SOStream: Self Organizing Density-Based Clustering Over Data Stream

Charlie Isaksson, Margaret Dunham, Michael Hahsler

Bobby B. Lyle School of Engineering,
Southern Methodist University, Dallas, Texas, USA

charlie.isaksson@tekcomms.com
{mhd, mhahsler}@lyle.smu.edu

This research was supported in part by the U.S. National Science Foundation under contract # IIS-0948893

Agenda

1. **Data streams and data stream clustering**

2. **SOStream algorithm**
   - Determine the clustering threshold
   - Online merging
   - Competitive-learning

3. **Experiments**
   - Synthetic data
   - Real-world data set
   - Sensitivity to parameters
   - Scalability and complexity

4. **Conclusions**

Data Streams

Data stream

- Unbounded sequence of data points
- Single pass restriction
- Data stream may be evolving over time

Applications

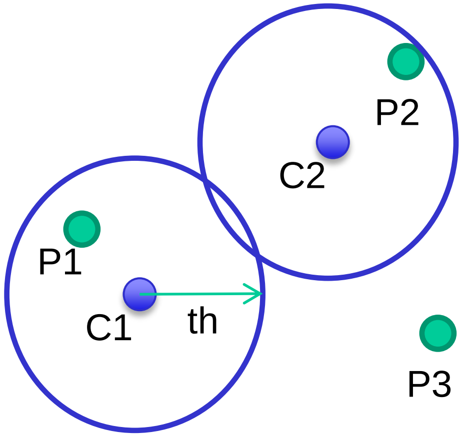
- Data streams:
  - Earth sciences (satellite data)
  - High energy physics (Large Hadron Collider), …
- Large sequence data:
  - Bioinformatics (genetics sequences), …

Data stream clustering

Typical approach:

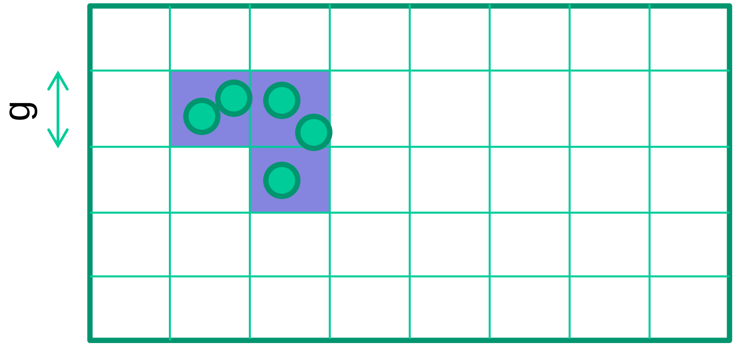
1. **Online**: Use micro-clusters (store cluster features or synopses: center, variance, weight)
2. **Offline**: Re-cluster micro-clusters into final clusters on demand.

Distance-based



e.g., CluStream, DenStream

Density-based



e.g., MR-Stream, D-Stream

SOStream

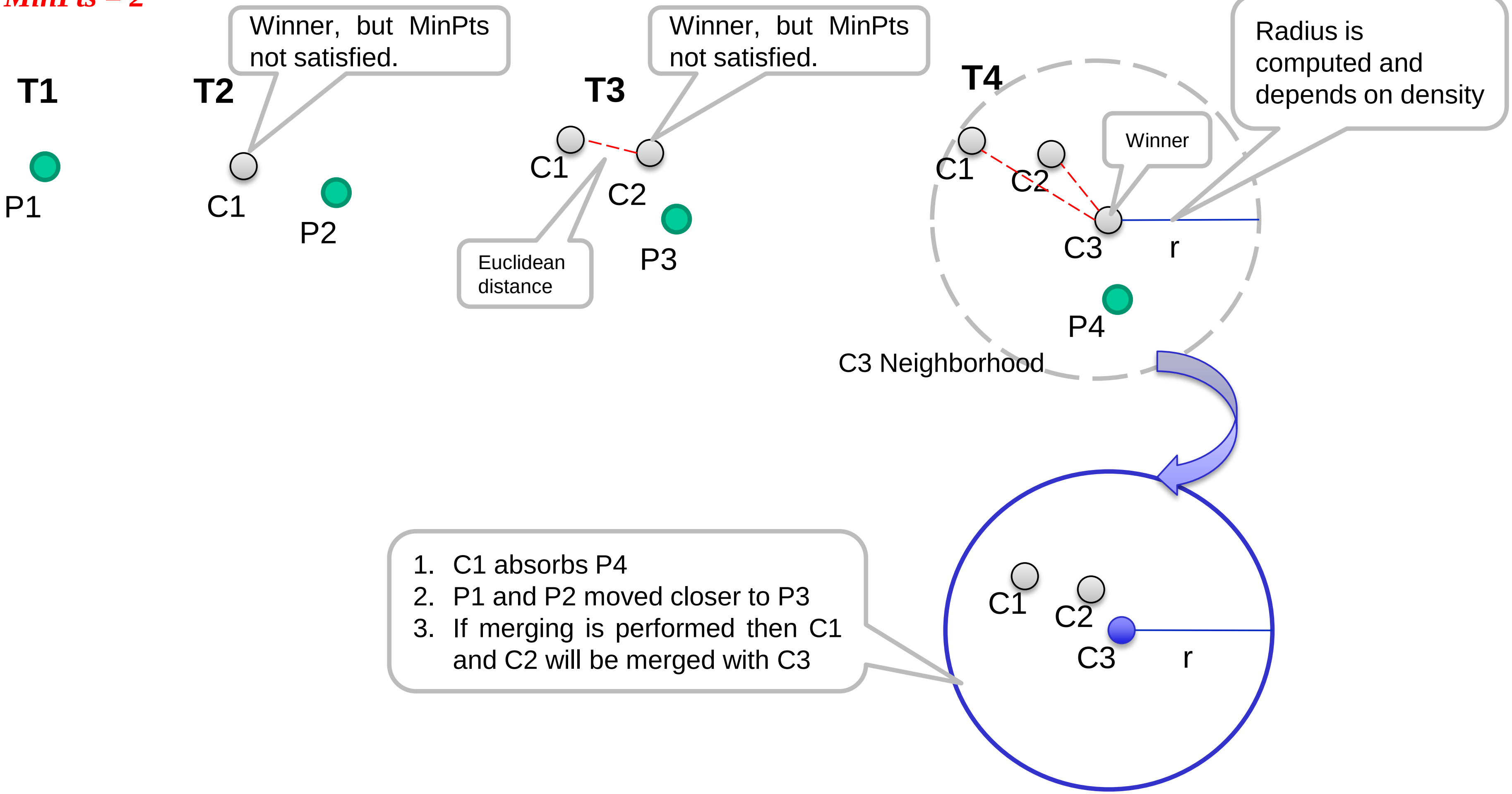
1. **How do we choose the threshold/radius or the grid size?**
2. **Can we merge micro-clusters online?**
3. **How do we deal with overlapping (real) clusters?**

SOStream uses the distance based approach.

1. Learn individual threshold for each micro-cluster using the density-based idea of the k-nearest neighbor distance (DBSCAN).
2. Use the radius for merging micro-clusters online.
3. Employ ideas from competitive learning (Self Organizing Maps).

Example of learning the radius and competitive learning

MinPts = 2



Updating clusters centroid to resemble the winning cluster

Motivated by Kohonen's SOMs [1], we propose that the centroid C_i of each cluster C_i that is within the neighborhood of the winning cluster C_{win} is modified to resemble the winner:

$$C_i(t+1) = C_i(t) + \alpha\beta\,(C_{win}(t) - C_i(t))$$

Where α is a scaling factor and β is a weight which represents the amount of influence of the winner on a cluster. We define β as:

$$\beta = e^{-\frac{d(C_i, C_{win})}{2(r_{win}^2)}}$$

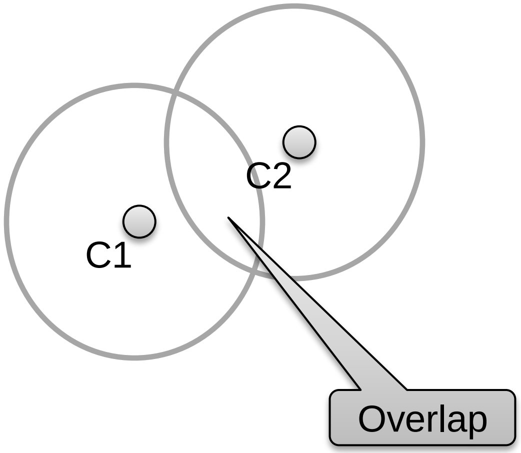
r_{win} denotes the radius of the winner. The definition of β ensures that $0 < \beta \leq 1$. This approach is used to aid in merging similar cluster and increasing separation between different clusters.

Online merging

Merging is performed online at each time step only considering the neighborhood of the winning cluster.

Clusters may change their original position over time and may result in overlap with other clusters. C_i and C_j overlap if

$$d(C_i, C_j) - (r_i + r_j) < 0$$



Online merging (cont.)

The new cluster C_y is created by finding the weight w_i and w_j of each cluster. This is achieved by:

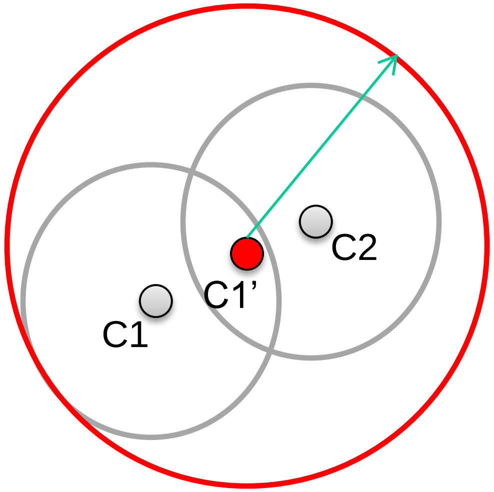
$$C_y = \frac{(w_i a_i + w_j b_i)}{(w_i + w_j)}$$

where a_i and b_i are the i^{th} dimension of the weighted centroids.

We compute the new cluster's radius r_y :

$$r_y = \max\{ d(C_y, C_i) + r_i, d(C_y, C_j) + r_j \}$$

where C_y is the new cluster centroid.



Evolving data stream

Fading of cluster structure is used to discount the influence of old data points. SOStream uses exponential decay :

$$f(t) = 2^{\lambda t}$$

where, λ define the rate of decay of the weight over time and $t = (t_c - t_0)$, t_c denote the current time and t_0 is the creation time of the cluster.

The frequency count n determines the weight of each cluster. Aging is accomplished by reducing the count over time. Any cluster that reach a defined minimum weight can be removed:

$$n_{i+1} = n_i 2^{\lambda t}$$

Experiments

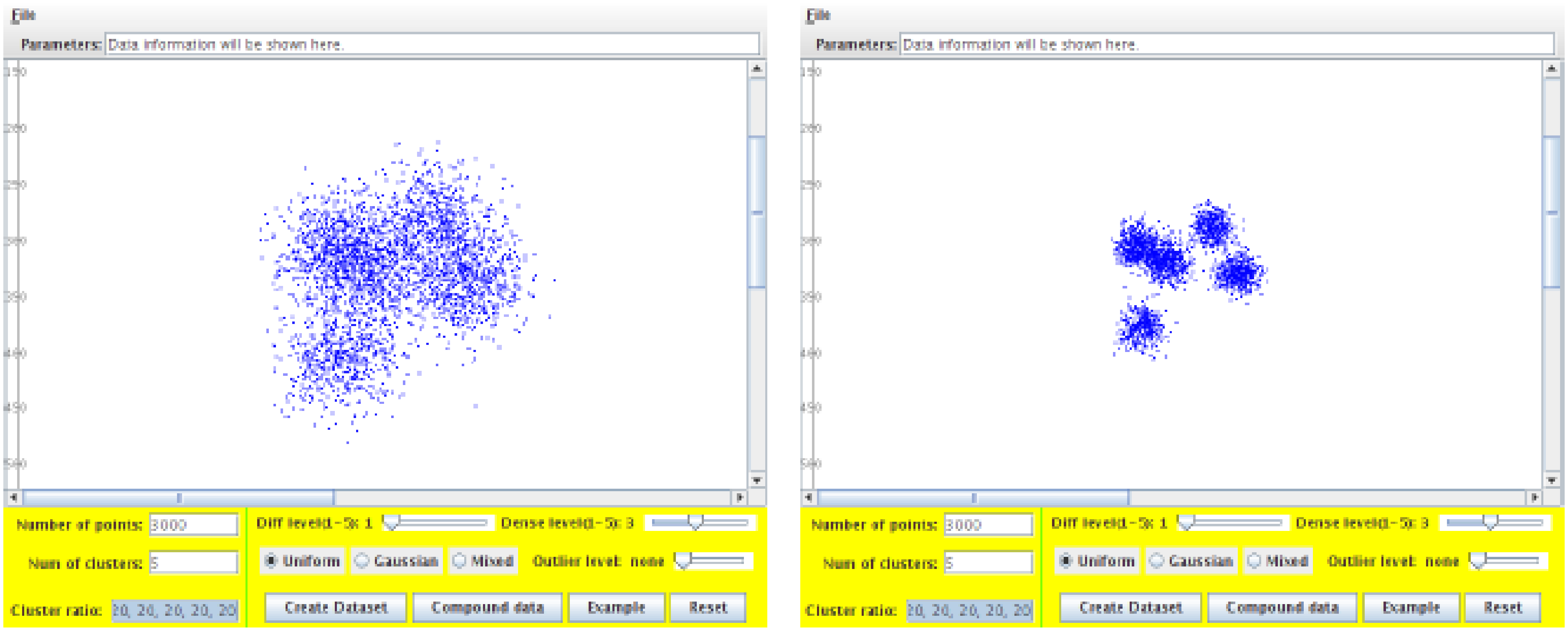
Synthetic data

- Java based dataset generator described in [2].
- 3000 data points (no added noise).
- 5 convex-shaped clusters that overlap.

Real-world dataset

- KDD CUP'99 dataset [3].
- Realistic network attacks in a Air Force base network.
- 494,000 labeled records with 34 continuous attributes

Synthetic data



(a) (b)

(a) Data points of stream with 5 overlapping clusters and
(b) show SOStream capability to distinguish overlapped cluster
($\alpha = 0.1$ and $MinPts = 2$). **No Fading or Merging where utilized**

Real-world dataset clustering quality

To compute the purity of the arriving data points are divided into 500 windows (known as horizon [5]). Average purity in window id defined as:

$$purity = \frac{1}{K} \sum_{i=1}^{K} \frac{N_i^d}{N_i}$$

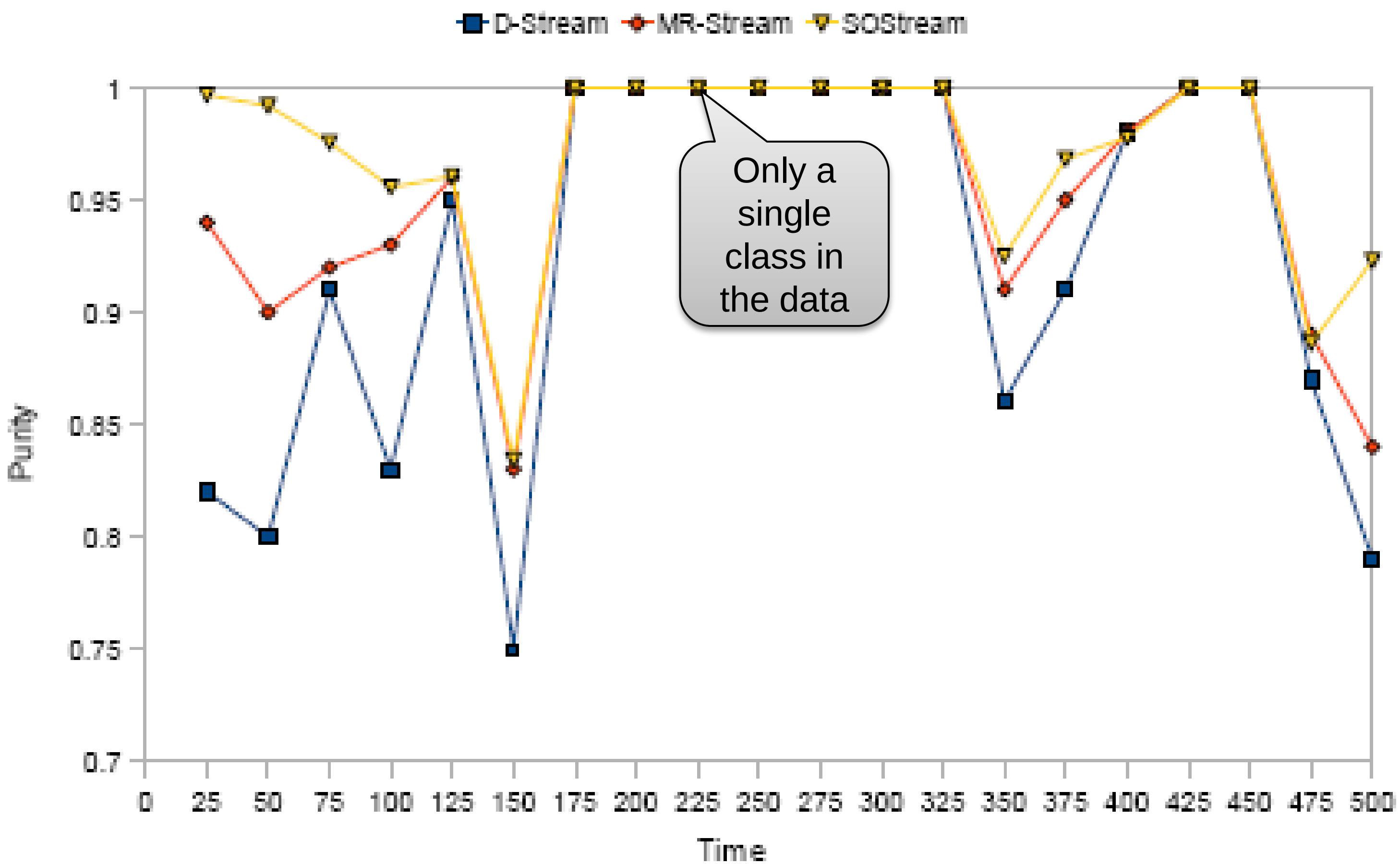
Where:

K = The number of real clusters

$|N_i^d|$ = The number of points that dominate the cluster label within each cluster

$|N_i|$ = The total number of points in each cluster

Real-world dataset quality evaluation



SOStream clustering quality evaluation, where horizon = 1K, Stream speed = 1K, $\alpha = 0.1$, $\lambda = 0.1$ and $MinPts = 2$.

Sensitivity to parameter changes

Using real-world dataset [3], we tested SOStream parameters performance with different α and *MinPts*.

	$\alpha = 0.1$			
Data Points	MinPts = 3	MinPts = 5	MinPts = 10	Mean
25000	0.983	0.990	0.921	0.965
75000	0.917	0.982	0.968	0.955
125000	0.907	0.973	1.000	0.960
175000	0.876	0.974	0.937	0.929
225000	0.876	0.974	0.937	0.929
275000	0.876	0.974	0.937	0.929
325000	0.876	0.974	0.937	0.929
375000	0.895	0.975	0.919	0.929
425000	0.907	0.975	0.963	0.949
475000	0.934	0.977	0.935	0.949
Mean	0.899	0.976	0.932	0.936

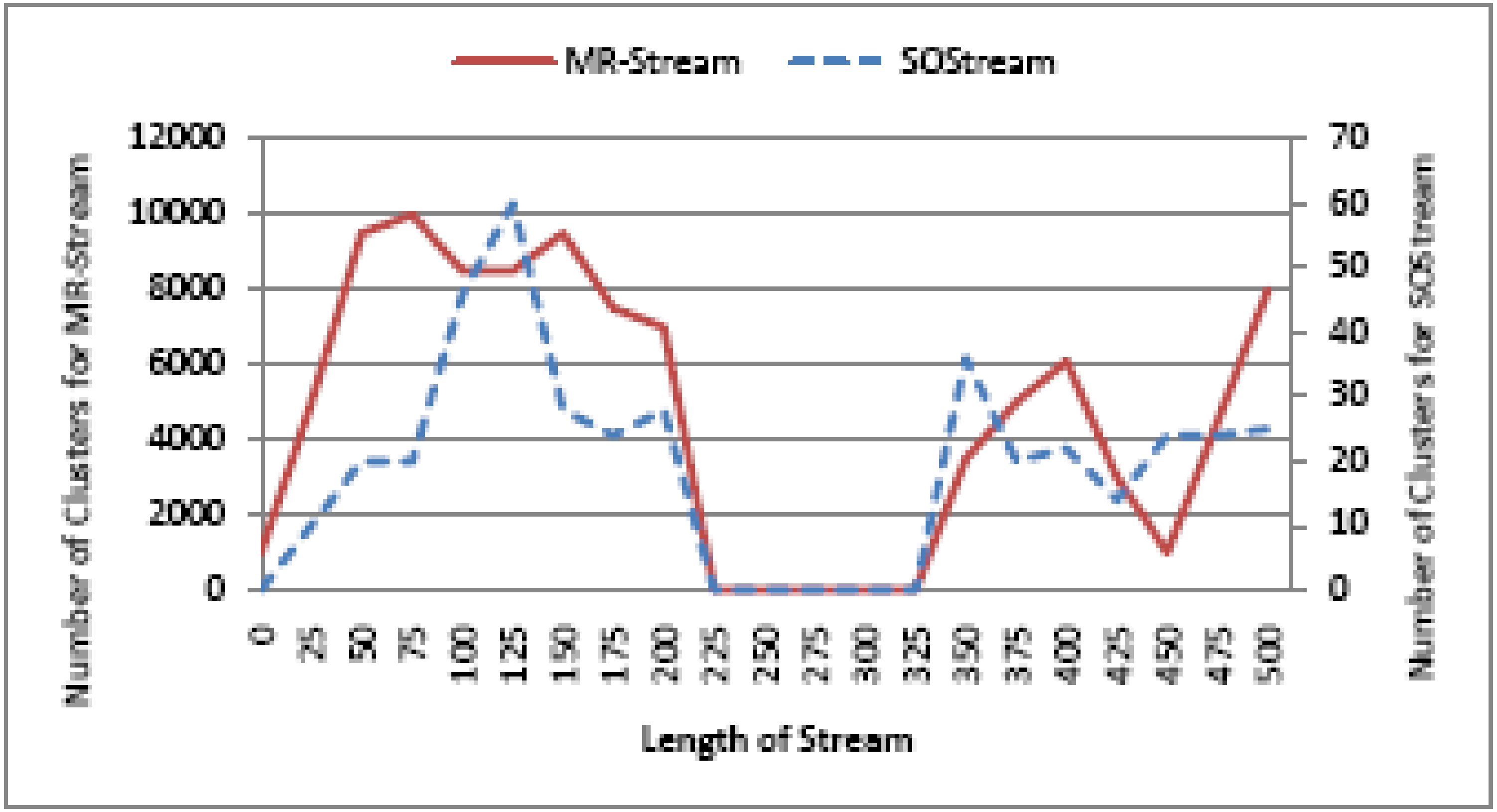
	$\alpha = 0.3$			
Data Points	MinPts = 3	MinPts = 5	MinPts = 10	Mean
25000	0.999	0.938	0.914	0.950
75000	0.998	0.996	0.962	0.985
125000	0.998	0.997	0.890	0.961
175000	0.995	0.993	1.000	0.996
225000	0.995	0.993	1.000	0.996
275000	0.995	0.993	1.000	0.996
325000	0.995	0.993	1.000	0.996
375000	0.996	0.991	0.877	0.955
425000	0.996	0.992	0.941	0.977
475000	0.997	0.993	0.946	0.979
Mean	0.996	0.991	0.943	0.977

Sensitivity to parameter changes (cont.)

Over D-Stream, SOStream improves by an average purity of 5.0% and over MR-Stream it improved by 2.1%.

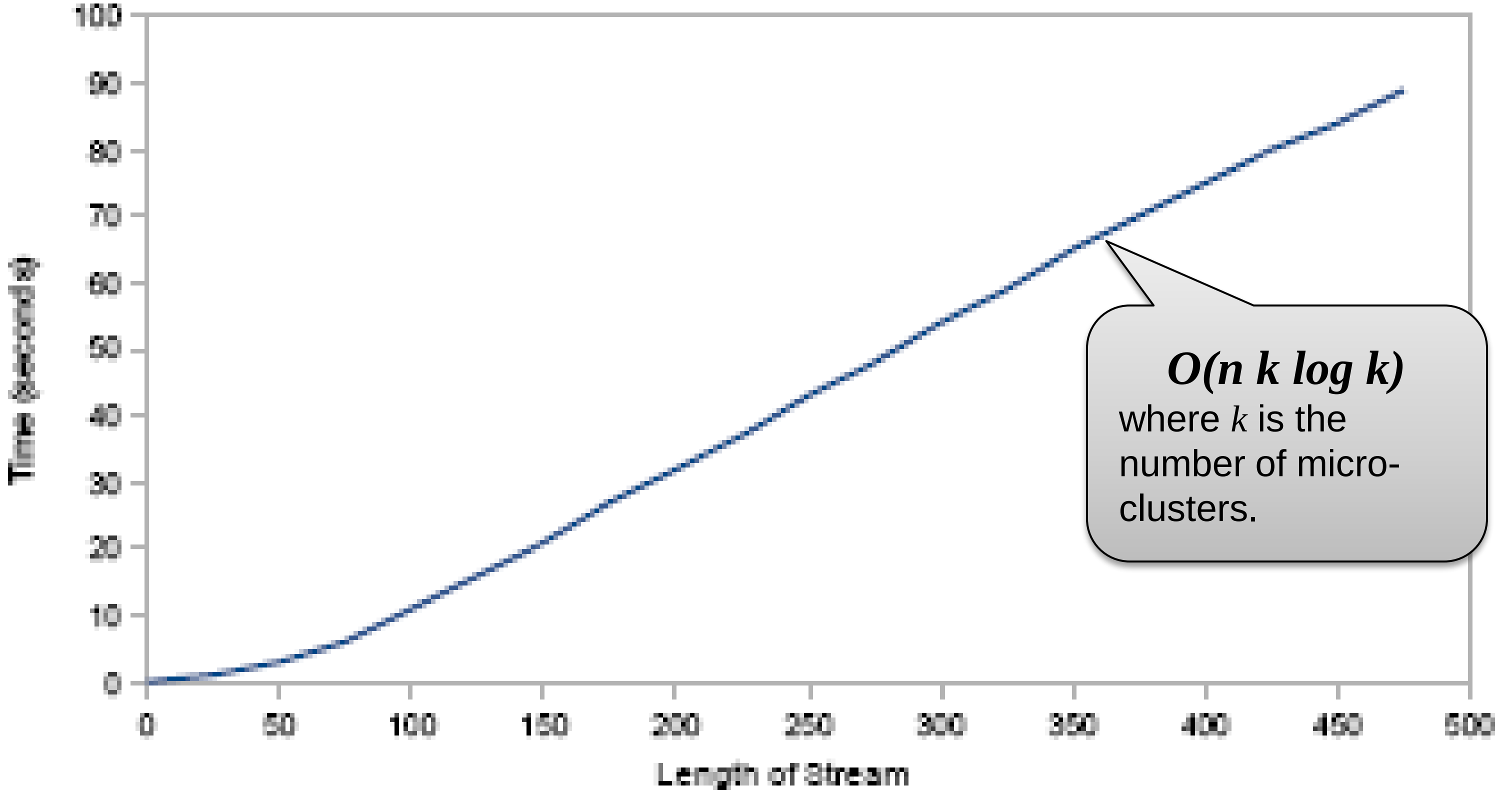
Data Points	SOStream ($\alpha = 0.1$)	SOStream ($\alpha = 0.3$)	MR-Stream	Improvement to MR-Stream%	D-Stream	Improvement to D-Stream%
25000	0.965	0.950	0.94	2.592	0.82	15.027
75000	0.955	0.985	0.92	6.646	0.91	7.661
125000	0.960	0.961	0.96	0.000	0.95	1.182
175000	0.929	0.996	1	0	1	0
225000	0.929	0.996	1	0	1	0
275000	0.929	0.996	1	0	1	0
325000	0.929	0.996	1	0	1	0
375000	0.929	0.955	0.95	0.000	0.91	4.688
425000	0.949	0.977	1.00	-2.387	1.00	-2.387
475000	0.949	0.979	0.89	9.056	0.87	11.100
Mean	0.936	0.977	0.96	2.081	0.93	5.020

Scalability and complexity of SOStream



SOStream memory cost over the length of the data stream ($\alpha = 0.1$, $MinPts = 2$, fading and merging threshold = 0.1). MR-Stream is retrieved from [7]

Scalability and complexity of SOStream (cont.)



SOStream execute time using high dimensional KDD CUP99 dataset with 34 numerical attributes. The sampling data rate is every 25K points.

Conclusions

We explored a set of techniques for data stream clustering

- Automatic threshold selection
- Using online merging
- Using competitive learning to deal with overlapping clusters

In our prototypical implementation called SOStream the new techniques show promise compared to MR-Stream and D-Stream.

Future work will deal with more thorough evaluation and handling noise in data.

References

[1] Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43 (1982) 59–69

[2] Pei, Y., Zaiane, O.: A synthetic data generator for clustering and outlier analysis. Technical report, Computing Science Department, University of Alberta, Edmonton, Canada T6G 2E8(2006)

[3] Hettich, S., Bay, S.D.: The UCI KDD Archive, University of California, Department of Information and Computer Science, Irvine, CA, USA. (1999) http://kdd.ics.uci.edu.

[4] Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th international conference on Very large data bases -Volume 29. VLDB ’2003, VLDB Endowment (2003) 81–92

[5] Cao, F., Ester, M., Qian,W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: In 2006 SIAM Conference on Data Mining. (2006) 328–339

[6] Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD ’07, New York, NY, USA, ACM (2007) 133–142

[7] Wan, L., Ng,W.K., Dang, X.H., Yu, P.S., Zhang, K.: Density-based clustering of data streams at multiple resolutions. ACM Trans. Knowl. Discov. Data 3 (July 2009) 14:1–14:28




Thank you!