USC University of Southern California

USC Viterbi School of Engineering

Reading Report 5

Streaming Hierarchical Clustering for Concept Mining

Introduction

- Concept Mining:

  Extracts meaningful associations from large, unstructured data streams

- Problem Statement:

  Existing systems struggle with real-time clustering for vast data, especially in bioinformatics and network analysis

- Streaming Hierarchical Clustering:

  Enables dynamic, real-time document organization into a hierarchy

- Key Innovation:

  Hardware-compatible, handles high data ingestion rates without needing pre-trained models, making it ideal for intelligence and evolving data streams

Key Approach

- Hierarchical Clustering:
  - Automatically organizes incoming data into a multi-level structure, capturing relationships between documents as they are processed in real time
  - This method enables a continuously evolving, flexible document hierarchy
- Real-time Adaptation:
  - The system dynamically adjusts clusters as new data arrives, effectively managing concept drift by reshuffling and reorganizing clusters to reflect the most current associations in the data stream
- Hardware Efficiency:
  - Leverages specialized hardware, such as Xeon processors and FPGA implementations, to process high data ingestion rates while maintaining computational efficiency
  - This approach ensures low-latency clustering, even with large datasets

Hierarchical Partitioning

- Dynamic Partitioning
  - The method organizes data into hierarchical partitions based on similarity, uncovering patterns at various granularity levels
- Adaptive Reconstruction
  - Clusters continuously adjust as new data arrives, keeping the hierarchy up-to-date, essential for fast-paced environments
- Scalability
  - Built to efficiently manage large datasets, this approach supports applications in social media and real-time network monitoring without sacrificing performance

Experimental Results

- K-means Clustering
  - Groups data into K clusters, minimizing intra-cluster distance
  - Uses distance metrics like Minkowski, Manhattan, Euclidean, and Cosine theta
- Algorithm Steps
  - Assign documents to K groups
  - Calculate centroids
  - Update assignments until stable
- Bisection K-means
  - Splits clusters iteratively until reaching the desired number
  - Confusion matrices show k-means underperformance compared to hierarchical methods

Hardware Design

- FPGA Implementation
  - Optimized for integer arithmetic to boost parallelism and avoid precision loss
- Performance Optimization
  - Bitmap packing reduces memory bandwidth and enhancements for vector summation and dot product tasks yield up to eightfold speedup
- Future Enhancements
  - Potential for up to 45 times faster performance using 64-bit instructions and signal processing extensions

Streaming Hierarchical Partitioning

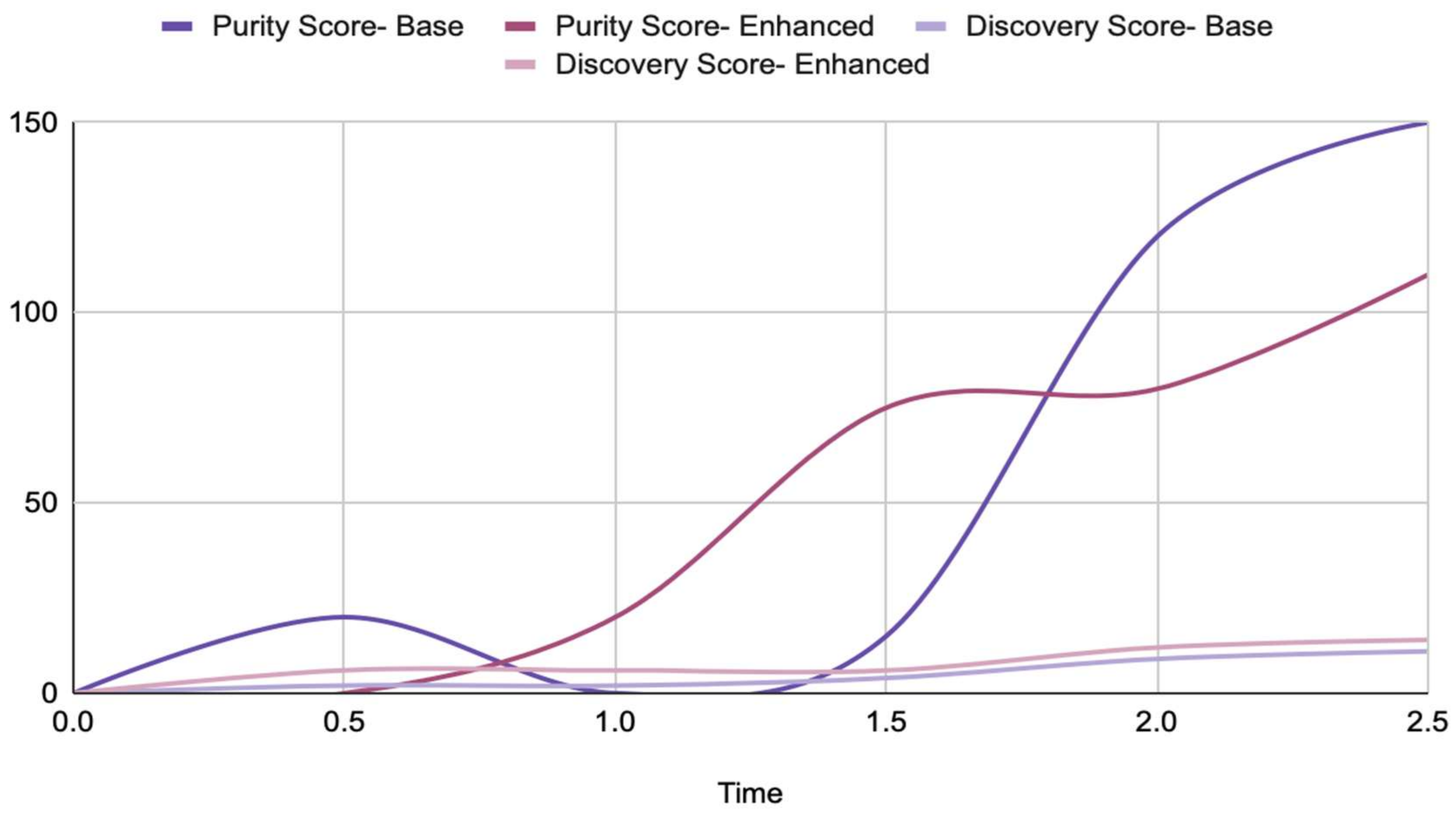
- Overview
  - Adapt hierarchical partitioning for evolving document streams
- Assumptions
  - Limited memory (m documents) and processing capacity
  - New document requires removal of an existing one
- Clustering Strategy
  - Recluster every tt time-steps
  - Efficient insertion/removal using cosine similarity and greedy matching

Streaming Experimental Results

- Setup
  - Newsgroup data streamed to simulate concept drift
- Parameters
  - Max memory: 1000 documents
  - Reclustering: After processing 1000 documents
- Evaluation Metrics
  - Purity Score: % of documents in pure clusters
  - Discovery Score: Counts unique label clusters over time
- Results Comparison
  - Non-naive method outperforms naïve in concept discovery while maintaining purity

USC Viterbi
School of Engineering

Streaming Experimental Results



Quiz Question

- **Question:** What is a primary attribute of Streaming Hierarchical Partitioning when used for document clustering?
- **Answer:**
  - Streaming Hierarchical Partitioning does not store an infinite number of documents in memory. Instead, it uses cosine similarity to efficiently insert documents and requires the removal of existing documents to make space for new ones. This approach does not perform clustering only once at the end of the document stream; rather, it continuously updates the clusters as new documents arrive. Additionally, it does not rely on traditional k-means clustering without adaptations; it is designed to handle streaming data effectively.