Data Stream Mining- Lecture 5

Clustering of Data Streams

Chandresh Kumar Maurya, Research Assistant Professor

Eötvös Loránd University, Budapest, Hungary

October 22, 2019

Clustering of data streams

Clustering of static data- several algorithms exists such as K-means, DBSCAN, OPTICS, ODAC etc.
Clustering of streaming data is non-trivial due to the following challenges:

1. Concept drifts
2. High data rate
3. Low memory footprints
4. Low processing time
5. Single (or 2-3 passes)
6. Algorithms needs to be robust
7. and so on.

Traditional algorithms needs to be adapted to account for the aforementioned issues.

What is clustering?

Clustering is an unsupervised technique to group (clusters henceforth) the data such that the examples in the group satisfy two requirements:

- Examples in the same cluster are similar
- Examples in the different cluster are dissimilar

Examples:

- Customer segmentation
- Cluster genes based on protein compositions
- Assign patients to doctors based on diagnosis
- Time-series clustering
- Cluster documents based on some similarity and so on.

Examples

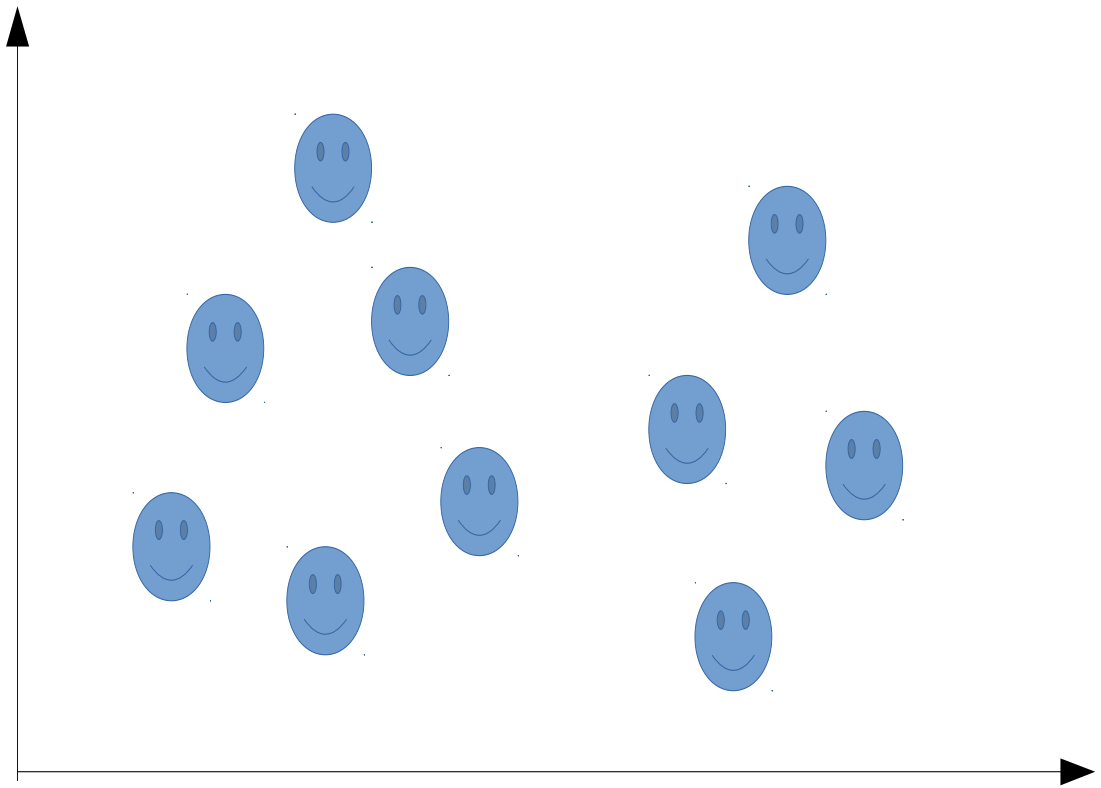

Figure: Clustering users based on behavior

Contd...

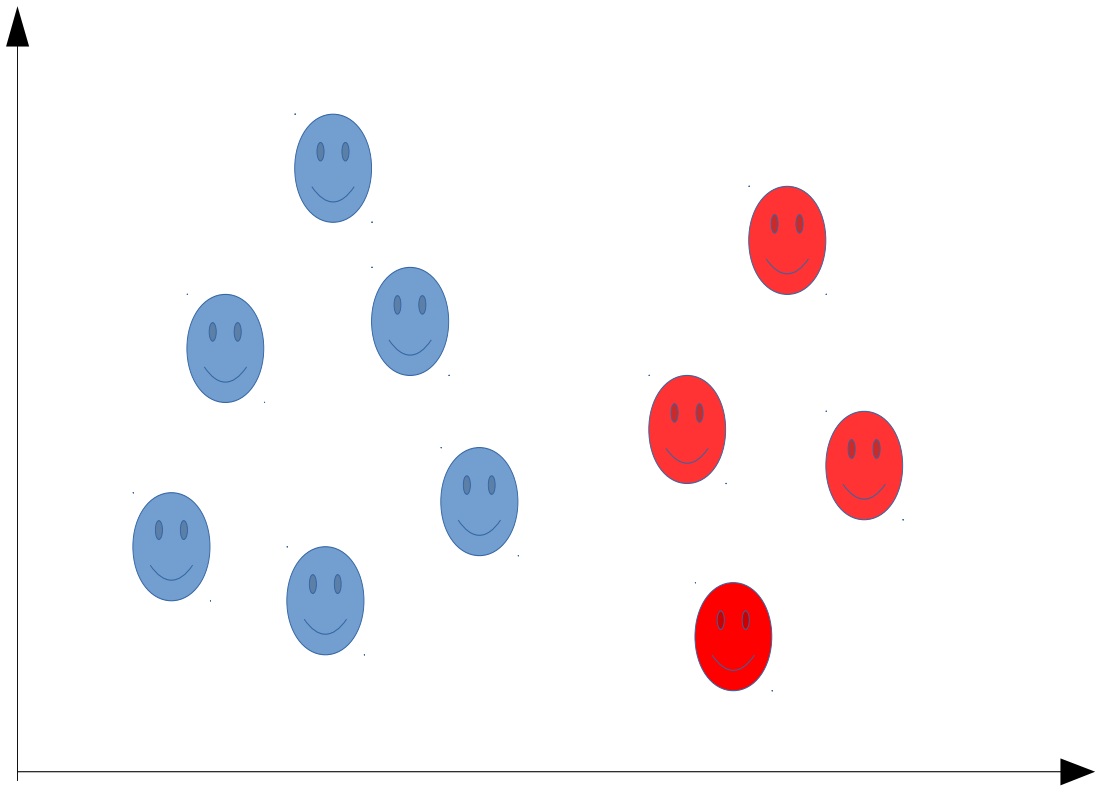

Figure: Clustering users based on behavior

Types of clustering

Should we cluster examples or features?

- Partition based clustering (e.g. K-means, k-medoids)
- Hierarchical clustering (e.g. Agglomerative vs Divisive)
- Density-based clustering (e.g. DBSCAN, OPTICS)
- Grid-based clustering (e.g. CLIQUE and STING)
- Modal-based clustering(e.g. COWEB, SOM)

Basic concepts

Requirements of data stream clustering:-

1. Need a compact representation of the data stream (Why?).
2. Fast and incremental processing of incoming examples.
3. Tracking cluster changes.
4. Clear and fast identification of outliers.

Contd...

Definition

Cluster feature: A cluster feature is a triple (N, LS, SS) used to store the sufficient statistics of the data points.

where

- **N**–number of data points.
- **LS**–a vector to store linear sum of N points.
- **SS**–a sector to store sum of squares of the points.

We can easily maintain various operations on the above called properties of the cluster feature. These are:

- Incrementality
- Additivity
- Centroid
- Radius

Partitioning Clustering

We will look at

- The Leader algorithm
- Single pass K-means

The Leader Algorithm

Algorithm 1: The Leader

Input : X: A sequence of Examples x_i
δ: Control distance parameter
Output : Centroid of k clusters
Initialization: Initialize the set of centroids $C = x_1$

```
for x_i ∈ X do
    Find the cluster c_r whose center is closest to x_i;
    if d(x_i, c_r) < δ then
        C = C ∪ x_i
    end
    else
        make a new leader with centroid x_i ;
    end
end
```

Single Pass $k-$Means Algorithm

Algorithm 11: Algorithm for Single Pass `k-Means` Clustering.

input : S: A Sequence of Examples
k: Number of desired Clusters.
output: Centroids of the k Clusters
begin
  Randomly initialize cluster means.;
  Each cluster has a discard set that keeps track of the sufficient statistics.;
  while *TRUE* **do**
    Fill the buffer with examples ;
    Execute iterations of k-means on points and discard set in the buffer, until convergence. ;
    `/* For this clustering, each discard set is treated like a regular point weighted with the number of points in the discard set. */`
    ;
    foreach *group* **do**
      update sufficient statistics of the discard set with the examples assigned to that group;
    Remove points from the buffer;

Figure: Single pass k-means

Hierarchical clustering

- Create clusters of various sizes
- two types: Agglomerative and divisive clustering
- An example is BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
- Extension of BIRCH for data streams is CluStream [Aggarwal et al., 2003]

CluStream

Key Ideas:

- It is a two-phase algorithm
- In the first phase, it stores summary statistics of the data points seen so far into **micro-clusters**.
- In the second-phase, which is an offline phase, actual clustering happens.
- It maintains a time dimension in the CF which is called as CFT. CFT stores sum of timestamps, and sum of squares of timestamps.
- **Pyramidal Time Frame**: In order to facilitate the analysis of cluster evolution the information about micro-clusters are periodically stored away in a permanent storage medium. This period varies in a pyramidal fashion. This storing is called "snapshot".

Contd...

How to store the snapshots?

- Snapshots are classified into different orders which can vary from 1 to $log(T)$ where $T =$ clock time elapsed since the beginning of the stream.
- The snapshot of the i^{th} order occurs every α^i time units where α is an integer and $\alpha > 1$.

For example when T=8 and $\alpha = 2$, stored snapshots order and times are as follows,

- Order 0 snapshot (2^0 x time units)- 0, 1, 2, 3, 4, 5, 6, 7, 8
- Order 1 snapshot (2^1 x time units)- 0, 2, 4, 6, 8
- Order 2 snapshot (2^2 x time units)- 0, 4, 8
- Order 3 snapshot (2^3 x time units)- 0, 8

Note: at any given moment of time, only the last $\alpha + 1$ snapshots of order i are stored.

Contd...

- the maximum order of any snapshot stored at T is $\log_\alpha(T)$
- the maximum number of snapshots at t is $(\alpha + 1)\log_\alpha(T)$
- for any user-specified time window of h, at least one snapshot can be found within $2h$ units of current time (time horizon approximation).

Assignment: prove the above results.

Contd...

As an example if we want to find a snapshot within a factor of 2 of any user-specified time window for a data stream running for 100 years with a time granularity of 1 second the total number of snapshots needed to be maintained is (2+1)*log2(100*365*24*60*60)=95, which is a modest storage requirement.

Online Cluster Maintenance

How to maintain clusters online?

- CluStream maintains q initial micro-clusters created by applying k-means algorithm on CF in the offline phase
- Whenever a new data points arrive, calculate distance of it from existing CFs, and if the distance is within the maximum boundary (RMS of deviations of the points) of the CF, absorb it.
- Otherwise create a new micro-cluster with a unique ID associated to it.
- Whenever, number of micro clusters exceeds more than q, either i) delete the oldest micro-cluster or ii) merge two cluster sharing some similarity.

Offline Macro Cluster Creation

Suppose you want to assess the data stream for higher level clusters. Two inputs:

- Time horizon h
- Number of clusters k.

Use stored snapshots up to time horizon h for macro cluster creation. How to find micro clusters which are specific to user=specified time horizon?

Property1: Let C_1 and C_2 be two sets of points. Then the following holds:

$$CFT(C_1 \cup C_2) = CFT(C_1) + CFT(C_2) \text{(Additive property)}$$

Property2: Let C_1 and C_2 be two sets of points. Then the following holds:

$$CFT(C_1 - C_2) = CFT(C_1) - CFT(C_2) \text{(Subtractive property)}$$

Contd...

We use the subtractive property to determine the number of approximate number of micro-clusters for a given time horizon h as follows:

- Let current time be t_c.
- We need to find micro-clusters stored just before time $t_c - h$.
- Pyramidal property ensures that there is always a snapshot at time $t_c - h'$ where $h' < h$.
- Denote the micro-clusters at time t_c and $t_c - h'$ by $S(t_c)$ and $S(t_c - h')$ respectively.
- Use subtractive property to determine the micro-clusters created in time horizon h,i.e., subtract CFT of micro-clusters in $S(t_c)$ and $S(t_c - h')$.
- This ensures that the micro-clusters created before time horizon h do not dominate the clusters.
- Denote by $\mathcal{N}(T_c, h')$ the set of micro-clusters created as such.

Contd...

Run modified k-means algorithm to cluster points in the set $\mathcal{N}(t_c, h')$ as follows:

1. At the initialization stage, the seeds are no longer picked randomly, but are sampled with probability proportional to the number of points in a given micro cluster. The corresponding seed is the centroid of that micro-cluster.
2. At the partitioning stage, the distance of a seed from a given pseudo-point (or micro-cluster) is equal to the distance of the seed from the centroid of the corresponding micro-cluster.
3. At the seed adjustment stage, the new seed for a given partition is defined as the weighted centroid of the micro-clusters in that partition.

Bibliography I

Aggarwal, C. C., Watson, T. J., Ctr, R., Han, J., Wang, J., and Yu, P. S. (2003).
A framework for clustering evolving data streams.
In *In VLDB*, pages 81–92.