PROJECT-7 PROPOSAL: INCREMENTAL CLUSTERING

Aryan Bansal
IIIT Hyderabad
aryan.bansal@research.iiit.ac.in
2021111018

Pratham Kumar Mishra
IIIT Hyderabad
pratham.mishra@students.iiit.ac.in
2021102036

ABSTRACT

In today's data-driven world, the ever-increasing volume of information necessitates a shift from traditional batch clustering to more agile incremental clustering techniques. These methods process data elements one at a time, thereby conserving resources and enabling real-time analysis. In our semester project, we delve into the formal analysis of incremental clustering methods, with a primary focus on their ability to identify distinct cluster structures. Our project aims to implement the incremental clustering algorithm as proposed in *Ackerman, M. and Dasgupta, S., 2014*[1]. The incremental setting exhibits limitations compared to the batch model, as it cannot readily detect certain cluster structures that are easily identifiable in batch processing. We compare the implemented algorithm against other incremental clustering algorithms using the standard evaluation datasets.

1 INTRODUCTION

In today's data-centric world, clustering is a foundational tool utilized across diverse domains, from scientific research to marketing. However, as data continues to grow exponentially, traditional offline batch clustering approaches become impractical due to the constant influx of new information. This semester project endeavors to address this challenge by implementing and studying incremental clustering methods for real-time data analysis.

Objectives:

1. **Algorithm Implementation:** We will implement the clustering algorithm proposed in the research paper, which is designed to perform clustering in an online, incremental manner.
2. **Visual Blog:** To disseminate our findings effectively, we will create a visually rich blog. This blog will not only explain how the algorithm functions but also provide insights into the key ideas, intuitions, and mathematical foundations behind it. This educational resource will make the complex topic of incremental clustering accessible to a wider audience.
3. **Comparative Analysis:** Our project will include a comprehensive comparison of the proposed incremental clustering algorithm against other incremental clustering methods. We will use standard evaluation datasets to assess the performance, efficiency, and effectiveness of our implementation in real-world scenarios.

2 PROOF OF CONCEPT OUTLINE

1. **Project Setup:**
   - Define the project objectives and goals, as outlined in the project description.
   - Establish a project timeline and milestones to ensure progress tracking.
   - Set up a version control system (GitHub) for collaborative development.

- Create a dedicated project repository to host code, documentation, and resources.

2. **Literature Review:**
   - Conduct a comprehensive review of relevant literature on clustering algorithms, especially focusing on incremental clustering methods.
   - Summarize key concepts, algorithms, and evaluation metrics related to clustering.
   - Identify the incremental clustering algorithm proposed in the selected research paper and understand its principles.
3. **Algorithm Implementation:**
   - Begin by implementing the incremental clustering algorithm based on the paper's description.
   - Ensure the algorithm can handle data arriving incrementally and update cluster assignments accordingly.
   - Develop code comments and documentation to aid understanding and future use.
4. **Data Preparation:**
   - Collect or obtain suitable datasets for evaluation.
   - Ensure the datasets represent scenarios where data is continuously added or updated.
   - Preprocess the data as needed, handling missing values or outliers.
5. **Blog Creation:**
   - Begin creating the visually rich blog that explains the functioning of the implemented algorithm.
   - Use visualizations, diagrams, and code snippets to illustrate key ideas, intuitions, and mathematical formulations.
   - Incorporate real data examples to demonstrate how the algorithm operates in practice.
   - Include explanations of any parameter tuning or configuration required.
6. **Comparative Analysis:**
   - Implement other incremental clustering algorithms for comparison.
   - Establish evaluation criteria and metrics, such as clustering accuracy, runtime efficiency, and memory usage.
   - Run experiments using the prepared datasets and record results for each algorithm.
   - Create visualizations or graphs to showcase the comparative performance of the algorithms.
7. **Documentation and Reporting:**
   - Document the implementation details, including code structure, dependencies, and usage instructions.
   - Compile the results of the comparative analysis into a report.
   - Discuss the strengths and weaknesses of the implemented algorithm and its performance relative to other methods.
   - Include references to the research paper and relevant literature.
8. **Review and Refinement:**
   - Conduct code reviews and seek feedback from peers or mentors to identify potential improvements.
   - Refine the implementation and documentation based on feedback.
   - Re-run experiments if necessary to ensure the validity of results.
9. **Final Deliverables:**
   - Submit the completed project, including the implemented algorithm, blog, and comparative analysis report.
   - Make the project repository and blog publicly accessible for others to learn from and use.

3 DATA & TECHNICAL REQUIREMENTS

Technical Requirements:

1. **Programming Language:** The primary programming language for this project will be Python, which is well-suited for data analysis and clustering tasks.
2. **Development Environment:** We will use Jupyter Notebooks (Anaconda) for code development, testing, and documentation. Jupyter Notebooks allow for interactive code execution and visualization, making it suitable for explaining the algorithm in the blog.
3. **Version Control:** We will use Git for version control and GitHub for repository hosting. This will enable collaborative development and easy tracking of code changes.
4. **Blog Framework:** The primary programming language for this project will be ReactJS (tentative), which is well-suited for dashboard creation.

Datasets:

1. **Simulated Datasets:** We plan on simulating datasets that mimic real-world scenarios where data is added incrementally. These datasets will serve as the primary input for testing and evaluating the incremental clustering algorithm.
2. **Public Datasets:** We plan on using publicly available datasets that exhibit incremental characteristics. These datasets will help us assess the algorithm's performance on real data.
3. **Synthetic Data:** Depending on the algorithm's requirements, we may generate synthetic data with known clustering structures to validate the algorithm's correctness.

REFERENCES

Margareta Ackerman and Sanjoy Dasgupta. Incremental clustering: The case for extra clusters. *Advances in neural information processing systems*, 27, 2014.