Association and Visualization of Clusters Using Multi Algorithm Voting

Roy Gelbard & Ran Bittmann - Bar-Ilan University - gelbardr@mail.biu.ac.il

Cluster Analysis Techniques

- Used in categorization problems, in the training stage, as analyzing method to find out the total number of categories and their profile.
- Daily applications such as user profiling, product, market profiling, for consumption recommendation, trends and hazard detections, medical diagnostics, etc. (mainly for problems with more than 2 categories).
- Several techniques for unsupervised clustering (i.e. classification into unknown number of categories), "white boxes" as well as "black boxes".

But . . .

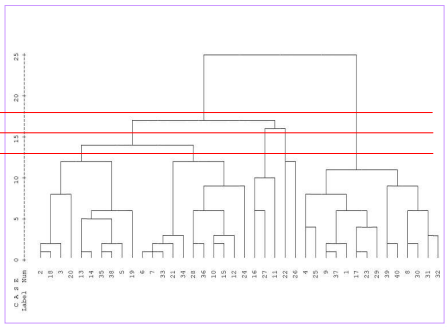
- Each algorithm yields different result.
- Results are hardly robust and affected by small changes in the similarity measure or scoring function.
- Number of categories is still an open issue.

Dendrogram for Visualization

No clear cut for Categories - Limited in Scale

- Entities, Items, Samples
- Features Representation
- Similarity Measures
- Scoring Functions
- Grouping Algorithms
- Evaluation
- Features Selection & Decision Rule

- Factor Analysis
- Discriminate Analysis
- Unsupervised Clustering



Similarity Measures

How similar are these strings ?

11000

10000

Different approaches:

4/5 = 80% - 4 common positions out of 5 existing positions

2/3 = 66.6% - 2 common positive bits ("1") out of 3 positive bits

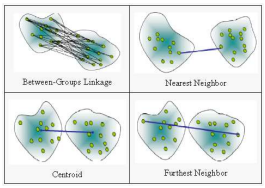
1/2 = 50% - 1 common positive position out of 2 positive positions

1/5 = 20% - 1 common positive position out of 5 existing positions

Require Agreed Similarity Philosophy

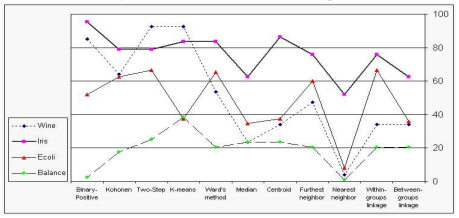
Scoring Functions

Min Within Group Distance, *Max* Between Group Distance

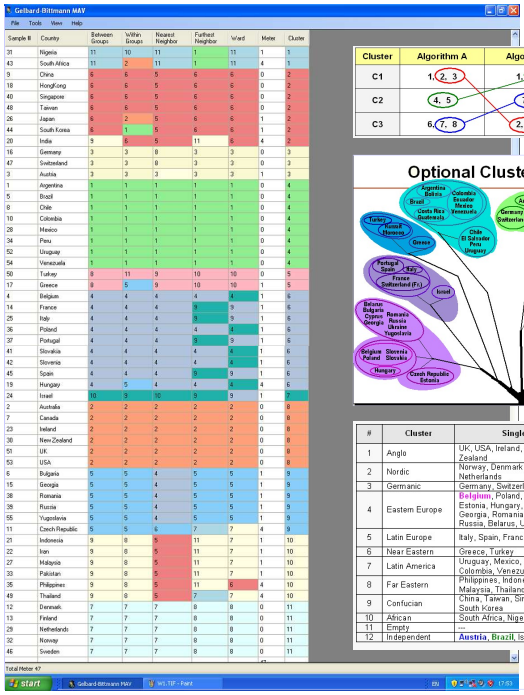


Dependencies

on Representation and Algorithm



Matching Rate of 44 Experiments
(10 Algorithms, 4 Datasets, 2 Representation forms)



Cluster	Algorithm A	Algorithm B
C1	1, 2, 3	1, 4, 5
C2	4, 5	7, 8
C3	6, 7, 8	2, 3, 6

#	Cluster	Single Algorithm	Multi Algorithm Voting
1	Anglo	UK, USA, Ireland, Canada, Australia, New Zealand	---
2	Nordic	Norway, Denmark, Finland, Sweden, Netherlands	---
3	Germanic	Germany, Switzerland	(+) Austria
4	Eastern Europe	Belgium, Poland, Slovakia, Slovenia, Estonia, Hungary, Czech Republic, Georgia, Romania, Bulgaria, Yugoslavia, Russia, Belarus, Ukraine, Cyprus	(-) Belgium, (-) Poland, (-) Slovakia, (-) Slovenia
5	Latin Europe	Italy, Spain, France, Portugal	(+) Belgium, (+) Poland, (+) Slovakia, (+) Slovenia
6	Near Eastern	Greece, Turkey	---
7	Latin America	Uruguay, Mexico, Chile, Argentina, Colombia, Venezuela, Peru	(+) Brazil
8	Far Eastern	Philippines, Indonesia, Pakistan, Malaysia, Thailand, Iran, India	(-) India
9	Confucian	China, Taiwan, Singapore, Honk Kong, South Korea	(+) Japan, (+) India
10	African	South Africa, Nigeria	---
11	Empty	---	---
12	Independent	Austria, Brazil, Israel, Japan	(-) Austria, (-) Brazil, (-) Japan

Method for Association and Visualization of Multiple Distributions

Multi-Algorithm-Voting for Better Decision Making

Classification and clustering decisions are frequently arise in business applications such as recommendation concerning products, markets, human resources, etc. Currently, decision makers must analyze diverse algorithms and parameters on an individual basis in order to establish preferences on the decision-issues they face; cause there is no supportive model or tool which enables comparing different result-clusters generated by these algorithms and parameters combinations.

A Multi-Algorithm-Voting (MAV) method was developed to analyze and visualize results of multi algorithms, were each one of them is pointing to any decision. The visualization uses a Tetris like format in which all distributions (decisions) are ordered in a Matrix, where each distribution suggested by a specific algorithm and/or parameters is presented in a column of the said Matrix, and each data component (case) is presented in a row of the Matrix. "Local decisions" (of each specific algorithm, concerning each case) are presented as "Tags" in the cells of the said Matrix.

The MAV method associates the "arbitrary Tags" to each other. Each association is presented in a visual form, for example using color codes. The colors are consistent over the said Matrix and similar colors, even on different rows, represent similar classification (decision).

The MAV method calculates the quality of each association for each row, representing a data component. The quality can be calculated, but is not limited to, as the Homogeneity (or Heterogeneity) of the association of a single data component over all the algorithms used in the analysis. Then it pinpoints the best association based on the quality meter used.

The MAV method enables not only visualization of results produced by diverse algorithms, but also as quantitative analysis of the results.

"Tetris-like" Visualization

Well-classified Case

A Hard-to-Classify Case

Effective Algorithm for a Certain Cluster

Wrongly Classified Cases

Heterogeneity Meter

- **Sample Heterogeneity Meters**

SVE (Squared Vote Error)

1 2 3 2 2 = 4

$$H = \sum_{i=1}^{n} (N - M_i)^2$$

Where:

H – is the Heterogeneity Meter

N – is the number of methods voting for the sample

M – is the maximum number of similar votes according to a specific association received for a single sample

i – is the sample number

n – is the total number of samples in the dataset

Local Search Optimization

Can be treated as a Local Search (Hill Climbing) problem:

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
  inputs: problem, a problem
  local variables: current, a node
                   neighbor, a node
  current ← MAKE-NODE(INITIAL-STATE[problem])
  loop do
    neighbor ← a highest-valued successor of current
    if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
    current ← neighbor
```

- Initial-State - Random permutation
- Neighbor - State after single permutation change
- Success - Better Homogeneity Meter
- Termination State - No success in neighborhood

Complexity

Multiple tries with random permutation.

Selection of the one with the best stop criteria.

Complexity: $O\,(S \bullet H \bullet C \bullet log(C!^{(M-1)}))$

Where:

- S - Number of samples
- H - Homogeneity Calculation (C•M)
- C - Number of clusters
  Permutation of cluster # association (C!) ordered by methods (algorithms)
- M - Number of Clustering Methods (always start with the same method (m-1))
- C - for each cluster
- log - Only succeed nodes are extracted as potential neighborhood of the next step; Therefore estimated as in each round it reaches the half containing the right solution. Therefore the log