K-means for Evolving Data Streams

Anikeit Sethi 190001003

Garvit Galgat 190001016

Yash Suvarna 190001068

Data Stream

Data Stream

- Data stream refers to a continuous, unbounded sequence of data elements that are generated over time and pose unique challenges for machine learning and data analysis.
- However, the data is assumed to be generated by the same underlying distribution throughout the stream.

Evolving Data Stream

- In an ***evolving*** *data stream*, the data not only arrives continuously, but the underlying distribution may change over time as well.
- The changes in the distribution are often attributed to external factors such as changes in the environment, user behaviour, or system dynamics.

Properties of Evolving Data Stream

- Volume
  - Data streams can generate large amounts of data, often at high velocities.
- Velocity
  - Data streams can arrive at high speeds, requiring fast processing and analysis.

Properties of Evolving Data Stream

- Variety
  - Data streams can come in different formats, structures, and types, such as text, audio, image, and numerical data.
- Variability
  - Data streams may exhibit temporal and spatial variations in the underlying distribution, leading to concept drift or changes in the data patterns.

Properties of Evolving Data Stream

- Unboundedness
  - Data streams may not have a fixed size or length, and can continue indefinitely.
- Noisy
  - Data streams may contain errors, outliers, or missing values due to various factors, such as sensor malfunction or transmission errors.

Datasets

Dataset

- Urban accidents
- Pulsar detection
- SUSY
- Gas sensors
- Anuran calls
- Google Reviews
- Gesture Segmentation
- Epilepsia

Concept Drift

A concept drift happens when there is a change in the fundamental pattern or characteristics of the data.
Ways to counter concept drift:

- **Active Approach:**
  - Dynamically adjusts stored batches depending on whether a concept drift has occurred or not
- **Passive Approach:**
  - More importance is given to recent batches.

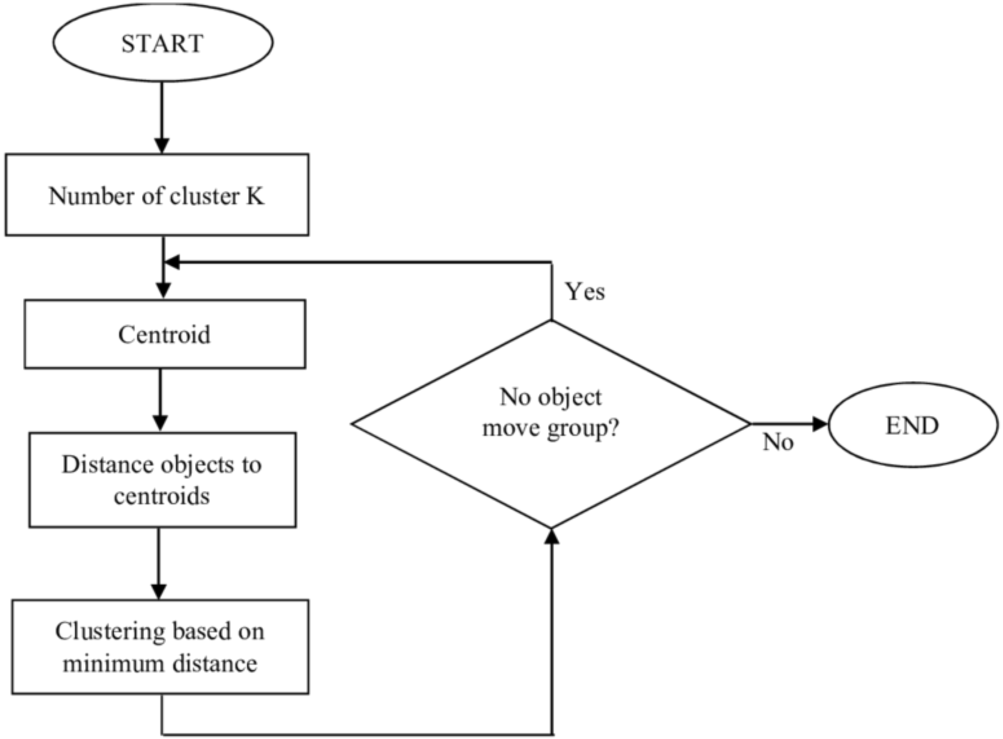
K Means

K means

K means is algorithm to cluster n objects based on attributes into k partition where k<n

Algorithm

Step 1. Select the Number of Clusters, *k*
Step 2. Select *k* Points at Random
Step 3. Make *k* Clusters
Step 4. Compute the New *Centroid* of Each Cluster
Step 5. Assess the Quality of Each *Cluster*
Step 6. *Repeat* Steps 3–5



FlowChart of K means

https://www.researchgate.net/publication/318341309_Application_of_k-means_clustering_algorithm_in_grouping_the_DNA_sequences_of_hepatitis_B_virus_HBV

K means++ (KM++)

- K-means++ is an extension of the original K-means algorithm designed to improve its initialization step
- K-means++ ensures that the initial centroids are well spread out and avoids the problem of initializing all the centroids in the same cluster or in close proximity to each other.
- K-means++ can also be more computationally efficient than the original K-means algorithm, as it often requires fewer iterations to converge.

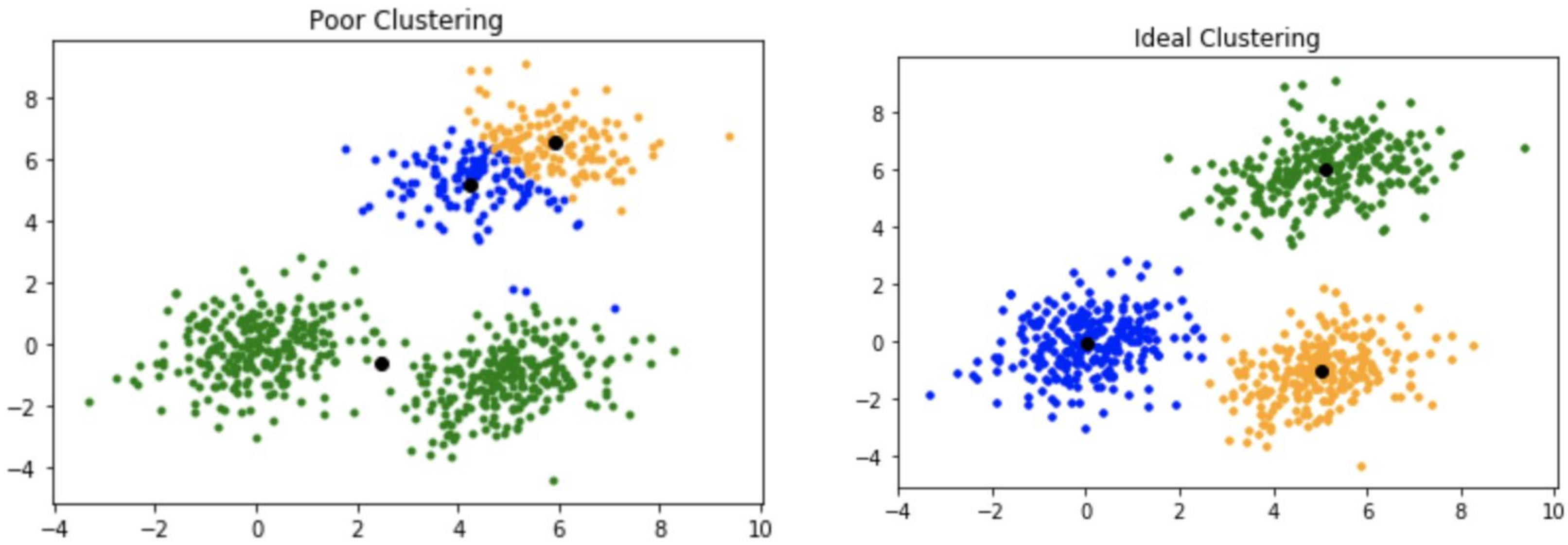
KM++ Algorithm

Step 1. Randomly select the first centroid from the data points.

Step 2. For each data point compute its distance from the nearest, previously chosen centroid.

Step 3. Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid.

Step 4. Repeat steps 2 and 3 until k centroids have been sampled



Comparison of poor clustering to ideal clustering

https://www.geeksforgeeks.org/ml-k-means-algorithm/

Streaming K-Means

STREAMING K-MEANS (SKM)

- It is a variant of the traditional K-Means algorithm that is specifically designed to deal with data streams.
- It processes data in batches, and can dynamically adapt to changes in the data distribution over time

SKM error function

$$E_*(\mathcal{X}, C) = \frac{1}{M_T} \cdot \sum_{t=0}^{T-1} \sum_{x \in B^t} \|\boldsymbol{x} - \boldsymbol{c}_x\|^2$$

$\mathcal{X}$ – Set of Batches of Datapoints

C – Set of centroids

E_* **– SKM error**

M_T - **the sum of each batch size**

t – **describes the antiquity of each batch**

Problem with SKM error function

When a concept drift occurs, the error function needs to be restarted.

Surrogate Error Function

A surrogate error function is a function that is used as an approximation of the SKM error function. The surrogate error is a weighted version of the K-means error for SD.

$$E_\rho(\mathcal{X}, C) = \frac{1}{M_{\mathcal{X}}} \cdot \sum_{t \geq 0} \rho^t \cdot \sum_{x \in B^t} \|\boldsymbol{x} - \boldsymbol{c}_x\|^2$$

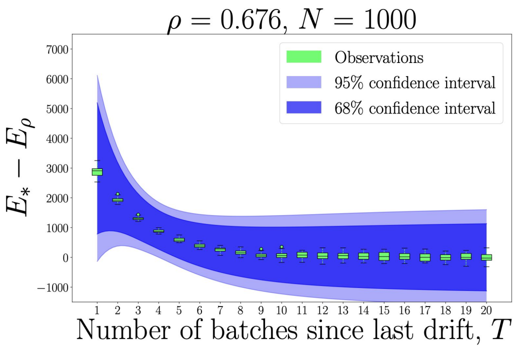
$\mathcal{X}$ - Set of Batches of Datapoints

C - Set of Centroids

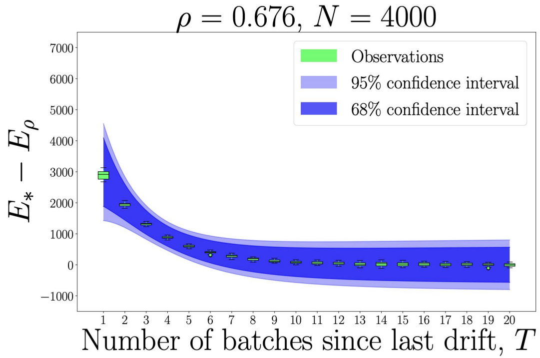
$\mathbf{M}_{\mathcal{X}}$ - total weighted mass of the set of batches

ρ - memory parameter

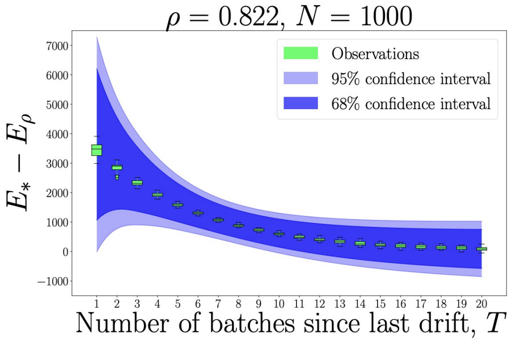
t - describes the antiquity of each batch



(a)



(b)



(c)

Difference between the SKM error E∗ and the surrogate error Eρ as T increases

References

1. D. Dua and C. Graff, “UCI Machine Learning Repository,” 2017.
2. R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles, “Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach,” Monthly Notices of the Royal Astronomical Society, vol. 459, no. 1, pp. 1104–1123, 2016.
3. P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” Nature communications, vol. 5, no. 1, pp. 1–9, 2014.
4. J. Burgues and S. Marco, “Multivariate estimation of the limit of ´ detection by orthogonal partial least squares in temperature-modulated MOX sensors,” Analytica Chimica Acta, vol. 1019, pp. 49–64, 2018.
5. J. Burgues, J. M. Jim ´ enez-Soto, and S. Marco, “Estimation of the limit ´ of detection in semiconductor gas sensors through linearized calibration models,” Analytica Chimica Acta, vol. 1013, pp. 13–25, 2018.
6. J. G. Colonna, M. Cristo, M. S. Junior, and E. F. Nakamura, “An ´ incremental technique for real-time bioacoustic signal segmentation,” Expert Systems with Applications, vol. 42, no. 21, pp. 7367–7374, 2015.
7. S. Renjith, A. Sreekumar, and M. Jathavedan, “Evaluation of partitioning clustering algorithms for processing social media data in tourism domain,” in 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 127–131, IEEE, 2018.
8. R. C. Madeo, C. A. Lima, and S. M. Peres, “Gesture unit segmentation using support vector machines: segmenting gestures from rest positions,” in Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 46–52, 2013.
9. R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” Physical Review E, vol. 64, no. 6, p. 061907, 2001.