CLUSTERING DATA STREAMS

Hadil Shaiba

February 27, 2012

Southern Methodist University

Data Streams

- Definition:
  - Tuples of data that arrive in an ordered and a continuous manner
    - Tuples = <a1, a2,……, am>
- Traditional data mining algorithms are not efficient
  - Algorithms for data streams are available

ISSUES

- Streams are of an infinite size
  - Handle limited size of memory
- Data arrives continuously
  - Clustering is processed in a single pass
  - Algorithm shall be fast enough to handle the fast arrival of streams
- Changes might happen over time
  - Traditional algorithms do not deal with this type of data

EXAMPLES

- Pattern recognition
  - Recognizing the relations between words.
- Data analysis
  - Tracking the behavior of stocks
- Image processing
  - Recognizing faces
- Knowledge discovery
  - Tracking the hurricanes using data received from sattelites

Clustering Data Streams

- Traditional Clustering Algorithms:
  - Unsupervised learning- groups are created based on certain measurements
    - Ex. Distance between attributes
  - Several passes over the dataset is required for final groups generation
  - Clustering algorithms:
    - K-means, Hierarchical, PAM clustering.
- New algorithms for clustering data streams:
  - On-line:
    - Algorithms that produce an efficient summery of the data in real-time
  - Off-line:
    - Summarized data are an input to traditional clustering algorithms
      - Ex. K-means

Performance

- Measuring the performance:
  - Number of passes over the tuple
  - The total time the algorithm takes to produce output

ALGORITHMS

- BIRCH:
  - Converts streams into a tree that contains the information needed for clustering
- STREAM:
  - Uses **Time Windows** to divide streams into windows
- CluStream:
  - Summarizes streams into **micro-clusters** to deal with memory limitations
- D-Stream:
  - Converts streams into micro-clusters using **Density based**
  - Produce arbitrarily shapes
  - Protects against outliers

Source: Mining Massive Data Streams by Michael Hahsler

Micro-cluster Cont'd

- Temporal extension of cluster feature (CF) vector
  - Produces a statistical summarization of the clusters
- Cluster feature (CF):

$$CF = (N, \overrightarrow{LS}, SS)$$

  - Summarization of the cluster where:
    - N: number of points
    - LS: linear sum of the N points
    - SS: square sum of the N points
  - If two clusters are joined the new cluster will look as follows:

$$CF_1 + CF_2 = (N_1 + N_2, \overrightarrow{LS}_1 + \overrightarrow{LS}_2, SS_1 + SS_2)$$

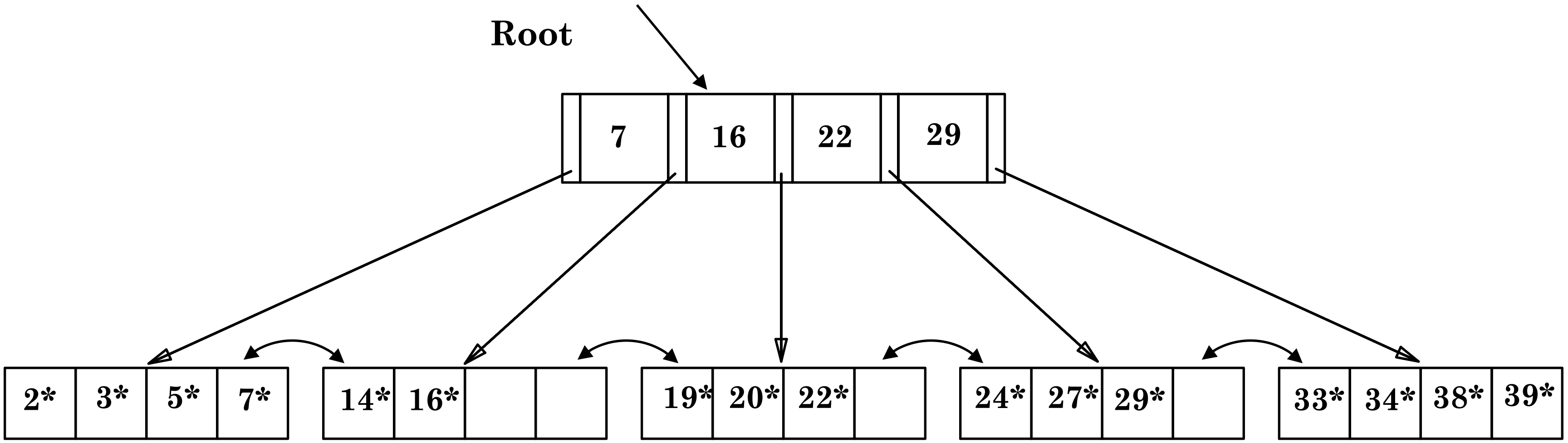
Micro-cluster Cont'd

- Extends the cluster feature by adding summaries about the time stamps
- It looks as follows:

$$(\overrightarrow{CF2^x}, \overrightarrow{CF1^x}, CF2^t, CF1^t, n).\ \overrightarrow{CF2^x}, \overrightarrow{CF1^x}$$

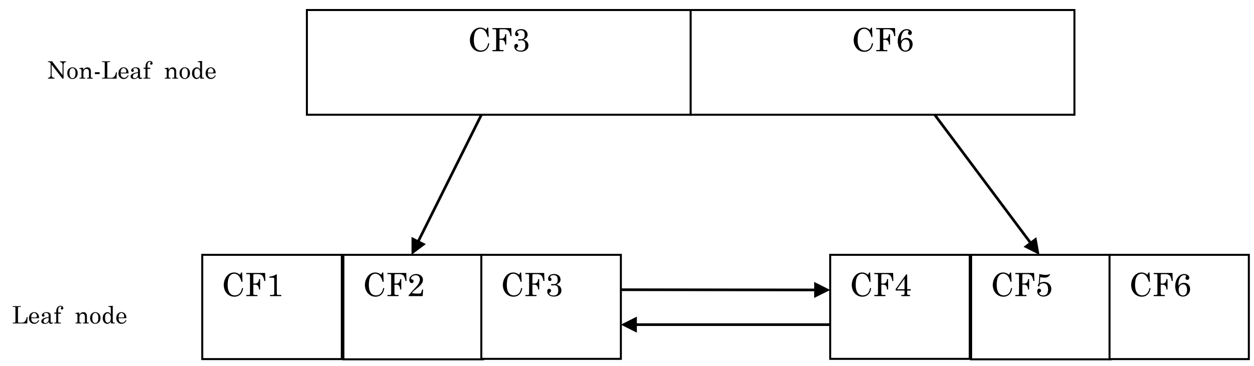
  - Where,
    - $\overrightarrow{CF2^x}$ = SS
    - $\overrightarrow{CF1^x}$ = LS
    - $CF2^t$ = sum of squares of time stamps
    - $CF1^t$ = sum of time stamps

B-Tree



CF TREE

- B-Tree, with a branch factor B, threshold T and L maximum number of entries in a leave node
- Threshold T can be adjusted so leaves can have more points.



BIRCH

- One of the first algorithms that deal with large data
- Hierarchal and incremental clustering
- Handles outliers
- Assumes we have limited size of memory
- Requires one scan to the database
- Builds a CF tree
  - carries a summary of the data
  - Searched from top to down
  - Leaves contain clusters that are created from the summary
- Complexity o(n)
- Worst case complexity o(n^2)
  - So many adjustments of threshold

Algorithm

- Create initial CF tree based on memory limits
- For each element find the best leaf cluster
  1. If less than threshold
     1. Add element to the leaf cluster
     2. update CF triple
  - Otherwise
     1. If there is enough place to insert element
        1. Insert element as a new cluster
        2. Update CF triple
     - Otherwise
        1. Split leaf node and redistribute CF features

Issues

- Each entry in the CF tree has a limited size which might not be the case
- The order of the input can affect the results in a negative way
- New and old entries have the same importance

CluStream

- Uses time stamps
- Snapshots of certain time stamps are taken
  - Different orders
    - Minimum: 1
    - Maximum: $\log_{\alpha}(T)$
  - Only the last $(\alpha^{\iota})+1$ snapshots are stored
- Micro-clusters
  - produced for each snapshot
    - Set of summarized statistics
  - Stored in a time following a pyramidal pattern

Snapshots order	Snapshots (clock times)
0	55 54 53 52 51
1	54 52 50 48 46
2	52 48 44 40 36
3	48 40 32 24 16
4	48 32 16
5	32

- Ex. $\alpha = 2$ and $\iota = 2$
- Number of snapshots for each order:
  - $(2^2)+1 = 5$

CluStream Cont'd

- Uses the same concept of k-means and nearest neighbor algorithms
- Q micro-clusters are stored at a certain time representing the current snapshot
  - $M_1, \ldots\ldots, M_q$ where each M has an id.
  - If M_a and M_b are merged:
    - List of ids is created
- Different clusters are produced when new streams arrive
- If clock time is divisible by (α^ι)
  - Micro-cluster is stored on disk
  - old micro-clusters are deleted if they exceed certain threshold

Algorithm

<u>CluStream Algorithm</u>:

1. Off-line process starts with q initial set of micro-clusters
2. On-line process goes through each new data point x
    1. Find for x the closest micro-cluster c
    2. If x is closer to c than a threshold δ
        1. update x to absorb c
    - Otherwise
        1. create a new micro-cluster for c with a unique id
        2. If any current micro-cluster is considered as outlier
            1. Delete the micro-cluster
        - Otherwise
            1. Merge two clusters

Algorithm Cont'd

- When creating a new cluster we need to get rid of an older cluster to save space
- Algorithm to find outliers:
  1. Estimate the average timestamp of the last m arrivals in each micro-cluster.
  2. If all time stamps are recent and above threshold δ
     1. Merge two closest micro-clusters
     2. Create an id list of the two micro-clusters ids
     - Otherwise
     1. Delete micro-cluster with the least resent time stamp

Advantages

- Very flexible for real time transactions
- Pyramidal time window guarantees efficient time and space

Conclusion

- There are different algorithms available for data streams
- Choosing the algorithm depends mainly on the type of application
  - Ex. D-Stream is good for finding arbitrarily shapes
- Clustering data streams is gaining a lot of importance and a lot of research is done on improving these algorithms to apply them on real time applications

Resources

- http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp410-414.pdf
- Aggarwal, C. C., Han J., Wang, J. & Yu, P. S. (2003). A Framework for Clustering Evolving Data Stream. In Proc. of the 29th VLDB Conference.
- Barbara, D. (2003). Requirements for Clustering Data Streams. SIGKDD Explorations, 3 (2), 23-27.
- Ester M., Kriegel H.-P., Sander J. & Xu X (1998). Clustering for Mining in Large Spatial Databases. Special Issue on Data Mining, KI-Journal, ScienTec Publishing, No. 1.
- http://cran.r-project.org/web/packages/birch/birch.pdf
- Margaret, H. D. (2003). "Data Mining Introduction and Advanced Topics."
- Michael, H. (2012). "Mining Massive Data Streams." http://michael.hahsler.net/SMU/8331/slides/datastream/datastream.pdf