Agglomerative 2-3 Hierarchical Clustering: theoretical improvements and tests

Sergiu Chelcea[1], Patrice Bertrand[1,2], Brigitte Trousse[1]

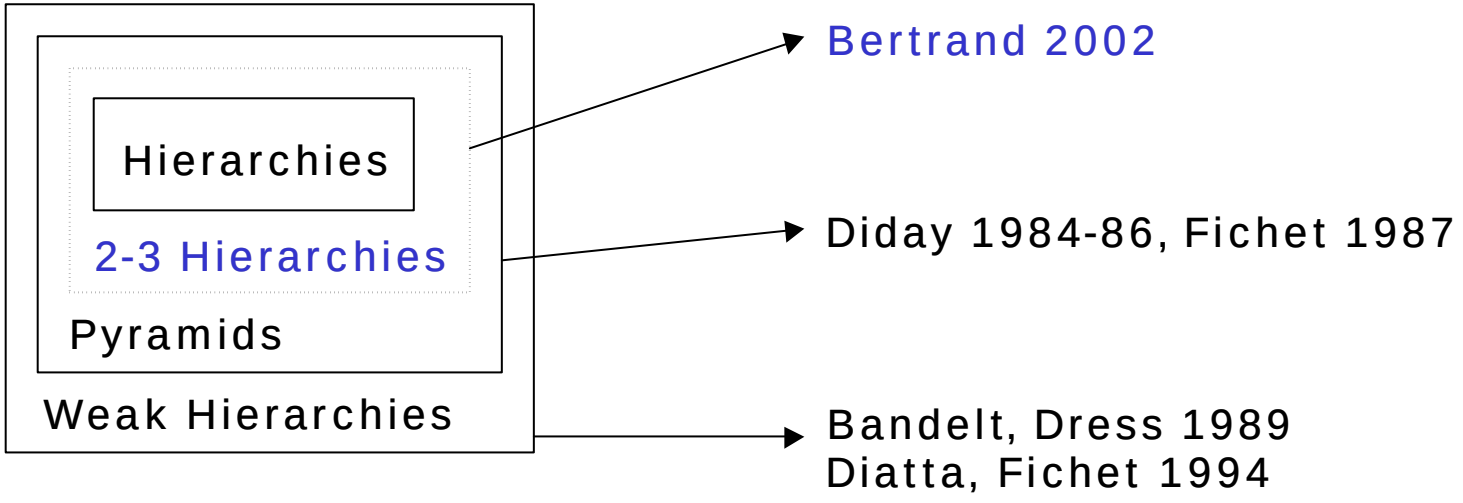
1. Action AxIS, INRIA Sophia-Antipolis, France
2. ENST Bretagne, France

LastName.FirstName@inria.fr

Outline

- The classical case of AHC
- 2-3 Hierarchies
  - Definitions
  - Properties
- Algorithm of 2-3AHC
- Analysis of complexity
- Application on simulated data
  - Experimental Validation of Complexity
  - Ongoing and Future Work

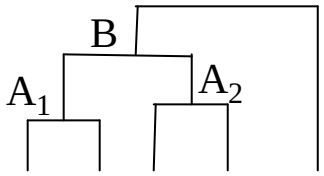
Context


Hierarchies
2-3 Hierarchies
Pyramids
Weak Hierarchies
Bertrand 2002
Diday 1984-86, Fichet 1987
Bandelt, Dress 1989
Diatta, Fichet 1994

Hierarchies (1/3)

We recall some definitions related to the hierarchical case that will be extended to the 2-3 hierarchies:

- Hierarchy:
  - each cluster is nonempty
  - *E* and the singletons are clusters
  - each pair of clusters (A,B) is hierarchical:
    $A \cap B \in \{\emptyset, A, B\}$



Remark: - admits at most n-1 non trivial clusters

Indexed hierarchy:

- each cluster is associated to a positive real number *f* ,
  where $\forall A, B \in S, A \subset B \Rightarrow f(A) < f(B)$

Agglomerative Hierarchical Classification (2/3)

Vocabulary:

- set inclusion order on the set of clusters:
  - predecessor/successor
  - comparable clusters
- candidate clusters (unmarked) = maximal clusters
- data input: dissimilarity $d : E \times E \to [0, \infty)$

$$d(a,b) = d(b,a) > d(a,a) = 0, \ \forall a, b \in E$$

- aggregation index (link between clusters), m :
  - single linkage
  - complete linkage
  - average linkage
- usually f(X È Y) = m(X,Y)

Algorithm AHC (3/3)

1. **Initialisation**: *iter* ← 0; Clusters are the singletons of set E.
   f ← 0;
2. *iter* ← *iter* + 1;

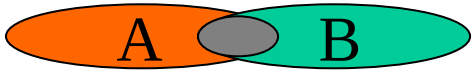
   Merge X and Y which are - in the sense of m - the two nearest clusters; compute f(XÈY)
3. **Reduction**: Eliminate the successors found on the same level *f* with their predecessor, if there are any
4. **Update** m , *predecessor* links, *successor* links

5. **Stopping rule**: Repeat step 2-4, until the set E becomes a cluster

2-3 Hierarchies: Definitions

- **Proper intersection**:

  - A *properly intersects* B, if AÇB Ï{Æ,A,B}



Concept: - in a 2-3 hierarchy, for any three clusters at least two pairs of them are hierarchical

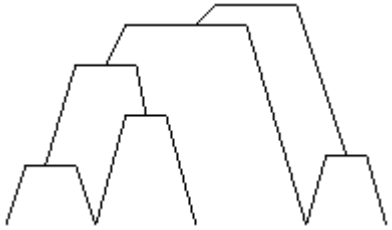
- **2-3 Hierarchy [Bertrand 2002]:**

  - each cluster is nonempty
  - *E* and singletons are clusters
  - the proper intersection of two clusters is also a cluster
  - each cluster properly intersects no more than one other cluster

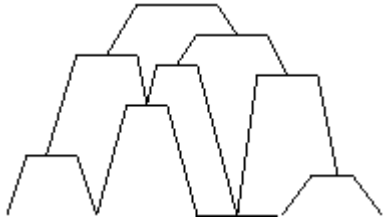
2-3 Hierarchies: Properties

[Bertrand 2002]

- The number of elements of a 2-3 hierarchy that are not reduced to singletons, is at most $\left[\frac{3}{2}(n-1)\right]$
- Each 2-3 hierarchical set system on E is a collection of intervals of some linear order defined on E.



2-3 Hierarchy



Pyramid

Algorithm of 2-3AHC

1. **Initialisation**: *iter* ← 0; Clusters are the singletons of set E.
   f ← 0;
2. *iter* ← *iter* + 1;

   Merge X and Y which are - in the sense of m - the two nearest *non-comparable* clusters, such that at least one of them is maximal; compute f(XÈY)

3. **Merge** XÈY and the other predecessor of X or Y, if it exists. compute f(XÈY)
4. **Reduction**: Eliminate the successors found on the same level *f* with their predecessor, if there are any
5. **Update** m , *predecessor* links, *successor* links

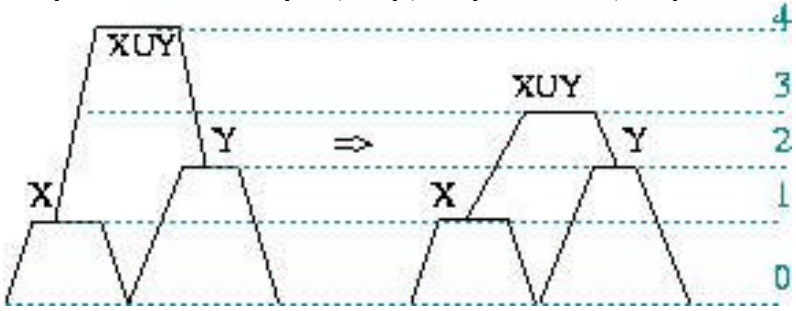
6. **Stopping rule**: Repeat step 2-5, until the set E becomes a cluster

Algorithm of 2-3AHC

- Generalizes the AHC:
  - a cluster can be merged with two different clusters

- Double single linkage [Jullien, Bertrand 2002]:

$$f(X \cup Y) = Min\{m(X,Y), m(X \cup Y, Z) : Z \text{ candidate cluster}\}$$



- Complexity: $O(n^2 \log n)$

Analysis of Complexity (1/3)

We use an ordered dissimilarity matrix on three levels:

- dissimilarity values
- cardinality of the two clusters
- lexicographical order

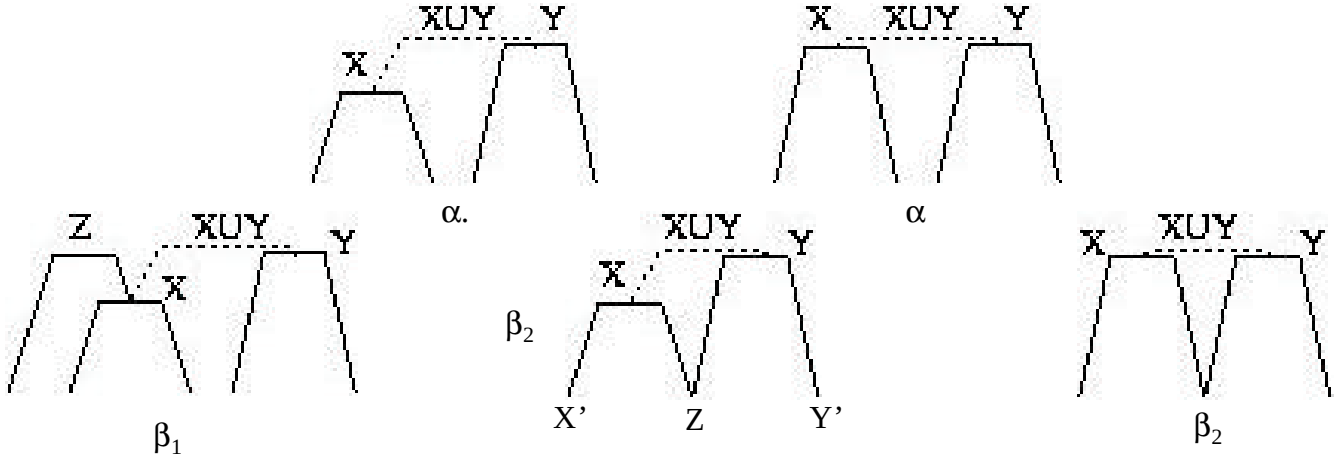
Step 1. Initialisation: Compute and order the dissimilarity matrix, $O(n^2 \log n)$

Step 2. Merge X and Y ... : Retrieve (X,Y) from the data structure, and create XÈY, O(1)

Step 3. Merge XÈY and ... : Intermediate merging with O(n) complexity

Analysis of Complexity (2/3)

Step 4. Reduction: We have five possible cases of reduction when merging a cluster:



- eliminate the successors found on the same level with their predecessor
- complexity O(n)

Analysis of Complexity (3/3)

Step 5. Update m :

- compute new dissimilarities and store them in the matrix, $O(n \log n)$

- eliminate dissimilarities containing non candidates clusters, $O(n \log n)$

Total complexity of the algorithm:

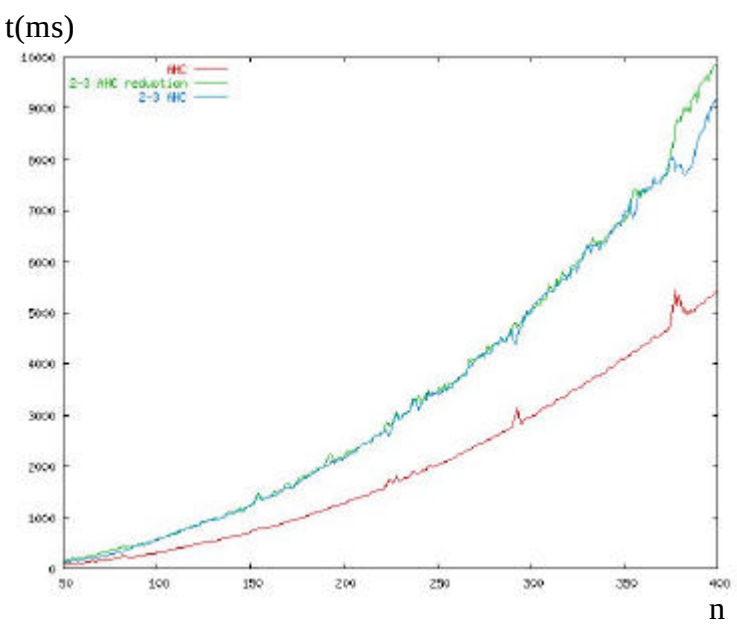
$$O(n^2 \log n) + n \times O(n \log n) \rightarrow O(n^2 \log n)$$

step 1. steps 2. - 5.

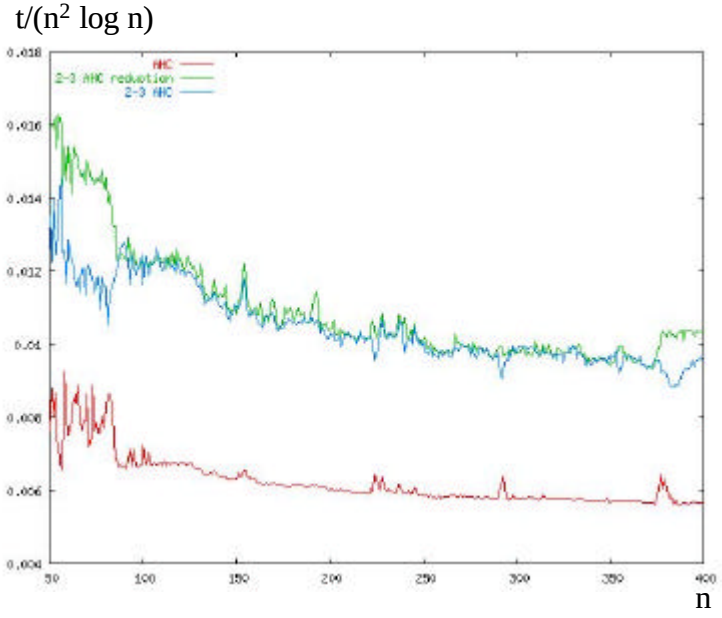
Application

Simulated data:

- complexity $\Rightarrow O(n^2 \log n)$



Execution times



Complexity

Conclusions

Contributions:

- a new formulation of the 2-3 AHC algorithm
- a reduction of complexity
- a first implementation of the 2-3 AHC algorithm (Java) and its integration in CBR*Tools, a Case Based Reasoning framework
- an experimental validation of the complexity on simulated data

Ongoing and Future work:

- study of the quality of the 2-3 AHC compared with AHC and other classification methods
- study of the applicability of 2-3 AHC in the context of Web Usage Mining