for downstream applications*, $e.g.$, process control

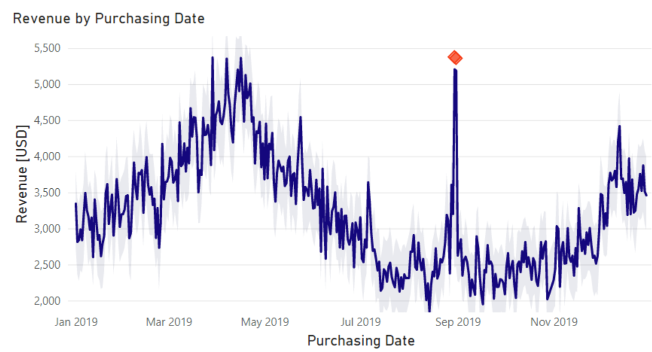




TS anomaly detection problems

- types of anomaly detection problems - given $x : T \to \mathbf{R}^n$
  - point anomaly - find x_{t_k} considerably different from other data
  - segment anomaly - find k_1 and k_2 s.t. TS segment $x_{t_k}|_{k=k_1}^{k_2}$ is considerably different from other data
  - sequence anomaly - given $x^1, \ldots, x^n : T \to \mathbf{R}$, find x^i considerably different from other TSs





TS segment anomaly detection algorithm

- use classification - given $x_{t_j}|_{j=k-l+1}^{k}$, $i.e.$, segment of length, l
  - training:
    - one classifier, c, and, p feature extractors, f_i
    - for each k
      - extract p features using extractors - $y_{i,k} = f_i \left(x_{t_j}|_{j=k-l+1}^{k} \right)$
      - train the classifier, c, with $(y_{1,k}, 1)$, $(y_{2,k}, 2)$, . . . , $(y_{p,k}, p)$, as training data
  - inferencing:
    - given new segment $x_{t_j}|_{j=k-l+1}^{k}$, apply c to the extracted features, $y_{i,k}$
    - if substantically different from $(1, 2, \ldots, p)$, it is anomaly
      - "difference" quantified by some *anomaly score*, $e.g.$, KL divergence or entropy

What really matters in productionization

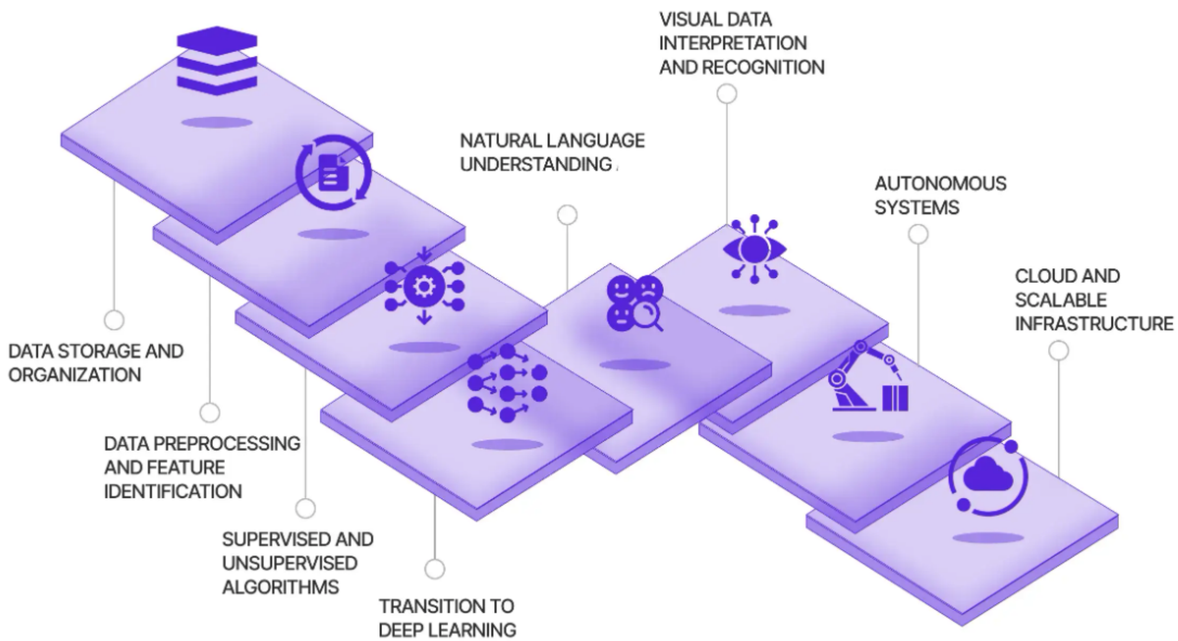
List of efforts required

- MLOps - for CI/CD
- data preprocessing - missing values, inconsistent names, difference among different systems
- feature extraction & selection
- monitoring & retraining
- notification, via messengers or emails
- main line merge approvals by humans
- data latency, data reliability, & data availability

Manufacturing AI Software System Development

Manufacturing AI Software System

- data, data, data! – store, persist, retrieve, data quality
- seamless pipeline for development, testing, running deployed services
- development envinroment should be built separately



Thank You! - sunghee.yun@erudio.bio