Collaborative Anomaly Detection on Blockchain from Noisy Sensor Data

Tsuyoshi (“Ide-san”) Ide (email: tide@us.ibm.com)
IBM T. J. Watson Research Center
IBM Research

Agenda

▪ Background: towards collaborative learning platform
▪ Problem setting
▪ Multi-task unsupervised learning for anomaly detection
▪ Updating global- and local state variables
▪ Concluding remarks
Development of Blockchain: From currency transfer to general business transaction

- **Blockchain 1.0: Bitcoin**
  - Specifically designed for currency transfer
  - Account identity is protected but transactional records are public
  - Verifying a transaction is trivial: just check the account balances
  - Futuristic consensus algorithm (“proof-of-work”) that lacks deterministic guarantees

- **Blockchain 2.0: Smart-Contract-enabled transactional platform**
  - Designed to be able to handle “general” business transactions
  - Public or semi-closed (membership, permissioned)
  - Verifying a transaction is not straightforward
  - Traditional consensus algorithm (e.g. PBFT) is typically used
Using Blockchain for IoT applications

- **Two major data types**
  - Traceability data: categorical, deterministic, may be incorrect but noise-free
    - Parts, inventories, work orders, SCM, CRM, etc.
    - Many attempts: food traceability (Walmart), shipping goods traceability (Maersk), etc.
  - Sensor data: real-valued, stochastic noise
    - Raw sensor signals such as temperature, pressure

- **Expectations towards novel business applications**
  - Decentralized SCM
  - Utility-based pricing of resources (sensors, algorithms, etc.)
  - etc.
Redefining Blockchain as collaborative learning platform

- Most of the existing Blockchain-based IoT applications are sort of static data storage. We want to go one step further

- “Blockchain 3.0”: Platform for collaborative learning
  - A platform to create new business insights through knowledge sharing among multiple parties in a Blockchain-specific way

- Key question: how can we create a new business value through data exchange on Blockchain?
Background: towards collaborative learning platform

Problem setting

Multi-task unsupervised learning for anomaly detection

Updating global- and local state variables

Concluding remarks
Sharing sensor data on Blockchain: Challenges

- Challenges to put sensor data onto Blockchain networks
  - Validation
  - Consensus

- Validation
  - What if a new observation shared is incorrect? ✓ This is a general issue for most of smart contracts  
  - Need automatic down-weighting mechanism for less informative observations

- Consensus
  - Most of the existing Blockchain system do NOT assume noisy sensor signals ✓ (out of the scope of this work)
Collaborative condition-based monitoring of industrial assets:

Problem setting

- System: distributed competing industrial assets
  - Mining tools, manufacturing tools, etc.
  - They want to keep their data privately, but they want to exploit other data

- Data: real-valued multi-variate noisy sensor signals
  - e.g. temperature, pressure, ...

- Goal: Collaboratively build an anomaly detection model through Blockchain transactions
Collaborative condition-based monitoring of industrial assets: Requirements

- Capable of handling noisy data
- Capable of taking an optimal balance between individuality vs. commonality of the assets
- Capable of preserving data privacy
  - Assumption of competing assets: Do not want to share their own data but want to exploit other one's data
  - Happens when assets belong to different companies
Collaborative condition-based monitoring of industrial assets: Approach overview

- Capable of handling noisy data
  
  Probabilistic sample weighting scheme

- Capable of taking an optimal balance between individuality vs. commonality of the assets
  
  Multi-task learning for anomaly detection

- Capable of preserving data privacy
  
  Separation of global- and local state variables
IBM Research

Agenda

- Background: towards collaborative learning platform
- Problem setting
  - Multi-task unsupervised learning for anomaly detection
  - Updating global- and local state variables
- Concluding remarks
Definition of multi-task learning:
- A machine learning algorithm is said to be multi-task learning if the model consists a local part and a global part:

  \[(\text{prediction model}) = (\text{global/shared part}) + (\text{local/individual part})\]

A Smart Contract is characterized by a pair of (state variable, algorithm)

We map an MTL-based anomaly detection model [Ide+ ICDM 17] onto a Smart Contract by properly defining state variables.
Learn probability density under normal condition. Define anomaly score as deviation from the normal state.

\[ \{ \mathbf{x}^1(n) \in \mathbb{R}^M \} \quad \text{Data} \]

\[ \{ \mathbf{x}^S(n) \in \mathbb{R}^M \} \]

\[ \text{multi-task learning (MTL)} \]

\[ p^1(\mathbf{x}^1 | \mathcal{D}) \quad \text{Prob. density} \]

\[ \ln p^1(\mathbf{x}^1 | \mathcal{D}) \quad \text{Anomaly score} \]

\[ p^S(\mathbf{x}^S | \mathcal{D}) \]

\[ \ln p^S(\mathbf{x}^S | \mathcal{D}) \]

\[ = \sum_{k=1}^{K} \pi_k^S \mathcal{N}(\mathbf{x}^S | \mu_k^S, (\Lambda_k^S)^{-1}) \]

\[ \text{all data} \]

\[ \text{Client 1 (in Singapore)} \]

\[ \vdots \]

\[ \text{Client s} \]

\[ \vdots \]

\[ \text{Client S (in New York)} \]
Each model is represented as a linear combination of shared dependency models.

Client 1 (in Singapore)

\[ \ldots \]

Client s

\[ \ldots \]

Client S (in New York)

Local state variable

probability

\[ \ldots \]

Global state variable
(or pattern dictionary)

\[ \text{dependency model } 1 \]

\[ \text{dependency model } 2 \]

\[ \ldots \]

\[ \text{dependency model } K \]

\[ \times \]

Monitoring model for client 1

\[ \ldots \]

Monitoring model for client S
Learning model parameters from data

- Employ an EM algorithm for model inference
  - See the text for the detail

- The resulting algorithm is **iterative**:

  ![Diagram showing iterative process]

  - Local state variable update
  - Global state variable update
  - Iteration
  - Shared pattern dictionary
Agenda

- Background: towards collaborative learning platform
- Problem setting
- Multi-task unsupervised learning for anomaly detection
  - Updating global- and local state variables
- Concluding remarks
Local and global state variables are iteratively updated as Smart Contract

- Anomaly score function is written in terms of global and local state variables

$$a^s(x^s \mid \theta_{gl.}, \theta_{lo.}) = -\ln p(x^s \mid \theta_{gl.}, \theta_{lo.})$$

Client side

Global state variable update

Shared pattern dictionary

Endorser (consensus node) side
The derived EM algorithm is naturally mapped into the local-global update framework.

- Anomaly score function is written in terms of global and local state variables:

\[ a^s(x^s \mid \theta_{\text{gl.}}, \theta_{\text{lo.}}) = -\ln p(x^s \mid \theta_{\text{gl.}}, \theta_{\text{lo.}}), \]

**Client side**

\[ N_k^s \leftarrow \sum_{n=1}^{N^s} r_k^{s(n)} \]

\[ m_k^s \leftarrow \sum_{n=1}^{N^s} r_k^{s(n)} x^{s(n)} x^{s(n)\top} \]

\[ \pi_k^s \leftarrow \frac{N_k^s}{\sum_{l=1}^{K} N_l^s} \]

**Endorser (consensus node) side**

\[ N_k \leftarrow \sum_{s=1}^{S} N_k^s \]

\[ \mu_k \leftarrow \frac{1}{\lambda_0 + N_k} \sum_{s=1}^{S} m_k^s \]

\[ \Sigma_k \leftarrow \frac{1}{N_k} \sum_{s=1}^{S} C_k^s + \mu_k \mu_k^\top \]

- Aggregated quantities:
- raw sample
How this algorithm meets the practical requirements

- **Validating transactions for real-valued noisy data**
  - EM algorithm automatically down-weights less informative observations
  - This can be viewed as automated validation of transactions

- **Balancing between individuality vs. commonality**
  - This is the very core concept of multi-task learning

- **Preserving data privacy**
  - Raw data is never shared beyond each client
  - Only aggregated statistics are shared with endorsers (consensus nodes)
Background: towards collaborative learning platform

Problem setting

Multi-task unsupervised learning for anomaly detection

Updating global- and local state variables

Concluding remarks
Conclusion

- We redefined Blockchain network as collaborative learning platform
- We showed that multi-task learning nicely fits the notion of Smart Contract by separating global and local state variables
- As a concrete IoT example, we wrote down an MTL-based dictionary learning algorithm for collaborative condition-based maintenance of industrial assets
Limitations of the current model and our on-going work

- Lack of an explicit consensus building mechanism
  - Traditional Byzantine Fault Tolerance mechanisms are not appropriate to IoT data
    - They implicitly assume categorical and deterministic data
  - Our recent approach has solved this issue

- Lack of theoretical guarantees on privacy preservation
  - We recently developed an improved version that has a mathematical privacy guarantee

- Lack of a realistic business model that motivates companies to participate in this network
  - On-going work is looking at an approach to incentivizing or penalizing clients based on the immutable Blockchain data, depending on contribution to dictionary learning
Thank you!