Lecture 12. Advanced Topics and Applications

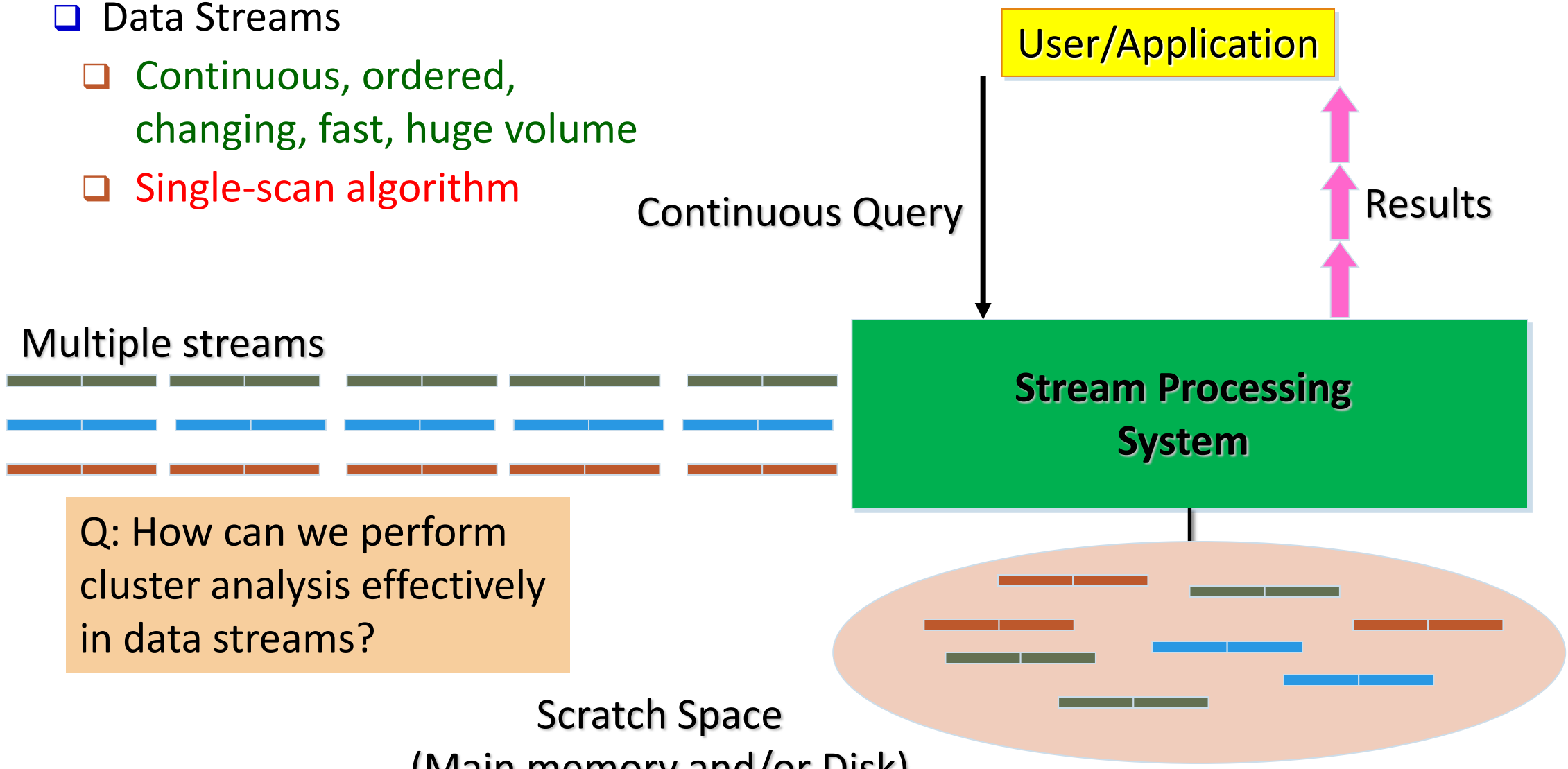
Lecture 12. Advanced Topics and Applications

- Clustering Data Streams: A K-Median-Based Approach
- Clustering Data Streams: The CluStream Approach
- Other Advanced Themes of Cluster Analysis
- Application of Cluster Analysis: Name Disambiguation
- Application of Cluster Analysis: Evolution of Heterogeneous Networks
- Exploration of Broad Applications of Cluster Analysis
- Summary

Session 1. Clustering Data Streams: A K-Median-Based Approach

Stream Data Processing: An Architecture

- Data Streams
  - Continuous, ordered, changing, fast, huge volume
  - Single-scan algorithm



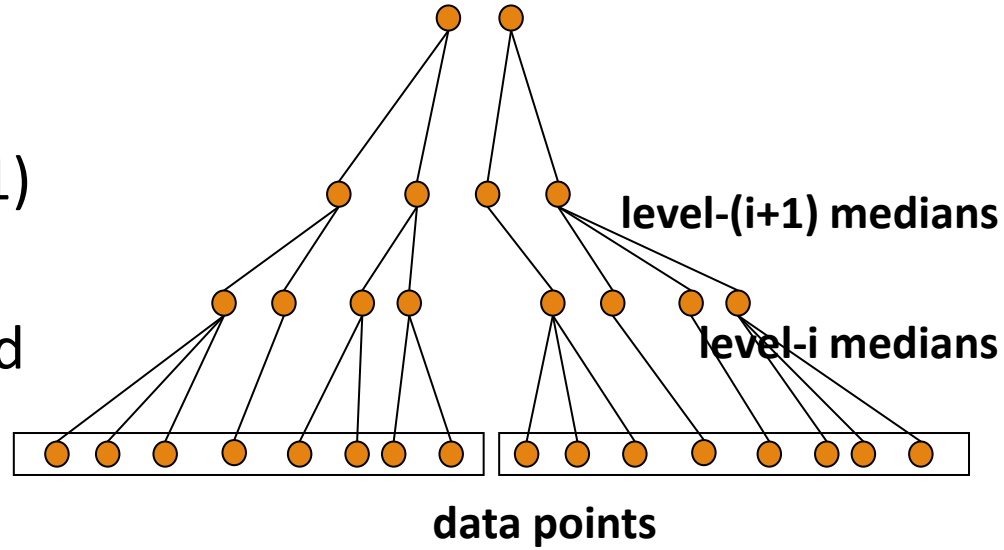
Q: How can we perform cluster analysis effectively in data streams?

Stream Clustering: A K-Median Approach

- O'Callaghan et al. Streaming-Data Algorithms for High-Quality Clustering (ICDE'02)
- Base on the *k-median* method
  - Data stream points are from metric space
  - Find *k* clusters in the stream such that the sum of distances from data points to their closest centers is minimized
- A constant factor approximation algorithm
  - In small space, a simple two-step algorithm
  - For each set of *M* records, S_i, find O(*k*) centers in $S_1, \ldots, S_l$
    - Local clustering: Assign each point in S_i to its closest center
  - Let *S'* be centers for $S_1, \ldots, S_l$ with each center weighted by the number of points assigned to it
    - Cluster *S'* to find *k* centers

Hierarchical Clustering Tree

- **Hierarchical Clustering Tree Method**:
  - Maintain at most *m* level-*i* medians
  - On seeing *m* of them, generate O(*k*) level-(*i*+1) medians of weight equal to the sum of the weights of the intermediate medians assigned to them



- Concerns:
  - Quality will suffer for evolving data streams (maintaining only *m* level-*i* medians)
  - Limited functionality in discovering and exploring clusters over different portions of the stream over time

Session 2. Clustering Data Streams: The CluStream Approach

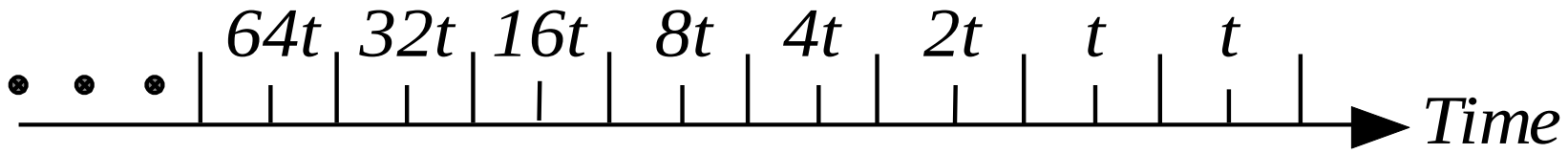
CluStream: A Framework for Clustering Evolving Data Streams

- C. Aggarwal, J. Han, J. Wang, P. S. Yu, A Framework for Clustering Data Streams, VLDB'03
- Design goal of CluStream
  - High quality for clustering evolving data streams with rich functionality
  - Stream mining: One-pass over the stream data, limited space usage, high efficiency
- The CluStream Methodology
  - **Tilted time frame work**: otherwise, will lose dynamic changes
  - **Micro-clustering**: better quality than *k-means*/*k-median*
    - Incremental, online processing, and maintenance
  - **Two stages: micro-clustering and macro-clustering**
    - With *limited overhead* to achieve high efficiency, scalability, quality of results, and power of evolution/change detection

Time Dimension: A Tilted Time Model

- **Tilted time frames:** A trade-off between space and granularity of time
  - Decide at what moments the snapshots of the statistical information are stored
- **Design**: *Natural, logarithmic* and *pyramidal* tilted time frames
  - **Natural tilted time frame:**
    - Ex: Minimal: 15min, then 4 * 15mins → 1 hour, 24 hours → day, …

  - **Logarithmic tilted time frame:**
    - Ex. Minimal: 1 minute, then 1, 2, 4, 8, 16, 32, …



Pyramidal Tilted Time Frame Adopted by CluStream

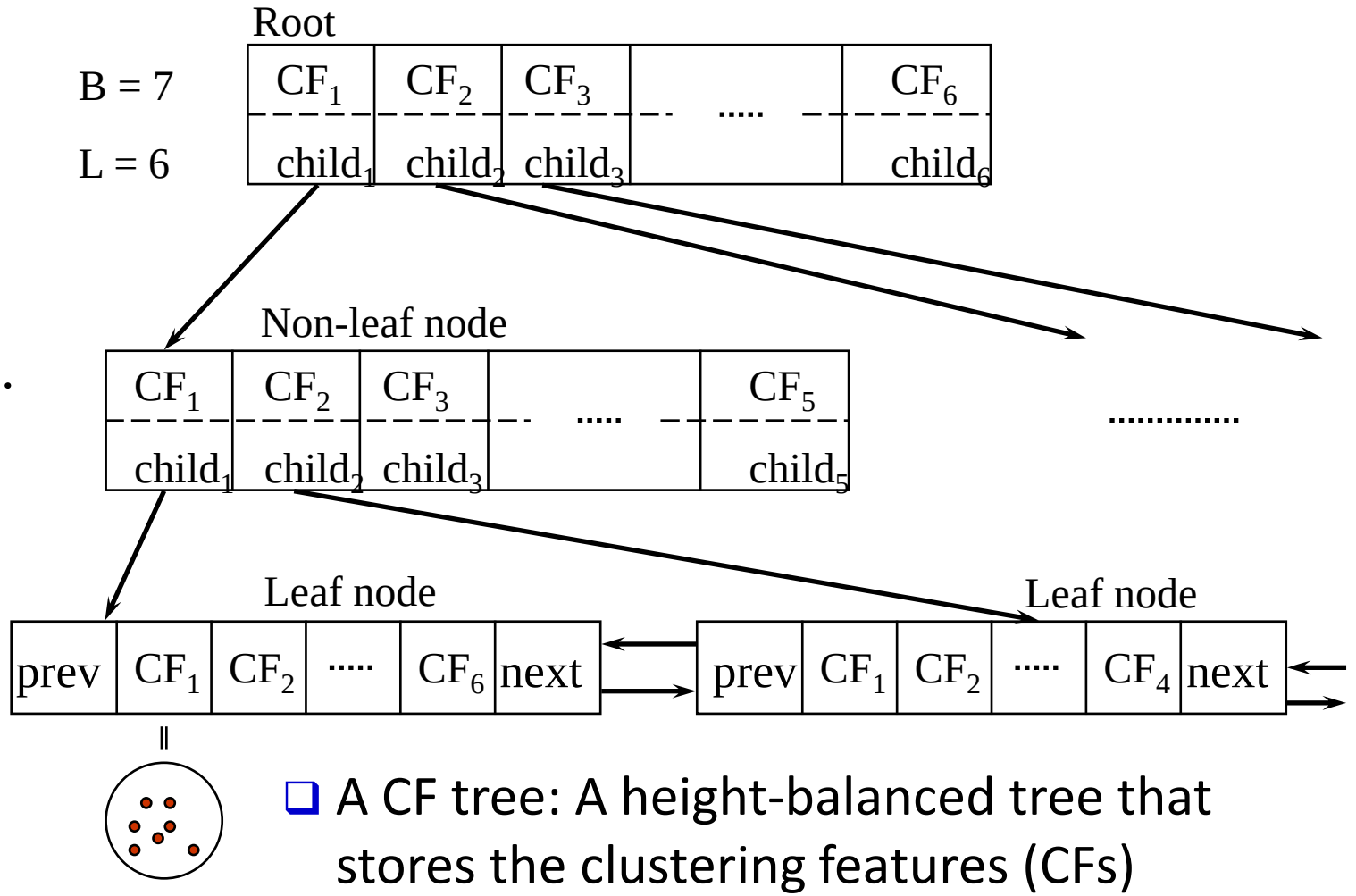
- **Pyramidal tilted time frame:**
  - Example: Suppose there are six frames and each takes a maximal of three snapshots
  - Given a snapshot number *N*, if *N* mod $2^d = 0$, insert into the frame number *d*
    - If there are more than three snapshots, eliminate the oldest one

Frame no.	Snapshots (by clock time)
0	69 67 65
1	70 66 62
2	68 60 52
3	56 40 24
4	48 16
5	64 32

- Snapshots of a set of micro-clusters are stored following the pyramidal pattern
  - They are stored at differing levels of granularity depending on the recency
- Snapshots are classified into different orders varying from 1 to *log(T)*
  - The *i*-th order snapshots occur at intervals of α^i where $\alpha \geq 1$
    - Only the last $(\alpha + 1)$ snapshots are stored

The CluStream Framework: A Micro-Clustering Approach Using the BIRCH CF-Tree Structure

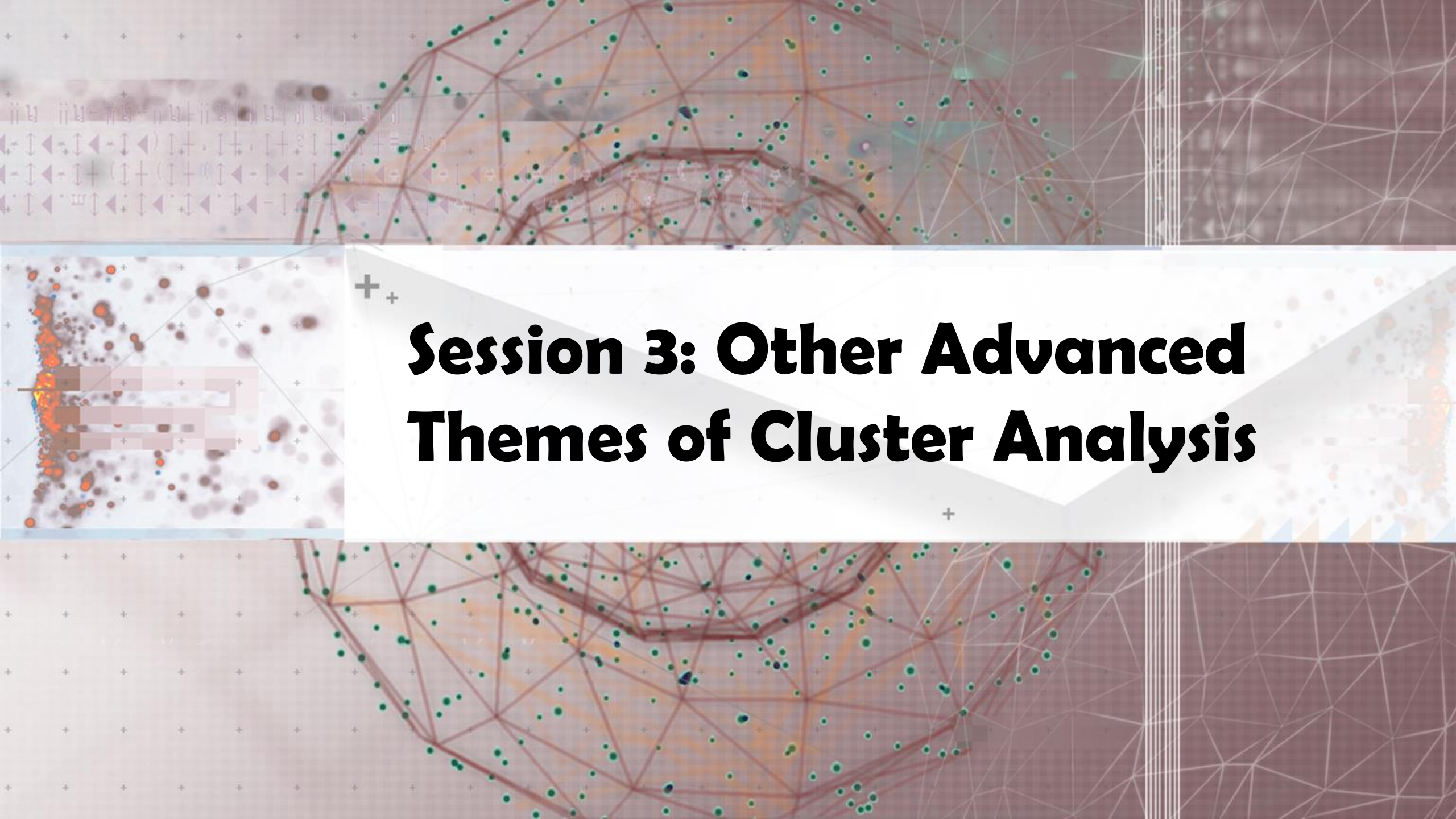
- Micro-clusters stored in CF-Tree
  - Statistical information about data locality
  - Temporal extension of the *cluster-feature vector* $\overline{X}_1 \ldots \overline{X}_k \ldots$
  - Multi-dimensional points with time stamps $T_1 \ldots T_k \ldots$
  - Each point contains *d* dimensions, i.e., $\overline{X}_i = \left(x_i^1 \ldots x_i^d\right)$
  - A micro-cluster for *n* points is defined as a (2*d* + 3) tuple $\left(\overline{CF2^x}, \overline{CF1^x}, CF2^t, CF1^t, n\right)$



- A CF tree: A height-balanced tree that stores the clustering features (CFs)
- The non-leaf nodes store sums of the CFs of their children

CluStream: Clustering Evolving On-Line Data Streams

- Divide the clustering process into *online* and *offline* components
  - **Online component** (**micro-cluster maintenance**)
    - Periodically store summary statistics about the stream data
      - Initially, create *q* micro-clusters
        - *q* is usually significantly larger than the number of natural clusters
      - Online incremental update of micro-clusters
        - If new point is within max-boundary, insert into the micro-cluster
        - Otherwise, create a new cluster
        - May delete obsolete micro-clusters or merge two closest ones
  - **Offline component** (**query-based macro-clustering**)
    - Answers various user questions based on the stored summary statistics
    - Based on a user-specified time-horizon *h* and the number of macro-clusters *k*, compute macro-clusters using the *k*-means algorithm



Session 3: Other Advanced Themes of Cluster Analysis

Ensemble Clustering

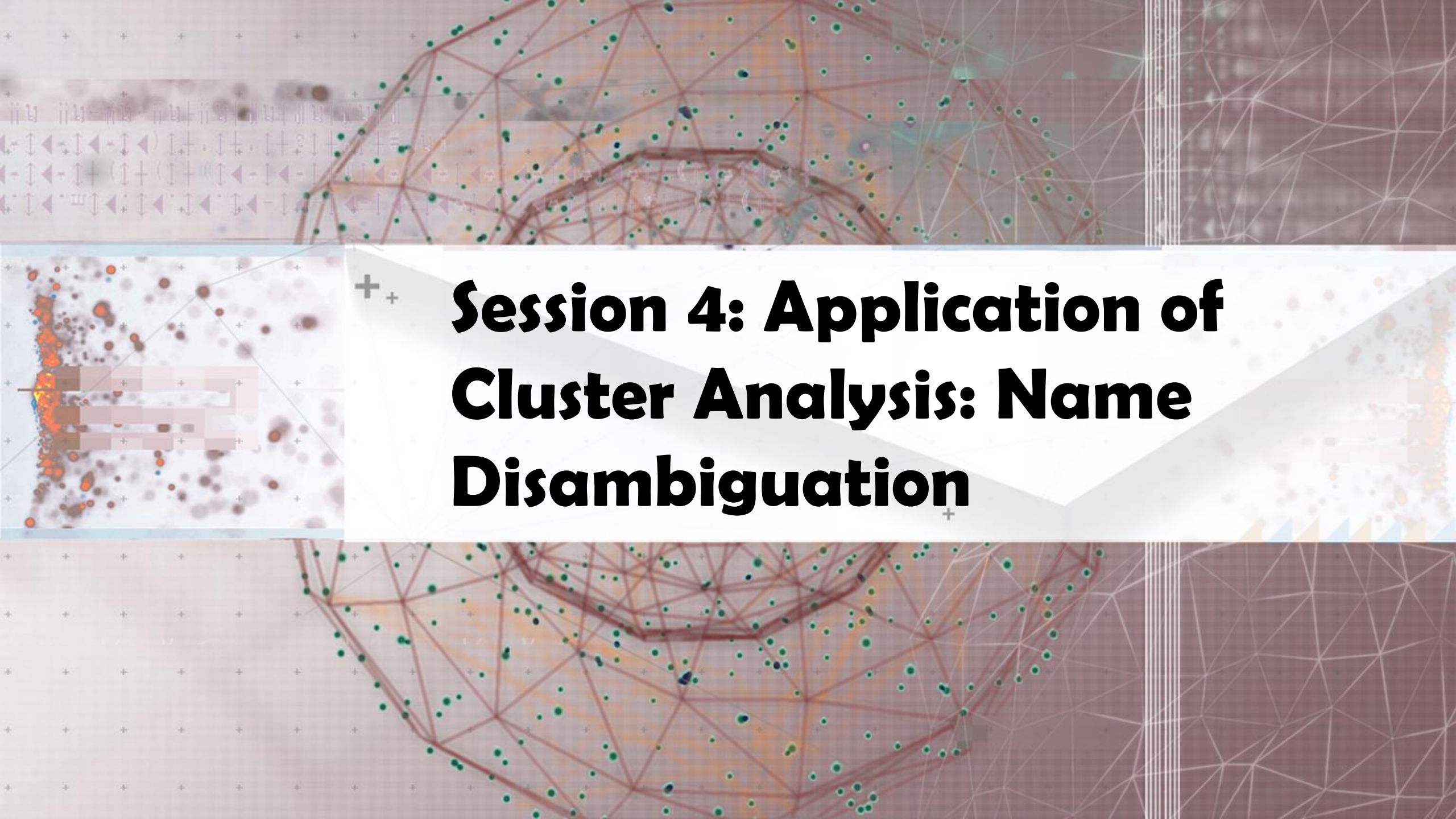
- **Ensemble clustering**: Combine the results of many clustering models to create a more robust clustering
  - No single model or criterion truly captures the optimal clustering, but an ensemble of models will provide a more robust solution
- **Methodology**
  - **Generate *k* different clusterings (i.e., *ensemble components*)**
  - **Combine the different results into a single and more robust clustering**
- Selection of ensemble components: Model-based or data selection-based
- Combining different ensemble components
  - Hypergraph partitioning: Each data point is a vertex, and a cluster in any of the ensemble components is represented as a *hyper-edge*
  - Meta-clustering: A graph-based approach, except that vertices are associated with each cluster in the ensemble components

Clustering for Effective Data Mining

- **Clustering for Data Summarization**
  - A natural form of summarization based on the notion of similarity
- **Outlier Analysis**
  - Outliers: Data points that are far from any particular cluster
- **Classification**
  - Clustering can help speed up KNN classification by replacing the data points with centroids of fine-grained clusters belonging to a particular class
- **Dimensionality Reduction**
  - NMF, spectral clustering, probabilistic latent semantic indexing
- **Similarity Search and Indexing**
  - Hierarchical clustering may create indexing structures, such as CF-Tree

Clustering Big Data

- Apache Spark: A fast and general engine for large-scale data processing
  - The most active project in big data (500+ contributors and 500+ production deployments)
- User-friendly and expressive APIs in Python, Java, Scala, and R
- Ultra fast and fault-tolerant in-memory and on-disk computation
  - Hold data in memory and provide fast access during iterations
  - Recover node failures via Resilient Distributed Dataset (RDD)
- Feature-rich standard components: machine learning, graph computation, SQL and DataFrames, and streaming processing
  - Clustering algorithms: *k-means* and *streaming k-means*, *power iteration clustering*, *Gaussian mixtures*, *latent Dirichlet allocation*, and more coming
- Resources: Documentation (clustering), events and meetups, MOOCs and workshops

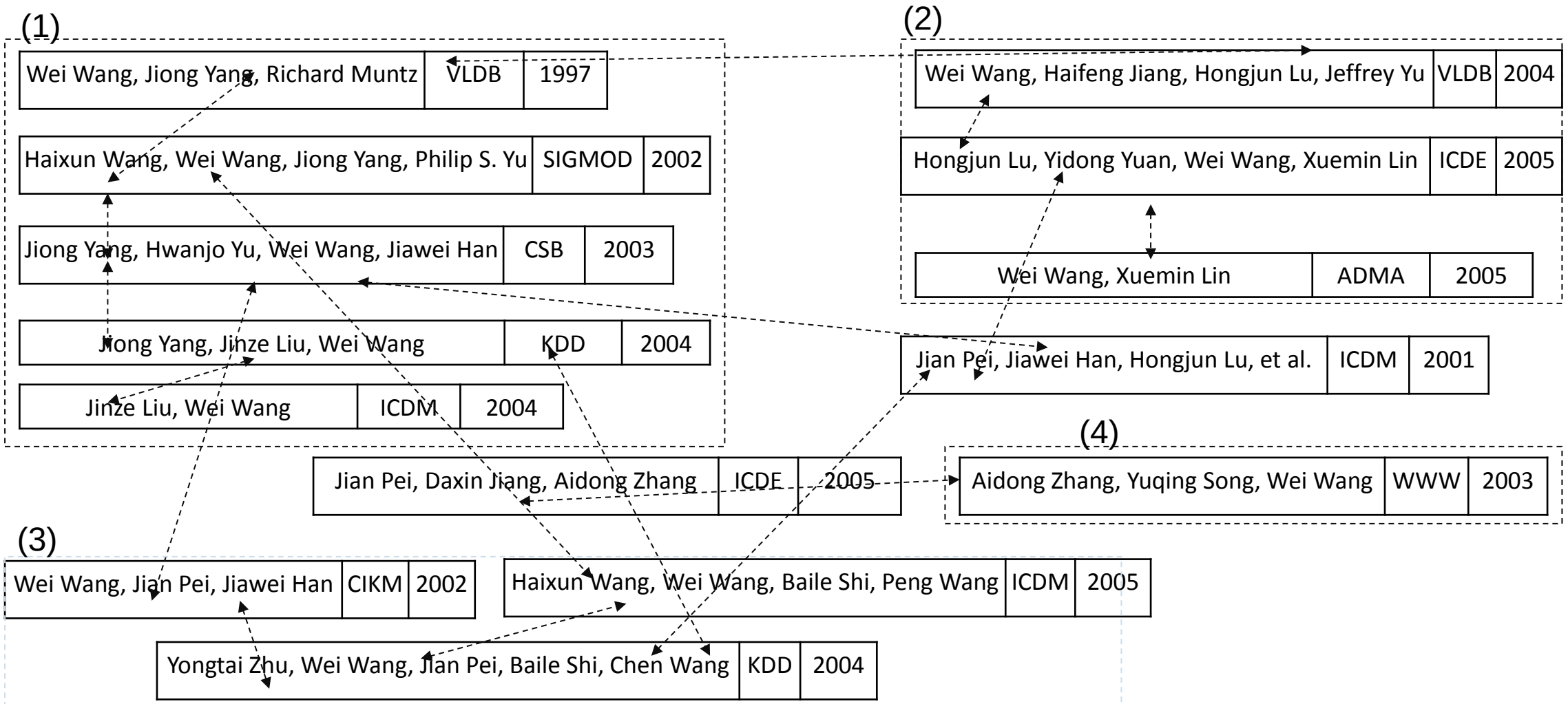


Session 4: Application of Cluster Analysis: Name Disambiguation

Data Cleaning by Link-Based Cluster Analysis

- Object reconciliation vs. object distinction as data cleaning tasks
- Object distinction: Different people/objects do share names
  - In AllMusic.com, 72 songs and 3 albums named “Forgotten” or “The Forgotten”
  - In DBLP, 141 papers are written by at least 14 different people with the same name: *Wei Wang*
- New challenges of object distinction:
  - Textual similarity cannot be used since they have the exact same name in DBLP
- Link analysis may take advantages of redundancy to facilitate entity cross-checking and validation
- Distinct: Object distinction by information network-based cluster analysis
  - X. Yin, J. Han, and P. S. Yu, “Object Distinction: Distinguishing Objects with Identical Names by Link Analysis”, ICDE'07

Entity Distinction: The "Wei Wang" Challenge in DBLP



(1) Wei Wang at UNC (2) Wei Wang at UNSW, Australia
(3) Wei Wang at Fudan Univ., China (4) Wei Wang at SUNY Buffalo

The DISTINCT Methodology

- **Measure similarity between references**
  - **Link-based similarity**: Linkages between references
    - References to the same object are more likely to be connected (Using random walk probability)
  - **Neighborhood similarity**
    - Neighbor tuples of each reference can indicate similarity between their contexts
- **Self-boosting: Training using the *same* bulky data set**
- **Reference-based clustering**
  - Group references according to their similarities

Training with the "Same" Data Set

- Build a training set automatically
  - Select distinct names, e.g., Johannes Gehrke, Mike Stonebraker
  - The collaboration behaviors within the same community share some similarity
  - Training parameters using a typical and large set of *unambiguous* examples
- Use SVM to learn a model for combining different join paths
  - Each join path is used as two attributes (with link-based similarity and neighborhood similarity)
  - The model is a weighted sum of all attributes

Clustering: Measure Similarity between Clusters

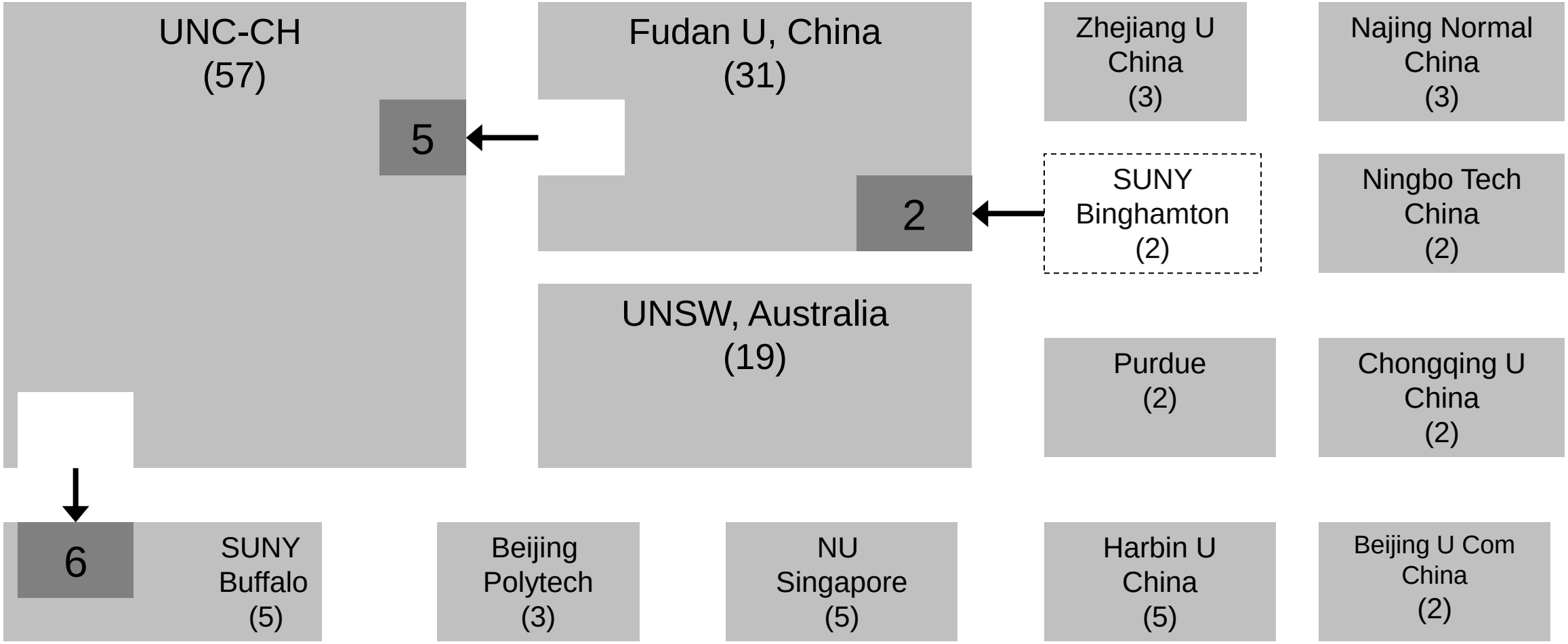
- **Single-link** (highest similarity between points in two clusters) ?
  - No, because references to different objects can be connected
- **Complete-link** (minimum similarity between them)?
  - No, because references to the same object may be weakly connected
- **Average-link** (average similarity between points in two clusters)?
  - A better measure
  - *Refinement: Average neighborhood similarity* and *collective random walk probability*

Test on Real Cases: DBLP Popular Names

Name	#authors	#refs	Accuracy	Precision	Recall	F-measure
Hui Fang	3	9	1.0	1.0	1.0	1.0
Ajay Gupta	4	16	1.0	1.0	1.0	1.0
Joseph Hellerstein	2	151	0.81	1.0	0.81	0.895
Rakesh Kumar	2	36	1.0	1.0	1.0	1.0
Michael Wagner	5	29	0.395	1.0	0.395	0.566
Bing Liu	6	89	0.825	1.0	0.825	0.904
Jim Smith	3	19	0.829	0.888	0.926	0.906
Lei Wang	13	55	0.863	0.92	0.932	0.926
Wei Wang	14	141	0.716	0.855	0.814	0.834
Bin Yu	5	44	0.658	1.0	0.658	0.794
average			**0.81**	**0.966**	**0.836**	**0.883**

Distinguishing Different People Named *Wei Wang*

- Quality clustering—grouping majority cases correctly



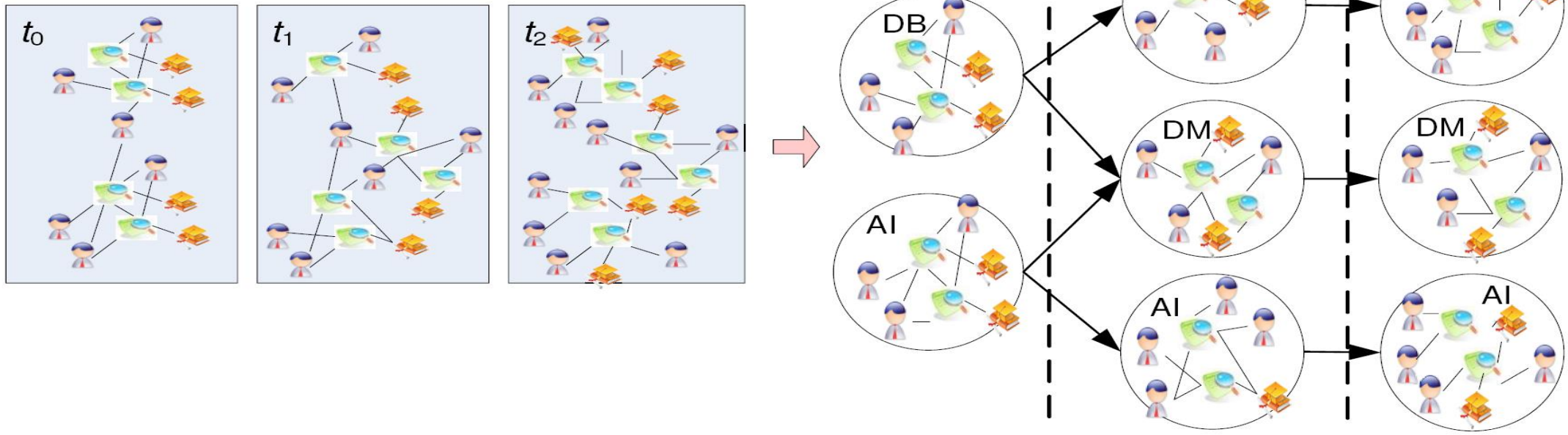
Session 5: Application of Cluster Analysis: Evolution of Heterogeneous Networks

Mining Evolution and Dynamics of Heterogeneous Networks

- Networks evolve with time
  - DBLP networks: Network sequences formed based on paper publication years
- Motivation: Model evolution of communities in heterogeneous networks
  - Automatically detect the best number of communities in each timestamp
  - Model the smoothness between communities of adjacent timestamps
  - Model the evolution structure explicitly: Birth, death, split of subnetworks
- EvoNetClus: Clustering for modeling evolution of dynamic heterogeneous networks
  - Co-evolution within a community
    - Heterogeneous multi-typed object/links
  - Discovery of evolution structures among different communities
- Y. Sun, et al., "Studying Co-Evolution of Multi-Typed Objects in Dynamic Heterogeneous Information Networks", MLG'10 (later version appears at TKDE'15)

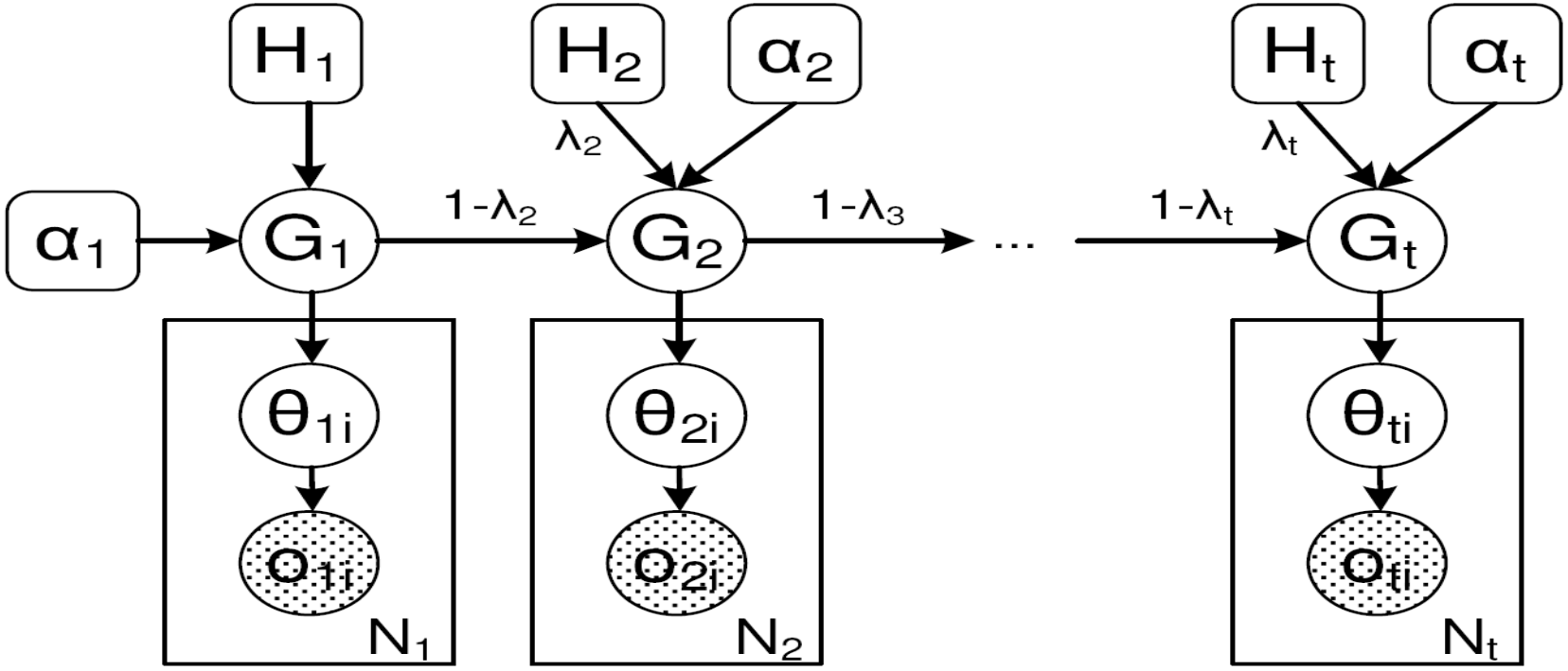
Evolution: Idea Illustration

- From network sequences to evolutionary communities (i.e., multi-typed object clusters)



Graphical Model: A Generative Model

- **Dirichlet Process Mixture Model-based generative model**
  - At each timestamp, a community is dependent on historical communities and background community distribution



Generative Model & Model Inference

- To generate a new paper o_i
  - Decide whether to join an existing community or a new one
    - Join an existing community k with probability $n_k/(I-1+\alpha)$
    - Join a new community k with probability $\alpha/(I-1+\alpha)$: Decide its prior, either from a background distribution (λ) or historical communities ($(1-\lambda)\pi_k$), with different probabilities, and draw the attribute distribution from the prior
  - Generate o_i according to the attribute distribution

$$
\begin{aligned}
p(o_{i,t}|z_{i,t}=k,\mathbf{\Theta}_t) &= p(o_{i,t}|\theta_{k,t}) \\
&= p(\mathbf{a}_{i,t}|\theta^A_{k,t})p(\mathbf{c}_{i,t}|\theta^C_{k,t})p(\mathbf{d}_{i,t}|\theta^D_{k,t}) \\
&= \prod_{j=1}^{|A|}\theta^A_{k,t}(j)^{a_{ij,t}}\prod_{j=1}^{|C|}\theta^C_{k,t}(j)^{c_{ij,t}}\prod_{j=1}^{|D|}\theta^D_{k,t}(j)^{d_{ij,t}}
\end{aligned}
$$

- Greedy inference for each timestamp: Collapse Gibbs sampling, which is trying to sample cluster label for each target object (e.g., paper)

Community Evolution Discovery: Algorithm

- Step 1: Generating Prior Groups
  - Calculation of the posterior probabilities of hidden labels

$$p(z_{t,i} = k | o_{t,i}, \mathbf{z_{t-1}}, \mathbf{O_{t-1}}, H_t) \propto n_{t-1,k} f_k^{-o_{t,i}}(o_{t,i})$$

$$p(z_{t,i} = k_{new} | o_{t,i}, H_t) \propto \gamma f_{k_{new}}(o_{t,i})$$

- Step 2: Iterative Hidden Community Label Assignment
  - Using an EM Algorithm

$$p(z_{ji} = k, k \notin \{K_{t-1}\} | \mathbf{z}_{-ji}, \mathbf{o}) \propto (n_{j\cdot k}^{-ji} + \alpha\eta_k) f_k^{-o_{ji}}(o_{ji})$$

$$p(z_{ji} = k, k \in \{K_{t-1}\} | \mathbf{z}_{-ji}, \mathbf{o}) \propto (n_{j\cdot k}^{-ji} + \lambda \frac{n_{t-1,k}}{N_{t-1}} + \alpha\eta_k) f_k^{-o_{ji}}(o_{ji})$$

$$p(z_{ji} = k_{new} | o_{ji}) \propto \alpha\eta_u f_{k_{new}}(o_{ij})$$

Input: Network $G_t, G_{t-1}, \mathbf{z}_{t-1}$; γ, α, β, λ ;
Output: The community assignment vector $\mathbf{z}_t$; the parameters $\mathbf{\Theta}_t = \{\theta_t^A, \theta_t^C, \theta_t^W\}$;

Assign each object into prior groups;
repeat
  for *each object* o_{ji} **do**
    1. E-step: Assign o_{ji} to the community with the maximum posterior probability in either existing community k or a new community $k+1$;
    2. M-step: Update relevant statistics ;
    3. if community k_{old} for o_i contains no objects, remove the community;
  end
until *reaches cluster change threshold*;
Estimate parameters $\mathbf{\Theta}_t$ for each community;

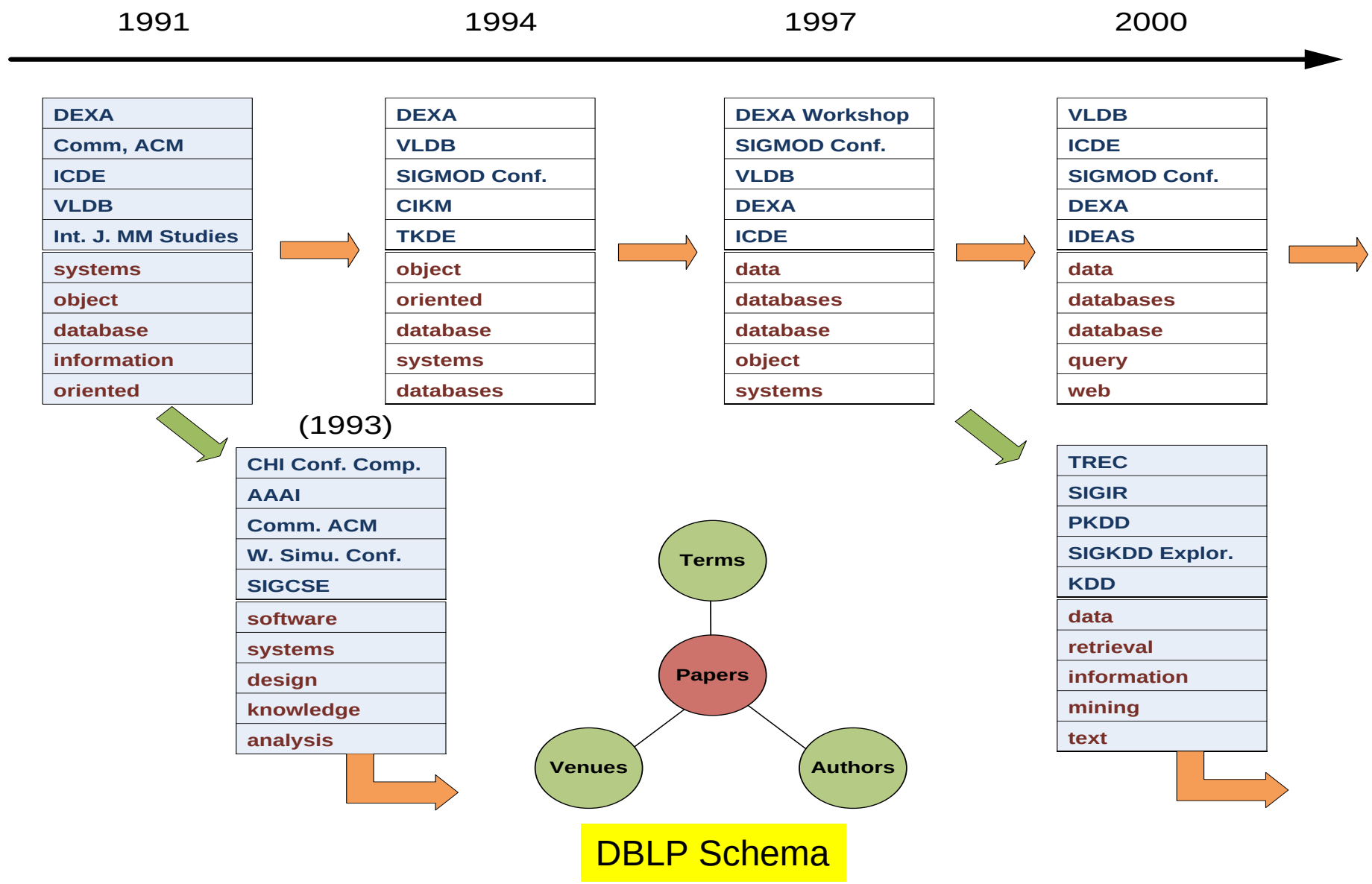
Algorithm 1: Parameter Estimation Algorithm.

- Step 3: Community Distribution Estimation:

$$\theta_k^A(j) = \frac{\beta_A + n_j^A}{\beta_A |A| + n^A}; \theta_k^C(j) = \frac{\beta_C + n_j^C}{\beta_C |C| + n^C}; \theta_k^W(j) = \frac{\beta_D + n_j^W}{\beta_D |W| + n^W}$$

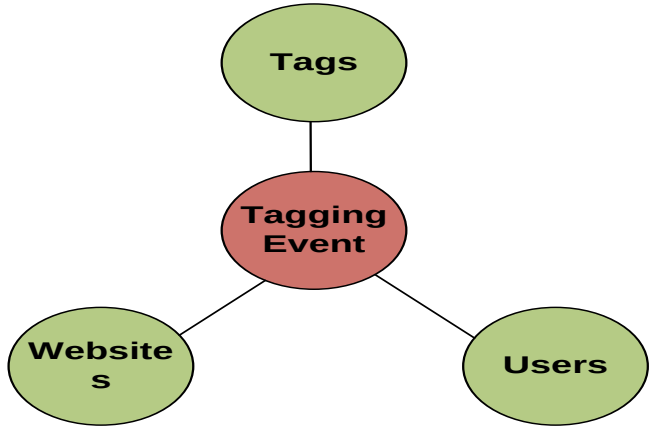
Case Study on DBLP

- Tracking database community evolution
- Accuracy study: Using perplexity as a measure
  - The more types of objects used, the better accuracy
  - Historical prior results in better accuracy

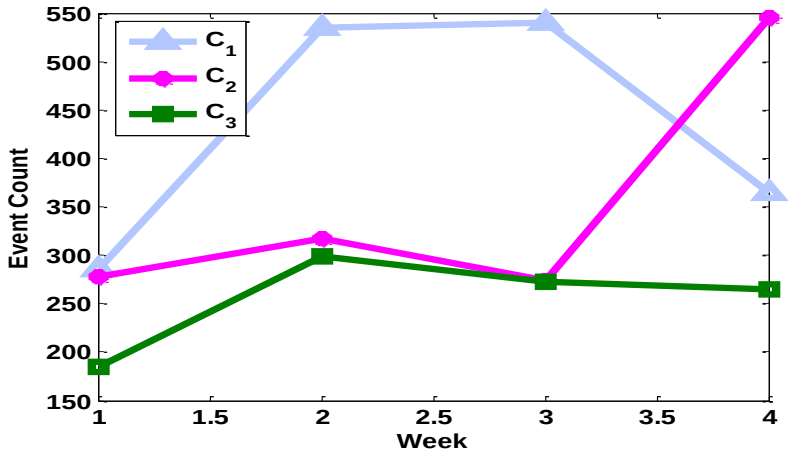


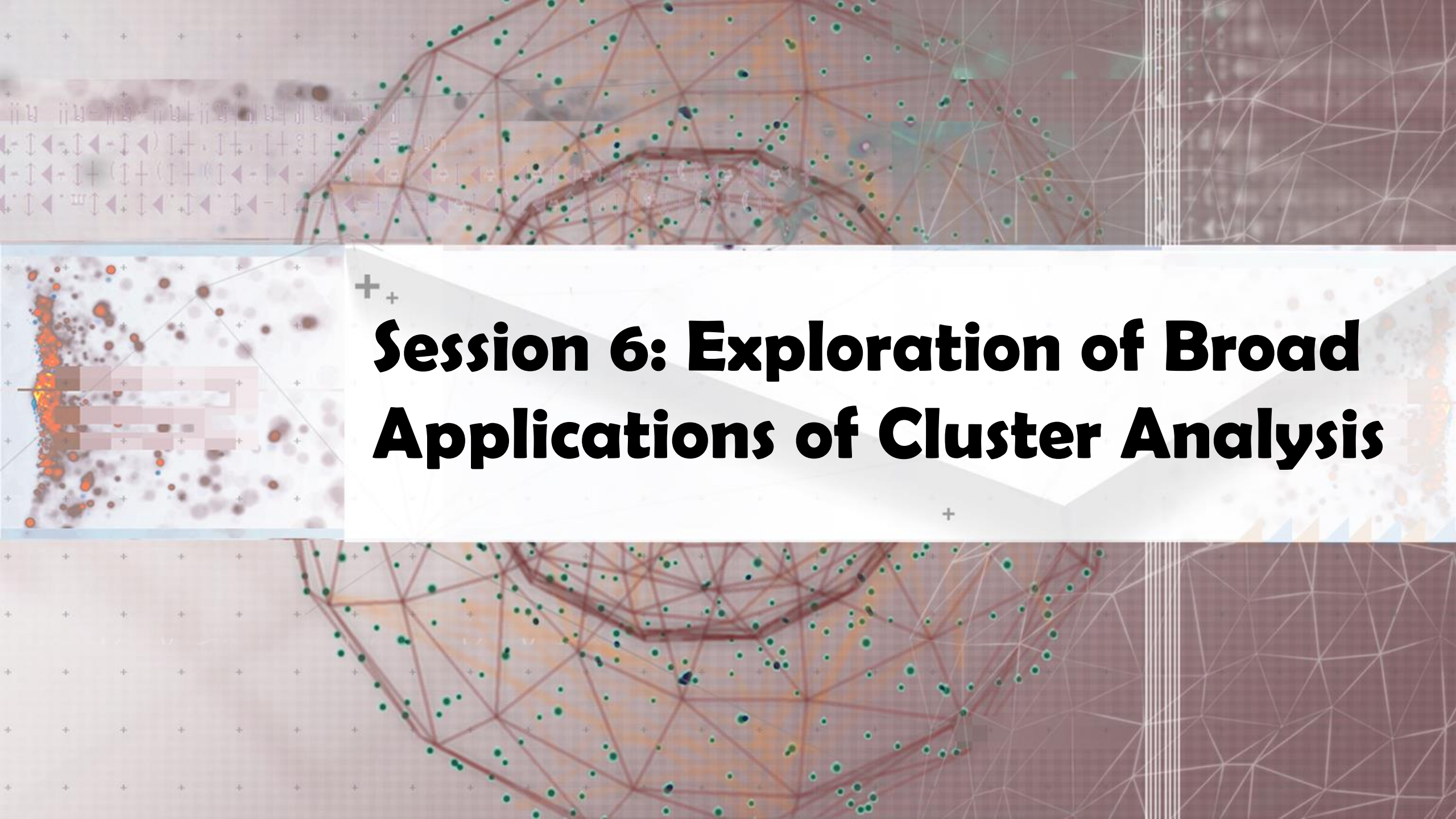
Evolution of Social Events: Case Study on Delicious.com

	Jan. 1 - Jan. 7	Jan. 8 – Jan. 14	Jan. 15 – Jan. 21	Jan. 22 – Jan. 28
C_1:	Security	Google	Security	Google
	Terrorism	China	Google	Security
	Politics	Security	China	China
	Travel	Internet	Internet	Internet
	Usa	Privacy	Microsoft	Privacy
	Airport	Politics	Privacy	Digg
	Israel	Censorship	Censorship	Politics
	Obama	Facebook	Politics	Datenschutz
	CIA	Business	Browser	Facebook
	Afghanistan	Terrorism	USA	USA
C_2:	Mac	Iphone	Iphone	Ipad
	Apple	Apple	Apple	Apple
	Iphone	Twitter	Mac	Iphone
	Windows	Mac	Mobile	Technology
	Tablet	Mobile	Twitter	Tablet
	Ipod	Apps	Software	Mac
	Tips	Ratio	Apps	Mobile
	Macbook	Blog	Business	Newspapers
	Tutorial	Newspapers	Osx	Kindle
	Drm	Technology	Radio	Media
C_3:	Health	Weather	Haiti	Haiti
	Depression	UK	Photography	BBC
	Sleep	Photography	BBC	Photography
	Teenagers	Photo	Earthquake	Animals
	Dubai	Haiti	Photos	Earthquake
	Tallest	Photos	UK	2010
	BBC	2010	2010	Photos
	Building	BBC	Disaster	Nature
	Architecture	Snow	Travel	Funny
	Mentalhealth	Earthquake	Wildlife	Theonion



Delicious.com Schema





Session 6: Exploration of Broad Applications of Cluster Analysis

Cluster Analysis for Broad Applications (I)

- **Customer Segmentation and Collaborative Filtering**
  - Clustering of customers based on the similarities of their profiles and shopping behaviors
  - *Collaborative filtering* and *recommendation systems* based on customer ratings of products
- **Text Mining**
  - Topic modeling: Co-clustering of words and documents
  - Clustering, typing, profiling, and in-depth analysis of large collections of text/web documents
- **Clustering of multimedia and social media**
  - Summarizing multimedia data (e.g., videos) and integrated analysis of news and tweets

Cluster Analysis for Broad Applications (II)

- **Temporal and Sequence Applications**
  - Clustering Web log click streams to find patterns of users and reorganize Web page structures
  - Clustering biological sequence data to construct biological models and discover anomalies
- **Social Network Analysis**
  - Clustering for finding hidden social communities—*community detection*
  - Subsequent analysis for anomaly detection, classification, influence analysis, and link prediction
- **Many more emerging applications**
  - Waiting for you to contribute!

Session 7: Summary

Summary: Advanced Topics and Applications

- Clustering Data Streams: A K-Median-Based Approach
- Clustering Data Streams: The CluStream Approach
- Other Advanced Themes of Cluster Analysis
- Application of Cluster Analysis: Name Disambiguation
- Application of Cluster Analysis: Evolution of Heterogeneous Networks
- Exploration of Broad Applications of Cluster Analysis
- Summary

Recommended Readings

- L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, S. Guha, “Streaming-Data Algorithms for High-Quality Clustering”, ICDE’02
- C. Aggarwal, J. Han, J. Wang, P. S. Yu, “A Framework for Clustering Data Streams”, VLDB'03
- X. Yin, J. Han, P. S. Yu, "Object Distinction: Distinguishing Objects with Identical Names by Link Analysis", ICDE'07
- J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd ed. , 2011
- C. Aggarwal and C. K. Reddy (eds.). Data Clustering: Algorithms and Applications. CRC Press, 2014
- C. Aggarwal. Data Mining: The Textbook. Springer, 2015
- Y. Sun, J. Tang, J. Han, C. Chen, M. Gupta, "Co-Evolution of Multi-Typed Objects in Dynamic Heterogeneous Information Networks", IEEE Trans. Know. & Data Eng., 2015