



Unlocking Insights: Streamlining Concept Mining in Big Data

Revolutionizing Data Analysis through
Scalable Methods and Hardware Acceleration

Background





Scalability Challenge

- Traditional methods for concept mining face scalability issues.
- Fields experiencing information explosions like network traffic analysis, bioinformatics, and intelligence are particularly affected.
- Extension of concept mining methodology is necessary to improve performance and permit hardware implementation.



Hardware Acceleration for Classification

- Hardware-accelerated systems have proven effective in automatically classifying content.
- Especially successful when topics are known in advance.
- Signifies the importance of novel algorithmic approaches leveraging hardware acceleration for real-time processing of large data streams.



Objectives

System is designed to achieve three primary goals

- Real-time concept mining of high-volume data streams
- Dynamic organization of concepts into a relational hierarchy
- Adaptive reorganization of the concept hierarchy in response to evolving circumstances and user feedback.



Introduction to Streaming Hierarchical Clustering

Novel Algorithmic Approach:

- Developed algorithm for extracting semantic content from large data streams.
- Applicable to multilingual documents and various encodings.

Motivation:

- Increasing data volume surpassing software processing capabilities.
- Need for hardware-accelerated approaches for efficient data analysis.

Hardware Acceleration:

- Emphasized development of hardware-accelerated methods for content detection at high speeds.



Introduction to Streaming Hierarchical Clustering

Extension to Hierarchical Clustering:

- Expanded previous work to include hierarchical clustering without prior training

Streaming Hierarchical Clustering:

- Introduction of novel concept for real-time hierarchical clustering of documents.
- Represents a subfield in emerging areas like "streaming AI" and "AI in hardware."

Example Applications:

- Network traffic analysis
- Bioinformatics

Hierarchical Partitioning





Introduction to Hierarchical Partitioning

Historical Context:

- Principles of hierarchical organization date back to Aristotle and have been refined over time.

Document Clustering Overview:

- Documents tend to cluster hierarchically based on similarity, mirroring natural categorization principles.

Agglomerative vs. Divisive Approach:

- Agglomerative-Cluster: Bottom-up merging of closest clusters.
- Divisive-Cluster: Top-down division of the collection into smaller parts recursively.



Superiority of Divisive Approach

Empirical Findings:


- Studies show that divisive clustering, considering global statistics, often outperforms agglomerative clustering in document clustering tasks.

Distance Metrics and Normalization:

- Preference for normalized similarity measures to accommodate varying document lengths and ensure robust performance.

Specifics of Hierarchical Partitioning:

- Representation with low dynamic range and binary dimensions.
- Expected presence of chaff documents necessitates creation of large, loose "junk" subtrees.



Cluster Quality Measurement and Partitioning Heuristic

Centroid Computation:

- Centroid vector aids in cluster analysis and classification, representing the overall makeup of a cluster.

Cluster Quality Measurement:

- Affinity and score functions quantify the quality of clusters and partitioning, respectively.

Partitioning Heuristic:

- Simple iterative approach maximizes division quality, ensuring termination and efficiency.



Experimental Comparison and AFE Document Vector Representation

Experimental Comparison:

- Hierarchical partitioning compared to k-means and bisection k-means clustering algorithms using the CMU-20 dataset mixed with "chaff" documents from talk.origins newsgroup.

AFE Document Vector Representation:

- **Dimensions and Counters:**
 - Document vectors have fixed dimensionality (e.g., 4000 dimensions).
 - Each dimension contains counters representing word presence.
- **Word Mapping Table (WMT):**
 - Dynamically maps words to specific dimensions in vectors.
 - Training-based approach for adaptability to different datasets.
- **Role of WMT:**
 - Organizes words into dimensions, capturing semantic relationships.
 - Facilitates efficient representation by reducing dimensionality.
- **Vector Construction:**
 - Consults WMT to construct document vectors.
 - Dimensions incremented based on word presence, resulting in sparse representations.

Experimental Results





Clustering Algorithms Comparison

k-Means Clustering Algorithm:

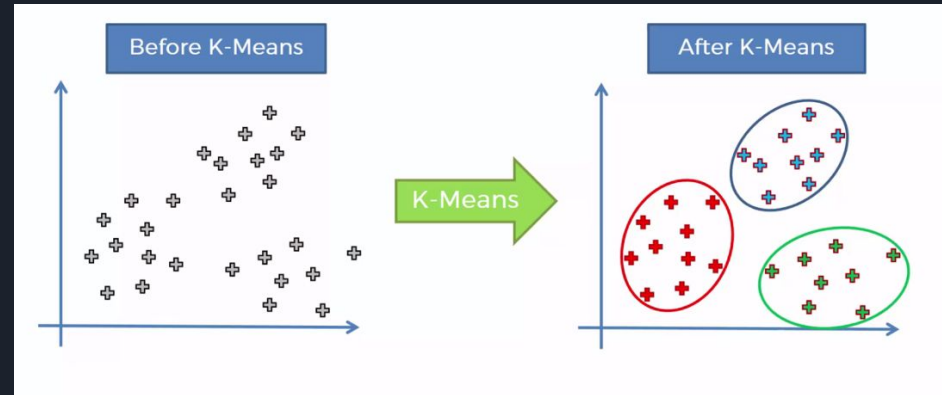
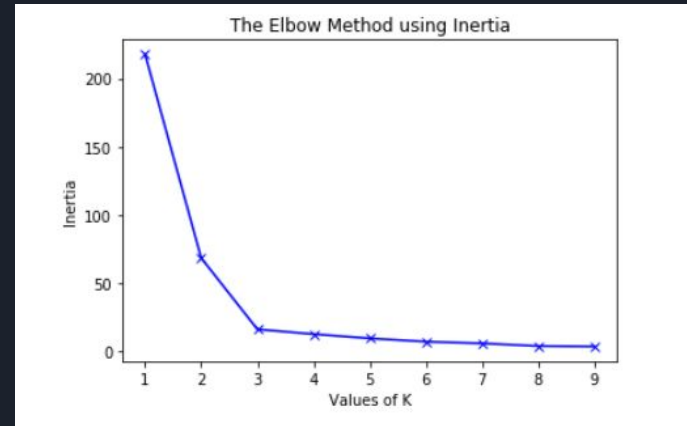
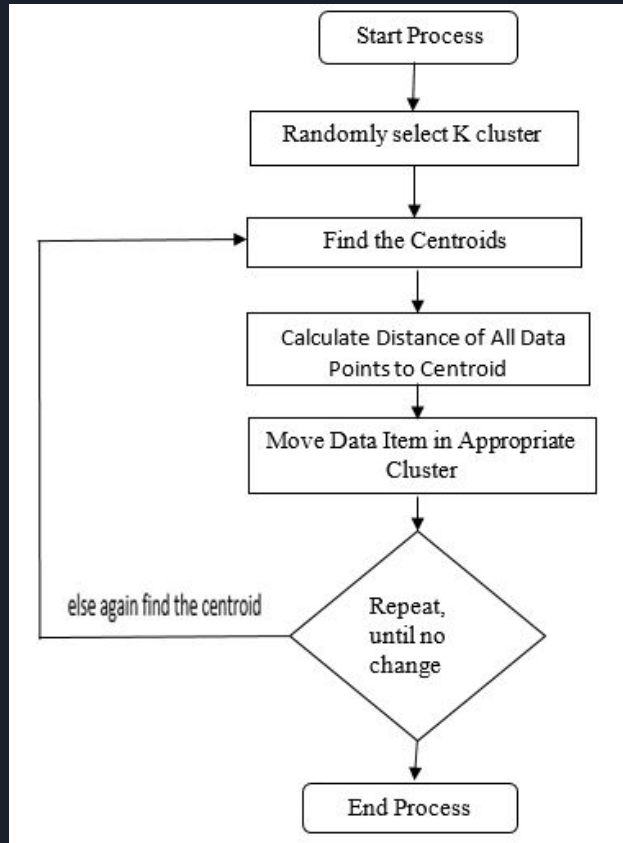
- Separates data into K groups with predefined K.
- Documents assigned to clusters based on centroid proximity.
- Minimizes inner cluster distance and inter cluster distance.
- Distance calculation methods include Minkowski, Manhattan, Euclidean, and cosine theta.
- Cyclical algorithm iteratively adjusts cluster assignments until convergence.

Bisection k-Means Variant:

- Starts with a single cluster and splits clusters into two sub-clusters iteratively.
- Cluster selection for splitting typically based on the cluster with the most elements.
- Continues until the desired number of clusters is reached.

Hierarchical Partitioning:

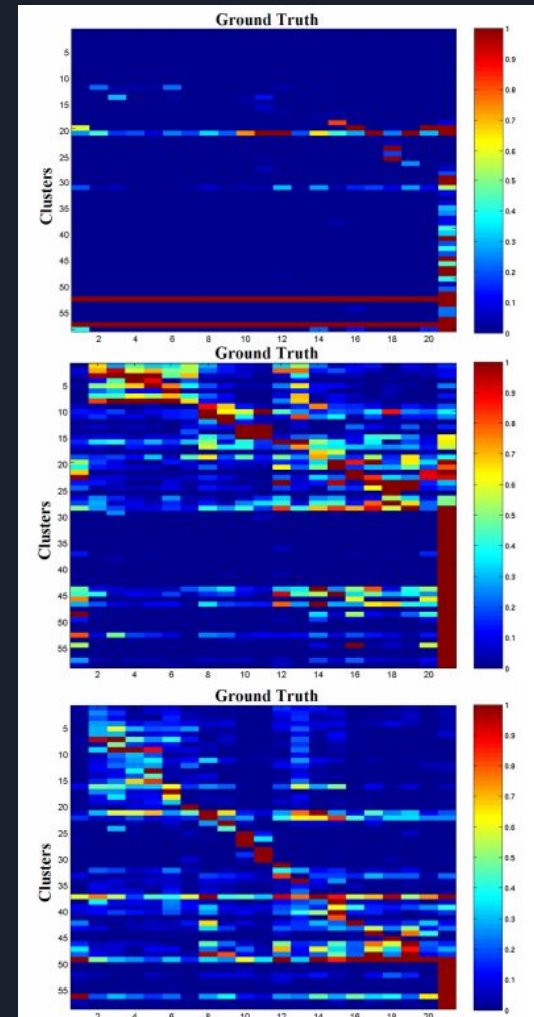
- Hierarchical clustering approach aiming to maximize cluster quality.
- Trees flattened to obtain k clusters for comparison with flat clustering.
- Subtrees chosen based on quality measures to form non-overlapping clusters.



k-Means Clustering Algorithm

Comparison Results

- Confusion matrices depict clustering results for k-means, bisectional k-means, and hierarchical partitioning on a dataset of 23,845 newsgroup postings.
- Hierarchical approaches outperform k-means, producing more diverse and informative clusters.
- Bisection k-means and hierarchical partitioning yield comparable results, but hierarchical partitioning generates fewer "junk" clusters dominated by chaff, enhancing usability for human analysts.



Hardware Design Optimizations





Design Optimizations

Objective:

- Enhance hierarchical partitioning algorithm for FPGA implementation without floating-point numbers.

Approach:

- Utilized integer arithmetic to increase parallelism on FPGAs.
- Mapped algorithms to integer arithmetic to prevent precision loss.

Optimization Techniques:

- Implemented classical optimization techniques.
- Applied bitmap packing, vector summation, and dot product calculation optimizations.

Performance Enhancement

Bitmap Packing:

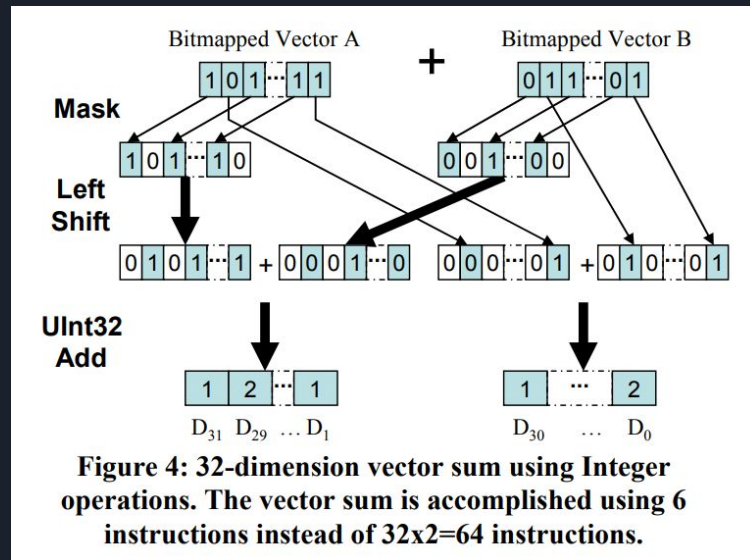
- Reduced storage and memory bandwidth requirements by 1/32th of the original size.

Accelerated Tasks:

- Vector summation and dot product tasks optimized for performance.

Significant Performance Gains:


- Enhanced performance particularly for sparse datasets.





What else can be done?

- Future directions include scalable optimizations and exploration of signal processing extensions.
- Reconfigurable devices like FPGAs offer potential for significant performance improvements.
- Continued research aimed at maximizing efficiency and performance in hardware implementation.



Streaming Hierarchical Partitioning

Overview

Objective:

- The objective is to adapt the hierarchical partitioning clustering approach for handling evolving document streams.

Assumptions:

- The document stream is effectively infinite and presented sequentially.
- The system has finite working memory capacity (m) and limited processing resources.

Approach:

- Reclustering of all documents occurs every t time-steps using hierarchical partitioning.



Document Insertion and Similarity Calculation

Document Insertion:

- Efficient insertion is achieved using tree-structured vector quantization, a method akin to greedy-descent matching.
- New documents are inserted into the concept hierarchy by finding similar leaf nodes and replacing them with internal nodes.

Similarity Calculation:

- Cosine-theta measure is utilized to compute similarity between document vectors and centroids.
- This measure ensures robust similarity calculation and is hardware-feasible, enhancing performance.

Document-Insert(Document D, Tree T):

Let $L = \text{find-Similar-Leaf}(D)$

Replace L with an internal node with children L and D

Find-Similar-Leaf(Document D, Tree T):

If T is a single leaf, return it

Compute similarity of D to

centroids of T 's left and right subtrees

Let $S =$ subtree with highest similarity

flipping a fair coin in case of a tie

Return $\text{Find-Similar-Leaf}(D, S)$



Document Removal Strategies and Concept Drift

Concept Drift Regimes:

- The system assumes a gradual drift in the distribution and semantics of concepts over time.

Document Removal Heuristics:

- Strategies for document removal are based on two main criteria: similarity and recency.
- Documents are considered for removal if they are very similar to others in memory or have not been seen for a long time.
- Probabilistic removal based on similarity and recency can be implemented to strike a balance between the two criteria, managing the tradeoff effectively.

Balancing Similarity and Recency:

- The tradeoff between similarity and recency is crucial in deciding which documents to remove.
- By probabilistically selecting documents for removal based on both criteria, the system can maintain a balance between memory efficiency and retaining relevant documents.

User Feedback Approach:

- User feedback may be solicited to guide the system in directing computational effort and improving clustering performance.

Streaming Experimental Results





Setup & Evaluation

Experimental Setup:

- Newsgroup data is streamed with a simulated concept drift regime.
- Documents are randomly shuffled, gradually introducing new newsgroups into the distribution while reducing the density of old newsgroups.
- Chaff (unwanted or irrelevant documents) is uniformly distributed throughout the datastream.

Parameters and Evaluation:

- Two main parameters are set for streaming hierarchical partitioning: the maximal number of documents stored in memory and the frequency of reclustering.
- Parameters are set to 1000 documents stored in memory, reclustering after processing every 1000 documents.
- Evaluation focuses on concept quality and concept discovery over time.



Concept

Concept Quality:

- Measured by the purity of clusters, where clusters with at least 90% of documents from a single newsgroup are considered.
- Only clusters with more than 10 documents are included in the purity score.

Concept Discovery:

- Assessed by the number of pure clusters created over time, corresponding to unique non-chaff labels.
- Cumulative measure indicating how effectively new concepts are identified as concept drift occurs.

Comparison of Streaming Clustering Methods

Naïve vs. Non-naïve Streaming Clustering:

- Naïve streaming clustering involves random document removal
- Non-naïve clustering incorporates intelligent heuristics for document insertion and removal.
- Comparative purity and discovery scores are shown, with a focus on the effectiveness of concept discovery.

Results

- Both methods show similar effectiveness in terms of purity scores, as batch clustering determines this score.
- Non-naïve streaming hierarchical partitioning consistently outperforms the naïve variant in concept discovery, suggesting its superiority for concept mining applications with evolving concepts over time.

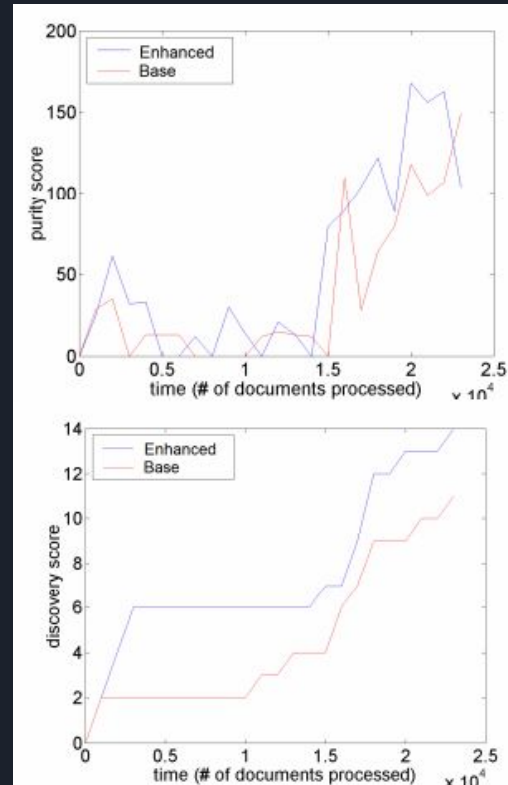


Figure 7. Purity (top) and discovery (bottom) for naïve (base) and non-naïve (enhanced) streaming hierarchical clustering.



Conclusion

System Extension:

- Developed a clustering system, streaming hierarchical partitioning, as an extension to prior work on semantic content extraction from unstructured text document streams.
- Designed to handle high ingestion rates and implemented for hardware deployment.

Hardware Implementation:

- Provided detailed hardware-ready design with performance predictions for clustering quality and concept discovery.
- Prototyped and tested on both Xeon processor and PowerPC embedded within a Xilinx Virtex2 FPGA.



Future Work

Integration with Classification System:

- Clustering to be integrated into existing classification system for enhanced data analysis.

Continuous Concept Discovery:

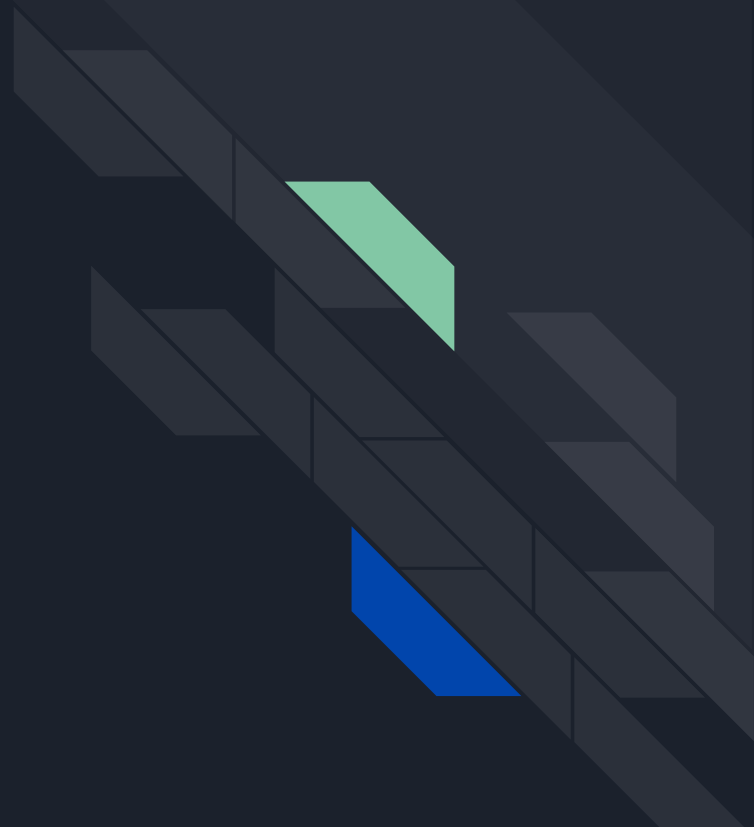
- System to continually search for new and emerging concepts in data streams.

Adaptation for Infinite-Length Data Sets:

- Allow resolution of concepts to fade over time to accommodate streaming with infinite-length data sets.

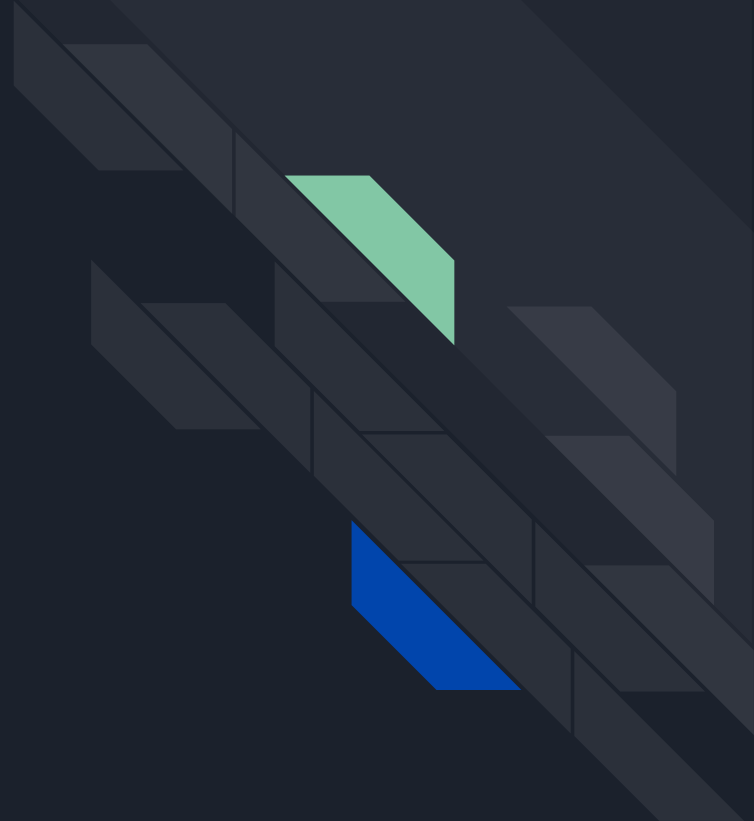
In the context of streaming hierarchical partitioning, how is concept drift defined?

1. Gradual change in the distribution and semantics of concepts over time
2. Abrupt shift in concepts with no continuity
3. Uniform distribution of concepts over time
4. Gradual introduction of new concepts



A

Gradual change in the distribution and semantics of concepts over time



THANK YOU

