Clustering-Based Anomaly Detection in Multi-View Data
Alejandro Marcos Alvarez*, Makoto Yamada†, Akisato Kimura* and Tomoharu Iwata*
* University of Liège  † Yahoo Labs  ‡ NTT Communication Science Laboratories
amarcos@ulg.ac.be

Highlights
- New anomaly score for multi-view data based on affinity propagation and HSIC
- 28% average improvement over state-of-the-art method
- No parameter tuning
- Fast and robust

Motivation and Problem Setting
Goal: detect anomalies in multi-view data
Anomaly = object with inconsistent behavior across multiple views
Problem Setting
- n i.i.d. objects denoted by \( i = 1, \ldots, n \) and \( V \) views denoted by \( v = 1, \ldots, V \)
- Object \( x \) seen through view \( v \) has a feature representation \( z_v \) such that \( z_v^i \in X^v \subset \mathbb{R}^p \)
- \( X^v \) is the domain of view \( v \) and \( p \) is the dimensionality of that domain
- \( D \) is the set containing the \( V \) feature representations of each object \( i \): \( D = \{ (z_1^i, \ldots, z_V^i) \}^n_{i=1} \)

Proposed Method
Step 1: perform clustering separately in each view
Clustering is performed
1) separately in the different views
   - \( V \) different clusterings \( \{ c_i^v \}^V_{i=1} \)
   - \( c_i^v \in \{ 1, \ldots, n \} \) = cluster of object \( i \)
2) with affinity propagation (AP)
   - fast, simple & parameter-free (almost)
   - automatically estimates the number of clusters
   - input: affinity matrix \( L^v_{ij} \)
Example of AP clustering
- two clusters are found
- objects 3 and 7 are the centers of the clusters
   \( \mathbb{Z}^n = \{ 3, 3, 3, 3, 7, 7, 7, 7 \} \)

Step 2: compute affinity vectors in each view
Clustering-based affinity vector \( z_v^i \in \mathbb{R}^p \) of object \( i \) in view \( v \):
\[
z_v^i(j) = \begin{cases} \frac{1}{Z} \exp \left( \frac{-1}{Z} \sum_{j'} \frac{1}{L_{j'}^v} + L_{j'}^v - 2 \right) & \text{if } i = j, \\ \text{otherwise,} & \text{with } Z = \sum_{j'} z_v^i(j) \end{cases}
\]
Idea: combine clustering information \( L_{ij}^v \) and objects affinities \( z_v^i \)

Step 3: compute anomaly scores
1) Distance based \( A(z_v^i, z_v^j) = z_v^i \cdot L_{ij}^v \)
2) Pearson’s correlation based
3) Spearman’s correlation based
\[
A(z_v^i, z_v^j) = \Delta_v, \text{ with } \Delta_v = H(Z_1^i + Z_1^j) \cdot H(Z_2^i + Z_2^j)
\]

Experiments
- spectral clustering performed in all the views at the same time
- clustering vectors compared with cosine similarity
- two parameters: \( k \) (number of expected clusters) \& \( m \) (a penalty parameter) \rightarrow tuning required
Setup
- compare AP vs HOAD with area under ROC curve (AUC)
- mono-view \rightarrow multi-view: split features into two sub-groups
- create anomalies: swap both sub-groups

Affinity matrices: negative L2-norm and Gaussian kernel
Datasets: UCI ML repository ("iris", "letter", "waveform", "zoo") & synthetic datasets

Results

Conclusions
Summary
- new anomaly detection approach for multi-view data
- combines affinities and objects’ neighborhoods in different views
Advantages
- robust & parameter-free
- fast & effective
Open question: Clustering method and anomaly score are different from those of HOAD. Which part is more significant to improve the performance?

Affiliations and Acknowledgments