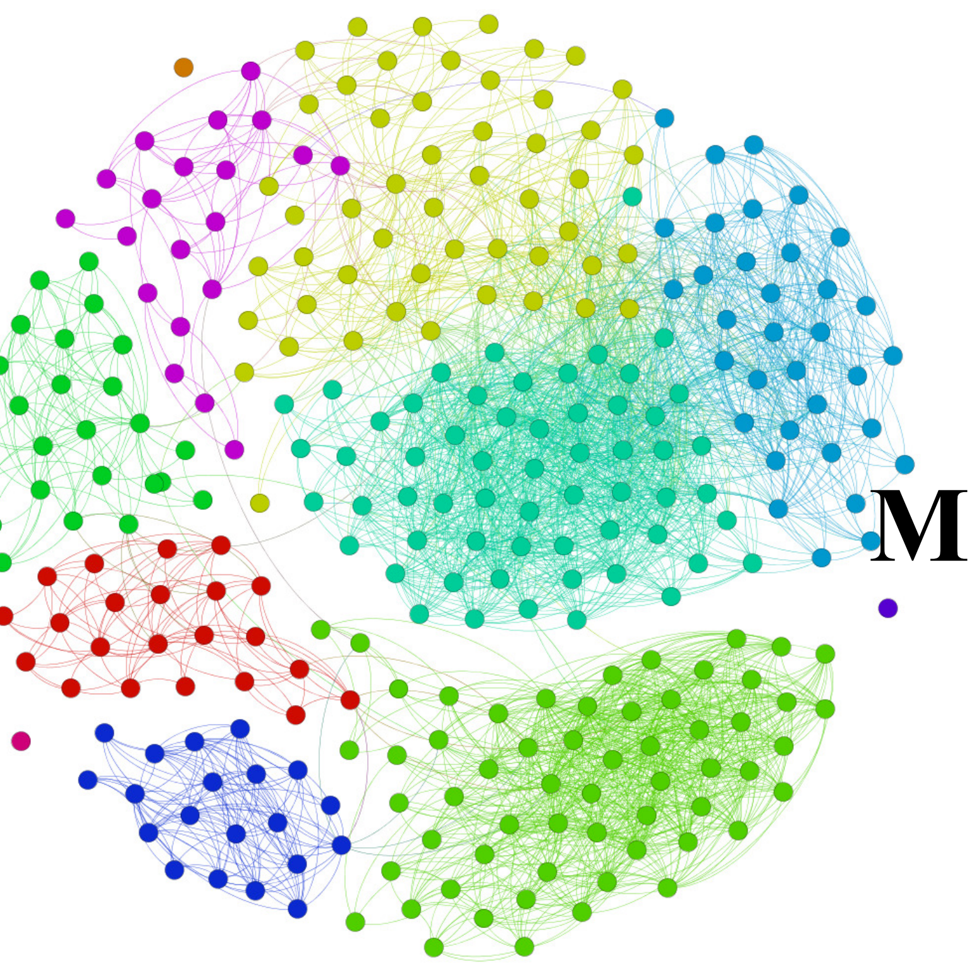
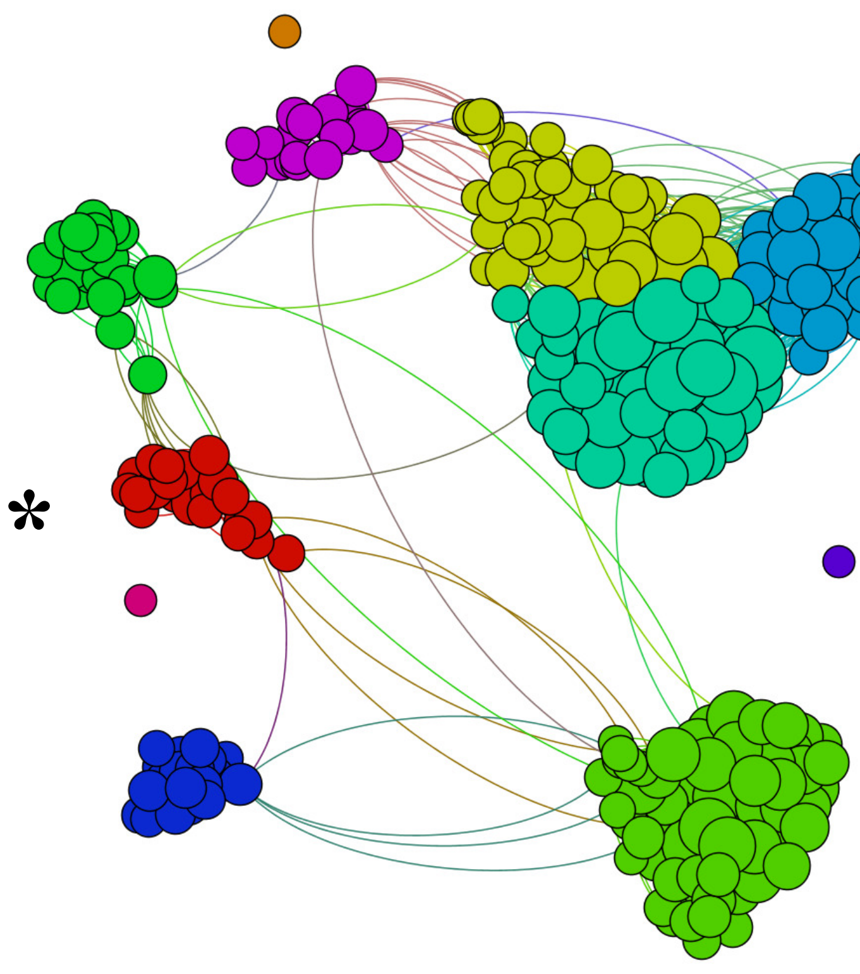


Streaming Clustering

Margareta Ackerman, Suhib Sam Kiswani*, and Jarrod Moore*
Computer Science Department, Florida State University



- Clustering is a fundamental data mining technique, used to identify similar groups in data. It is widely applied in many different fields, spanning from astronomy and biology to marketing and zoology.
- However, with the unprecedented amounts of data which recently became available for analysis, data analysts are turning away from classical batch clustering due to insufficient space in main memory.
- We consider the efficacy of *streaming algorithms*, which present data one elements at a time as well as allow multiple passes over a data stream. As such they can be used on very large data sets.
- We show that additional streams enable the identification of some fundamental clustering structures. In addition, we present a streaming algorithm that is identical to the classical *k*-means algorithm in terms of its input-output behavior.
- On the other hand, we demonstrate the existence of elementary cluster structures that cannot be detected by any streaming clustering algorithm without resorting to O(n) streams.
- These results can aid in algorithm selection, helping users decide when they may successfully utilize streaming methods by considering the cluster structures that they wish to uncover.

We begin with our positive results, presenting an algorithm that can be used to detect cluster structure in the streaming model.

Algorithm 3.1 Multipass Subsample

```
Require: $\ell \geq k$
  Choose $\mathcal{S} = (s_1, \ldots, s_\ell)$ to be a random set of points from a stream of $X$
  loop
      for $x_i$ in a stream of $X$ do
          With probability $(\ell/t)$:
              Remove an element from $\mathcal{S}$ uniformly at random and add $x_i$ to $\mathcal{S}$
      end for
      Stream $X$ to calculate the cost $\Phi(\mathcal{S})$ of the clustering induced by $\mathcal{S}$
      if $\Phi(\mathcal{S}) < \Phi^*$ then
          $\Phi^* \leftarrow \Phi(\mathcal{S})$ and $\mathcal{S}^* \leftarrow \mathcal{S}$
      end if
  end loop
```

We rely on the following formal definition of cluster structure:

Definition 2.2 ((c, ϵ)-Uniqueness of Optimum [1]). A data set (X, d) satisfies (c, ϵ)-Uniqueness of Optimum for objective function Φ if for every k-clustering $\mathcal{C}$ of X where $\Phi(\mathcal{C}) \leq c \cdot \mathrm{OPT}_\Phi$, $dist(\mathcal{C}, \mathcal{C}^*) < \epsilon$.

The following is our main result, showing that the above algorithm can detect cluster structure in the streaming model:

Theorem 3.1. *Let (X, d) be a data set that satisfies $(1 + \alpha, \epsilon) - UO$ for the k-medians objective function Φ with optimal k-clustering $\mathcal{C}^*$. If the size of every cluster $C_i^* \in \mathcal{C}^*$ is greater than $2\epsilon n$ and $0.2 \leq \epsilon \leq 1$, then Algorithm 3.1 can approximate C^* within distance $(1 + 5/\alpha)\epsilon n$ using M streams with probability $\theta \geq 1 - \left[ke^{-(1-5/\alpha)\epsilon k}\right]^{M/3}$.*

In addition, we present the following streaming algorithm, which is able to attain precisely the same output as the classical batch *k*-means algorithm.

Algorithm 5.1 Streaming Lloyd's

```
  Use the first stream to choose $\mathcal{S} = (s_1, \ldots, s_k)$ random points from $X$
  Let $\mathcal{T} = (t_1, \ldots, t_k)$, where these initialize values are arbitrarily and $\mathcal{S} \neq \mathcal{T}$
  loop(until $\mathcal{S} = \mathcal{T}$)
      Unless it is the first iteration, set $\mathcal{S} = \mathcal{T}$
      Set $mass(t_i) = 0$ for all $t_i \in \mathcal{T}$
      for $x_i$ in a stream of $X$ do
          Find $x_i$'s closest center $s_i \in \mathcal{S}$:
              Set $t_i = \frac{mass(t_i)t_i + s_i}{mass(t_i) + 1}$
              $mass(t_i) = mass(t_i) + 1$
      end for
  end loop
```

Lastly, we show our negative result, proving that a natural cluster structure cannot be detected in the streaming model without resorting to *O(n)* streams.

A clustering is *nice* if each point is closer to all elements in its cluster than to all other data. Let *n* denote the number of elements in the data.

Theorem 4.1. *No streaming algorithm $\mathcal{A}$ with memory capacity 3c is nice-detecting with fewer than $\frac{n-4}{3c}$ passes.*

mackerman@fsu.edu

* represents computer science FSU students