Recent advances in sensor data analytics

Tsuyoshi Ide ("Ide-san")
PhD, Senior Technical Staff Member
IBM Thomas J. Watson Research Center
Agenda

- General challenges in industrial sensor data analytics

- Solution examples:
  - Change detection using directional statistics
  - Multi-task multi-modal models for collective anomaly detection
  - Tensorial change analysis

- Discussion: deep learning, Blockchain, and future directions
IBM T. J. Watson Research Center
Center for Computational and Statistical Learning

Accelerated breeding of energy crops
Joint modeling/clustering for GWAS
Panicle detection and counting

Lead conversion prediction
Insurance cost prediction & attribution

Fire frontline prediction
Deep Learning model for reservoir simulator

Oil field Planning & Control
Transfer learning for anomaly detection

On-going Agenda

Automated phenotyping
Integrated genotype-phenotype analysis
Accurate metric forecasting

Enterprise revenue forecasting
Yield & production modeling

Fast reservoir modeling
Deep sequential decision making

Industrial collaborative learning

Capabilities

Satellite, drone image data analytics
Structural and Adaptive Learning
Structured time series analysis
Integration of ML/statistics with physics

Deep Reinforcement Learning
Distributed learning and Blockchain

© 2018 IBM Corporation
Machine learning from sensor data is one of the major research focuses

- Anomaly and change detection is a major topic in sensor data analytics
- Recently published two textbooks (in Japanese)
Basics: General problem setting in machine learning

- **Supervised learning**
  - Given a data set
    \[ \{ (x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}) \} \]
  - find the probability distribution of \( y \) given \( x \):
    \[ p(y \mid x) \]

- **Unsupervised learning**
  - Given \( \{ x^{(1)} \ldots, x^{(N)} \} \)
  - find \( p(x) \)

- **Typical assumptions**
  - \( x \) is a vector
  - \( y \) is a scalar
  - Samples are independently and identically distributed (i.i.d.)

- **What makes sensor data analytics interesting?**
General challenges: No “one-size-fits-all” algorithm

- Example in anomaly detection
  - "Happy families are all alike; every unhappy family is unhappy in its own way." - Anna Karenina, Leo Tolstoy

Examples of anomalies:
- Outliers (from i.i.d. samples)
- Outliers (from auto-correlated samples)
- Change points
- Discords

General challenges: Business requirements often drive extensions of existing approaches

- Example: corporate-level asset management with anomaly detection
  - Typically assets are managed as a cohort
    - 10s of off-shore oil production systems
    - 100s of industrial robots
    - 1000s of electric vehicles in a certain area
  - How can we leverage the commonality between assets to build an anomaly detection solution for individual assets?

General challenges: Complex internal structure may exist in one measurement

Example from semiconductor manufacturing (etching)


Each wafer pass is a higher-order tensor, rather than a vector.
General challenges: Ready-to-use solution to your problem might not even exist

- Example: Charge retention (~ battery life) prediction of electric vehicle batteries
  - Depends on the entire history of battery usage
  - Battery usage is represented as a complex trajectory of a multi-dimensional space

- Charge retention prediction task should be formulated as “trajectory regression”

\[ y = f(\text{“trajectory”}) \]

General challenges: Ground truth may not be available. Some degrees of freedom are usually latent

- Example: sensor data of a compressor of oil production system
  - Data taken under a normal operational condition
  - Noisy, nonstationary, heterogeneous, high-dimensional …
- Hard to pinpoint what is indicative of malfunction
General challenges in industrial sensor data analytics

Solution examples:
  - Change detection using directional statistics (Ide et al., IJCAI 17)
  - Multi-task multi-modal models for collective anomaly detection
  - Tensorial change analysis

Discussion: deep learning, Blockchain, and future directions
Continuous operation of conveyor systems is critical in the mining industry

- **Business goal**: Ensure continuous operation of conveyor system ("apron feeder") by detecting early indications of failures

- **Data**: Physical sensor data from conveyors and motors
  - Every several seconds over ~ 1 year
  - Sensors include: Gearbox temperatures, motor power consumptions, apron speed, etc.

- **Challenge**: Conveyor system is subject to significant fluctuation in load. Hard to characterize the normal operation
  - Mined crude ore never come in a uniform size
Problem setting: change detection from multivariate noisy time-series data

- Change = difference between $p(x)$ and $p_t(x)$
  - $x$: $M$-dimensional i.i.d. observation
  - $p(x)$: p.d.f. estimated from training window
  - $p_t(x)$: p.d.f. estimated from the test window at time $t$

- Assume a sequence of i.i.d. vectors
  - Training data in the training window
    \[
    \{x^{(1)}, \ldots, x^{(n)}, \ldots, x^{(N)}\}
    \]
Problem setting: change detection from multi-variate noisy time-series data

- Question 1: What kind of model should we use for the probability density?
- Question 2: How can we quantify the difference between the densities?
We use von Mises-Fisher distribution to model $p(x)$ and $p_t(x)$

- vMF distribution: “Gaussian for unit vectors”
  \[ p(z \mid u, \kappa) = c_M(\kappa) \exp(\kappa u^\top z) \]
  - $z$: random unit vector of $\|z\| = 1$
  - $u$: mean direction
  - $\kappa$: “concentration” (~ precision in Gaussian)
  - $M$: dimensionality

- We are concerned only with the direction of observation $x$:
  - \[ z = \frac{x}{\|x\|} \]
    - Normalization is always made
    - Do not care about the norm
  - same direction = same input
Normalization is useful to suppress multiplicative noise

- Real mechanical systems often incur multiplicative noise
  - Example: two belt conveyors operated by the same motor

- Normalization of vector is simple but powerful method for noise reduction
Mean direction $u$ is learned via maximum likelihood. Introduce sample weight to down-weight contaminated ones

- Weighted likelihood function

$$L(u, \kappa) = \sum_{n=1}^{N} w^{(n)} b^{(n)} \{ \ln c_M(\kappa) + \kappa u^\top z^{(n)} \}$$

- Regularization over sample weights

$$R(w) = \frac{1}{2} \| w \|_2^2 + \nu \| w \|_1$$

- Parameters are learned by solving

$$(u^*, w^*) = \arg \max_{u, w} \{ L(u, \kappa) + \lambda R(w) \}$$
Multiple patterns (directions) can be obtained by coupling maximum likelihood equations

- Find orthogonal sequence of the mean direction \( \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m \) by coupling the weighted regularized maximum likelihood

\[
\begin{align*}
  (u_1^*, w_1^*) &= \arg \max_{u_1, w_1} \{ L(u_1, \kappa) + \lambda R(w_1) \} \\
  (u_2^*, w_2^*) &= \arg \max_{u_2, w_2} \{ L(u_2, \kappa) + \lambda R(w_2) \} \\
  & \vdots \\
  (u_m^*, w_m^*) &= \arg \max_{u_m, w_m} \{ L(u_m, \kappa) + \lambda R(w_m) \}
\end{align*}
\]

Orthogonality condition

\[
\mathbf{u}_i^\top \mathbf{u}_j = \delta_{i,j}
\]

Kronecker delta
Iterative sequential algorithm for the coupled maximum likelihood

For each $i$, $w_i$ and $u_i$ are solved iteratively until convergence

Analytic solution exists in each step

Results in very simple fixed point equations

\[ w_1 \quad u_1 \]
\[ w_2 \quad u_2 \quad \downarrow \quad u_1 \]
\[ \vdots \]
\[ w_m \quad u_m \quad \downarrow \quad u_i \]

\[(i = 1, \ldots, m-1)\]
Derived fixed-point iteration algorithm

- Example: $i = 1$

  Given $w_1$, solve
  \[
  \max_{u_1} \{ \kappa u_1^T X w_1 \} \text{ s.t. } u_1^T u_1 = 1
  \]

  Given $u_1$, solve
  \[
  \min_{w_1} \left\{ \frac{1}{2} \|w_1 - \frac{q}{\lambda} \|_2^2 + \nu \|w_1\|_1 \right\}
  \]
  
  \[
  q \equiv \ln c_M b + \kappa X^T u_1
  \]

  This Lasso problem is solved analytically

---

**Algorithm 1** RED algorithm.

**Input:** Initialized $w$. Regularization parameters $\lambda, \nu$. Concentration parameter $\kappa$. The number of major directional patterns $m$.

**Output:** $U = [u_1, \ldots, u_m]$ and $W = [w_1, \ldots, w_m]$.

for $j = 1, 2, \ldots, m$ do
  while no convergence do
    
    \[
    u_j \leftarrow \kappa [I_M - U_{j-1}^T U_{j-1}] X w_j
    \]
    \[
    u_j \leftarrow \text{sign}(u_j^T X w_j) \frac{u_j}{\|u_j\|_2}
    \]
    \[
    q_j \leftarrow \gamma b + \kappa X^T u_j
    \]
    \[
    w_j \leftarrow \text{sign}(q_j) \odot \max \left\{ \frac{|q_j|}{\lambda} - \nu 1, 0 \right\}
    \]
  end while
end for

Return $U$ and $W$.  

Theoretical property: The algorithm is reduced to the “trust-region subproblem” in $\nu \rightarrow 0$

**Theorem 2.** When $\nu$ tends to 0, the nonconvex problem (5) is reduced to an optimization problem in the form of

$$\min_u \{u^\top Qu + c^\top u\} \quad s.t. \quad u^\top u = 1,$$

(23)

which has a global solution obtained in polynomial time.

*Proof.* The non-convex optimization problem (23) is known as the trust region subproblem. For polynomial algorithms to the global solution, see [Sorensen, 1997; Tao and An, 1998; Hager, 2001; Toint et al., 2009]. Here we show how the algorithm is reduced to the trust region subproblem.
With extracted directions, define the change score at time $t$ as

$$a^{(t)} = \min_{f, g} \int \text{d}x \, \mathcal{M}(x | U f, \kappa) \ln \frac{\mathcal{M}(x | U f, \kappa)}{\mathcal{M}(x | U^{(t)} g, \kappa)}$$

where $f^\top f = 1, g^\top g = 1$

Concisely represented by the top singular value of $U^\top U^{(t)}$

- $U \equiv [u_1, \ldots, u_m]$  
- $U^{(t)} \equiv [v_1, \ldots, v_r]$
Experiment: Failure detection of ore belt conveyors

- vMF formulation successfully suppressed very noisy non-Gaussian noise of multiplicative nature

- ~40% of samples were automatically excluded from the model

- Better than alternatives
  - PCA, Hoteling $T^2$
  - Stationary subspace analysis [Blythe et al., 2012]
Agenda

- General challenges in industrial sensor data analytics

- Solution examples:
  - Change detection using directional statistics
  - Multi-task multi-modal models for collective anomaly detection (Ide et al, ICDM 17)
  - Tensorial change analysis

- Discussion: deep learning, Blockchain, and future directions
Wish to build a collective monitoring solution

- You have many similar but not identical industrial assets

- You want to build an anomaly detection model for each of the assets

- Straightforward solutions have serious limitations
  - 1. Treat the systems separately. Create each model individually
    - Suffers from lack of fault examples
  - 2. Build one universal model by disregarding individuality
    - Model fit is not good
Practical requirements: Need to capture both commonality and individuality

- Capture both individuality and commonality
- Automatically capture multiple operational states
  - Real-world is not single-peaked / single-modal
- Be robust to noise
- Be highly interpretable for diagnosis purposes
Formalizing the problem as multi-task density estimation for anomaly detection

Data

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

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

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

Prob. density

\[ p^1(\mathbf{x}^1 | \mathcal{D}) \] all data

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

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

Anomaly score

\[ a^1(\mathbf{x}^1) \]

\[ a^s(\mathbf{x}^s) \]

\[ a^S(\mathbf{x}^S) \]

• overall

• variable-wise
Use Gaussian graphical model (GGM)-based anomaly detection approach as the basic building block.

Sparse graphical model:

\[
a(x) = \begin{cases} 
-\ln p(x | D) \\
-\ln p(x_i | x_{-i}, D)
\end{cases}
\]

Overall score

Variable-wise score

\[
\max_{\Lambda} \left\{ \ln \det \Lambda - \text{tr}(\Sigma \Lambda) - \rho \|\Lambda\|_1 \right\}
\]

Sample covariance

Multi-variate data

Sparse graphical model

Anomaly score

[Ide+ SDM09] [Ide+ ICDM16]

Sample covariance

Overall score

Variable-wise score

Training data
Basic modeling strategy: Combine common pattern dictionary with individual weights

System 1 (in New Orleans)

... System s ...

System S (in New York)

Individual sparse weights

Common dictionary of sparse graphs

Monitoring model for System 1

Monitoring model for System 2

Monitoring model for System S

GGM=Gaussian Graphical Model
Basic modeling strategy: Resulting model will be a sparse mixture of sparse GGM

System $s$:

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

- **Sparse mixture weights**
  - (= automatic determination of the number of patterns)

Sparse Gaussian graphical model
Propose a Bayesian multi-task model with two sparsity-enforcing priors

- **Observation model (for the s-th task)**
  - Gaussian mixture with task-dependent weight
    \[
    \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_s \mid \boldsymbol{\mu}^k, (\Lambda^k)^{-1}) z^s_k
    \]

- **Sparsity enforcing priors (non-conjugate)**
  - Laplace prior for the precision matrix
  - Bernoulli prior for the mixture weights
    \[
    p(\Lambda^k) = \left(\frac{\rho}{4}\right)^{M^2} \exp\left(-\frac{\rho}{2}\|\Lambda^k\|_1\right)
    \]
    \[
    p(\pi^s) = p_0^{\|\pi^s\|_0} (1 - p_0)^{G - \|\pi^s\|_0}
    \]

- **Conjugate prior on \{\boldsymbol{\mu}^k\} and \{z^s\}**
    \[
    p(\boldsymbol{\mu}^k \mid \Lambda^k) = \mathcal{N}(\boldsymbol{\mu}^k \mid \mathbf{m}^0, (\lambda_0 \Lambda^k)^{-1})
    \]
    \[
    p(z^s \mid \pi^s) = \prod_{k=1}^{K} (\pi_k^s)^{z^s_k}
    \]
Maximizing log likelihood using variational Bayes combined with point-estimation

- Complete log likelihood

\[
L = \sum_{s=1}^{S} \sum_{n=1}^{N_s} \sum_{k=1}^{K} \ln \mathcal{N}(x_s^{(n)} | \mu_k^z^{(n)}) + \sum_{k=1}^{K} \text{Lap}(\Lambda_k | \rho)p(\mu_k | \Lambda_k) + \sum_{s=1}^{S} z_s^{(n)} \ln \pi_k^{s} + \sum_{s=1}^{S} \ln p(\pi^s)
\]

- Likelihood by the obs. model

- Prior distributions

- Use VB for \( \{\mu_k^z\}, \{z_s^{(n)}\} \)

- Use point-estimate for \( \{\Lambda_k^z\}, \{\pi^s\} \)
  - Results in two convex optimization problems
Maximizing log likelihood using variational Bayes combined with point-estimation

- Update sample weights
- Update cluster weights
- Update precision matrices
- Update other parameters

Use new semi-closed form solution

\[
\max_{\pi^s} \left\{ \sum_{k=1}^{K} c_k^s \ln \pi_k^s - \tau \| \pi^s \|_0 \right\}
\]

The ratio of samples assigned to the \( k \)-th cluster

\[
\text{s.t.} \quad \| \pi^s \|_1 = 1.
\]

Solved by graphical lasso [Friedman 08]

\[
\max_{\Lambda^k} \left\{ \ln |\Lambda^k| - \text{Tr}(\Lambda^k Q^k) - \frac{\rho}{N_k} \| \Lambda^k \|_1 \right\}
\]

total # of samples assigned to the \( k \)-th cluster
Solving the $L_0$-regularized optimization problem for mixture weights

- Conventional VB approach without $L_0$ regularization on $\pi_k^s$ is problematic
  - Claimed to get a sparse solution [Corduneanu+ 01]
  - But mathematically $\pi_k^s$ cannot be zero due to logarithm

- We re-formalized the problem as a convex mixed-integer programming
  - A semi-closed form solution can be derived (→ see paper)

$$\max_{\pi^s} \left\{ \sum_{k=1}^{K} c_k^s \ln \pi_k^s \right\}$$

subject to $\sum_{k=1}^{K} \pi_k^s = 1$, $y_k^s \geq \pi_k^s - \epsilon$, $y_k^s \in \{0, 1\}$ for $k = 1, \ldots, K$, $\|\pi^s\|_1 = 1$. 
## Comparison with possible alternatives

<table>
<thead>
<tr>
<th>Method</th>
<th>Interpretability</th>
<th>Noise reduction</th>
<th>Fleet-readiness</th>
<th>Multi-modality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our work [Ide et al. ICDM 17]</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>(single) sparse GGM</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gaussian mixtures</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Multi-task sparse GGM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multi-task learning anomaly detection</td>
<td>No</td>
<td>(depends)</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Notes:**
- [Ide et al. SDM 2009, Ide et al. ICDM 2016]  
- [Yamanishi et al., 2000; Zhang and Fung, 2013; Gao et al., 2016]  
- [Varoquaux et al., 2010; Honorio and Samaras, 2010; Chiquet et al., 2011; Danaher et al., 2014; Gao et al., 2016; Peterson et al., 2015].  
- [Bahadori et al., 2011; He et al., 2014; Xiao et al., 2015]
Experiment (1): Learning sparse mixture weights

- Conventional ARD approach sometimes gets stuck with local minima
  - ARD = automatic relevance determination
  - Often less sparse than the proposed convex $L_0$ approach

- Typical result of log likelihood vs VB iteration count

![Graph showing log likelihood vs number of iterations for conventional ARD and proposed convex $L_0$ approaches. The proposed approach shows a higher log likelihood and avoids getting stuck with a local minimum.](image-url)
Experiment (2): Learning GGMs and detecting anomalies

- “Anuran Calls” (frog voice) data in UCI Archive
  - Multi-modal (multi-peaked)
  - Voice signal + attributes (species, etc.)
- Created 10-variate, 3-task dataset
  - Use the species of “Rhinellagranulosa” as the anomaly
- Results
  - Two non-empty GGMs are automatically detected starting from K=9
  - Clearly outperformed single-modal MTL alternative in anomaly detection
    ✓ Group graphical lasso, fused graphical lasso
Agenda

▪ General challenges in industrial sensor data analytics

▪ Solution examples:
  o Change detection using directional statistics
  o Multi-task multi-modal models for collective anomaly detection
  o Tensorial change analysis

▪ Discussion: deep learning, Blockchain, and future directions
Developing a system for change diagnosis when input data is a tensor (multi-way array)

- Real application example: Condition-based monitoring of reactive ion etching tool
  - Tools deteriorate over time due to debris in the etching chamber
  - Degradation process is implicit and subtle. Quantification is challenging

- Basic problem setting: Compare a test period with a reference period to explain what really is the difference in terms of observable variables

```
- ~ 100s wafers processed
- ~ 30 sensors
- ~ 20 etching steps
- ~ 10 statistical quantities
```

“golden period” (or reference period)  test period in question
The input is a tensor (multi-way array) associated with a goodness metric

- Semiconductor etching example
  - y: (one of) quality measurements
    - electric resistance, line widths, ...
  - X: “trace data” (sensor recordings)
    - pressure, temperature, electric current, ...

- One etching round of trace data is most naturally represented as a tensor (multi-way array)
  - Typically 3-way array
    - variable x etching step x statistics used x time
    - variable x etching step x etching metal layer
  - Often summarized as 2-way tensor by e.g. taking the mean over time in each step
The task addressed: (1) Detect a change in X-y relationship. (2) Explain which mode/dimension is most responsible

- (1) Compute the anomalousness of a single or a set of etching round(s) in a test period
- (2) Compute the responsibility of the dimensions of each mode that explains the anomalousness of the test period
Technical challenges

Tensor regression is not well-studied
- Regression is the task to learn a function $y = f(X)$ from training data
- Existing techniques mainly use vectorization of tensors

Probabilistic prediction is even harder
- Non-subjective change scoring requires probabilistic prediction.
- Existing probabilistic tensor regression methods are impractical

Vectorized probabilistic model cannot be the solution
- Not very interpretable – it destroys the tensor structure of the input

Tensorial change diagnosis framework using probabilistic tensor regression algorithm
Agenda

▪ General challenges in industrial sensor data analytics

▪ Solution examples:
  o Change detection using directional statistics
  o Multi-task multi-modal models for collective anomaly detection
  o Tensorial change analysis

▪ Discussion: deep learning, Blockchain, and future directions
Does deep learning mean the end of journey? Probably not. Factors that make deep learning work

- **Well-defined and well-accepted task**
  - No need to tell why

- **Huge amount of labeled training data**
  - Typically needs millions labeled samples

- **Minimum uncertainty in data representation**
  - Pixels, words, Mel-filterbank

- **Good applications meeting these criteria**
  - Image recognition
  - (Some of) natural language processing
  - Speech recognition

- **How about industrial dynamic systems?**
  - Interesting research topic
One caveat: Automated feature learning from noisy sensor signal is still challenging

- Image recognition and NLP (natural language processing) are an ideal area for deep learning
  - Huge annotated datasets exist
  - Established preprocess method

- A little secret in speech recognition: State-of-the-art deep-learning-based systems use handcrafted features

- The situation will be much tougher in general industrial sensor data analytics

“State-of-the-art speech recognition systems rely on fixed, handcrafted features such as mel-filterbanks to preprocess the waveform before the training pipeline”
Implications for sensor data analytics

- Deep learning (especially RNN such as LSTM) will be a powerful tool when
  - we know how to read the data (and thus a good amount of labeled training data exists)
  - we know limitations of linear models (state-space models)
  - we have a lot of GPU!
Discussion: Will Blockchain bring in any value on sensor data analytics?

- What is Blockchain?
  - Distributed decentralized database characterized by a hash chain data structure and a consensus algorithm

- Blockchain generations
  - 1st generation (Bitcoin)
    - De-centralized, secure platform for money transfer
  - 2nd generation (Ethereum, Hyperledger, Corda)
    - Extended to handle general business transactions beyond money transfer

- Expected to be a useful platform for IoT (internet-of-things) systems
  - “Device democracy”
Discussion: Will Blockchain bring in any value on sensor data analytics?

- Blockchain should be generalized as a collaborative learning platform
  - “Blockchain 3.0”
  - The particular hash chain data structure can be viewed as just one instance of implementation

- Example: privacy-preserving multi-task learning on Blockchain
Industrial sensor data have many interesting features that call for new machine learning formulation.

- Introduced a few recent works on anomaly detection:
  - Change detection using directional statistics
  - Multi-task anomaly detection algorithm
  - Tensorial change analysis

Ongoing/future work:
- Prediction/anomaly detection from novel data types
  - Tensors, functions, graphs, trajectories, events, etc.
- Multi-x / cross-x learning
  - Multi-task, view, domain, modality
- Deep learning for dynamic systems
Thank you!