Exploring K-means Clustering Over Sliding Window in Continuous Data Stream

Presented by Team 6:

Purva Kulkarni

Rujuta Palande

Kishan Pawar

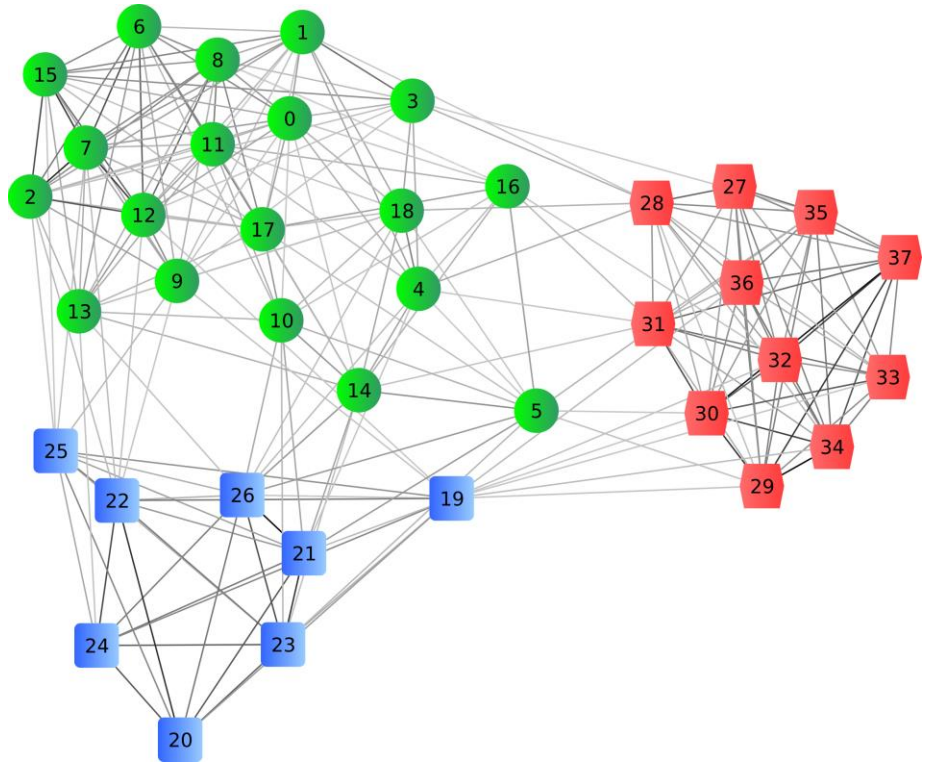
Siddharth Utgikar

Contents

- Terminology
- Previous Work
- Project
- Conclusion

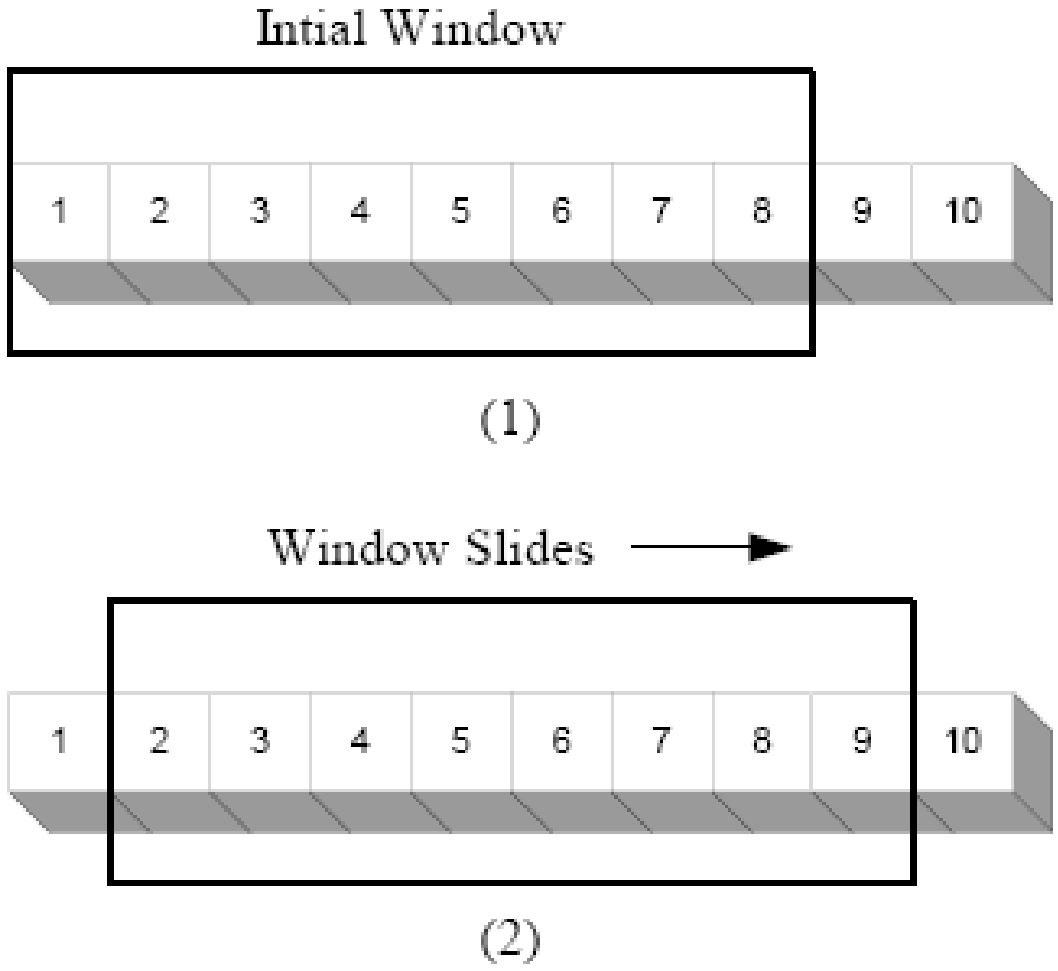
Clustering

- Method to partition data on the base of similarity
- Need
  - *Large datasets stored on secondary memory*
  - *Linear access better than random access.*
- Clustering in data streams is to find k centers such that distance from each point to its nearest center is minimized.



Sliding Window

- Stream of large data
- Data can be pipelined
- Transmit/ Process Window
- Use the latest data for processing

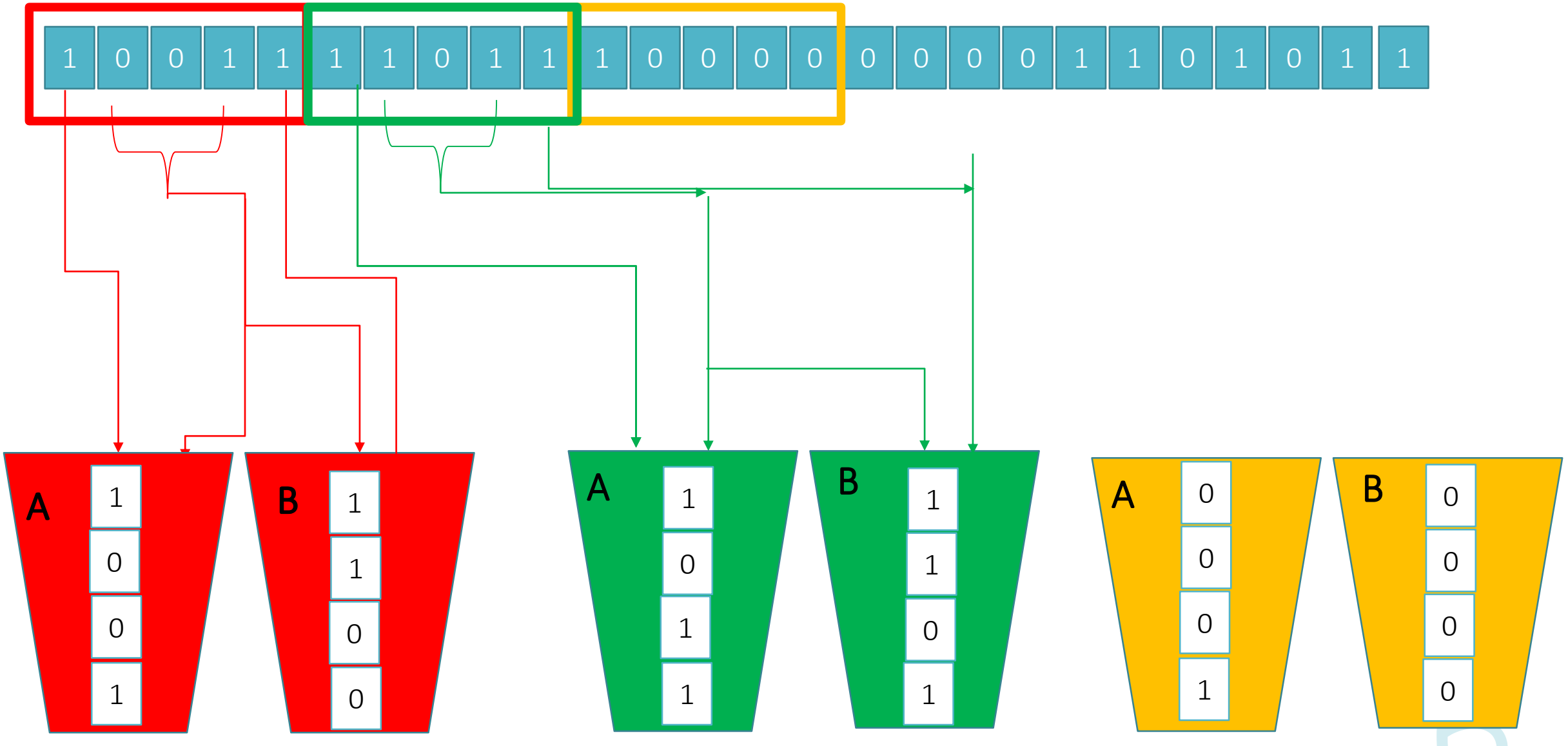


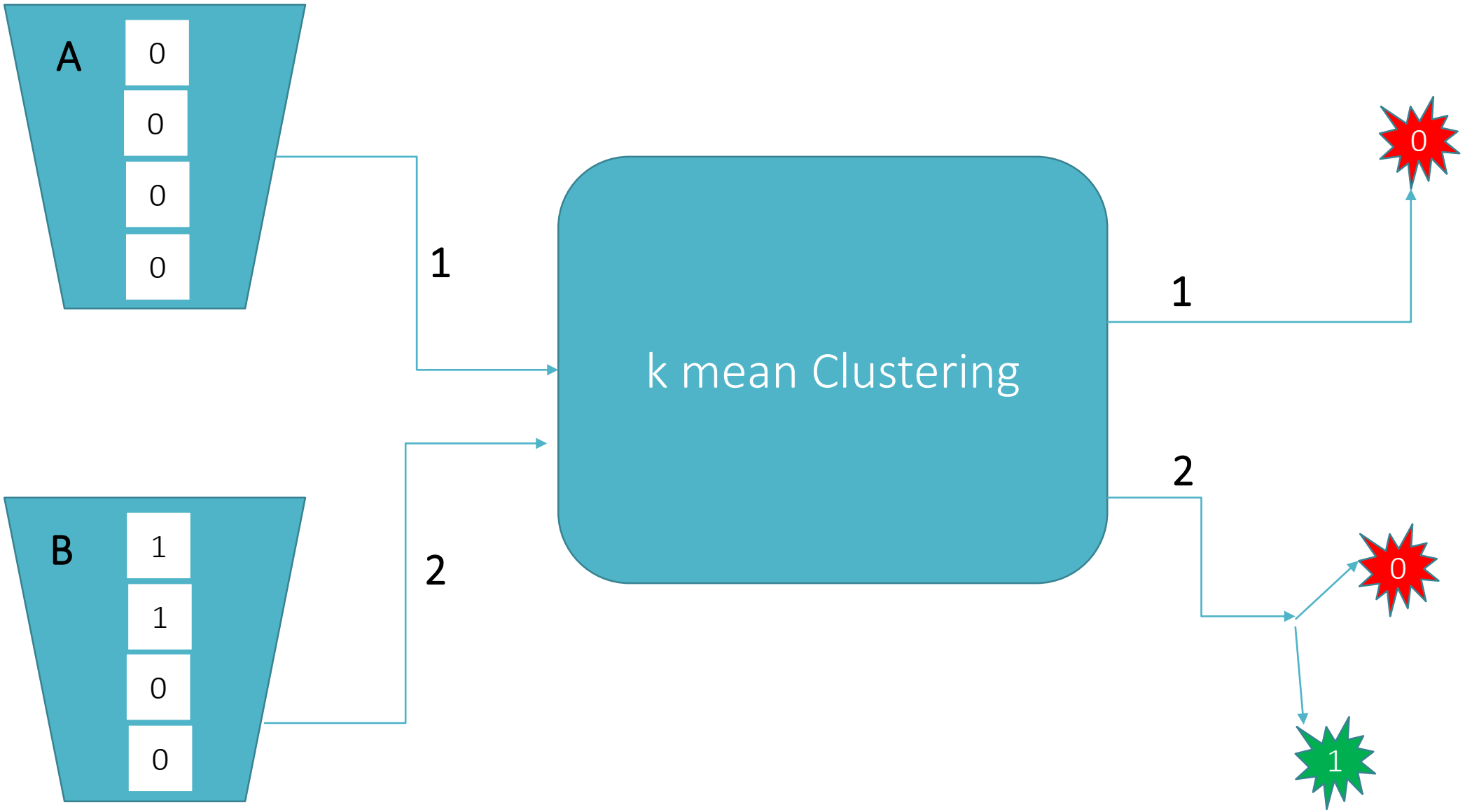
Previous Work: Clustering Problems in Sliding Windows

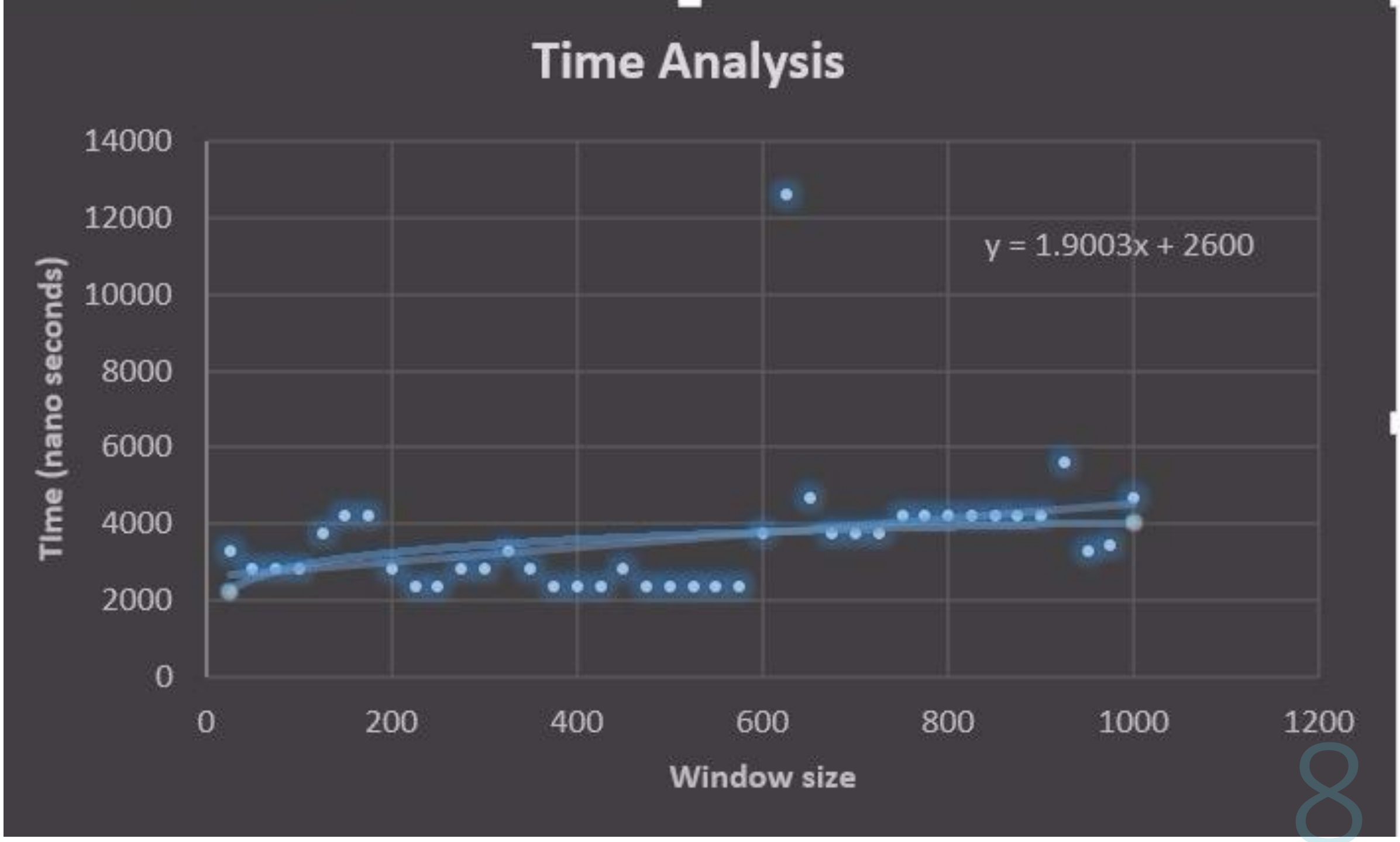
Vladmir Braverman Harry Lang Keth Levin Morteza Monemizadeh

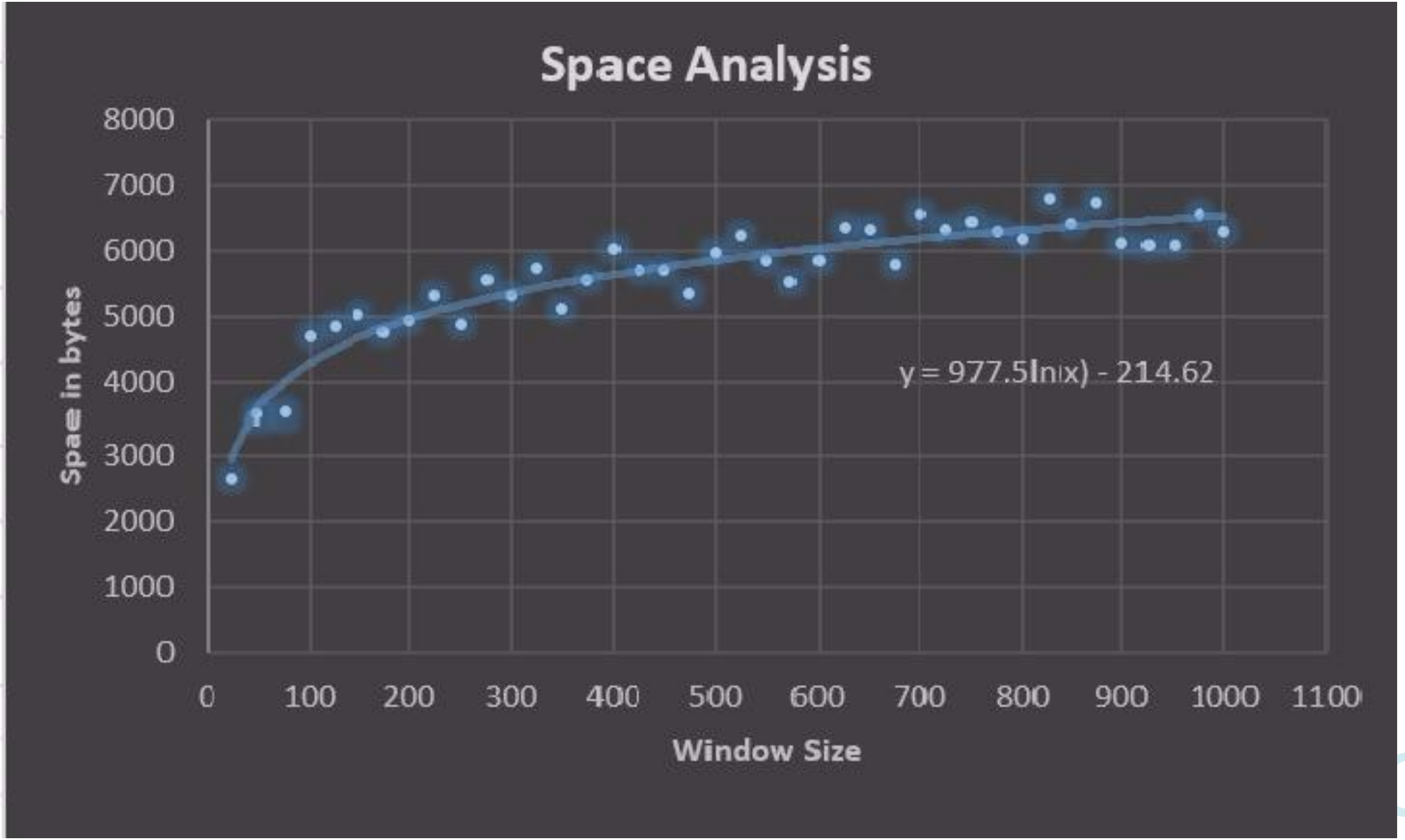
- **Main Result:**
  There exists an O(1)-approximation for the metric k-median problem on sliding windows that operates in $O(k^3 \log^6 n)$-space.
- First polylogarithmic solution to this problem
- Previous algorithm of Babcock et al. (Sliding Window Computations over Data Streams (2003)) provides solution $O(\frac{k}{\tau^4} N^{2\tau} \log^2 N)$

Bucket Algorithm




A
0
0
0
0
B
1
1
0
0
1
2
k mean Clustering
1
2
0
0
1


Time Analysis
14000
12000
10000
8000
6000
4000
2000
0
Time (nano seconds)
y = 1.9003x + 2600
0
200
400
600
800
1000
1200
Window size


Space Analysis
8000
7000
6000
5000
4000
3000
2000
1000
0
Spae in bytes
y = 977.5ln(x) - 214.62
0
100
200
300
400
500
600
700
800
900
1000
1100
Window Size

Conclusion

- Proposed Bucket Algorithm for k-means clustering over sliding window on continuous data streams
- Implemented Bucket Algorithm with value based clustering
- Regression analysis of outputs by Bucket Algorithm