Automatic subspace clustering of high-dimensional and streaming data

Seminar Multimedia Retrieval and Data Mining

István Sárándi

Advisor: Marwan Hassani

RWTH Aachen University

istvan.sarandi@rwth-aachen.de

January 16, 2014

What is the topic?

My paper was about the CLIQUE algorithm:

Agrawal et al. *Automatic subspace clustering of high-dimensional data for data mining applications* ACM, 1998.

I will explain the algorithm and apply it to streams as an extra.

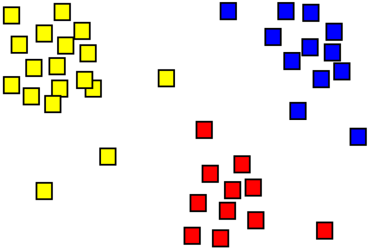
What follows:

- Intro to clustering
- Intro to high-dimensional data
- Intro to streaming data
- In-depth description of CLIQUE
- Description of CluStream, DenStream
- Evaluation in SubspaceMOA
- Discussion

Clustering

Clustering task

Find **groups** of data objects that are **similar** to each other **in the group**, but **dissimilar** to objects in **other groups**.



Clustering

Uses

- Gain new insight about structure in the data
- Compress the data by storing clusters instead of objects
- Classification in abscence of labeled data

Applications

- Marketing (e.g. customer grouping)
- Computer network analysis
- Biomedical research (e.g. computational genomics)
- Computer vision (e.g. image segmentation)
- ...

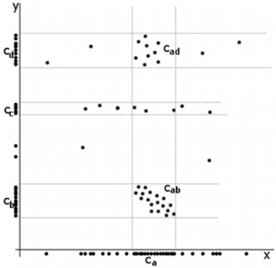
High-dimensional data

Recently, datasets in many dimensions (∼10,000)

- Computational genomics
- Text mining, etc...

New challenges

- Traditional clustering not effective, distance measures are meaningless
- Nearest neighbor distance indistinguishable from farthest "neighbor"
- Look for clusters in subspaces
- Subspace clustering, projected clustering



On another front: Streaming data

Data streams became also widespread
Infinite stream, process as generated. Consequences:

- No random access
- Keep up with input speed (be fast on average)
- Adapt to varying input speed (flexible trade-off: accuracy vs. proc. time)
- Compress unlimited amout of past data (memory management)
- Concentrate on recent data (aging)

Overview

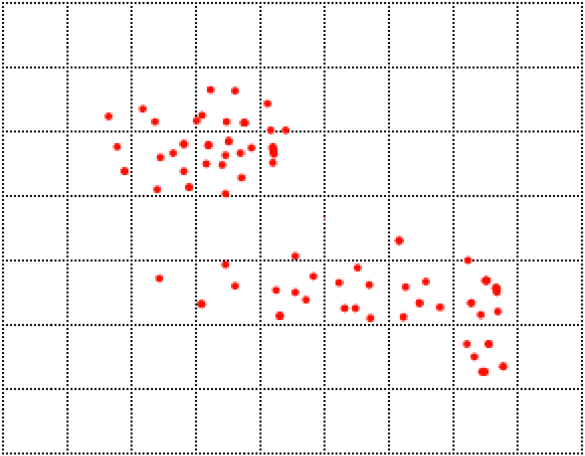
CLIQUE

CLIQUE is a subspace clustering algorithm (subspace = set of dimensions)

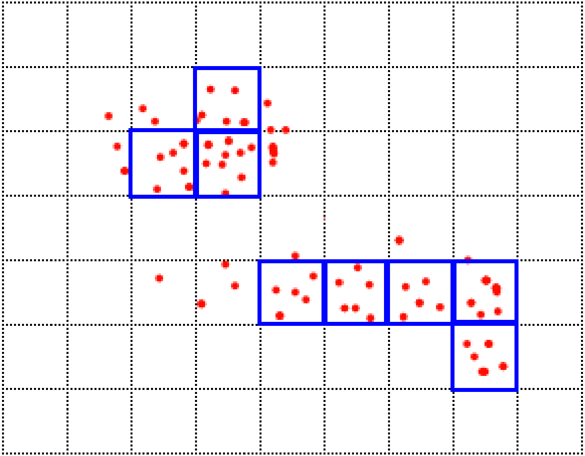
- Looks for clusters in all subspaces, efficiently
- Grid-based
- Cluster = set of (subspace) grid units having at least τ objects (dense)
- Describes clusters with disjunctive normal form formulas $(A \wedge B) \vee (C \wedge D \wedge E) \vee (...)$
- Suitable for exploratory (insight) analysis

CLIQUE - Main idea

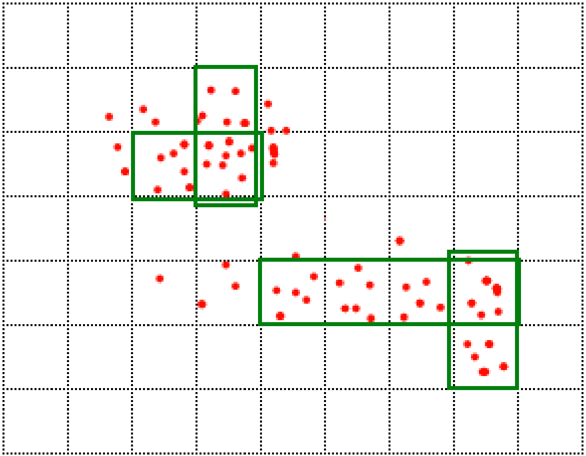
CLIQUE - Main idea



CLIQUE - Main idea



CLIQUE - Main idea



CLIQUE - Steps

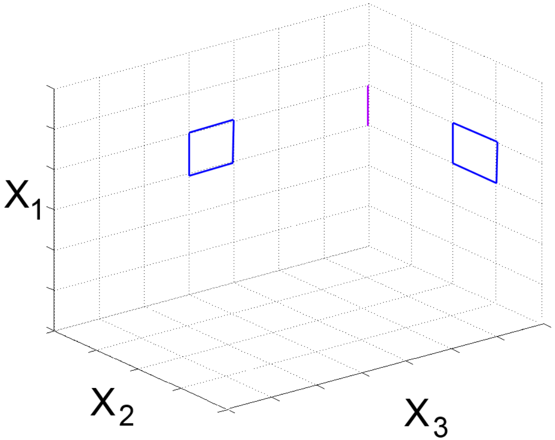
- Find dense cells efficiently
- Collect connected dense cells to clusters
- Cover clusters with few hyper-rectangles

CLIQUE - Finding dense cells

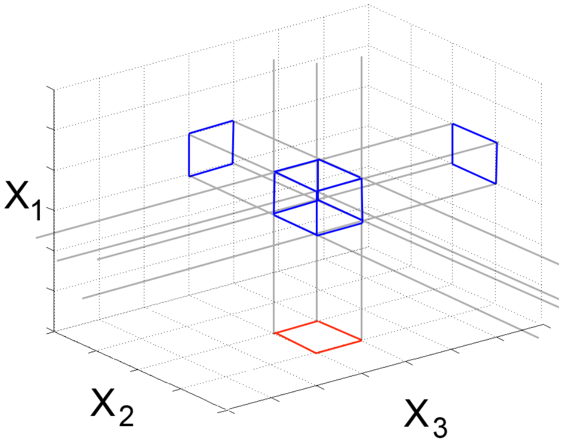
- Bottom-up search (apriori-like)
- Uses monotonicity: lower-dimensional projections of a dense unit are dense
- Find 1, 2, ... dimensional dense units sequentially
- Only inspect k-dim. cells if all $(k-1)$ dim. projections dense
- Candidate generation (Join procedure)
- Filter out those with a non-dense projection unit
- Filter out non-dense units: Pass over the data, build histograms

CLIQUE - Candidate generation

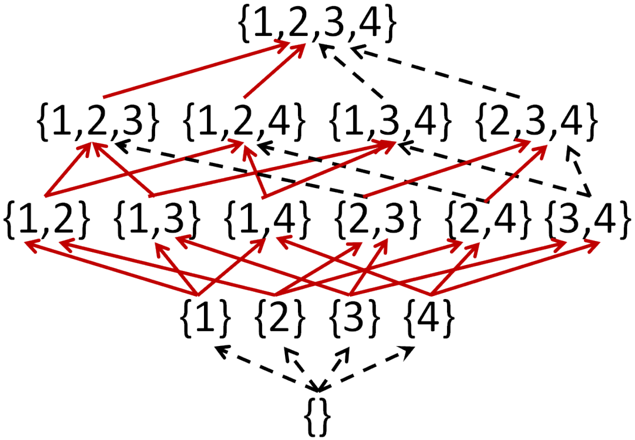
- Take 2 units that share their first $k-2$ dimensions and also the projection to that subspace
- Create intersection unit



CLIQUE - Candidate generation (Join)



CLIQUE - Candidate generation (Join)



Red arrows = Joined to create next level candidates

CLIQUE - Some formal notation

- Dimension (attribute): $A = (A.name, [A.min, A.max])$
- Data space: $\mathcal{V} = [A_1.min, A_1.max] \times ... \times [A_D.min, A_D.max]$
- Set of subspaces: $\mathcal{S} = 2^{\{1,...,D\}}$
- k-dimensional grid: $\mathcal{G}_k = \{0, ..., \xi - 1\}^k$
- k-dimensional units: $\mathcal{U}_k = \mathcal{S}_k \times \mathcal{G}_k$
- Selectivity is the number of objects in the unit:
  $sel(u) = \left| \left\{ \mathbf{v} \in V \;\middle|\; contains(u, \mathbf{v}) \right\} \right|$

CLIQUE - Pseudocode

1: **function** FINDDENSEUNITS
2: make one pass over V and build a histogram $hist_i$ for each dimension A_i
3: $Den_1 \leftarrow \{(\{i\},(g)) \mid hist_i[g] \geq \tau\}$
4: // dense units in 1D

5: **for** $k \leftarrow 2; k \leq D; k \leftarrow k+1$ **do**
6: $Cand \leftarrow$ JOIN(Den_{k-1})
7: // See Alg. 2

8: // check all projections to $(k-1)$ dimensions
9: **for all** $(S, \mathbf{g}) \in Cand$ **do**
10: $(d_1, ..., d_k) \leftarrow sorted(S)$
11: **for all** $d \in \{d_1, ..., d_{k-2}\}$ **do**
12: $u_{proj} \leftarrow (S \setminus d, (g_1, ..., g_{d-1}, g_{d+1}, ..., g_k))$
13: **if** $u_{proj} \notin Den_{k-1}$ **then**
14: // u has a non-dense projection
15: $Cand \leftarrow Cand \setminus u$
16: **continue** loop of line 9
17: **end if**
18: **end for**
19: **end for**

...

CLIQUE - Pseudocode

```
        for all u ∈ Cand do
            selectivity[u] ← 0
            // initialize frequency counters for candidate cells
        end for

        for all v ∈ V do
            // pass over the data
            for all u ∈ Cand do
                if contains(u, v) then
                    // See Eq. 1 for the definition of contains
                    selectivity[u] ← selectivity[u] + 1
                end if
            end for
        end for

        Den_k ← {u | selectivity[u] ≥ τ}
        Den_k ← PruneMDL(Den_k, selectivity)
        // See Alg. 3
        if |Den_k| < 2 then
            break
        end if
    end for

    return ⋃_{k=1}^{D} Den_k
end function
```

CLIQUE - Pseudocode

function $\textsc{Join}(Den_{k-1})$
    $Cand \leftarrow \emptyset$
    for all $((S, \mathbf{g}), (S', \mathbf{g}')) \in Den_{k-1} \times Den_{k-1}$ **do**
        $(d_1, ..., d_{k-1}) \leftarrow sorted(S)$
        $(d'_1, ..., d'_{k-1}) \leftarrow sorted(S')$
        if $(\forall i \in \{1, ..., k-2\} : d_i = d'_i \wedge g_i = g'_i) \wedge d_{k-1} < d'_{k-1}$ **then**
            // if same projection to the subspace of their first $(k-2)$ dimensions
            $c \leftarrow (S \cup S', (g_1, ..., g_{k-2}, g_{k-1}, g'_{k-1}))$
            $Cand \leftarrow Cand \cup \{c\}$
        end if
    end for
    return $Cand$
end function

CLIQUE - Further pruning

Still too complex to be feasible (at least in 1998...)

- Too many candidates
- Throw away dense units in "uninteresting" subspaces
- "Interestingness" = coverage of a subspace

$$cov(S) = \left|\left\{\mathbf{v} \in V \mid \exists \mathbf{g} : sel((S, \mathbf{g})) \geq \tau \wedge contains((S, \mathbf{g}), \mathbf{v})\right\}\right|$$

- Establish a cutting point and throw away lower coverage subspaces
- Cutting point: Minimum Description Length principle

CLIQUE - Further pruning

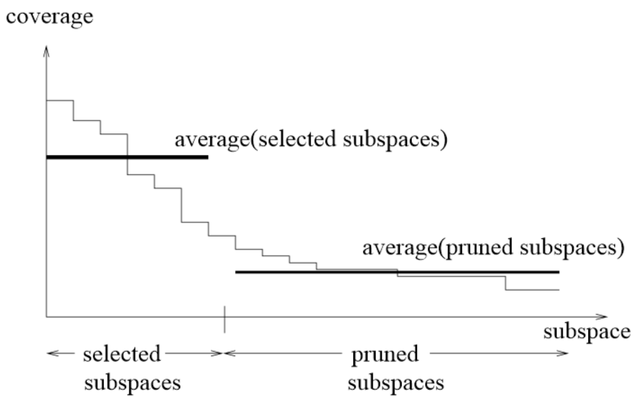
Minimum Description Length principle

- We sort subspaces and split them to 2 groups: kept and pruned
- Imagine storing the following as rounded integers:
  - Mean coverage of kept subspaces
  - Mean coverage of pruned subspaces
  - Absolute deviation of each subspace from the group mean
- How many bits are needed?

$$CL(i) = \log_2 \mu_{keep}(i) + \sum_{j=1}^{i} \log_2 |cov(S_j) - \mu_{keep}(i)| + \log_2 \mu_{prune}(i) + \sum_{j=i+1}^{n} \log_2 |cov(S_j) - \mu_{prune}(i)|$$

- Select i that minimizes this!

CLIQUE - Pruning



(source: Agrawal, 1998)

CLIQUE - Pseudocode

function PRUNEMDL($Den, selectivity$)

  $\sigma \leftarrow \{S \mid \exists \mathbf{g} : (S, \mathbf{g}) \in Den\}$

  for all $S \in \sigma$ **do**

    $cov[S] \leftarrow \sum_{(S,\mathbf{g}) \in Den} selectivity[(S, \mathbf{g})]$

    // Sum of selectivities of dense units

  end for

  $n \leftarrow |\sigma|$

  $(S_1, ..., S_n) \leftarrow$ sort σ by $cov[.]$ to descending order

  $i^* \leftarrow \arg\min_{1<i<n} CL(i)$

  $Den' \leftarrow \{(S, \mathbf{g}) \in Den \mid S \in \{S_1, ..., S_{i^*}\}\}$

  return Den'

end function

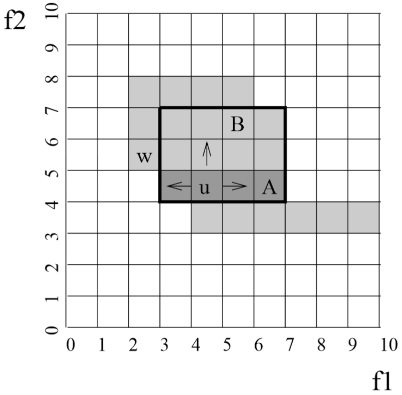
CLIQUE - Connecting dense units

- Done with finding dense units!
- Next step is to find which are connected
- Connected component labeling by depth first search in each subspace separately
- Recursive code in original paper

CLIQUE - Covering (DNF description)

- Now we have our clusters as sets of units in the same subspace
- Better representation needed for intuition
- Authors suggestion: cover cluster with union of hyper-rectangles
- Goal: minimal number of hyper-rectangles for each cluster
- Optimal solution NP-hard
- Use greedy heuristic instead

CLIQUE - Covering



(source: Agrawal, 1998)

CLIQUE - Covering

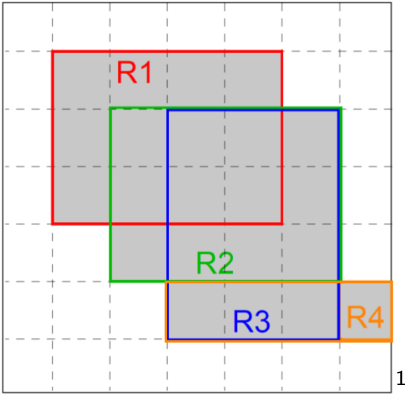
Greedy covering with maximal regions:

- Take an uncovered unit
- Expand a region around it in one dimension as far as dense units allow
- Expand the resulting region in another dimension etc.
- Order of dimensions is randomized

CLIQUE - Covering pseudocode

function GreedyCover(C)
  $(S, G) \leftarrow C$
  $uncovered \leftarrow G$
  $\mathcal{R} \leftarrow \emptyset$
  while $uncovered \neq \emptyset$ **do**
    pick $\mathbf{g} \in uncovered$
    $R \leftarrow Rect(min : \mathbf{g}, max : \mathbf{g})$
    for all $d \in S$ in random order **do**
      $R.min_d \leftarrow \min\left\{x \mid \forall \mathbf{g}' \in Rect(R.min[d \leftarrow x], R.max) : \mathbf{g}' \in G\right\}$
      $R.max_d \leftarrow \max\left\{x \mid \forall \mathbf{g}' \in Rect(R.min, R.max[d \leftarrow x]) : \mathbf{g}' \in G\right\}$
    end for
    $uncovered \leftarrow uncovered \setminus R$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
  end while
  return $\mathcal{R}$
end function

CLIQUE - Covering, redundancies



- Note that R3 is unnecessary. Idea: redundancy elimination
- That is also NP-hard in itself! Use greedy heuristic again.

[1] T. Seidl. *Data Mining Algorithms II - Subspace clustering - slide 19*

CLIQUE - Covering, redundancy elimination

function GreedyEliminateRedundancy($\mathcal{R}$)
    for all $R \in \mathcal{R}$ in ascending order by size **do**
        if $\forall \mathbf{g} \in R : \; \exists R' \in \mathcal{R} \setminus \{R\} : \mathbf{g} \in R'$ **then**
            $\mathcal{R} \leftarrow \mathcal{R} \setminus \{R\}$
        end if
    end for
    return $\mathcal{R}$
end function

CLIQUE - Summarized

CLIQUE is complete now. Summarized:

- Find dense units level by level: candidate generation, histogram building, MDL pruning
- Collect connected units by DFS (clusters)
- Greedily cover clusters and eliminate redundancy

Since CLIQUE

CLIQUE was one of the first subspace clustering algorithms. Since then:

- MAFIA: adaptively spaced grid, no MDL pruning[4]
- SCHISM: selectivity threshold depends on dimensionality of subspace[6]
- Non-redundant subspace clustering: Discard clusters that are sufficiently explained by other, higher dimensional clusters (INSCY, RESCU, OSCLU, STATPC...)
- SUBCLU: no grid, based on DBSCAN ("transitive clustering"), also Apriori-like
- DUSC: extension of SUBCLU, unbiased for dimensionality differences

Overview

CLIQUE and Streaming

Let's turn back to CLIQUE.

- Not suitable for streaming data (multi-pass, no compression or aging)
- How to make it suitable?
  - Derive totally new algorithm based on CLIQUE. (e.g. SOStream[7])
  - Two-phase approach (online/offline separation)
    - Online: maintain statistics about "microclusters" (compression)
    - Offline: use non-streaming algorithm on the microclusters when requested

Streaming data - Microcluster approaches

Multiple such approaches exist. Now concentrate on those available in the SubspaceMOA Framework: CluStream[1] and DenStream[3].

- CluStream incrementally updates q microclusters with the following data:
  - Number of objects
  - Linear sum of objects
  - Squared sum of objects
  - Linear sum of timestamps
  - Squared sum of timestamps
  - (list of identifiers of previous clusters merged into this)
- An incoming object is either merged into an existing microcluster (if there is one sufficiently nearby), or new mcluster is formed
- To keep constant number of mclusters, discard a mcluster with old timestamps (if old enough) or merge the two nearest mclusters.
- Offline algorithm should run on recent data! Therefore: keep snapshots of the situation regularly (pyramidal timeframe)

Streaming data - Microcluster approaches

- DenStream incrementally updates p-microclusters and o-microclusters with the following data:
    - Time-weighted number of objects (weighting by exponential decay)
    - Time-weighted linear sum of objects
    - Time-weighted squared sum of objects
    - (Time of creation for o-microclusters)
- An incoming object is either merged into an existing p or o-microcluster (if its variance would stay low enough), or new o-microcluster is formed. (When merging to o-mcluster, promote to p-mcluster if time-weighted object count high enough)
- To keep a bouned number of mclusters, periodically discard p-microclusters with low time-weighted object count, and o-microclusters created long ago

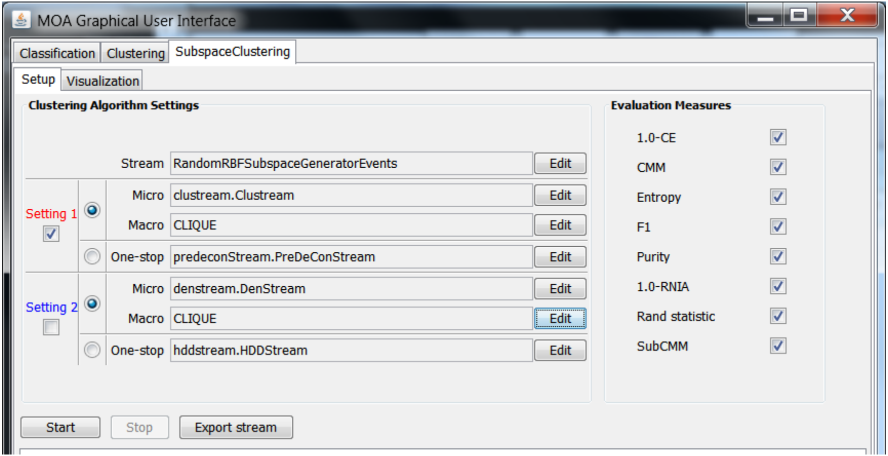
CLIQUE with microclusters

CluStream and DenStream yield microcluster statistics, how to use it in CLIQUE?

- Modify pass over data to pass over microclusters?
- Simpler: (re)generate objects from a distribution fitted to the microcluster

Overview

SubspaceMOA



Evaluation datasets

A real and a synthetic dataset was used

- KDDCup'99: network intrusion data
- Synthetic dataset with two 2D clusters evolving in 3D space

Two algorithms tested

- CluStream+CLIQUE
- DenStream+CLIQUE

KDDCup'99

- Dataset with $\sim$ 5 million objects in 41-dimensional space
- I used a corrected, newer version called NSL-KDD
- Symbolic attributes and ones that are constant over a long horizon had to be removed
- Still 15 dimensions left

Results on synthetic data

Table : Results of CluStream+CLIQUE on the synthetic dataset

1.0-CE	CMM	Entropy	F1	Purity	1.0-RNI	Rand st	SubCMM
0.2656	0.71489	0.35838	0.75783	0.65952	0.31056	0.72356	0.65854

Table : Results of DenStream+CLIQUE on the synthetic dataset

1.0-CE	CMM	Entropy	F1	Purity	1.0-RNI	Rand st	SubCMM
0	0.58571	0.3152	0.60149	0.60331	0	0.72356	0

Table : Results of CluStream+CLIQUE on the real dataset

1.0-CE	CMM	Entropy	F1	Purity	1.0-RNI	Rand st	SubCMM
4.3E-4	0.86318	0.7008	0.66651	0.78043	0.00125	0.99635	0.92804

Table : Results of DenStream+CLIQUE on the real dataset

1.0-CE	CMM	Entropy	F1	Purity	1.0-RNI	Rand st	SubCMM
0.0	0.89291	0.62465	0.66651	0.7168	0.0	0.99635	0.0

Discussion

- Overall CluStream had slightly better results
- However, some metrics were not calculated correctly by SubspaceMOA (1-CE, 1-RNI, SubCMM)
- Better calibration of parameters may be necessary
- Other concern: CluStream and DenStream were not designed for high-dimensional data. There already exist such algorithms:
  - HPStream (projected stream clustering, not only online) [2]
  - GCHDS (grid-based with its own offline component) [5]

Summary

We discussed

- relevance of clustering high-dimensional and streaming data
- details of the CLIQUE algorithm
- main ideas of microcluster approaches like CluStream and DenStream
- connecting microclusters with CLIQUE
- evaluation in SubspaceMOA

Thank you for your attention!

References I

C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.
A framework for clustering evolving data streams.
In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment, 2003.

C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.
A framework for projected clustering of high dimensional data streams.
In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 852–863. VLDB Endowment, 2004.

F. Cao, M. Ester, W. Qian, and A. Zhou.
Density-based clustering over an evolving data stream with noise.
In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 328–339, 2006.

References II

S. Goil, H. Nagesh, and A. Choudhary.
Mafia: Efficient and scalable subspace clustering for very large data sets.
In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 443–452, 1999.

Y. Lu, Y. Sun, G. Xu, and G. Liu.
A grid-based clustering algorithm for high-dimensional data streams.
In *Advanced Data Mining and Applications*, pages 824–831. Springer, 2005.

K. Sequeira and M. Zaki.
Schism: A new approach for interesting subspace mining.
In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 186–193. IEEE, 2004.

References III

S. Wang, Y. Fan, C. Zhang, H. Xu, X. Hao, and Y. Hu.
Subspace clustering of high dimensional data streams.
In *Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on*, pages 165–170. IEEE, 2008.