Hierarchical Clustering for Real-Time Stream Data with Noise

Philipp Kranen, Felix Reidl, Fernando Sanchez Villaamil, Thomas Seidl

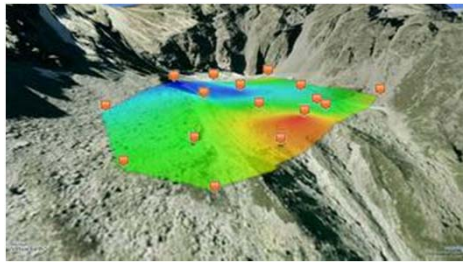
Data Management and Data Exploration Group
RWTH Aachen University, Germany

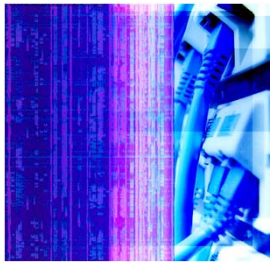
Agenda

- Motivation and problem statement
- The ClusTree algorithm [ICDM '09]
- The LiarTree algorithm
- Evaluation
- Conclusion

Motivation and problem statement

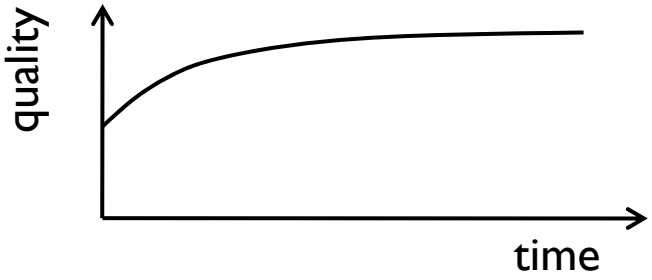
- Data streams are ubiquitous
  - Sensor measurements, network traffic, customer data, surveillance data, …
- Clustering is a frequently used technique
  - Reduces amount of data, provides an overview of the data distribution
- Stream clustering challenges:
  - Single pass, limited time, limited memory (yet least information loss)
  - **Noisy data & changing distributions**
  - **Varying time allowance**



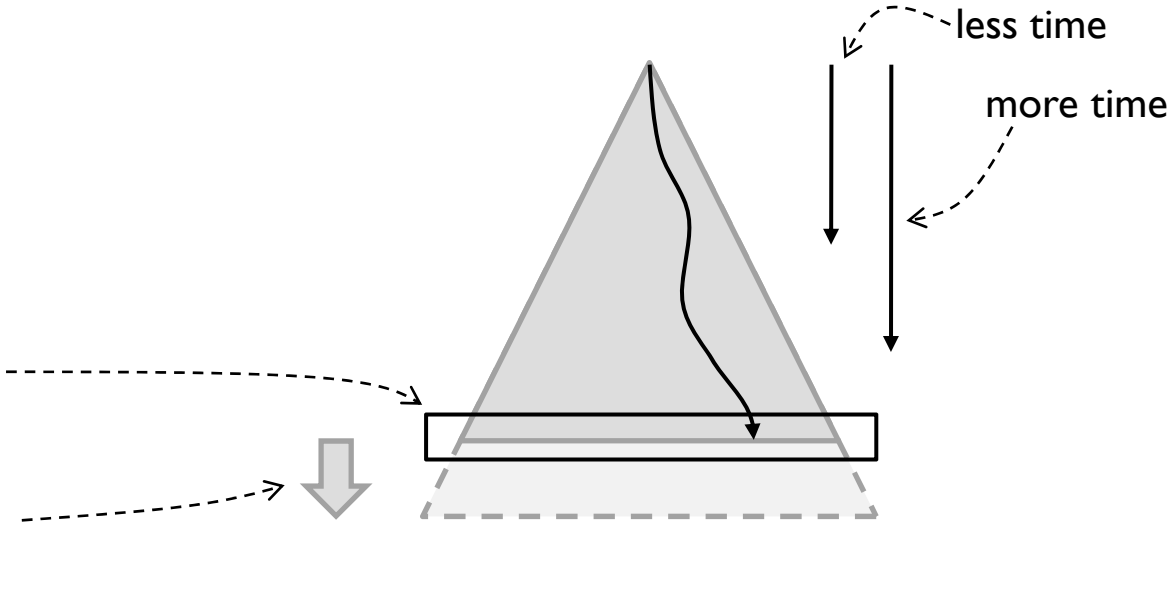
Motivation and problem statement

- Most approaches have to restrict themselves to the worst case time
- Anytime algorithms (cf. classification, learning)
  - **Provide a result after any point in time**
  - Improve quality with additional time



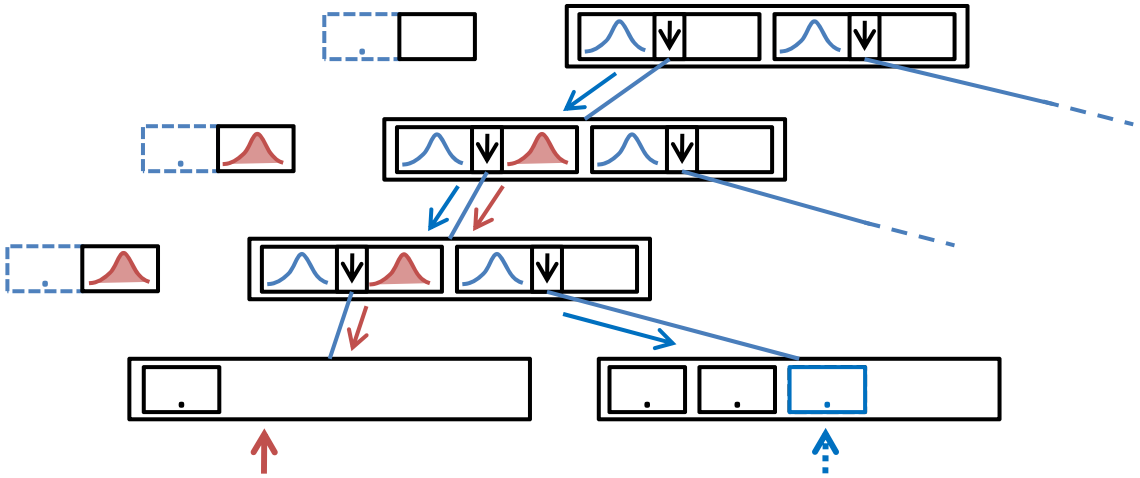
ClusTree[1] – basic idea

- Cluster features CF = (N, LS, SS) represent micro-clusters
- Maintain a balanced hierarchical data structure
  - Insert new object into the closest subtree
  - Insertion stops if next object arrives
  - Most detailed model is stored at leaf level
  - Tree (= model) grows if more time is available



[1] Kranen, Assent, Baldauf, Seidl. *Self-adaptive Anytime Stream Clustering*. ICMD 2009

Buffer and hitchhiker



- **Buffer: interrupt insertion** – aggregate objects on interrupt
- **Hitchhiker: resume insertion** – take buffer along (if same way)

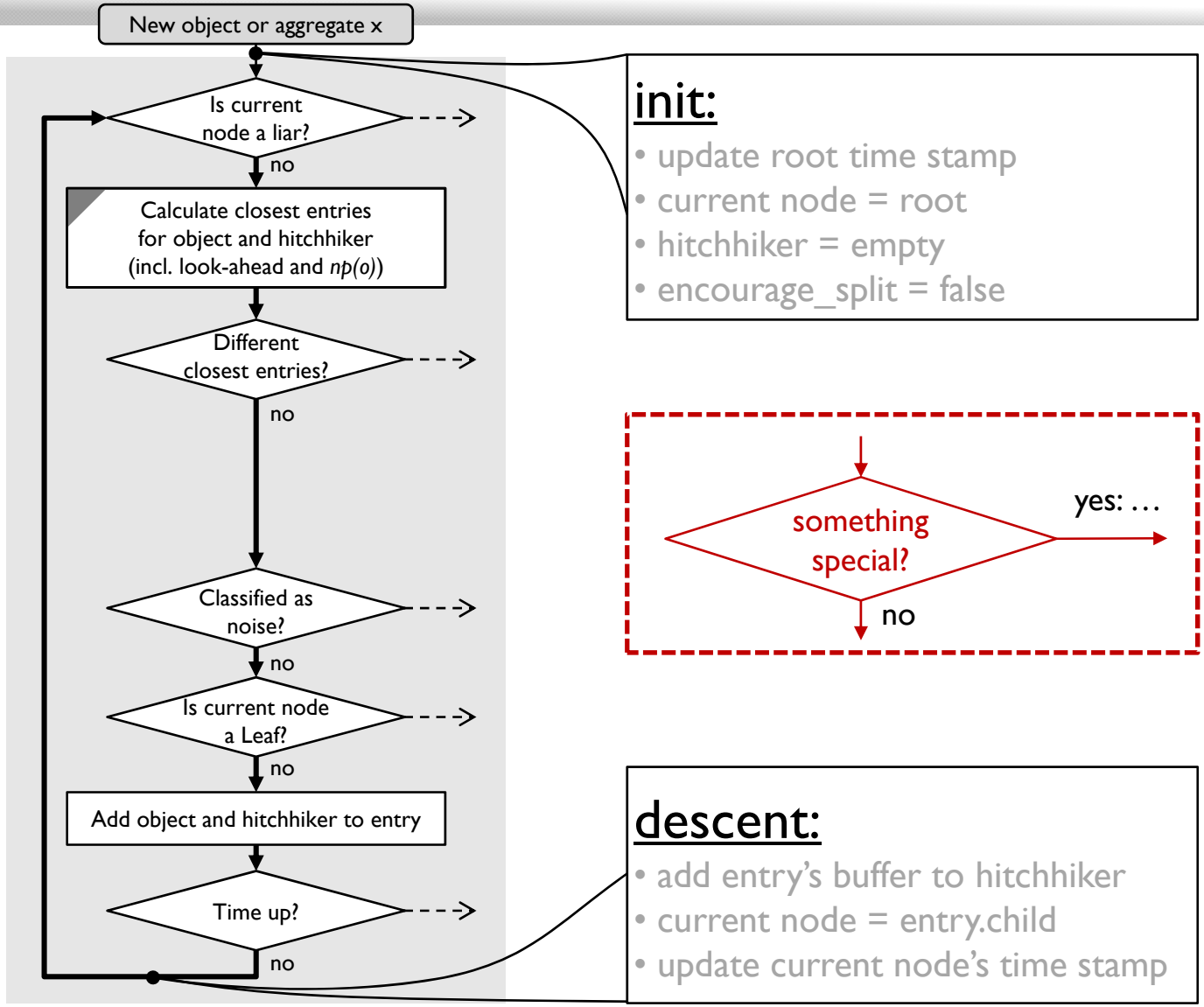
  → Maximally two objects to descend with
- Tree grows through splitting nodes starting from the leaf

Goals

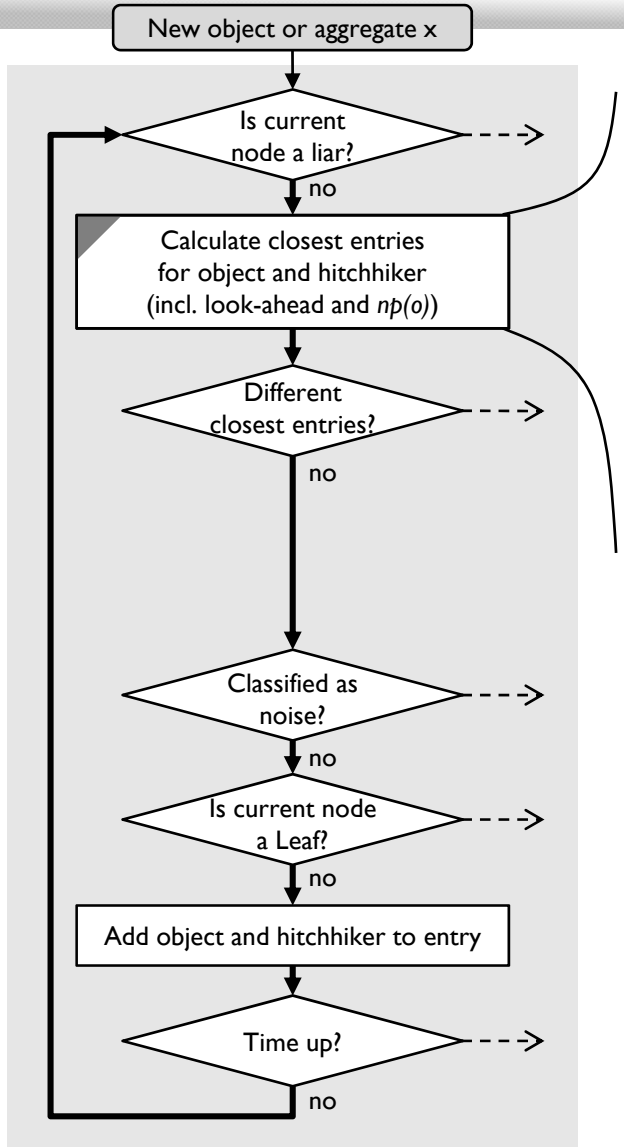
- The ClusTree does not adress
  - **Overlapping** of inner entries
  - **Noise**, every point is treated equally

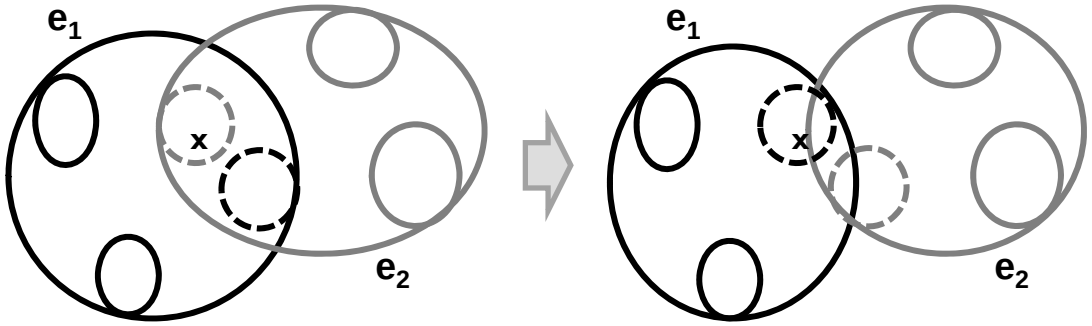
- The goals of the LiarTree are
  - Reduce overlapping of inner entries
  - Incorporate explicit noise handling
  - Allow the transition from noise to novel concepts
  - Maintain the advantages of the ClusTree
    - Anytime clustering
    - Self-adaptive model size

LiarTree algorithm – overview



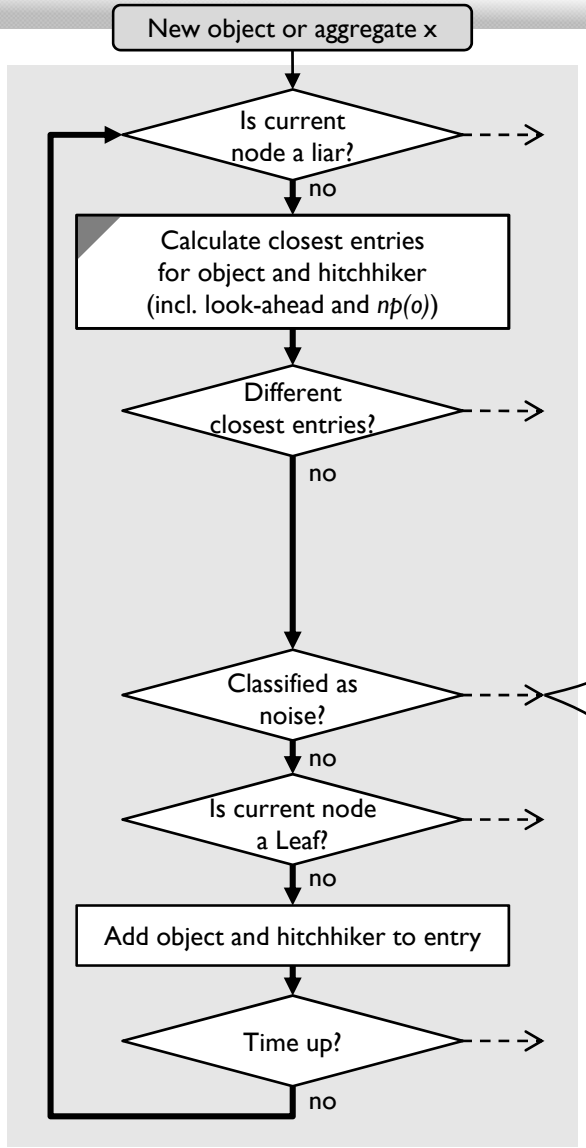
LiarTree algorithm – look ahead and *np(o)*





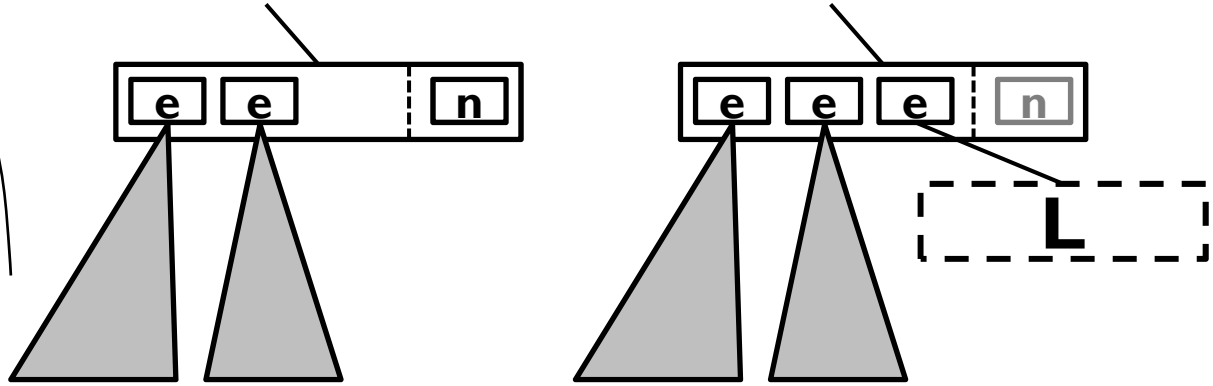
$$np(o) = \min_{e_i \in node} \{\{dist(o, \mu_{e_i})/r_{e_i}\} \cup \{1\}\}$$

LiarTree algorithm – noise-to-cluster event

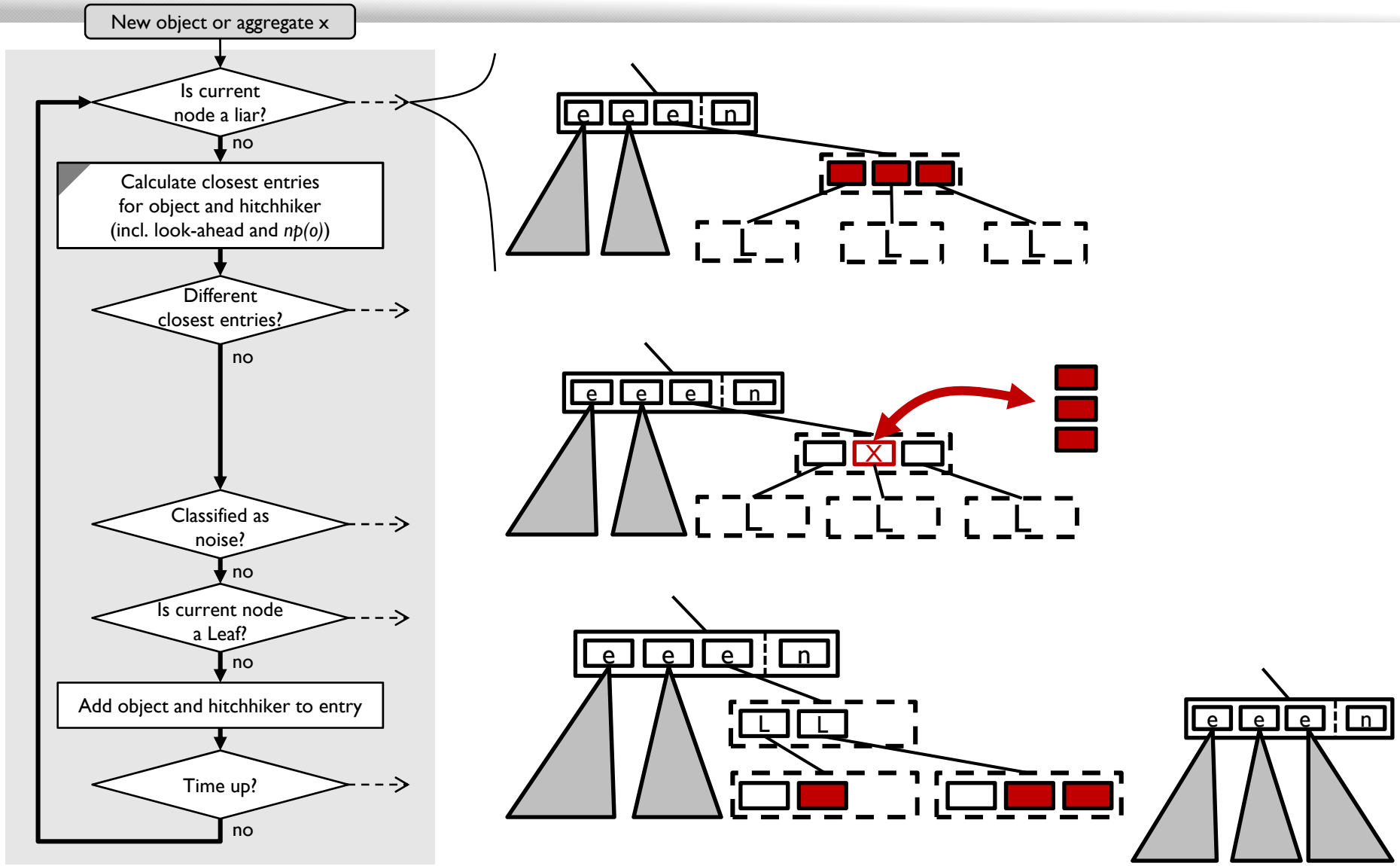


Definition 4. ***Noise-to-cluster event.*** *For a node* $node = (e_1, \ldots, e_k, CF_{nb}^{(t)})$ *with average weight* $n_{avg} = \frac{1}{k}\sum n_{e_i}^{(t)}$ *and average density* $\rho_{avg} = \frac{1}{k}\sum \rho_{e_i}$ *the noise buffer* $CF_{nb}^{(t)}$ *becomes a new entry, if*

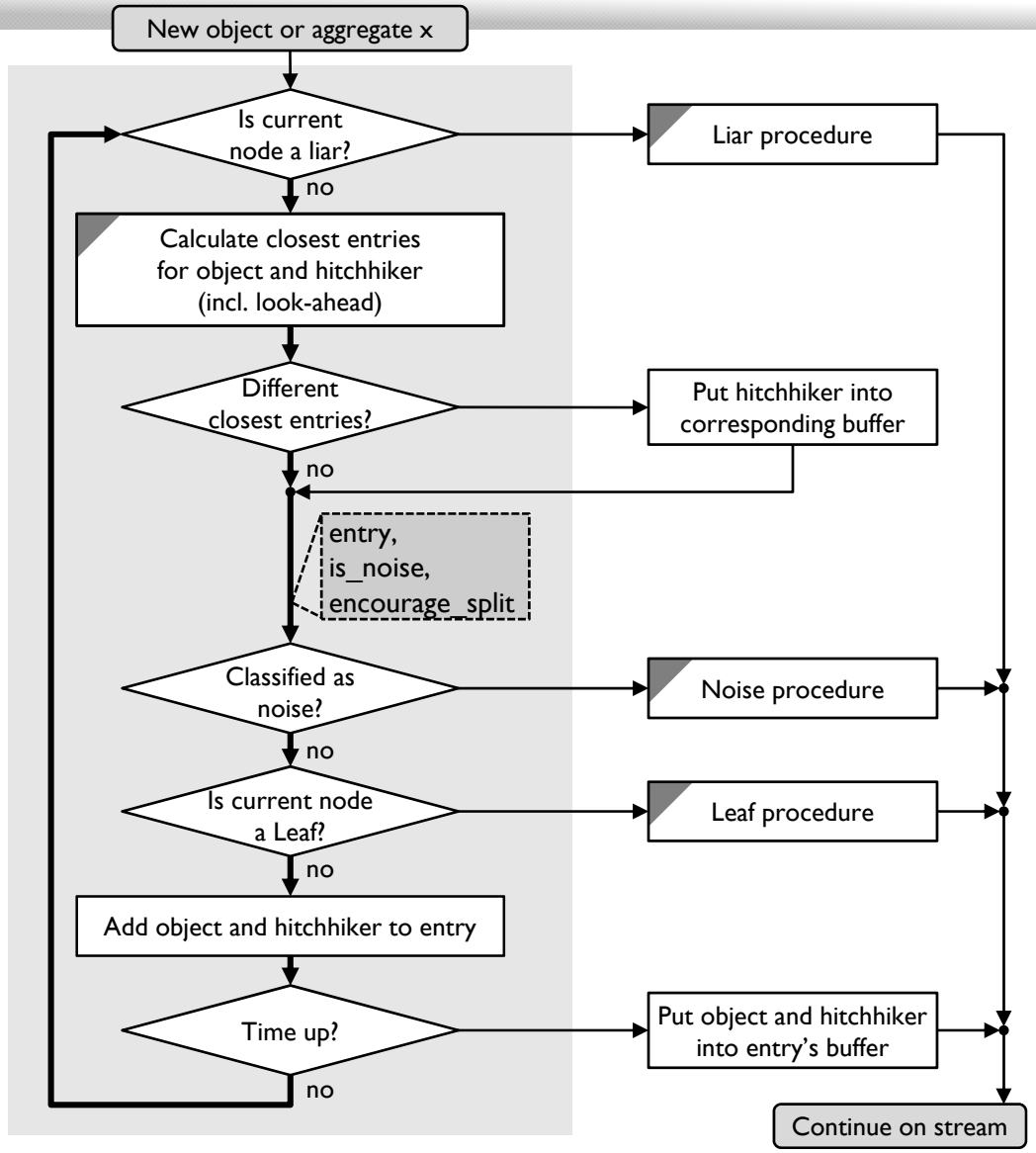
$$gompertz(n_{nb}^{(t)}, n_{avg}) \cdot \rho_{nb} \geq \rho_{avg}$$



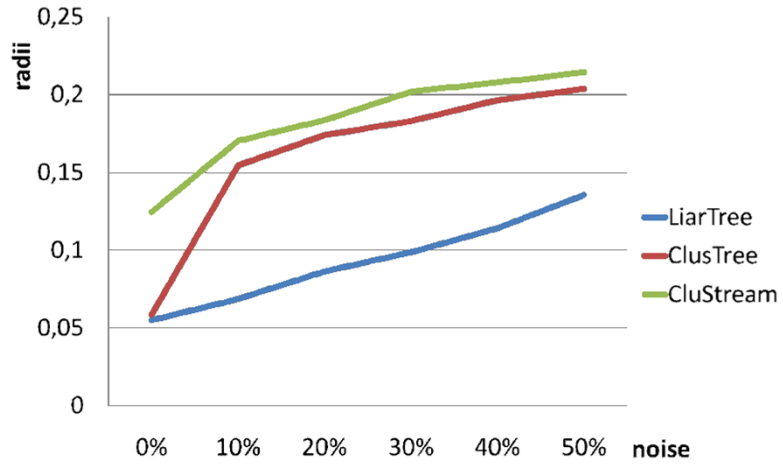
LiarTree algorithm – liar procedure

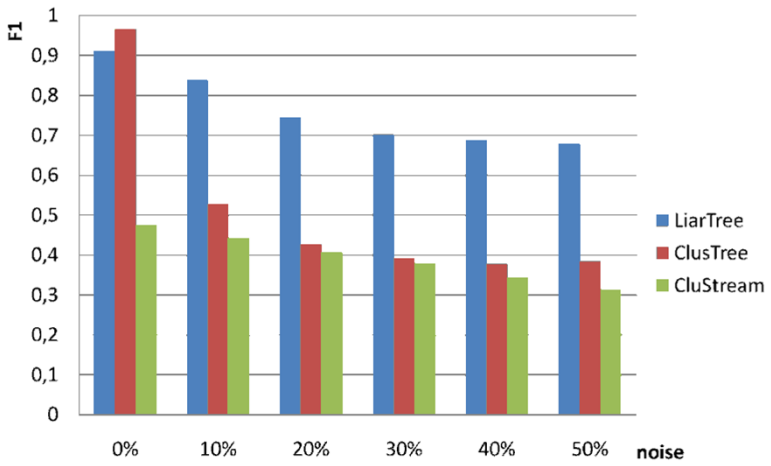


LiarTree algorithm – summary



Evaluation





- Measures
  - Cluster size: Radii of the resulting micro clusters
  - Clustering quality: Precision, Recall, F1 (Monte Carlo approach)
- More experimental results and algorithm detail @SensorKDD 2011
  - UCI Data, Sensor data, varying drift speeds, etc.

Conclusion

- LiarTree
  - Maintain advantages from the ClusTree, especially anytime clustering
  - Reduces overlapping of inner entries through local look ahead
  - Perform explicit noise handling on all levels of the tree
  - Allows for transitions from noise to new concepts

- ClusTree – Self-adaptive Anytime Stream Clustering
  - Hierarchical data structure → self-adaptive and fine grained (offline input)
  - Buffer & hitchhiker concept → enables anytime clustering
  - Aging → enables decay and allows reusing insignificant entries
  - Aggregation → improves quality on exceptionally fast streams
  - Cluster features → compatible with existing work for drift, novelty, shapes, …