**Clustering Data Streams Based on Shared  
Density between Micro-Clusters**

**Abstract**

As more and more applications produce streaming data, clustering data streams has become an important technique for data and knowledge engineering. A typical approach is to summarize the data stream in real-time with an online process into a large number of so called micro-clusters. Micro-clusters represent local density estimates by aggregating the information of many data points in a defined area. On demand, a (modified) conventional clustering algorithm is used in a second offline step to re-cluster the micro clusters into larger final clusters. For re-clustering, the centers of the micro-clusters are used as pseudo points with the density estimates used as their weights. However, information about density in the area between micro-clusters is not preserved in the online process and re-clustering is based on possibly inaccurate assumptions about the distribution of data within and between micro-clusters (e.g., uniform or Gaussian). This paper describes DBSTREAM, the first micro-cluster-based online clustering component that explicitly captures the density between micro-clusters via a shared density graph. The density information in this graph is then exploited for re-clustering based on actual density between adjacent micro-clusters. We discuss the space and time complexity of maintaining the shared density graph. Experiments on a wide range of synthetic and real data sets highlight that using shared density improves clustering quality over other popular data stream clustering methods which require the creation of a larger number of smaller micro clusters to achieve comparable results.

**Existing System**

Data stream clustering is typically done as a two-stage process with an online part which summarizes the data into many micro-clusters or grid cells and then, in an offline process, these micro-clusters (cells) are re-clustered/merged into a smaller number of final clusters. Since the re-clustering is an offline process and thus not time critical, it is typically not discussed in detail in papers about new data stream clustering algorithms. Most papers suggest using an (sometimes slightly modified) existing conventional clustering algorithm (e.g., weighted k-means in CluStream) where the micro-clusters are used as pseudo points. Another approach used in Den Stream is to use reach ability where all micro-clusters which are less than a given distance from each other are linked together to form clusters. Grid-based algorithms typically merge adjacent dense grid cells to form larger clusters (see, e.g., the original version of D-Stream and MR-Stream).

**Disadvantages:**

1. The number of clusters varies over time for some of the datasets. This needs to be considered when comparing to CluStream, which uses a fixed number of clusters.

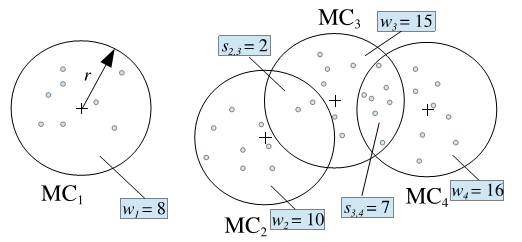
**Proposed System**

We develop and evaluate a new method to address this problem for micro-cluster-based algorithms. We introduce the concept of a shared density graph which explicitly captures the density of the original data between micro-clusters during clustering and then show how the  
graph can be used for re-clustering micro-clusters. This is a novel approach since instead on relying on assumptions about the distribution of data points assigned to a micro cluster (MC) (often a Gaussian distribution around a center), it estimates the density in the shared region between micro-clusters directly from the data. To the best of our knowledge, this paper is the first to propose and investigate using a shared-density-based re-clustering approach for  
data stream clustering.

**Advantages:**

1. This is an important advantage since it implies that we can tune the online component to produce less micro-cluster for shared-density re-clustering.
2. It improves performance and, in many cases, the saved memory more than offset the  
   memory requirement for the shared density graph.

**System Architecture**

****

**Fig. 2 MC1 is a single MC. MC2 and MC3 are close to each other but the density between them is low relative to the two MCs densities while MC3 and MC4 are connected by a high density area.**

**Modules**

1. Leader-Based Clustering
2. Capturing Shared Density
3. Micro-Cluster Connectivity
4. Noise Clusters

**Module Description**

1. **Leader-Based Clustering**

DBSTREAM represents each MC by a leader (a data point defining the MC’s center) and the density in an area of a user-specified radius r (threshold) around the center. This is similar to DBSCAN’s concept of counting the points is an eps-neighborhood.

1. **Capturing Shared Density**

The fact, that in dense areas MCs will have an overlapping assignment area, can be used to measure density between MCs by counting the points which are assigned to two or more MCs.

1. **Micro-Cluster Connectivity**

Less dense clusters will also have a lower shared density. To detect clusters of different density correctly, we need to define connectivity relative to the densities (weights) of the participating clusters.

1. **Noise Clusters**

To remove noisy MCs from the final clustering, we have to detect these MCs. Noisy clusters are typically characterized as having low density represented by a small weight.

# Configuration:-

# H/W System Configuration:-

# Processor - Pentium –III

Speed - 1.1 Ghz

RAM - 256 MB(min)

Hard Disk - 20 GB

Floppy Drive - 1.44 MB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

# S/W System Configuration:-

* Operating System :Windows95/98/2000/XP
* Programming Language : Java