Streaming-data Clustering: Challenges

- Possibly infinite data streams.
- New data arriving rapidly.
- Need to be able to provide an estimate of the model at any given time.
- Data statistics are usually non-stationary.
- Clusters may appear/disappear.
- Cluster properties (e.g., centroids) can change with time. 
- Cluster weights can change with time.

The Proposed Solution: ScStream

- Based on a DPM for each cluster.
- Uses a DPM for each cluster.
- Combines the iterative sampling with an additional iteration that uses a deterministic subroutine based on the predictive posterior.

ScStream Satisfies the Following Desiderata

- Fast.
- Does not need to retain previously processed data.
- Can readily estimate the number of clusters as needed.
- Supports non-stationary cluster statistics.
- No label switching.
- Efficient memory use.

Weighted Batched Sufficient Statistics

Consider Gaussian components with a Normal-Inverse-Wishart prior. In the DPM, the posterior for cluster \( k \), \( \pi_k \alpha, \omega_k \), is calculated using:

\[
\pi_k = \frac{1}{Z} \left( \frac{1}{\alpha_k} \right)^{d/2} \left( \omega_k \right)^{\frac{d+2}{2}} \left( \sum_{i=1}^{D} x_i \right)^{\frac{2d+2}{2}} \left( \sum_{i=1}^{D} x_i^2 \right)^{-\frac{d}{2}}
\]

where \( Z \) is the partition function. For the DPM, \( 
\text{NMI}(h, B) \) is the normalized mutual information.

The Algorithm

**Algorithm 1:** ScStream

Input: \( X_1, \ldots, X_T \) Output: \( \pi_k, \omega_k, \alpha_k \)

1. Pick a random subset of \( \mathbb{X}_1 \) and set the initial \( \pi_k, \omega_k, \alpha_k \).
2. Initialize \( M_k \).
3. for \( t \in [1, T] \)
4. for \( k \in [1, K] \)
5. for \( i \in [1, B] \)
6. Multiply \( \frac{1}{\alpha_k} \left( \omega_k \right)^{1/2} \left( x_i \right)^{1/2} \left( \sum_{j=1}^{D} x_j \right)^{1/2} \left( \sum_{j=1}^{D} x_j^2 \right)^{-1/2} \)
7. end for
8. end for
9. end for

Can the Dirichlet Process Mixture Model (DPM) be used for Clustering Streaming Data?

- Fast.
- Can handle different data types (e.g., categorical models can be Gaussian, multivariate, etc.).
- Not limited to constant data size.
- Can support non-stationary elements.
- Easy to add new data.
- Supports non-stationary cluster statistics.

References


Our ScStream Code is Publicly Available with Support for either Julia or Python

- Julia: github.com/SGU-CS-UL/DPMMals/cluster-streaming.jl
- Python: github.com/SGU-CS-UL/dpmm-clustering/ScStream

Table 1: Comparing our method (ScStream) with BIRCH [1], CluStream [2], DBSTREAM [3], and CluStream [4]. The best result for a metric is displayed in bold. *N/A: Not available.

**Experiments and Results**

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet100</th>
<th>CoverType</th>
<th>BIRCH CluStream DBSTREAM StreamKM++</th>
<th>StreamKM++</th>
<th>ScStream</th>
<th>DPM Sampler</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet100</td>
<td>4.64</td>
<td>4.41</td>
<td>6.00</td>
<td>6.64</td>
<td>6.00</td>
<td>6.00</td>
</tr>
<tr>
<td>CoverType</td>
<td>4.64</td>
<td>4.41</td>
<td>6.00</td>
<td>6.64</td>
<td>6.00</td>
<td>6.00</td>
</tr>
</tbody>
</table>

Figure 3: Visualization of the NMI results for each of the experiments.