Sequential Clustering Algorithm

Quanyu Zhu

April 16, 1999

Introduction:

Sequential Clustering Algorithm is the simplest type of clustering algorithm, uses straightforward greedy approach.

. Basic approach

- Each run of the algorithm produces a single clustering
- Frequently, output depends on the order the f.v's are pre ented to the algorithm and values of the parameters.

. Scheme disccussed below:

- Basic Sequential Algorithm Scheme(BSAS)
- Modified Basci Sequential Algorithm(MBSAS)
- Two Threshhold Scheme(TTSAS)

Sequential Algorithm Scheme description

1 BSAS

1.1 Intuition

- Required Parameter : Θ
- Start with $x_1$in first cluster C_1
- For each remaining f.v. x_i. find clusterC_k minimizing d(x_i, C_k) (maximizing s(x_i,C_k)) using desired DM/SM
- If d(x_i,C_k)$\leq\Theta$, addx_i to C_k; else create new cluster for x_i
- Optional parameter q limits total number of clusters to conserve computational resources

1.2 Pseudocode

- Input : Θ, q (q is optional)
- Initialize m = 1, C_m=$\{x_i\}$
- For i = 2 to N

  - C_k= $\underset{1 \leq j \leq m}{argmin}\{d(x_i, C_j)\}$
  - If $d(x_i,\, C_k) > \Theta$ and $m < q$ then m=m+1 and $C_m = \{x_i\}$
  - Else $C_k = C_k \bigcup \{x_i\}$ and update representatives if necessary

- If SM used, replace argmin with argmax and> with <.

1.3 Problem with BSAS

- Sensitive to value of Θ and order of f.v.
- BSAS assign f.v.'s to clusters before all clusters have been formed, so might the assignment to another cluster be better
- Improvement possible to assign most f.v. to clusters until all clusters have been formed.

2 Modified BSAS

2.1 Deal with the problem of BSAS

2.2 Basic Idea

- Tow phrase process

  1. Determine clusters by scanning list and create new clusters when necessary, but don't assign any f.v. to existing clusters
  2. Once clusters determinied (each represented by one f.v.), assign each remaining (unassigned) f.v. to its best cluster.

2.3 Pseudocode

- Input : Θ, q (q is optional)

  Phase 1.

- Initialize m = 1, $C_k = \{x_i\}$
- For i = 2 to N

  - C_k= $\underset{1 \leq j \leq m}{argmin}\{d(x_i, C_j)\}$
  - If $d(x_i,\, C_k)$> Θ and $m < q$ then m=m+1 and $C_m = \{x_i\}$

Phase 2.

- For each x_i not assigned to a cluster

  - $C_k = \underset{1 \leq j \leq m}{argmin} \{d(x_i, C_j)\}$
  - $C_k = C_k \bigcup \{x_i\}$ and update representatives if necessary

- If SM used, replace argmin with argmax and> with <.

2.4 Problem remains

- Still sensitive to value of Θ and order of f.v.

3 Two - Threshhold Scheme(TTSAS)

3.1 Introduction

- To reduce sensitive to Θ and order of f.v.'s, use two threshhold Θ_1 and Θ_2
- Find best cluster C_k for x_i, but assign to C_k only if $d(x_i, C_k) < \Theta_1$, and create new cluster only if $d(x_i,\, C_k) > \Theta_2$, otherwise defer decision on x_i

3.2 Pseudocode

- Input : Θ_1, Θ_2
- Initialize m = 1, $C_k = \{x_i\}$
- while there exist unassign f.v.'s

  - For each x_i not assigned to a cluster

    * $C_k = \underset{1 \leq j \leq m}{argmin} \{d(x_i, C_j)\}$
    * If $d(x_i,\, C_k)$ $<\Theta_1$ then C_k $=C_k \bigcup \{x_i\}$and update representatives if necessary
    * Else $d(x_i, C_k) > \Theta$ then m=m+1 and $C_m = \{x_i\}$

  - If no f.v. was assigned to a cluster in the previous For loop, then choose an arbitary unassigned f.v. x_j, m = m + 1, and C_m=$\{x_i\}$(try to avoid infinite loop)

- Variation : First find cluster using Θ_2, then run above code

3.3 Problem remains

- Must careful to avoid infinite loop, where some x_i alwyas has d$(x_i, C_k) \in$ [Θ_1,Θ_2]

4 Example

$x_1 = [2,5]^T$ $x_2 = [6,4]^T$ $x_3 = [5,3]^T$ $x_4 = [2,5]^T$
$x_5 = [1,4]^T$ $x_6 = [5,2]^T$ $x_7 = [3,3]^T$ $x_8 = [2,3]^T$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	$\sqrt{17}$	$\sqrt{13}$	3	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	2
x_2	$\sqrt{17}$	0	$\sqrt{2}$	$\sqrt{20}$	5	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$
x_3	$\sqrt{13}$	$\sqrt{2}$	0	$\sqrt{10}$	$\sqrt{17}$	1	2	3
x_4	3	$\sqrt{20}$	$\sqrt{10}$	0	$\sqrt{5}$	3	$\sqrt{2}$	1
x_5	$\sqrt{18}$	5	$\sqrt{17}$	$\sqrt{5}$	0	$\sqrt{20}$	$\sqrt{5}$	$\sqrt{2}$
x_6	$\sqrt{5}$	$\sqrt{5}$	1	3	$\sqrt{20}$	0	$\sqrt{5}$	$\sqrt{10}$
x_7	$\sqrt{5}$	$\sqrt{10}$	2	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	0	1
x_8	2	$\sqrt{17}$	3	1	$\sqrt{2}$	$\sqrt{10}$	1	0

MBSAS, $\Theta = 2.5$
TTSAS, Θ_1=2.2, Θ_2=4

5 Clusters Refining

5.1 Mergering Clusters

- If, after running clustering algorithm, two clusters are very close, can merge them togather
- Input: Parameter M and clusters $C_1, \cdots, C_m$
- Find closest pair of clusters:

  $(C_k, C_r) = \frac{argmin}{1 \leq k,r \leq m, k \neq r}\{d(C_k, C_r)\}$
- If d(C_i,C_j) $\leq$ M then merge C_j into C_i, update representatives if necessary, rename clusters, and repeat
- Else stop

5.2 Feature Vector Reassignment

- Even MBSAS and TTSAS can end up assigning f.v.'s to clusters that are (utimately) farther away than other clusters
- Possible because as clusters grow, representatives change, so distances to clusters change
- Hence might allow reassignment of f.v.'s after algorithm and merging completed

- Let b(i) denote cluster closest to x_i
- For each f.v. x_i

  - $C_k = \frac{argmin}{1 \leq j \leq m} \{d(x_i, C_j)\}$
  - b(i) = k

- For k=1 to m, $C_k = \{x_i \in X : b(i) = k\}$, and update representatives if necessary
- Can repeat process if desired, since again rlrepresentatives will change