Unsupervised learning in evolving environments

Group Project

S. Pattar[1] Z. Pasztori[2]

European Master in Advanced Robotics (EMARO)
University of Genova

EMARO, 2016-2017

Outline

1. Methodology
   - Anomaly detection
   - Detecton with membership
   - Detection with TEDA framework

Data streams

Optional Subtitle

- Great amount of data generated. Robot proximity sensors, encoders, cameras.
- The data changes over time, as the robot environment is changing, or when it is mapping a new location. Shift and drift.
- Methods must be both computationally and storage efficient. They have to be run onboard the robot, with limited capacity.

Anomaly detection

- Data point which does not conform to expected pattern
- Can be noise or the underlying distribution might have changed
- A really hard problem, since we have *bounded rationality*. Need to make decisions in the present, with limited data and computation time.

Anomaly detection

Two methods used:

- Anomaly detection based on membership, with a variation of fuzzy k-means clustering (PCM)
- σ gap principle, introduced by Dr Plamen Angelov

Outline

1. Methodology
   - Anomaly detection
   - Detecton with membership
   - Detection with TEDA framework

Clustering

- Unsupervised machine learning
- Reduces the data dimensionality
- k-means clustering, each cluster is represented by a point, reduces n data points to k
- k-means makes handling big-data easier

K-means clustering

- Divides the data space into clusters, the boundaries depend on the distance metric we use

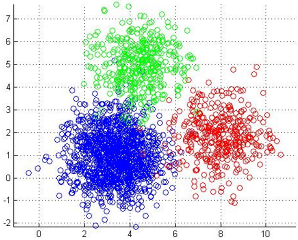


Figure: 3-means algorithm, where the data is 3 Gaussian distributions

Anomaly detection based on membership

This method detects anomalies during clustering. We make use of *possibilistic* fuzzy K-means clustering for this approach.

- Fuzzy K-means assigns each point partially to each clusters, i.e. how much a point belongs to one of the K clusters.
- Membership are values between 0 and 1.

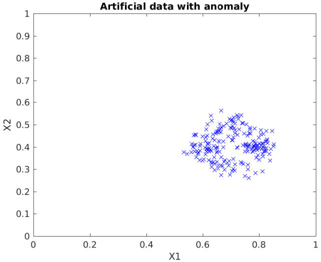
Anomaly detection based on membership

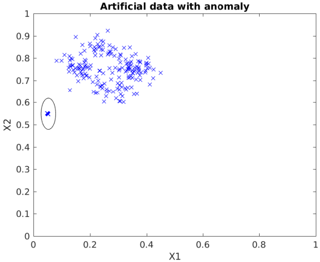
Algorithm:

1. Calculate membership of each data point to each cluster at each iteration.
2. Check if sum of membership to each cluster is greater than $1/K$.(Here K is number of clusters).
3. IF yes, then update the clusters at the next iteration.
4. ELSE ignore the data points with sum of membership less than $1/K$.

Anomaly detection based on membership

We made use of artificial drifting data with anomalies created at a specific time.

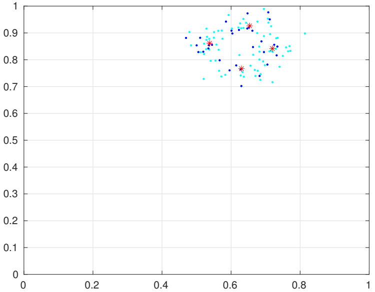




VIDEO

Anomaly detection based on membership

For fuzzy k-means clustering we use K = 4, which is one of the presumptions of our algorithm.



Anomaly detection based on membership

With the basic fuzzy k-means clustering, the centroids shift due to the presence of anomalies.

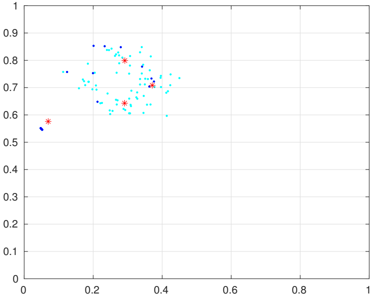


Figure: Anomalies causing centroids to shift

Anomaly detection based on membership

Once the algorithm is implemented, data points with sum of membership less than $1/K$ are detected and ignored.

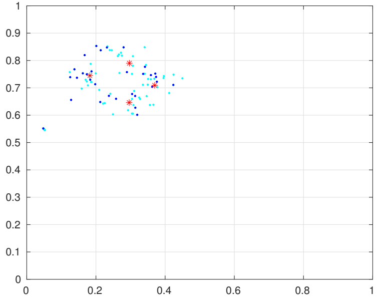


Figure: Anomalies ignored by the algorithm

VIDEO

Outline

1. Methodology
   - Anomaly detection
   - Detecton with membership
   - Detection with TEDA framework

σ gap principle

This algorithm is useful in detecting anomalies before clustering or any other process. The following characteristics are introduced in the algorithm from TEDA (Typicality and Eccentricity Data Analysis) framework:

- *Accumulated proximity*, π: sum of distaces to each data point from every data point.

  $$\pi_s(x_j) = \pi_{js} = \sum_{i=1}^{s} d_{ij} \quad s > 1$$

  where d_{ij} denotes a distance measure between data samples. We used eucledian distance.

- *Eccentricity*, ξ: Quotient of accumulated proximity of one point and sum of all accumulated proximities.

  $$\xi_{js} = \frac{2\pi_{js}^{s}}{\sum_{i=1}^{s} \pi_{is}} \quad \sum_{i=1}^{s} \pi_{is} > 0$$

σ gap principle (contd.)

- *Normalized eccentricity,* ζ

$$\zeta = \frac{(x_s - \mu_s)^2}{2s\sigma_s^2} + \frac{1}{2s}$$

where, VARIANCE, σ_s^2

$$\sigma_s^2 = \sum_{i=1}^{s} \frac{(x_i - \mu_s)^T (x_i - \mu_s)}{s}$$

σ gap principle

The σ gap condition is very intuitive and is defined as follows:

$$\textbf{IF}(\Delta\zeta^{1,2} > n/s)\textbf{THEN}(x^1 \text{ is an outlier})$$

σ gap principle

Algorithm:

1. Calculate normalized eccentricity of a point.
2. Arrange the points with the maximum normalized eccentricity, x^1 second maximum normalized eccentricity, x^2, etc. in decreasing order.
3. Check the "σ **gap**" condition.
4. If it is satisfied, declare the point x^1 an outlier.

σ gap principle

Its advantages over traditional "nσ" approaches are:

- It does not need any presumptions on the data.
- It can find anomalies with dataset as small as 3 samples *(Angelov 2014)*.

σ gap principle

We made use of the same data as used by the reference paper. As seen below, the traditional "nσ" approach fails to detect the anomaly.

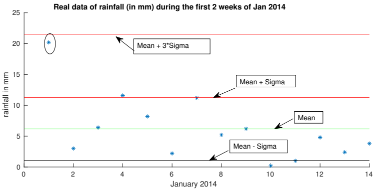


Figure: Real rainfall data from Bristol, UK, first two weeks of January, 2014 [7,14].

σ gap principle

But the σ gap principle successfully detects the anomaly even in a small data-set.

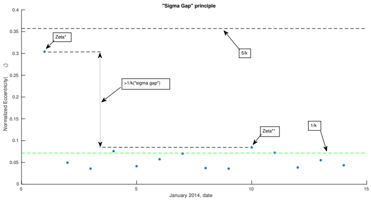


Figure: The σ gap principle is illustrated on the simple 1D rainfall data from the first couple of weeks in South-West UK.

σ gap principle

This algorithm was also tested on our own artificial dataset, but which needed a sliding window of size 31. And it successfully detected the anomaly.

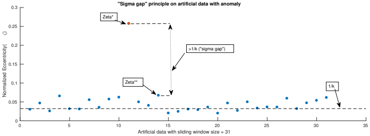


Figure: The σ gap principle illustrated on the artificial data at the time when anomalies are created.

Summary

Anomaly detection based on membership

- *Pros*
  - Detects anomalies during clustering of data.
  - Online detection of anomalies of live streaming data.
- CONS
  - Has all the problems associated with clustering algorithms, such as selection of number of clusters.

Summary

"σ gap" principle

- *Pros*
  - does not need any presumptions on the data such as used in the traditional "$n\sigma$" approaches.
  - It can find anomalies with datasets as small as 3 samples.
- CONS
  - To implement in on data streams, the window size needs to be pre-assigned.
  - Adds an extra step of computation.
  - Difficulties with live data streams where the data needs to clustered or classified on-line.

Thank you for your attention. Any questions?