Top-Down Clustering Method Based On TV-Tree

Zbigniew W. Ras

Heuristic TV-Tree Construction

Heuristic method

- A random attribute "a" is selected from system S=(X,A,V) to decompose S into systems $S_1=(X_1,A_1,V_1)$ and $S_2=(X_2,A_2,V_2)$ based on certain value from Dom(a) called a split point for S.
- $v_a \in$ Dom(a) is an optimal split point for S if it maximizes the total number of active dimensions in S_1 and S_2 [$\{S_1, S_2\}$ is a balanced set].
- Active Dimension: If the distance between any two objects in S with respect to attribute a is less than λ_a, the attribute a is considered an active dimension for S.
  - Definition: S_1 is λ_a - active if $|\max_a - \min_a| \leq \lambda_a$

    where $\max_a = \max\{a(x): x \in X\}$,

    $\min_a = \min\{a(x): x \in X\}$,

    λ_a is a user given threshold for attribute "a" to be active.

Heuristic TV-Tree Construction

Heuristic method

- Middle Value is calculated for each active dimension.
- S_1, S_2 equally weighted: Constructed TV-Tree has to satisfy so called weighted threshold γ defined as [(card(X_1) / card(X)) $\geq \gamma$ and (card(X_2) / card(X)) $\geq \gamma$]. Then, TV-Tree is called γ-balanced.
- Repeat the same process for S_1 and S_2 by picking another random attribute from S_1 and next from S_2.
- This recursive process will end once all dimensions in the smallest decomposed systems are active.
- Optimal structure of TV-Tree depends on the random attribute selected.
  - If the random attribute selected is strongly related to other non-active attributes, we may get more dimensions close to the top of tree.

Heuristic TV-Tree Construction

Figure 1

	a	b	***c***	d	e
X1	3	1	***5***	6	5
X2	3	4	***0***	0	5
X3	2	6	***8***	8	5
X4	3	1	***5***	6	5
X5	2	9	***6***	3	5
X6	3	4	***7***	1	5
X7	2	9	***6***	4	5
X8	3	1	***7***	7	5
X9	2	6	***8***	7	5
X10	3	4	***7***	1	5

S

	A	b	c	***d***	**E**
X1	3	1	5	***6***	5
X2	3	4	0	***0***	5
X4	3	1	5	***6***	5
X5	2	9	6	***3***	5
X7	2	9	6	***4***	5

S_1

	A	***b***	**C**	d	**E**
X3	2	***6***	8	8	5
X6	3	***4***	7	1	5
X8	3	***1***	7	7	5
X9	2	***6***	8	7	5
X10	3	***4***	7	1	5

S_2

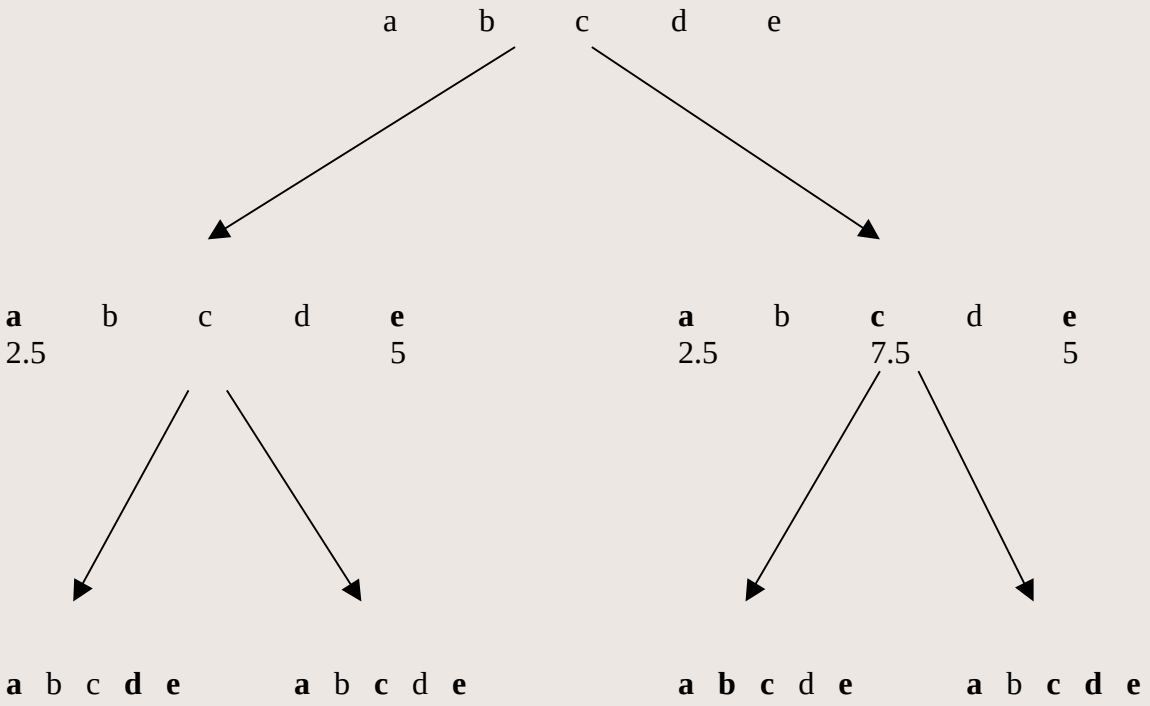
In Search For Optimal Split Point

Figure 2

Value	0		5		6		7		8
		↓		↓		↓		↓	
Split points		2.5		5.5		6.5		7.5	

Layout of TV-Tree

Figure 3

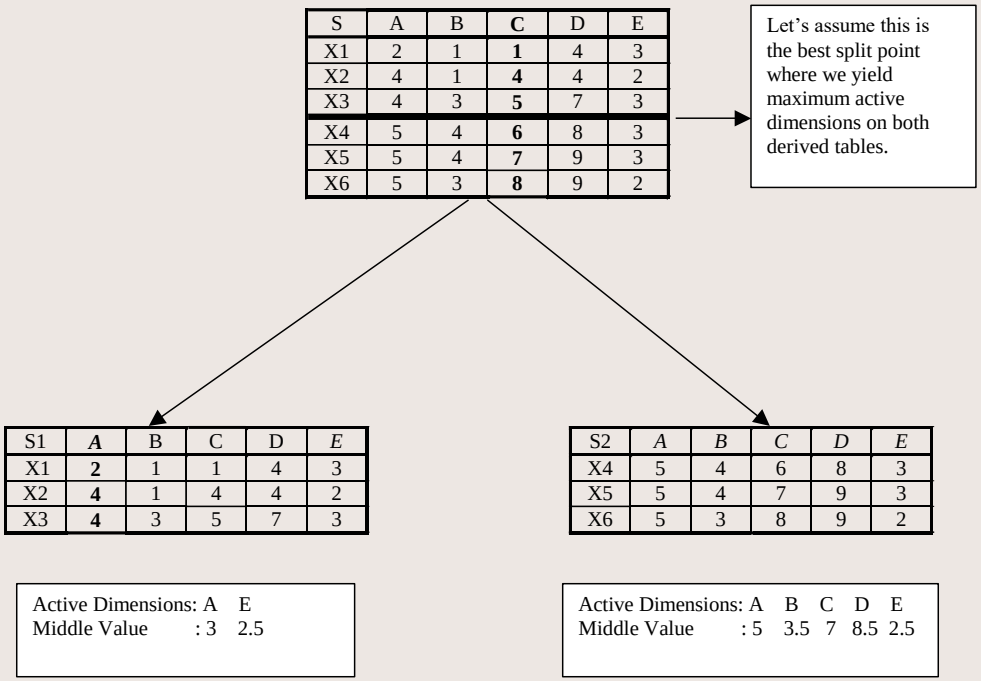


TV-Tree Construction

Figure 4

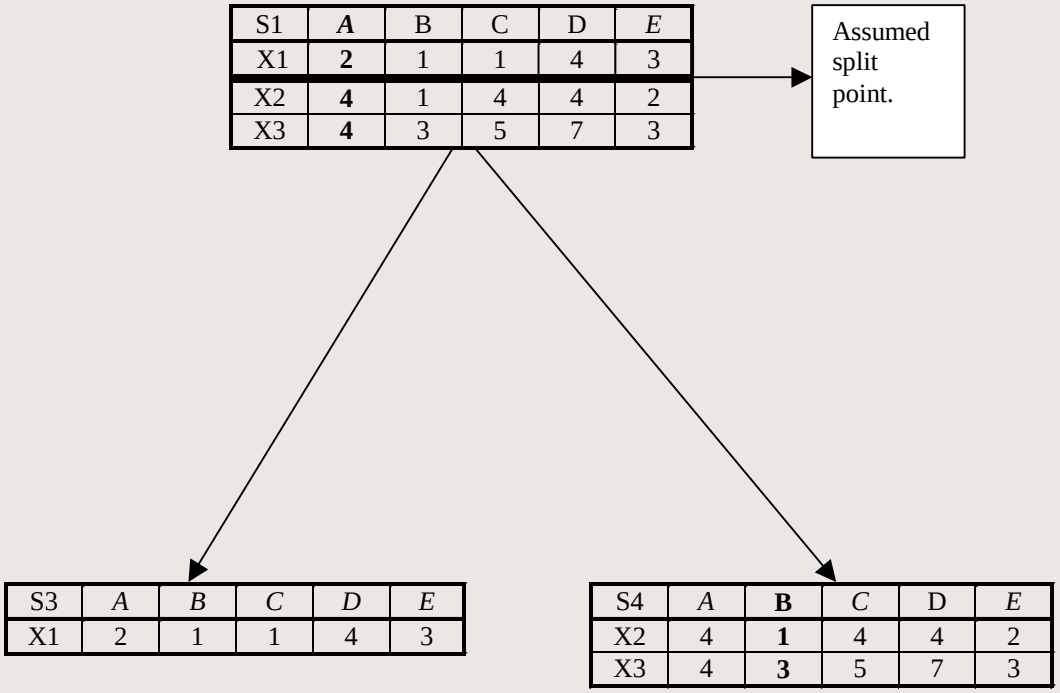
Heuristic Algorithm

Random Attribute: C



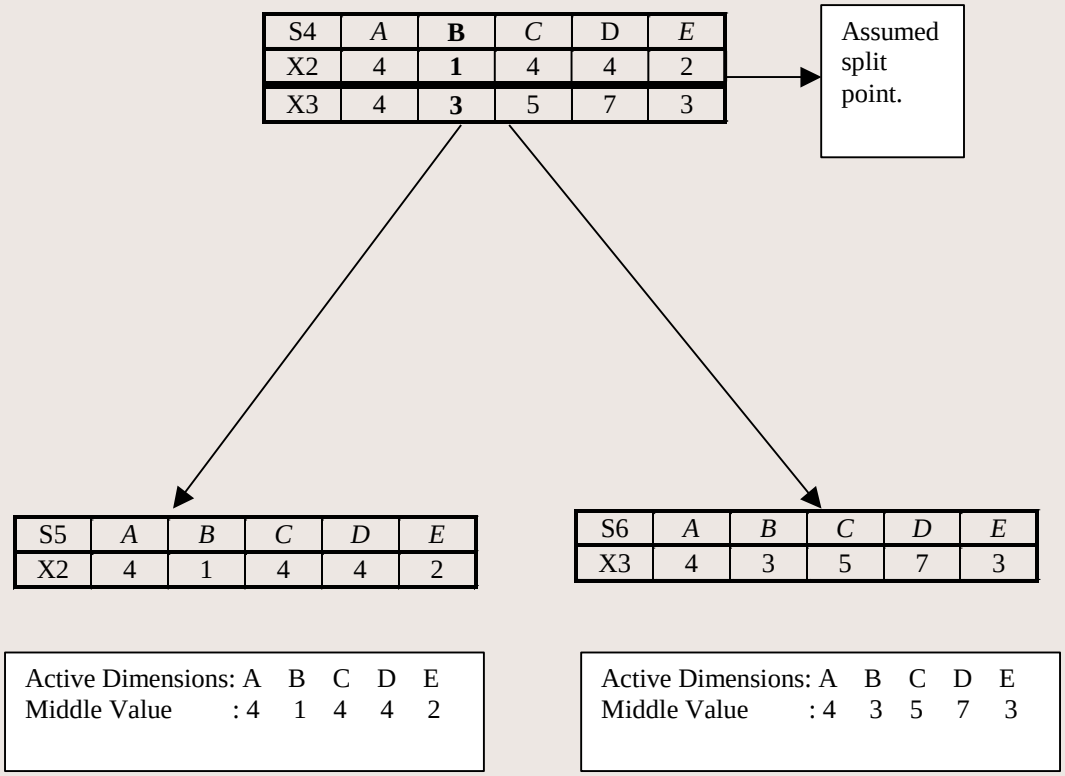
TV-Tree Construction

Random Attribute: A



TV-Tree Construction

Random Attribute: B



Non-Heuristic TV-Tree Construction

Non-Heuristic method

- Search through all attributes in A to look for the best split point for S to decompose it into S_1 and S_2 in a way to maximize the overall number of active dimensions in S_1, S_2 additionally assuming that $\{S_1, S_2\}$ is a well-balanced set.
- The remaining steps are similar to heuristic method.
- This method is more expensive than heuristic method but may produce more optimal TV-trees and therefore more efficient QAS.
- It works well if information system S does not have a large number of attributes.

TV-Tree Based Query Answering

Exact match query

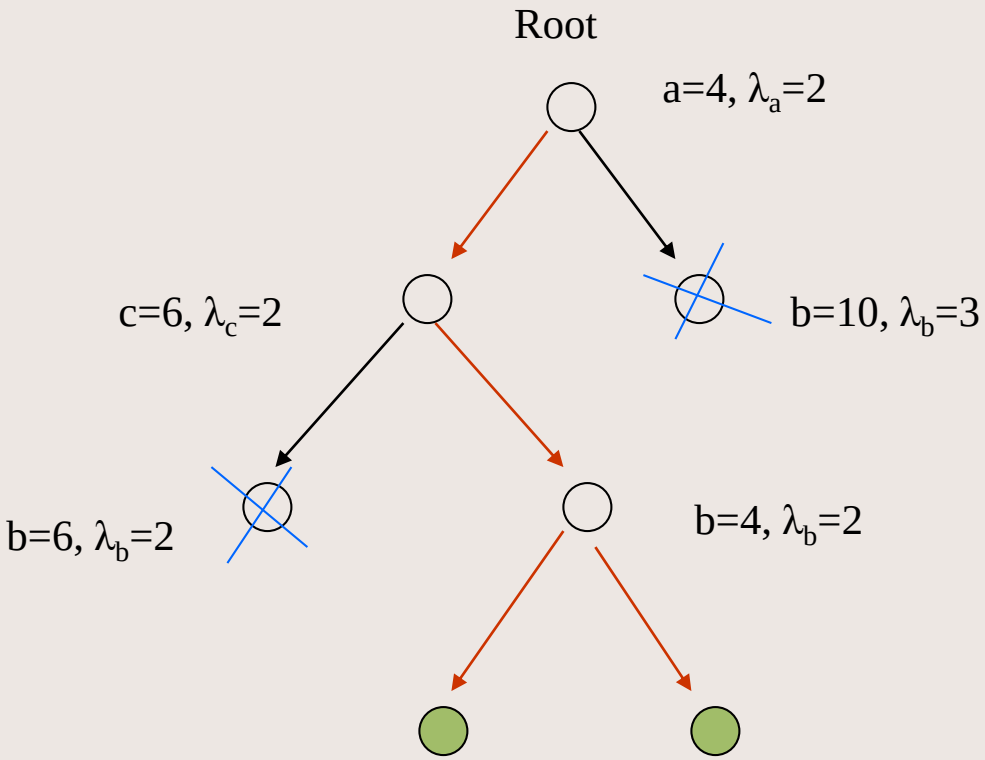
- User specifies attributes and their values.
  - Example query: q=[a=5].

Range query

- User specifies attributes, their values, and tolerance α_a for each specified attribute a (if α is a tolerance value for attribute a, then $\alpha \leq \lambda_a$).
  - Example query: q=[a=5,2].

TV-Tree Based Query Answering

- Q=[a=5]•[b=3] •[c=7,1]



TV-Tree Based Query Answering

TV-Tree can be built further from a leaf node by reducing system thresholds at the leaf node if most of queries reaching that leaf node require scanning large amount of data.