CS 403 Project Report

Aniket Sethi - 190001003
Yash Suvarna - 190001068
Garvit Galgat - 190001016

K-means for Evolving Data Streams

Introduction

In recent years, the analysis of streaming data has gained significant importance in the field of machine learning. A large proportion of available data streams are unlabeled, which presents a challenge in developing effective clustering techniques that can handle the unique characteristics of streaming data. Streaming data scenarios involve a continuous sequence of data batches, and these streams may exhibit a phenomenon known as concept drift, where the underlying data distribution changes over time.

In this work, we aim to address the problem of streaming K-means (SKM), which specifically deals with clustering in streaming data and incorporates the need to handle concept drift. The primary focus is on developing an approach that does not rely on explicit concept drift detection but rather restarts the error function whenever a concept drift occurs. This enables us to adapt to the changing data distribution and maintain accurate clustering results in streaming data scenarios. To achieve this, we utilized an approximated error function that is robust to concept drift. The surrogate error function serves as a substitute for the SKM error, providing a reliable measure of clustering quality without the need for drift detection. We Observed the effectiveness of the surrogate by proving that it is a good approximation of the SKM error, ensuring that it captures the relevant information for clustering.

Objective

The main objective of this research is to develop a k-means algorithm capable of efficiently clustering evolving data streams.

To achieve this, the following specific objectives will be pursued:

1. Formal Definition of Streaming K-means (SKM) Problem, which specifically addresses clustering in streaming data scenarios with the presence of concept drift.
2. The algorithm for SKM without Concept Drift Detection incorporates an approximation approach that assigns exponentially decaying weights to older batches. This allows for continuous adaptation to concept drift without the need for drift detection.
3. Initialization Techniques for SKM, aim to provide robust and effective initial centroids for the SKM problem, enhancing the quality of clustering results.

Data Stream

A data stream refers to a continuous and infinite flow of data that arrives sequentially over time. Unlike static datasets, where all the data is available at once, data streams present unique challenges due to their dynamic nature and high-speed data arrival rates. Stream processing involves analyzing and extracting valuable insights from these data streams in real-time or near real-time. Data streams can originate from various sources, including sensor networks, social media feeds, financial transactions, online user interactions, and more. They are characterized by their high volume, velocity, and variability. The data in a stream is typically transient and cannot be stored entirely, which necessitates real-time analysis to make timely decisions or identify patterns and trends.

Data stream analysis techniques aim to address the challenges posed by streaming data. These techniques include online learning algorithms, statistical methods, and adaptive models that can handle the high data rates, evolving patterns, and concept drift often observed in data streams. Stream processing frameworks, such as Apache Flink and Apache Kafka, provide tools and infrastructure to handle large-scale data streams and enable real-time analytics and decision-making. Data stream analysis finds applications in various domains, including fraud detection, network monitoring, recommendation systems, anomaly detection, and predictive maintenance. It enables organizations to leverage the valuable insights hidden within streaming data to make informed decisions, optimize operations, and gain a competitive edge.

Concept Drift

Concept drift refers to the phenomenon where the underlying distribution or patterns in a data stream change over time. In the context of K-means clustering applied to data streams, concept drift poses a significant challenge as the clustering solution needs to adapt to the evolving nature of the data. When concept drift occurs in a data stream, the assumptions made by the K-means algorithm regarding the distribution of the data may no longer hold true. The existing clusters may become obsolete, new clusters may emerge, or the characteristics of existing clusters may change. As a result, the clustering solution needs to be updated to reflect the current data distribution accurately.

One approach to address concept drift in K-means for data streams is to continuously monitor the data stream for drift detection. This involves detecting abrupt or gradual changes in the data distribution and triggering the re-clustering process when significant drift is detected. The challenge lies in accurately and timely detecting concept drift, as it may be subtle and occur gradually over time. Another approach is to employ adaptive K-means algorithms that can update the clustering solution incrementally as new data arrives. These algorithms dynamically adjust the cluster centers and sizes to accommodate concept drift. Adaptive techniques often utilize a sliding window mechanism or exponential decay to give more weight to recent data and gradually forget older data that may no longer represent the current distribution accurately. To effectively handle concept drift in K-means for data streams, a combination of drift detection mechanisms, adaptive algorithms, and appropriate parameter settings is necessary. This ensures that the clustering solution can adapt to the changing nature of the data stream and maintain accurate cluster representations over time.

One major disadvantage of concept drift in the context of K-means for data streams is that it can lead to inaccurate or outdated clustering results. As the underlying distribution changes, the existing clusters may become less representative or even irrelevant to the current data patterns. This can result in misclassification or decreased clustering performance. Detecting concept drift and adapting the clustering model in real-time adds complexity to the system and requires additional computational resources. Moreover, if concept drift is not properly handled, it can lead to suboptimal clustering solutions and reduce the overall effectiveness of K-means in analyzing evolving data streams.

K Means

K-means is a popular clustering algorithm used to partition a dataset into distinct groups or clusters based on similarity. It aims to minimize the within-cluster variance by iteratively assigning data points to the nearest cluster centroid and updating the centroids based on the mean of the assigned points. Applying the traditional K-means algorithm can pose several challenges and limitations in the context of evolving data streams with concept drift. The main disadvantage of using K-means in this context is that it assumes a fixed data set and stationary data distribution, which contradicts the dynamic nature of data streams.

One limitation of K-means for evolving data streams is that it requires storing the entire data stream or a significant portion to perform clustering. However, in streaming scenarios, the volume of data can be massive and continuously increasing, making it infeasible to store all the data. This leads to scalability and memory constraints when applying K-means directly. Another drawback of K-means is its sensitivity to outliers and noisy data. In data streams, concept drift can introduce abrupt changes, which can create outliers or noisy instances. K-means is not robust to such outliers and may produce suboptimal clustering results when confronted with data stream dynamics. Additionally, the concept drift phenomenon challenges the assumption of K-means regarding the static data distribution. As the underlying data distribution changes over time, the initial cluster centroids and assignments may become less representative of the current data patterns. K-means does not have built-in mechanisms to handle concept drift, which can lead to outdated or inaccurate clustering solutions. Several K-means adaptations have been proposed for evolving data streams to address these limitations.

K Means ++

K-means++ is an improvement over the traditional K-means algorithm that addresses the issue of selecting initial centroids, which greatly impacts the quality of the clustering solution. The initialization step in K-means++ aims to choose the initial centroids in a way that improves the chances of finding a globally optimal clustering. In the standard K-means algorithm, initial centroids are typically selected randomly from the dataset. However, this random initialization

can lead to poor clustering results or convergence to suboptimal solutions. K-means++ introduces a more intelligent and systematic approach to initializing centroids.

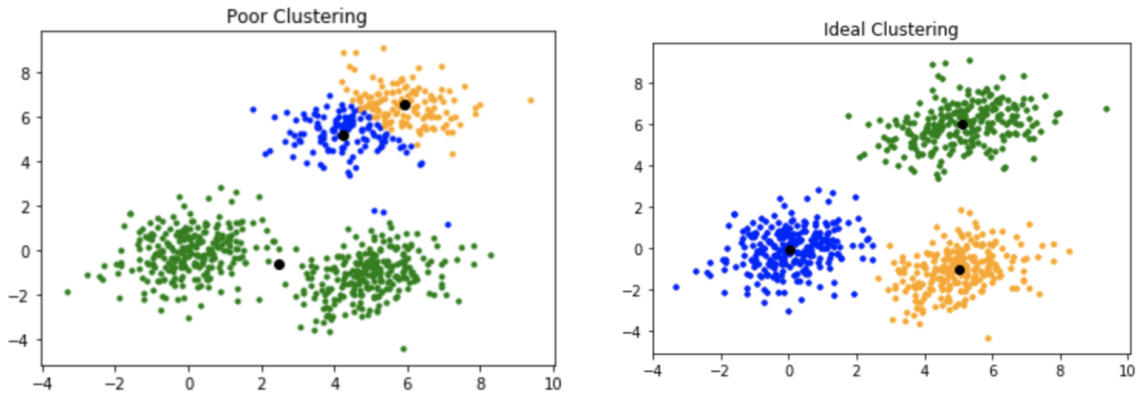
The initialization process in K-means++ consists of the following steps:

1. Select the first centroid randomly from the dataset. This initial centroid serves as the starting point.
2. Calculate the minimum distance (D(x)) for each data point to the nearest existing centroid. The selection of subsequent centroids is based on these distances.
3. Choose the next centroid with a probability proportional to D(x)^2. The rationale behind this probability is to favor data points that are farthest from existing centroids, promoting a more even spread of centroids.
4. Repeat steps 2 and 3 until the desired number of centroids is obtained.

K-means++ aims to select diverse and representative initial centroids using this initialization procedure. The distances D(x) play a crucial role in determining the probability of selecting each data point as a centroid, ensuring that data points far from existing centroids are given higher chances of being selected. The advantage of K-means++ initialization is that it significantly improves the chances of obtaining a better clustering solution compared to random initialization. By selecting initial centroids more intelligently, it helps overcome the problem of converging to suboptimal or degenerate solutions that can occur with traditional K-means.

Theoretical analysis has shown that the initialization step of K-means++ reduces the expected distortion, which measures the average squared distance between data points and their closest centroid. This reduction in distortion leads to more accurate and stable clustering results. Another advantage of K-means++ is its computational efficiency. The initialization process involves selecting a fixed number of centroids, regardless of the size of the dataset. This makes it scalable for large datasets, as the time complexity remains proportional to the number of centroids rather than the number of data points. However, it is important to note that K-means++ does not guarantee finding the globally optimal clustering solution. It provides a significant improvement over random initialization but is still sensitive to the initialization of the first centroid. Multiple runs of K-means++ with different random initializations are recommended to mitigate this issue. In conclusion, K-means++ offers an intelligent and effective approach for initializing centroids in the K-means algorithm. By selecting diverse and representative initial centroids, it improves the chances of finding a better clustering solution and reduces the

likelihood of getting stuck in suboptimal solutions. The computational efficiency and theoretical guarantees make K-means++ a popular choice for initializing K-means and enhancing the performance of the clustering algorithm.



Comparison of poor clustering to ideal clustering

Streaming K Means

Streaming K-means (SKM) is a clustering algorithm specifically designed to handle evolving data streams. It extends the traditional K-means algorithm to adapt to the sequential and dynamic nature of streaming data. SKM addresses two key challenges in streaming scenarios: the increasing volume of data and the occurrence of concept drift. In SKM, a sliding window approach is employed to efficiently process data streams. Only a fixed number of the most recent batches are considered for clustering, reducing memory requirements and computational complexity while still capturing the most up-to-date information. To handle concept drift, SKM introduces a restart mechanism. When a significant concept drift is detected, the error function is reset, and the clustering process starts anew. This allows SKM to adapt to changes in the underlying data distribution and maintain the accuracy of the clustering solution over time.

Mathematical Formulation

The Stream K Means error function is given by:

$$E(\chi, C) = 1/Mt \sum_{t=0} \sum_{x \in B^t} ||x - c||^2$$

$\chi = \{B^t\}$ $t \geq 0$

C *is the set of Centroids*

Mt *the sum of each batch size*

c *is the corresponding centroid for the data point* x

Approximation Error

In the context of Streaming K-means (SKM), the approximation error refers to the discrepancy between the true error of the clustering solution and the error estimated using an approximate surrogate error function. This surrogate error function is employed in SKM to handle concept drift without explicitly detecting it. The surrogate

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

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

C - Set of Centroids

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

ϱ - memory parameter

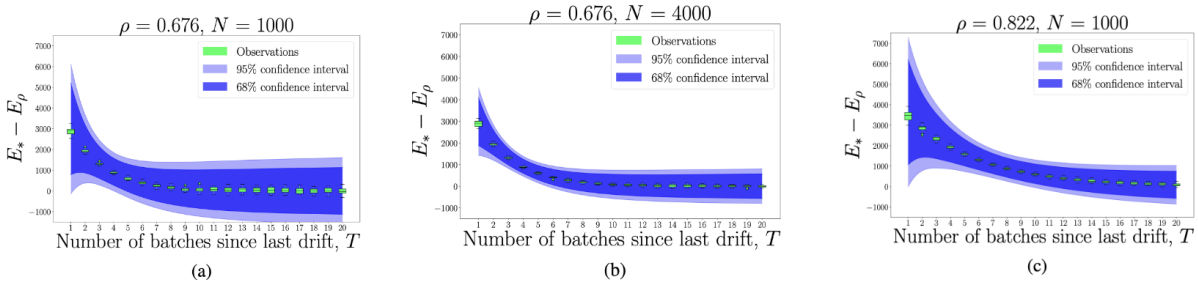
t - describes the antiquity of each batch

error function in SKM assigns exponentially decaying weights to older batches, giving higher importance to recent data. This approximation allows SKM to prioritize the most

up-to-date information while still considering past batches. By minimizing this surrogate error function, SKM guides the clustering process to adapt to the evolving data stream.

The key property of the surrogate error function is that it serves as a good approximation of the true SKM error. This approximation is established through theoretical analysis, often utilizing statistical tools such as Hoeffding's inequality. Theoretical proofs demonstrate that the surrogate error captures the underlying characteristics of the SKM error and exhibits similar behavior. By using this approximate error function, SKM avoids the need for explicit concept drift detection, which can be challenging and computationally expensive in streaming scenarios. Instead, the clustering algorithm can continuously update the clustering solution based on the weighted contributions of recent and past batches, adapting to changes in the data distribution over time. It is important to note that the quality of the approximation depends on the specific characteristics of the data stream and the chosen weighting scheme. The choice of the surrogate error function and the decay rate of the weights can impact the accuracy of the approximation.

Results



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

Skm error vs surrogate error the expected value of the alternative error function tends to the SKM error function exponentially fast with T for a single centroid lower values of ρ makes the expected difference between the SKM error and the approximation tends to zero faster. In other words, the bias of the surrogate as an estimate of the SKM error decreases faster for smaller values of ρ. , lower values of ρ implies broader bounds of E∗−Eρ. Thus, the variance of the surrogate estimate is higher as ρ decreases. IT is seen that the surrogate function can be used to approximate the error for a single centroid, thus applying this result to every subgroup of points and their respective centroids yields a good approximation of the SKM error Due to the exponential decrease of the weights as antiquity t increases, the contribution to the approximated error of older batches rapidly becomes negligible. Therefore, in practise, we can compute an arbitrarily accurate surrogate error function by considering the last Tmax batches. By using this approximation, we deal with the issue of indefinite increasing volume of data.

Future Scope

The Streaming K-means (SKM) algorithm presents a promising approach for clustering evolving data streams. However, there are several potential areas for future research and development to further enhance its effectiveness and applicability. One area of future research could focus on refining the approximation techniques used in SKM. Investigating alternative surrogate error functions and weighting schemes may lead to improved accuracy in capturing the true error of the clustering solution. Additionally, exploring advanced statistical techniques or machine learning algorithms to estimate the error in the presence of concept drift could provide more robust and adaptive solutions. Another important direction for future work is the development of efficient and scalable algorithms for handling large-scale streaming data. As data streams continue to grow in volume and velocity, SKM algorithms need to be optimized to handle such big data scenarios. This could involve exploring parallel processing techniques, distributed computing frameworks, or streaming data processing architectures to ensure real-time and scalable clustering performance.

Finally, empirical evaluations on real-world datasets and benchmarking SKM against other state-of-the-art clustering algorithms can provide valuable insights into its strengths and weaknesses. Comparative studies and performance evaluations can help identify scenarios where SKM excels and identify areas for further improvement. In conclusion, the future scope for SKM lies in advancing approximation techniques, addressing scalability challenges, incorporating domain knowledge, exploring extensions, and conducting comprehensive empirical evaluations. By addressing these areas, SKM has the potential to become a more powerful and robust tool for clustering evolving data streams and find applications in a wide range of domains and industries.

References

1. A. Bidaurrazaga, A. Pérez and M. Capó, "K-means for Evolving Data Streams," 2021 IEEE International Conference on Data Mining (ICDM), Auckland, New Zealand, 2021, pp. 1006-1011, doi: 10.1109/ICDM51629.2021.00114 .
2. Dua and C. Graff, “UCI Machine Learning Repository,” 2017.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.