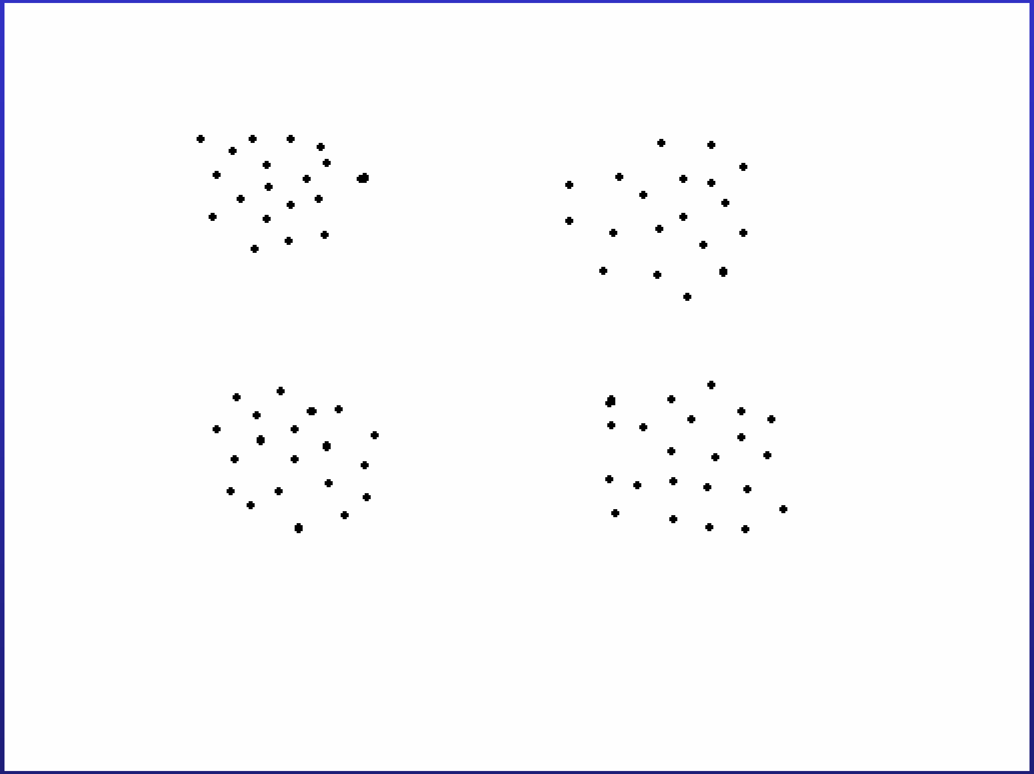
Fuzzy Decision tree and Tree-Structured Vector Quantizer

楊雄彬 (Shiueng-Bien Yang)

Contents

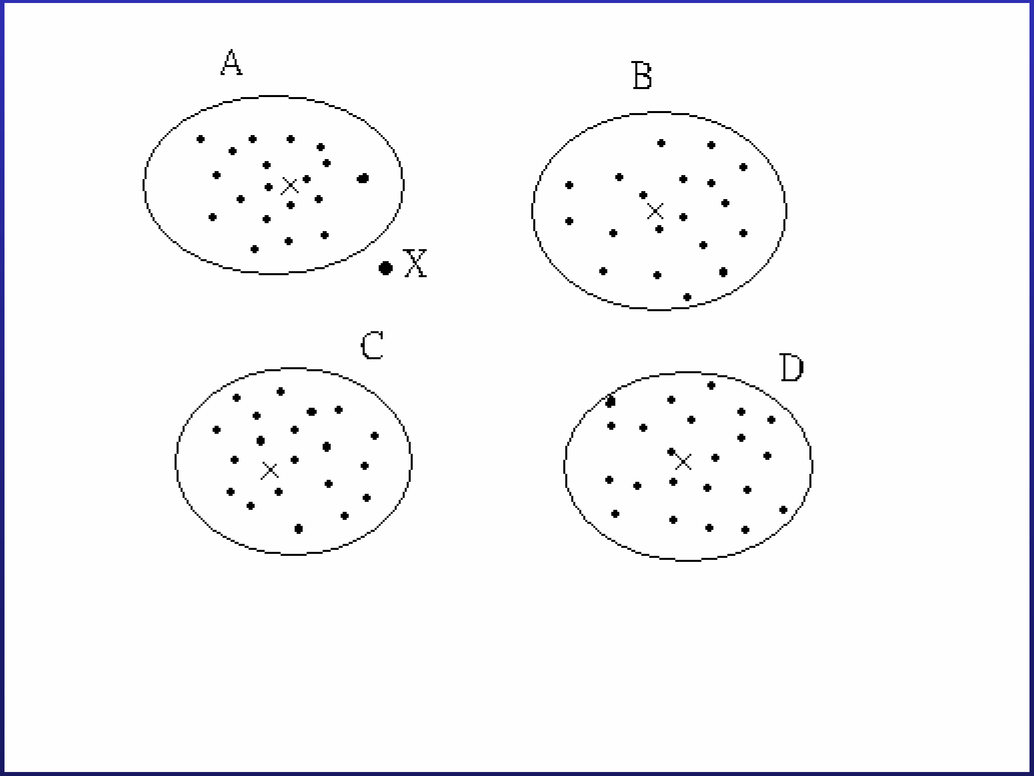
- Basic of Tree Classifier
- Design of Fuzzy Decision Tree
- Design of Tree-Structured Vector Quantizer

Data Set

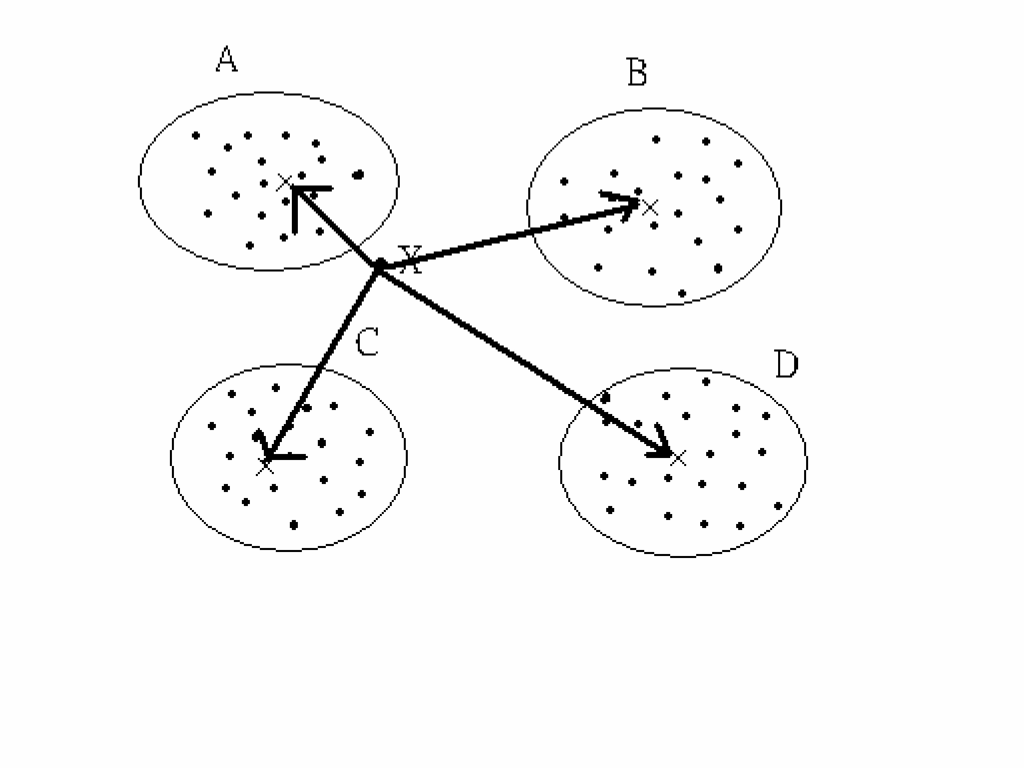


Full Search Method

- Deign Phase:
  - The data set is classified into four clusters by the K-means algorithm.

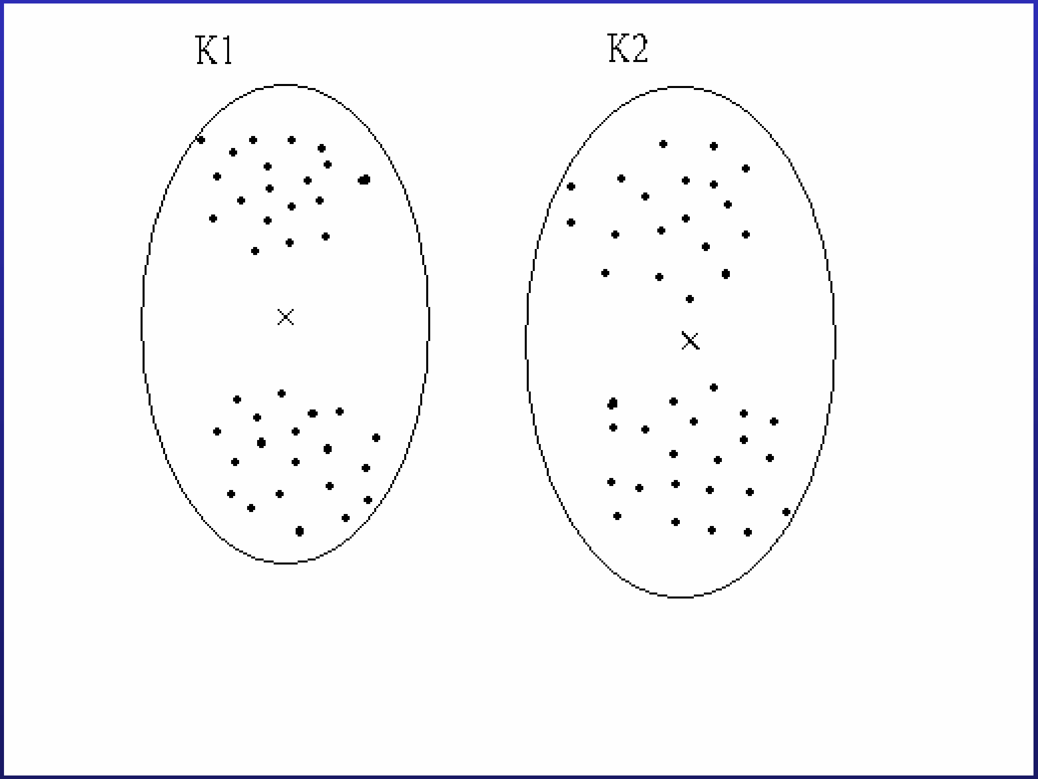


- Test Phase:
  - X is compared with the centers of these four clusters.

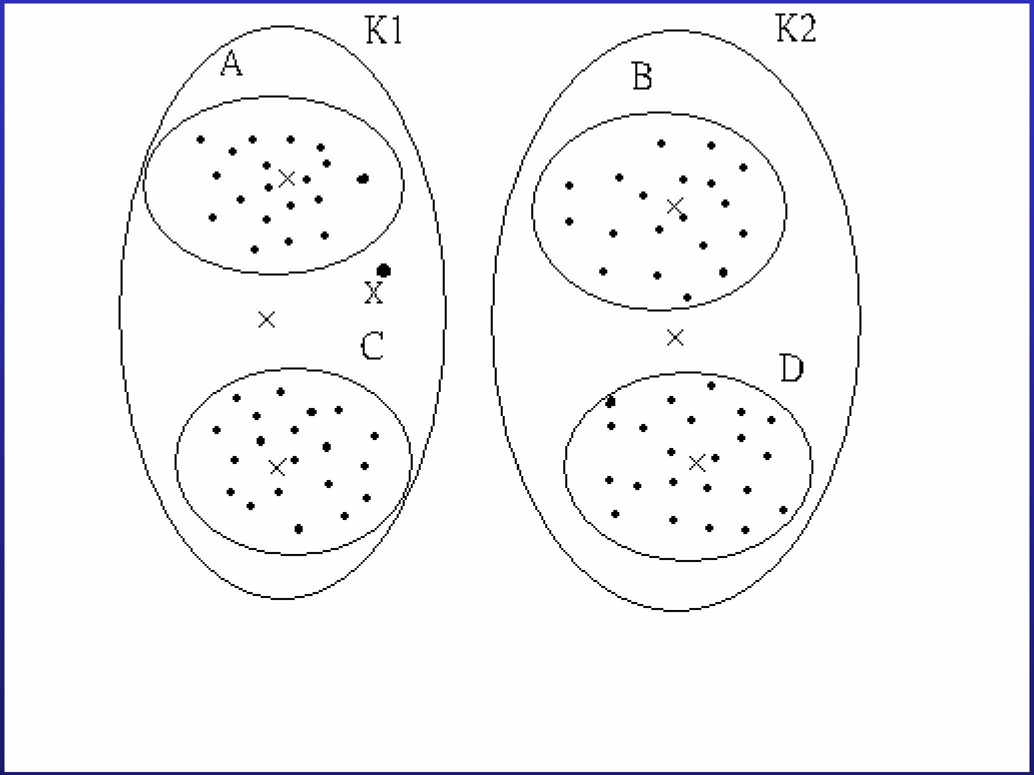


Tree Classifier

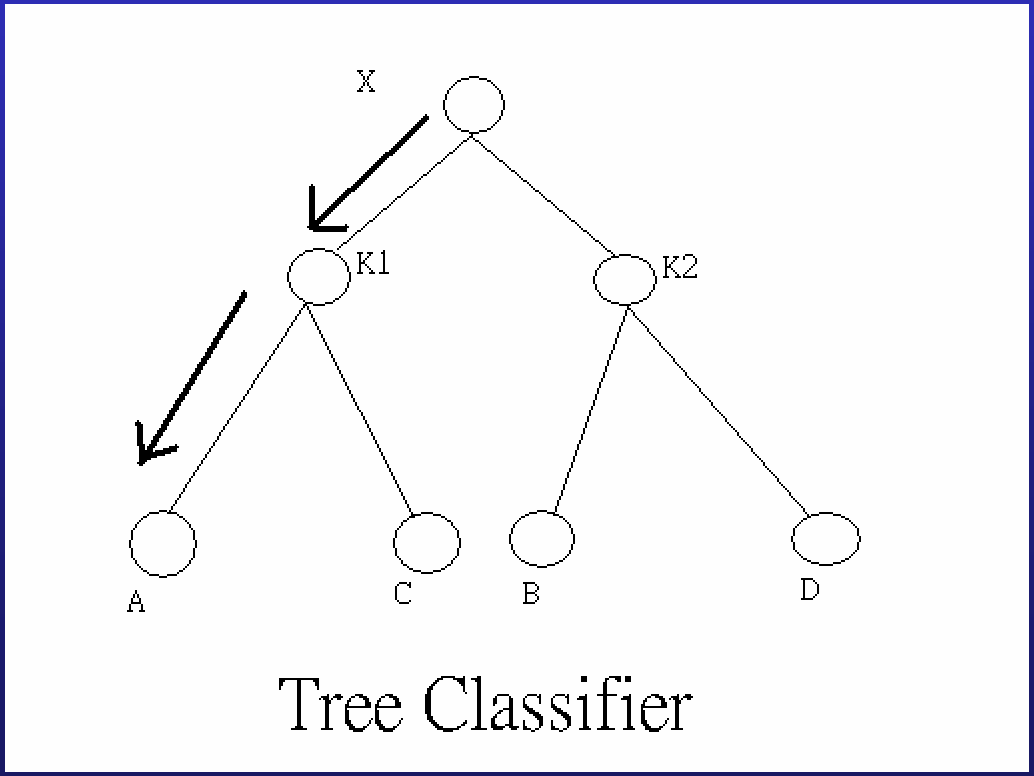
- Deign Phase:
  - The data set is classified into two clusters by the K-means algorithm.



- Each cluster is further classified into two clusters by the K-means algorithm, respectively.



- Test Phase:
  - X is compared with the centers of K1 and K2. X is closer to K1 than K2.
  - X is compared with the centers of A and C.



Comparison of Tree Classifier and the Full Search Method

- Testing time:

  n clusters (classes)

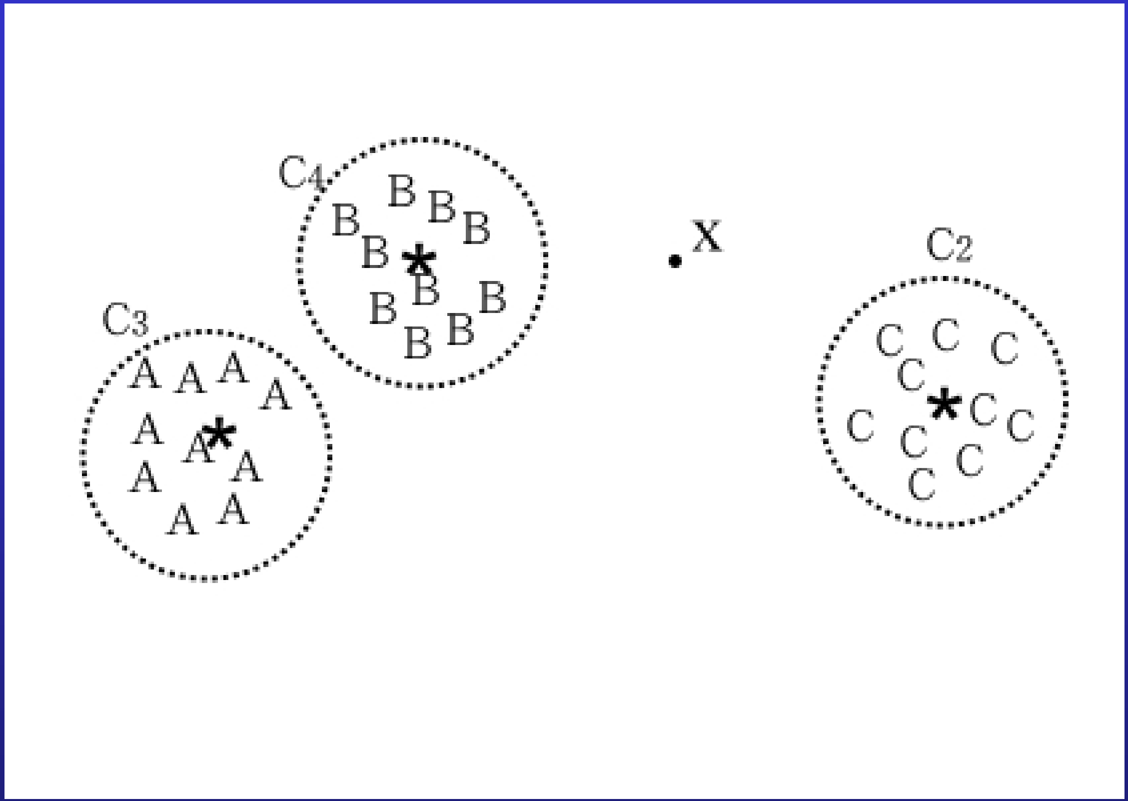
  Tree classifier takes O(*logn*). ☺

  Full search method takes O(*n*).

Comparison of Tree Classifier and the Full Search Method

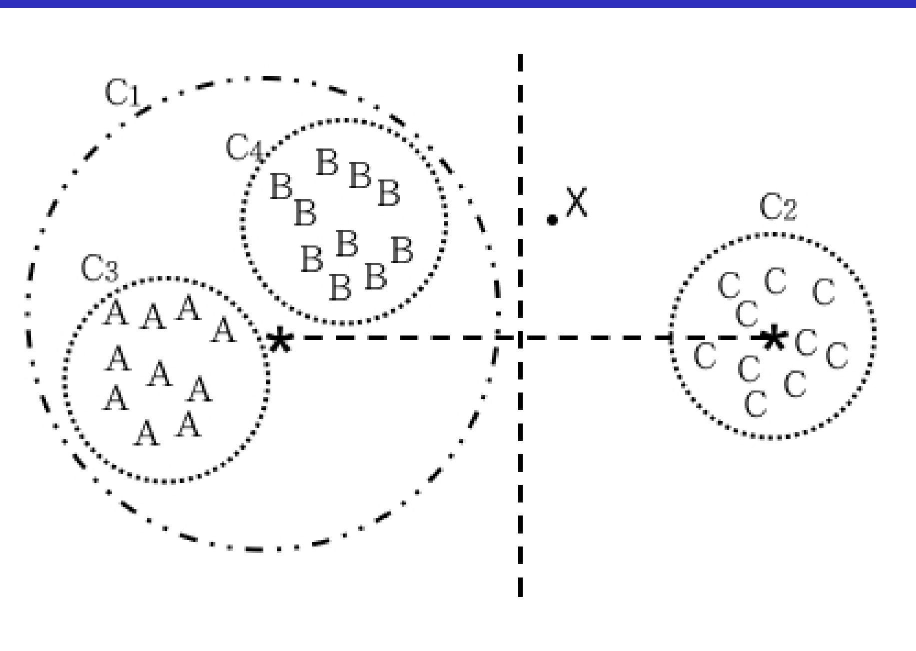
- Recognition rate:
  - ☺ Full search method > Tree classifier

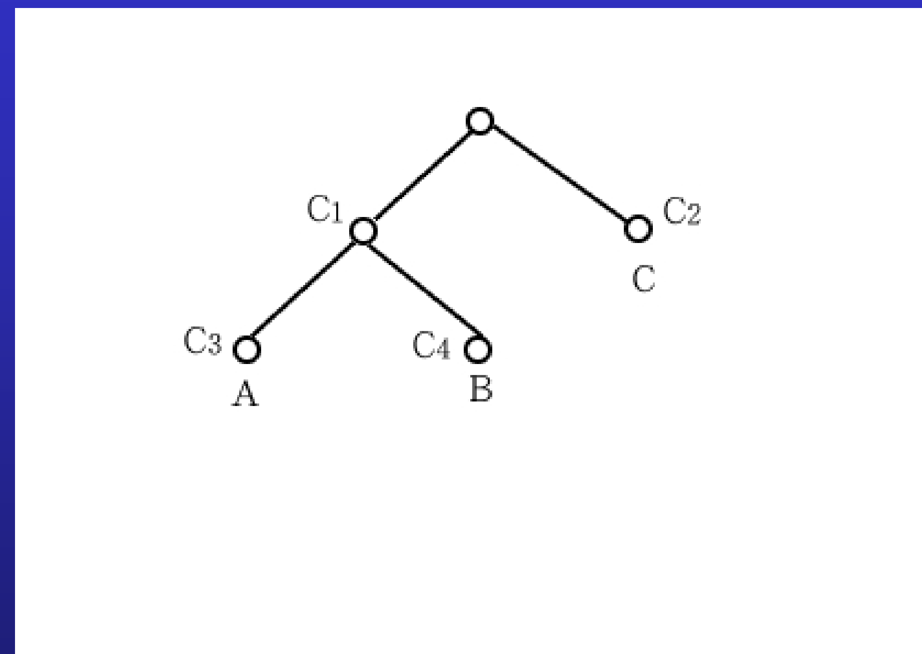
- Example: Full search method



- X is classified to C_4.

- Example: Tree classifier



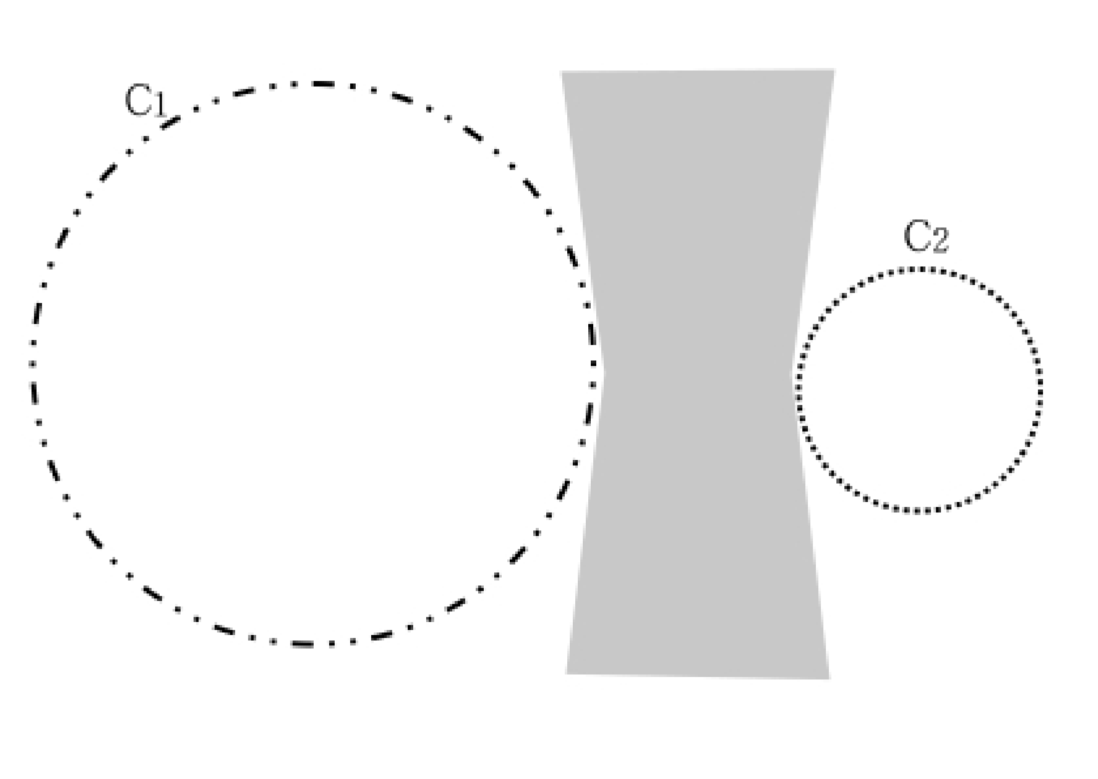


- X is classified to C_2.

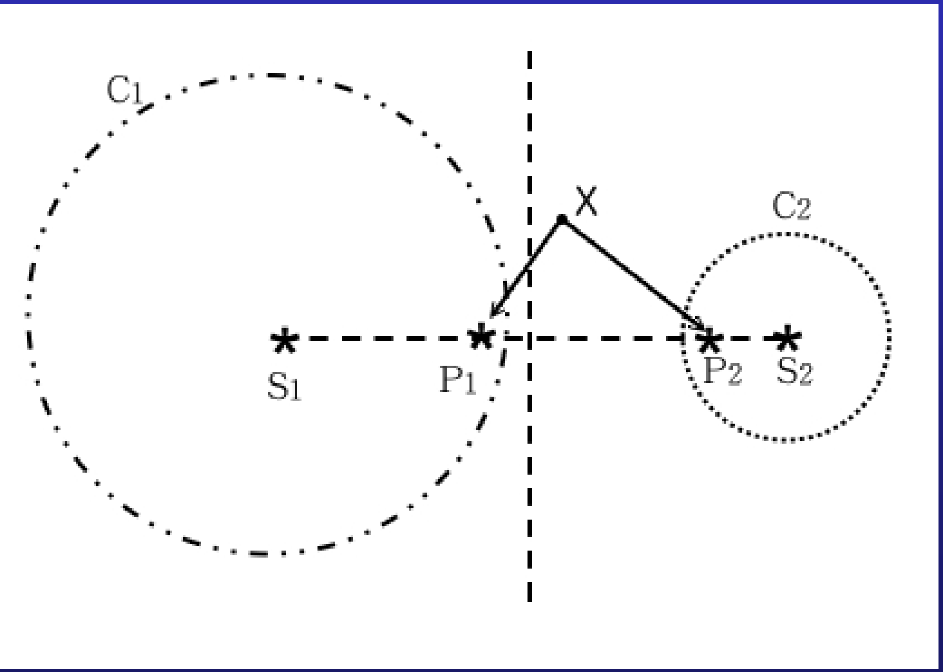
Two Problems in Tree Classifier

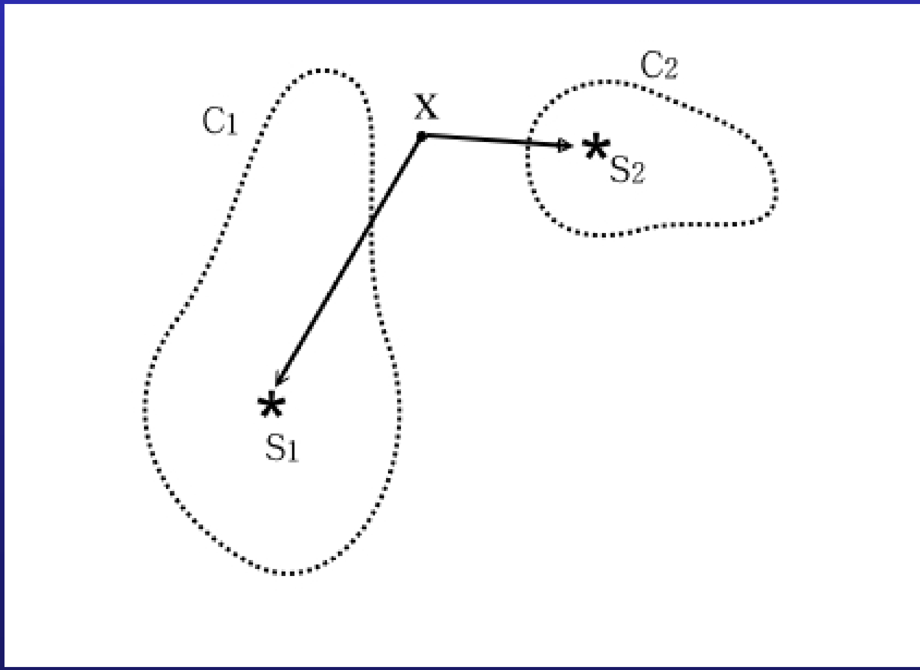
Problem 1

- The classification error usually occurs when the input vector is located in a region between the two clusters.



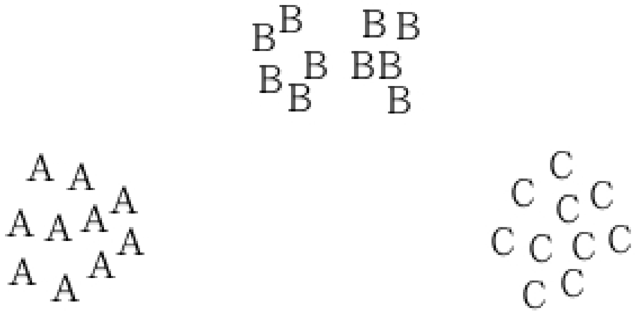
- The cluster centers, S_1 and S_2, are not appropriate for classifying the input vector.
- The cluster center can not represent all vectors contained in a cluster with non-spherical shape.

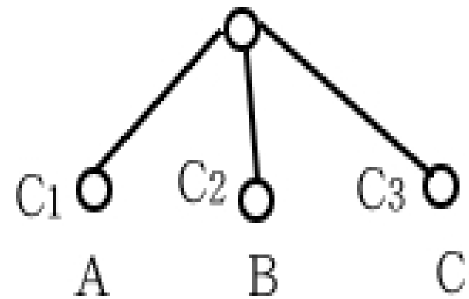


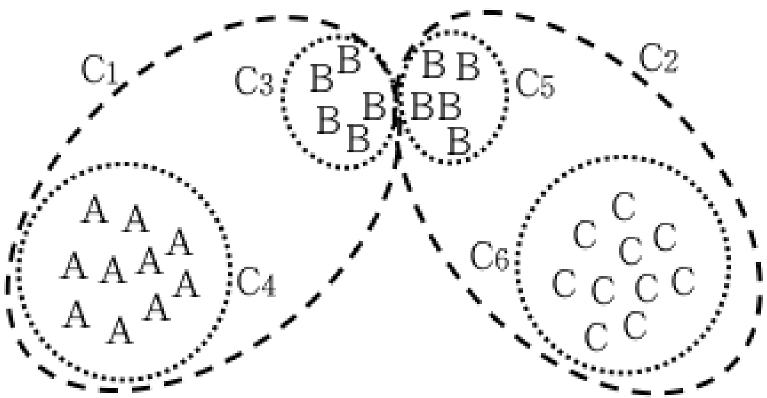


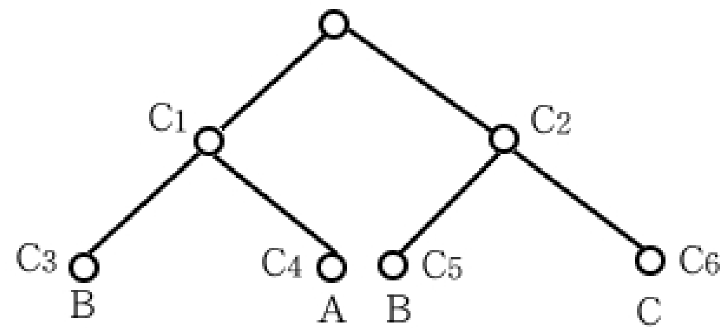
Problem 2

- The user is hard to determine the number of branches for each node in the tree classifier.




C1
C2
C3
A
B
C

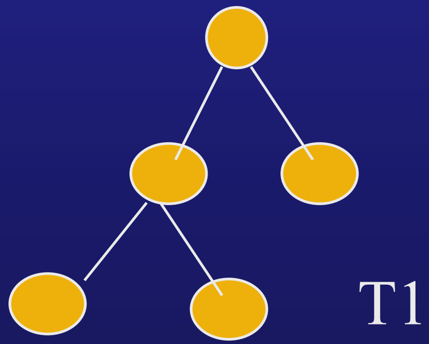

C1
C3
C5
C2
C4
C6

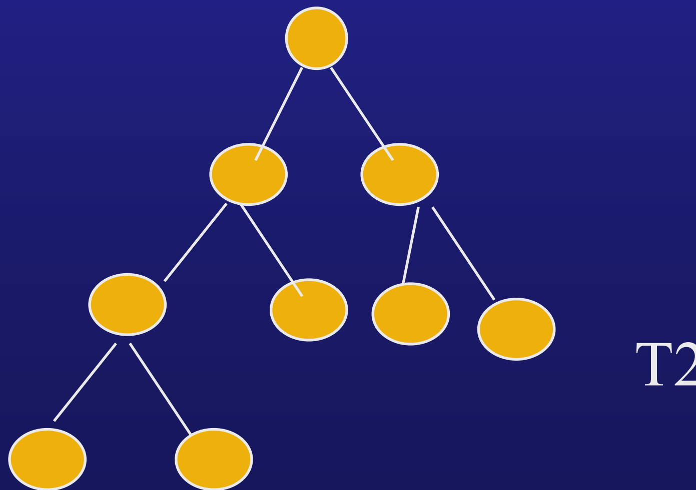

C1
C2
C3
C4
C5
C6
B
A
B
C

Design of Fuzzy Decision Tree

Two design issues:

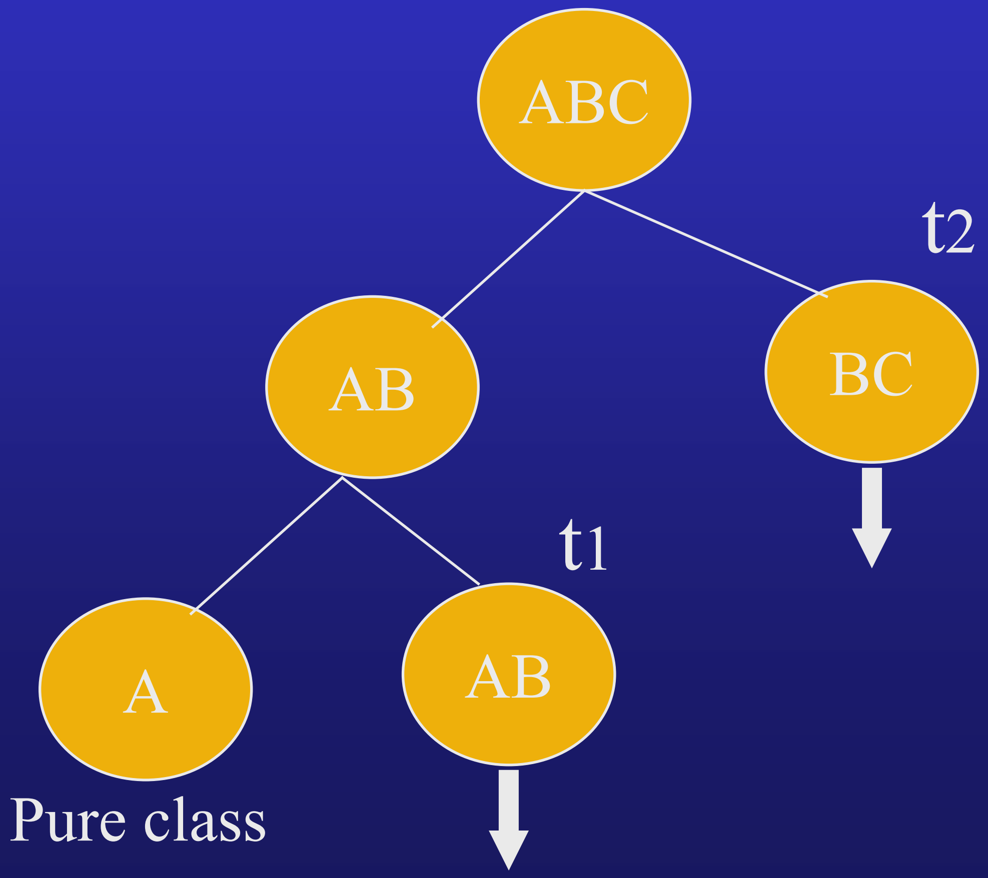
- *First:* The classification error rate of a decision tree is as small as possible. The growing method is used to grow the decision tree to reduce the classification error rate. Hence, a large decision tree is preferred.
- *Second:* The average number of comparisons for the input vector that traces the decision tree should be as small as possible. Thus, a small decision tree is preferred.





Example: C-fuzzy decision tree (CFDT) 2005

- The growing method selects the node to be split according to the classification error rate of the decision tree.



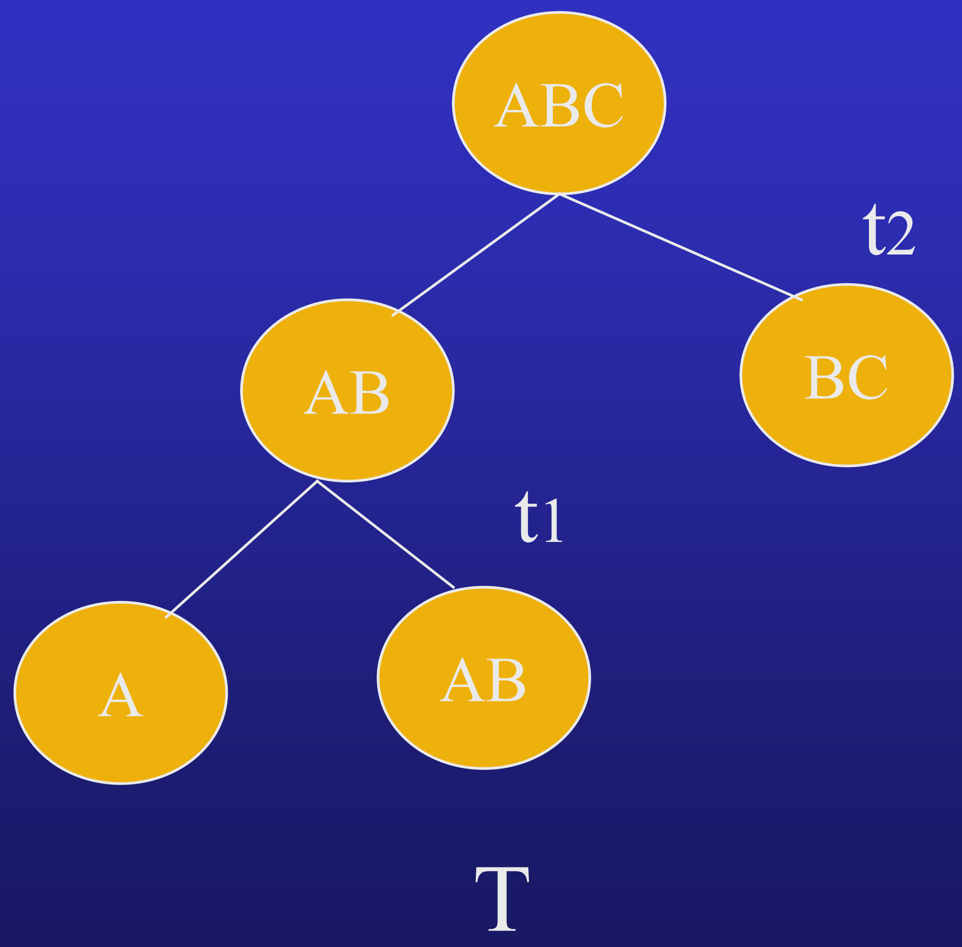
The Problem of CFDT

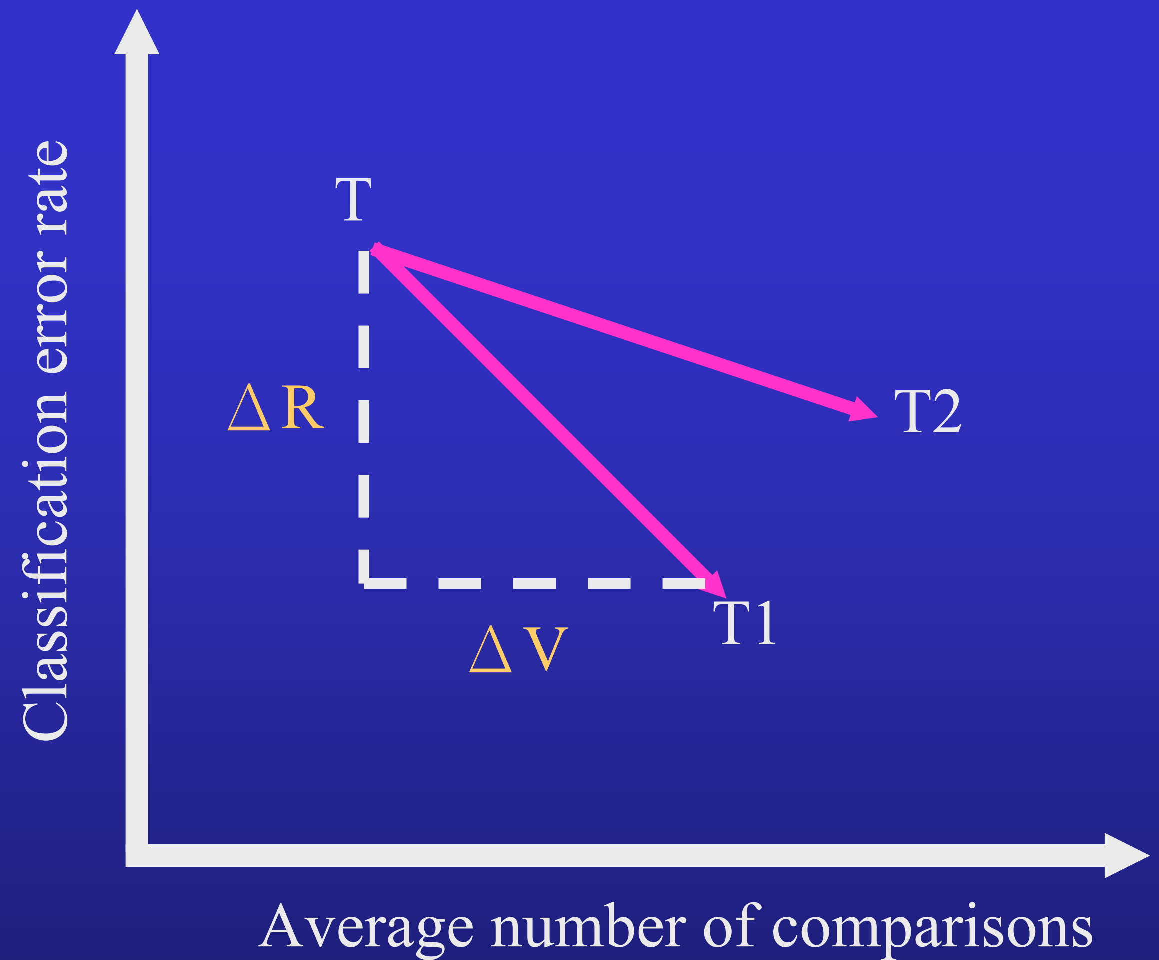
Two Problems:

- The average number of comparisons for the input vector classified in the decision tree is not considered during the design of CFDT, which departing the second design issue for the decision tree.

- The K-means algorithm is applied to classify the vectors into two clusters. However, the performance of CFDT is not considered in the K-means algorithm.

Improve the First Problem of CFDT





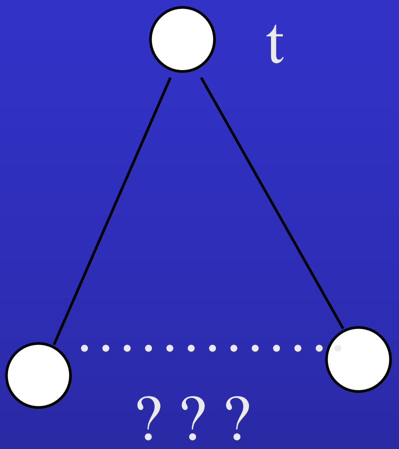
- T1 is better than T2. Thus, the node t1 is selected to be grown by the growing method according to $\lambda = \Delta R / \Delta V$.

Fuzzy Variable-Branch Decision Tree (FVBDT)

- The genetic algorithm (GA) replaces the K-means algorithm to design the decision tree.

- Design issues:
  - The GA can automatically finds the proper number of branches for each node in the FVBDT.

- How do the users decide the number of branches of t ?

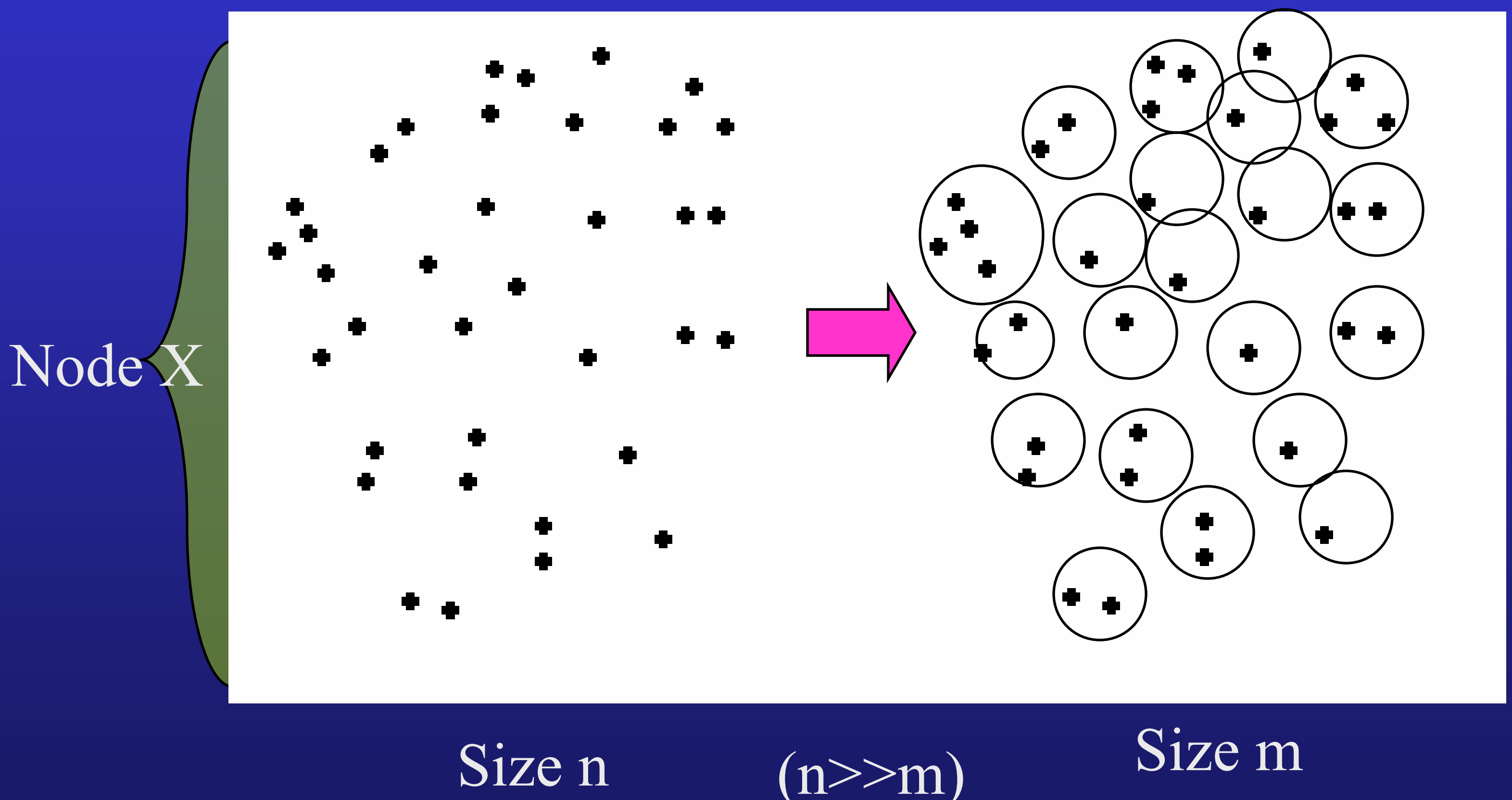


Design of the Genetic Algorithm

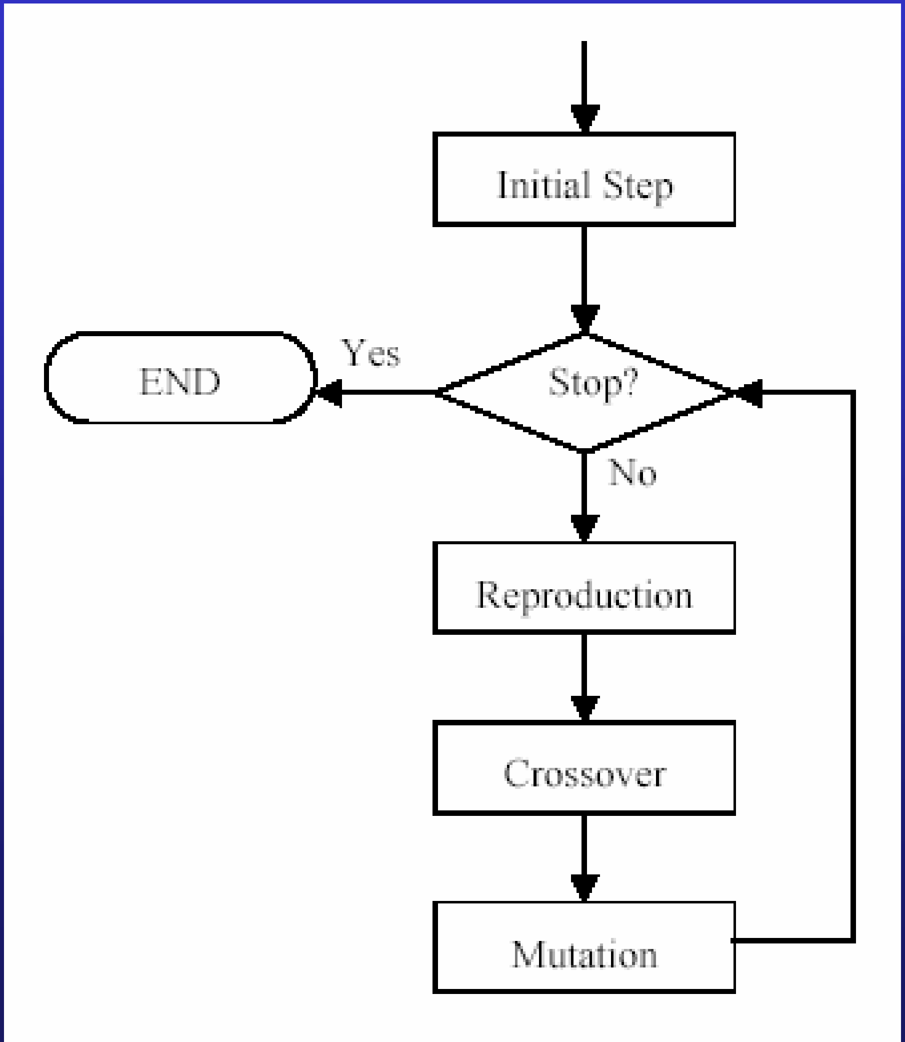
Two stages:

- *First:* Merging method
  - Deal with the large data set .
- *Second:* The genetic algorithm
  - Automatically search for the proper number of branches for each node in the decision tree.

▸ Merging Method: The closer vectors shall be merged into one component.



- The Genetic Algorithm:



Initialization step

m

01001.................

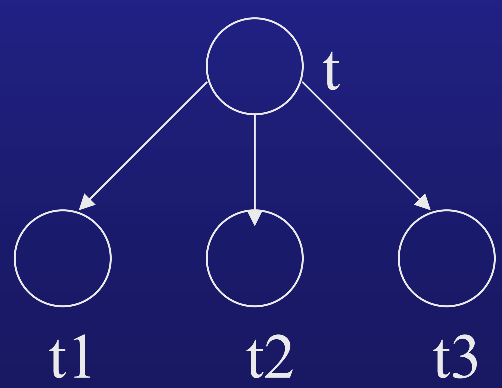
10110.................

00110.................

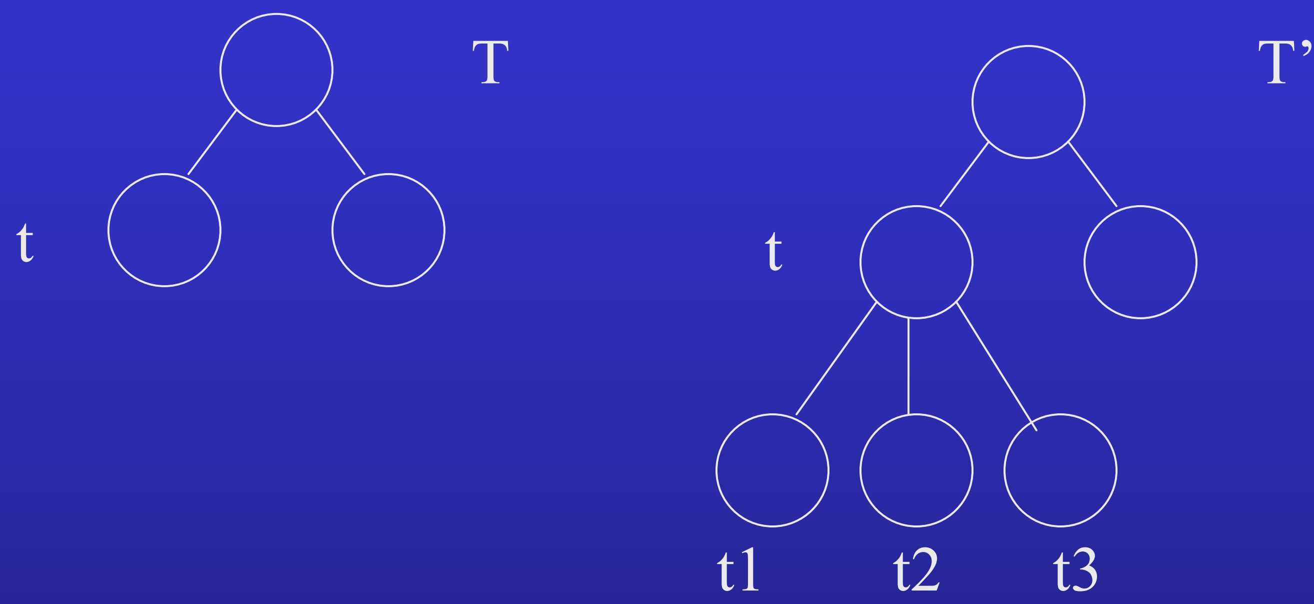
Ex: m=8

R=	0	1	0	0	1	1	0	0
	b1	b2	b3	b4	b5	b6	b7	b8

Three initial seeds, b2,b5 and b6, generate three clusters (branches).



Reproduction phase



Fitness(R)= λ (t)= ΔR / ΔV

Crossover Phase

0 1 0 1 | 0 0 0 1

1 0 0 1 | 0 0 1 1

Crossover

1 0 0 1 0 0 0 1

0 1 0 1 0 0 1 0

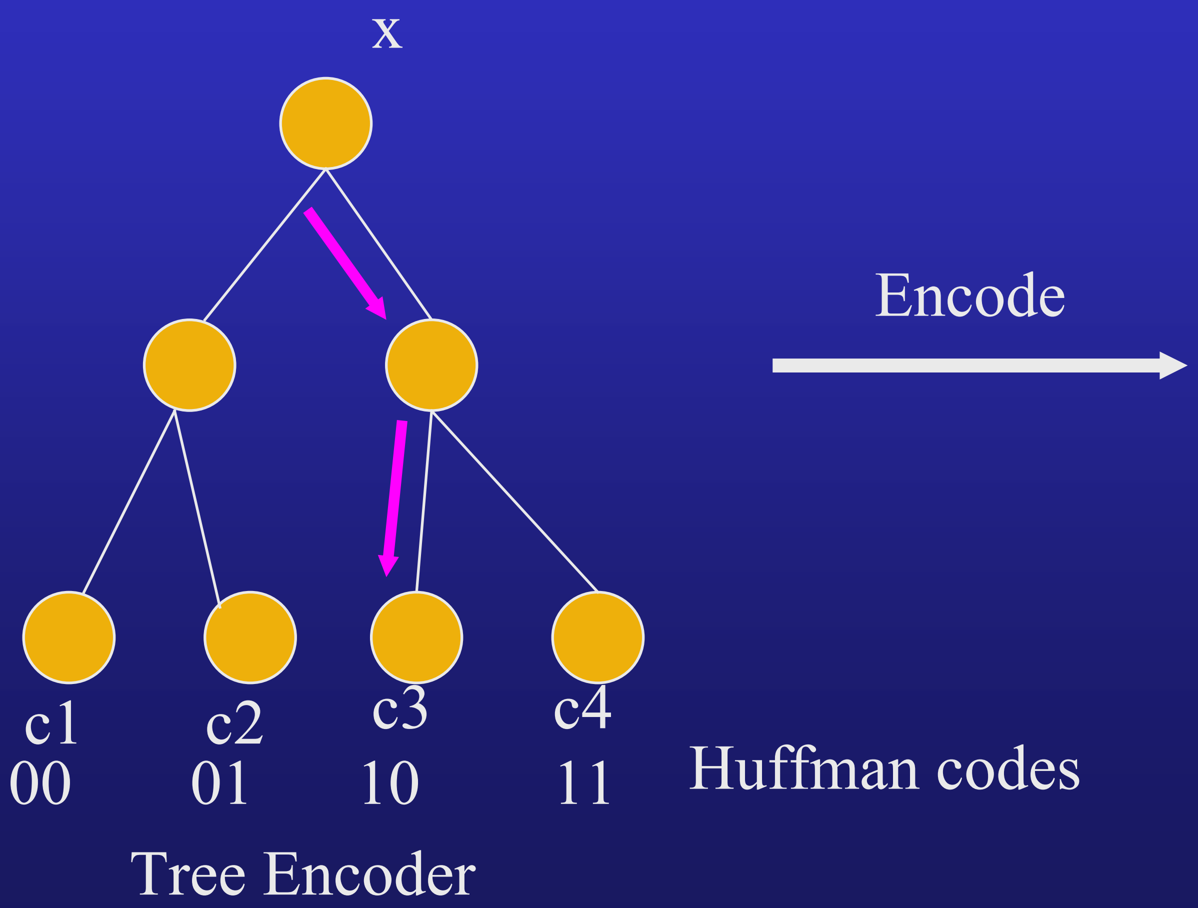
Mutation phase

0 1 0 1 0 0 0 1

Mutation

0 1 1 1 0 0 0 1

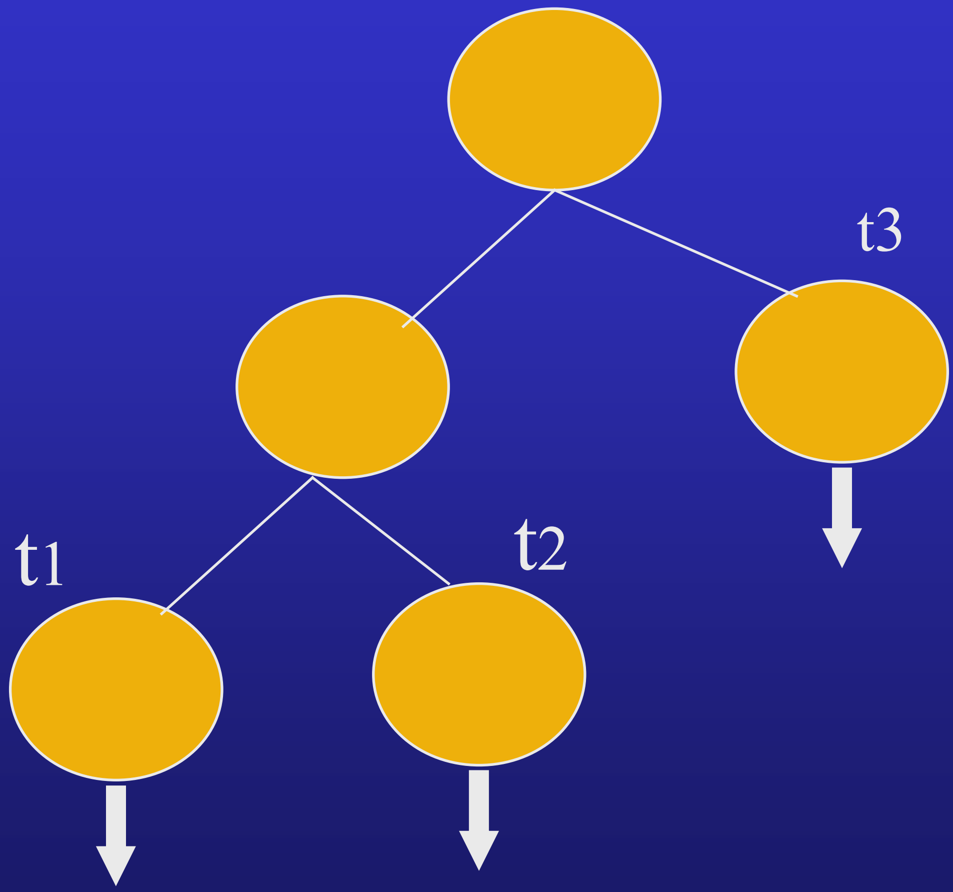
Design of Tree-Structured Vector Quantizer (TSVQ)



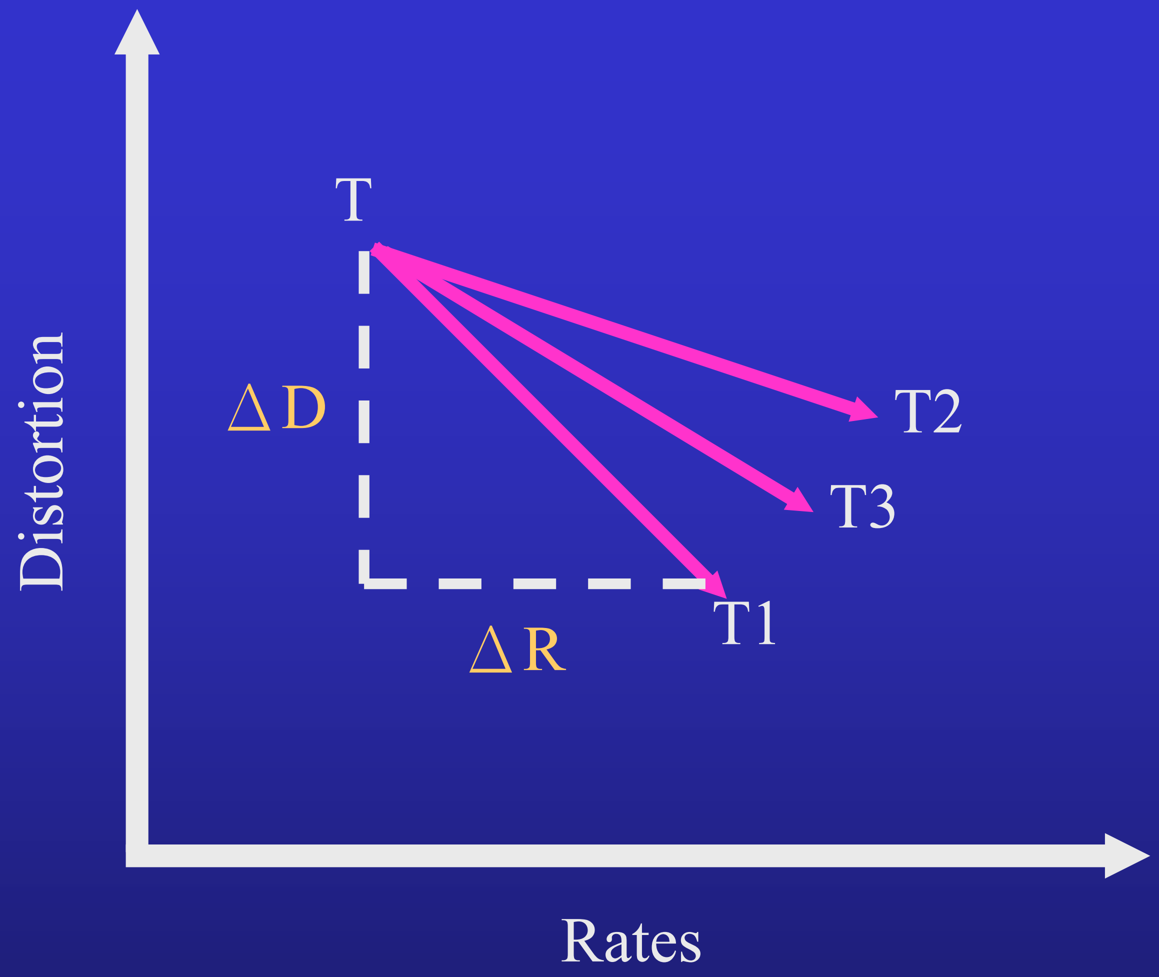
Code=10 $\xrightarrow{\text{Decode}}$

codeword	Huffman code
c1	00
c2	01
c3	10 ☺
c4	11

The Growing Method of TSVQ



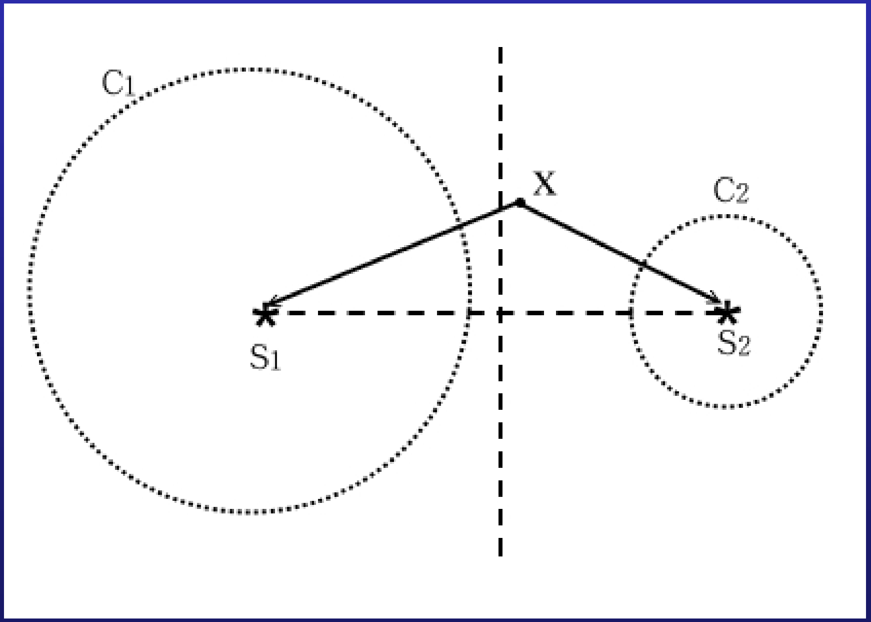
Tree Encoder

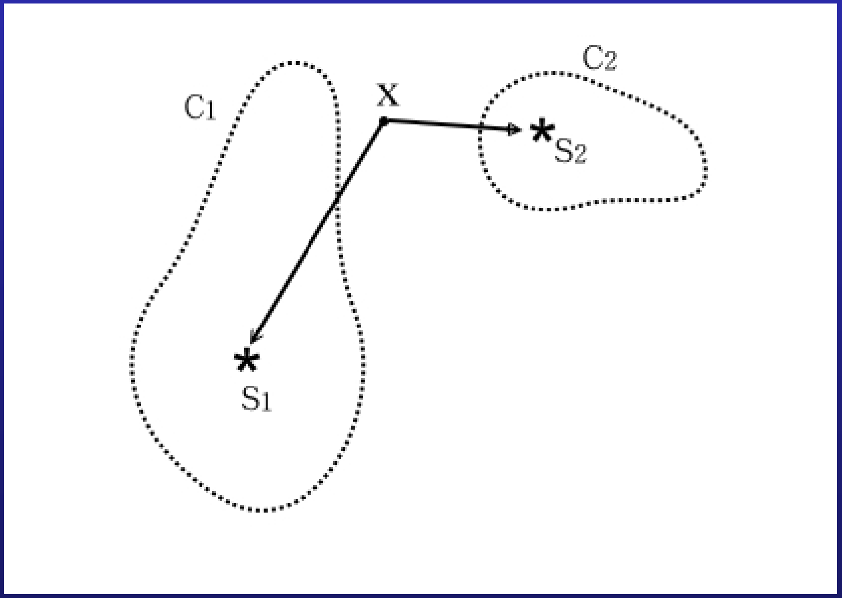


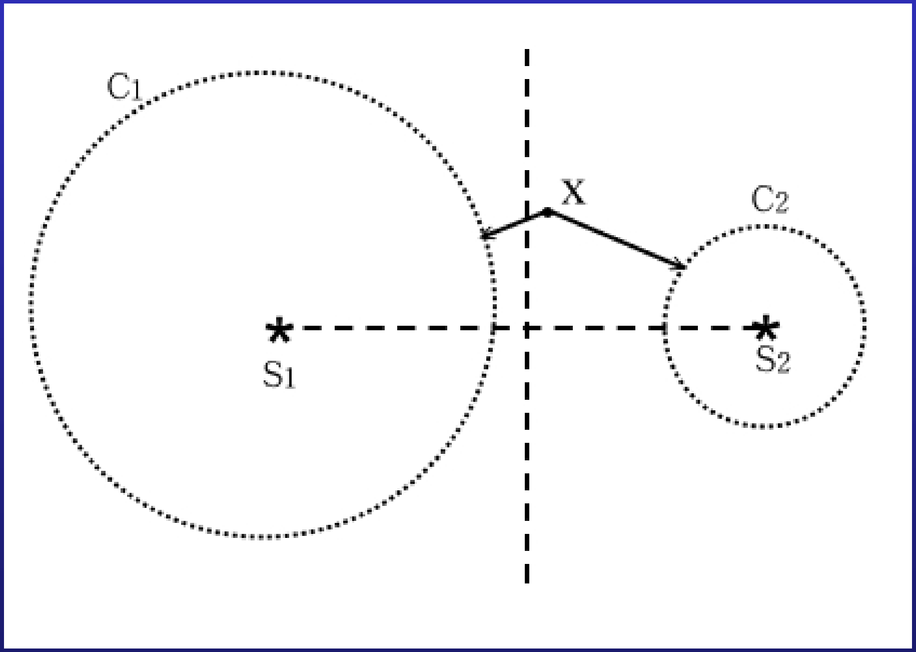
$\lambda(\mathbf{t}) = \Delta \mathbf{D} / \Delta \mathbf{R}$

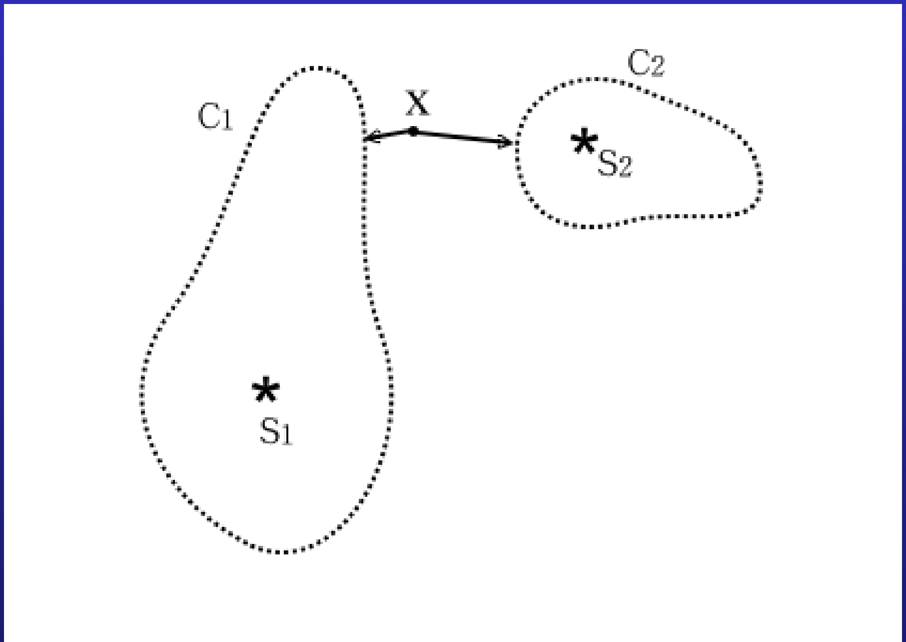
The Problem of TSVQ

- The cluster centers are not suitable to classify the input vector in TSVQ.





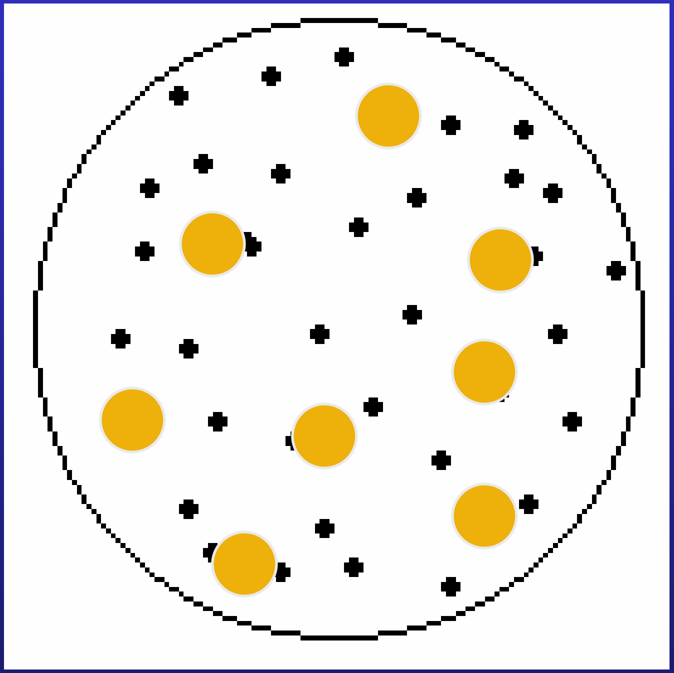

C1
X
C2
S1
S2


C2
C1
X
S2
S1

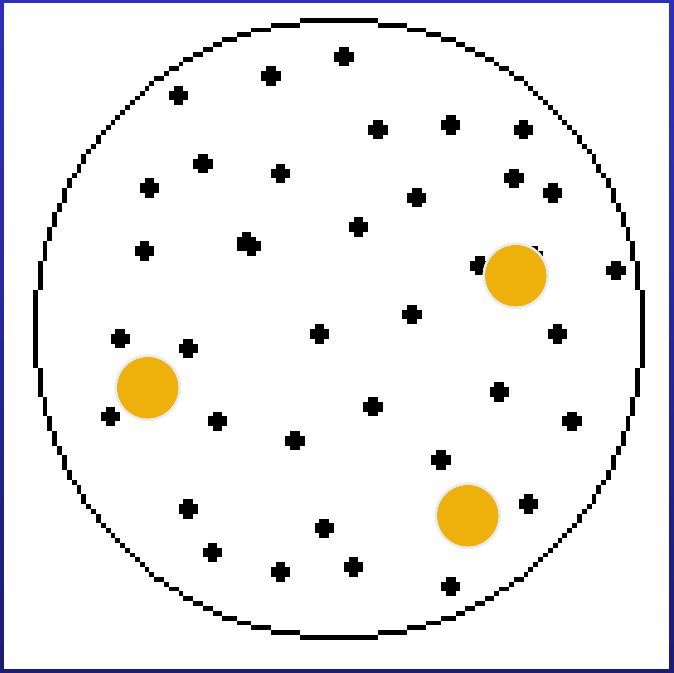
Classification Points in TSVQ

- The classification point replaces the cluster center to classify the input vector in TSVQ.
- How to find the classification points in a node?
- How to determine the number of classification points in a node?

- The classification points shall satisfy the following.
  (1) The number of classification components contained in a node shall be as small as possible.



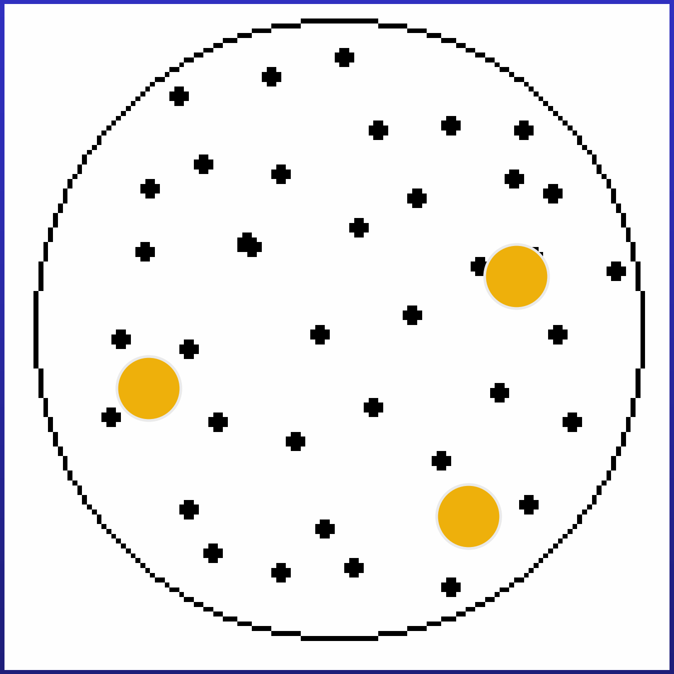
(A)



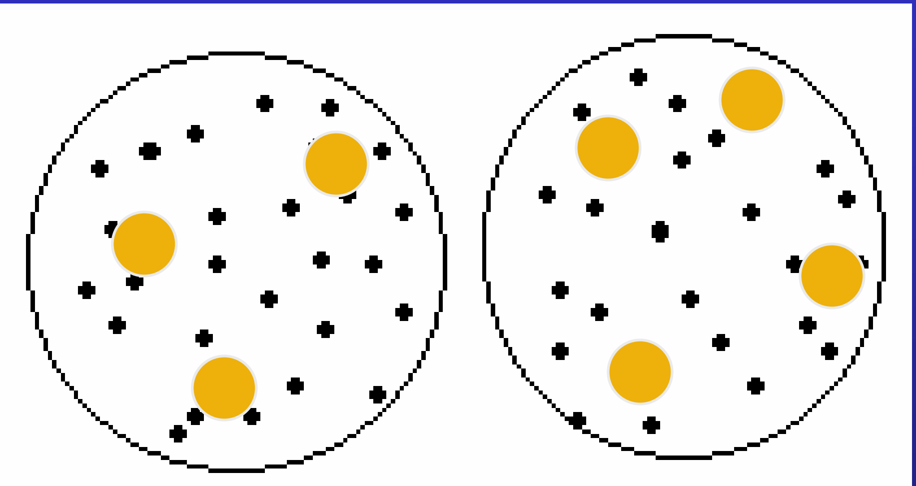
(B)

(B) is better than (A).

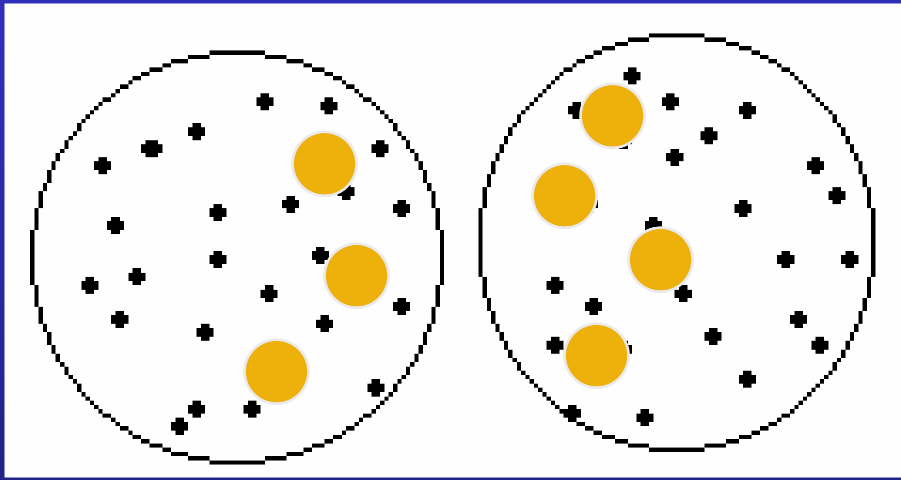
(2) The classification points shall represent all of the vectors in the cluster.



(3) The classification points shall close to the danger region.



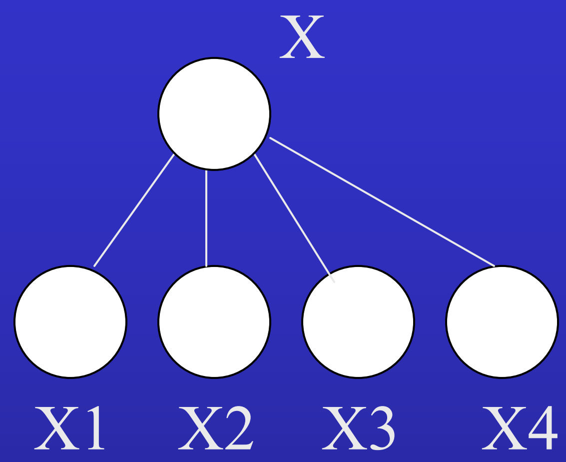
(A)



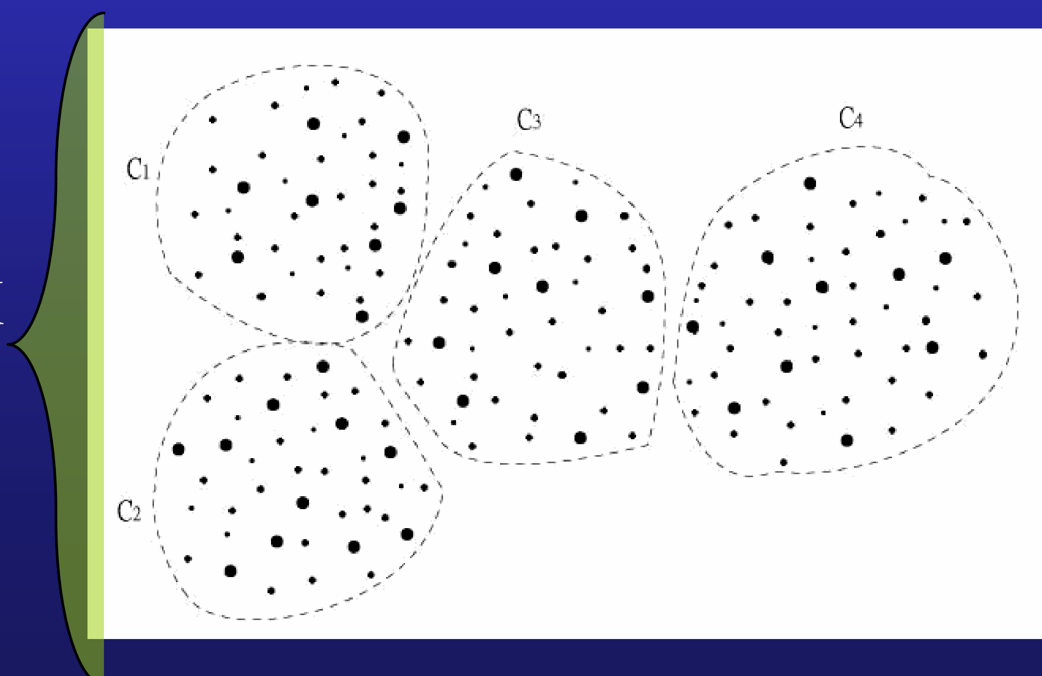
(B)

(B) is better than (A).

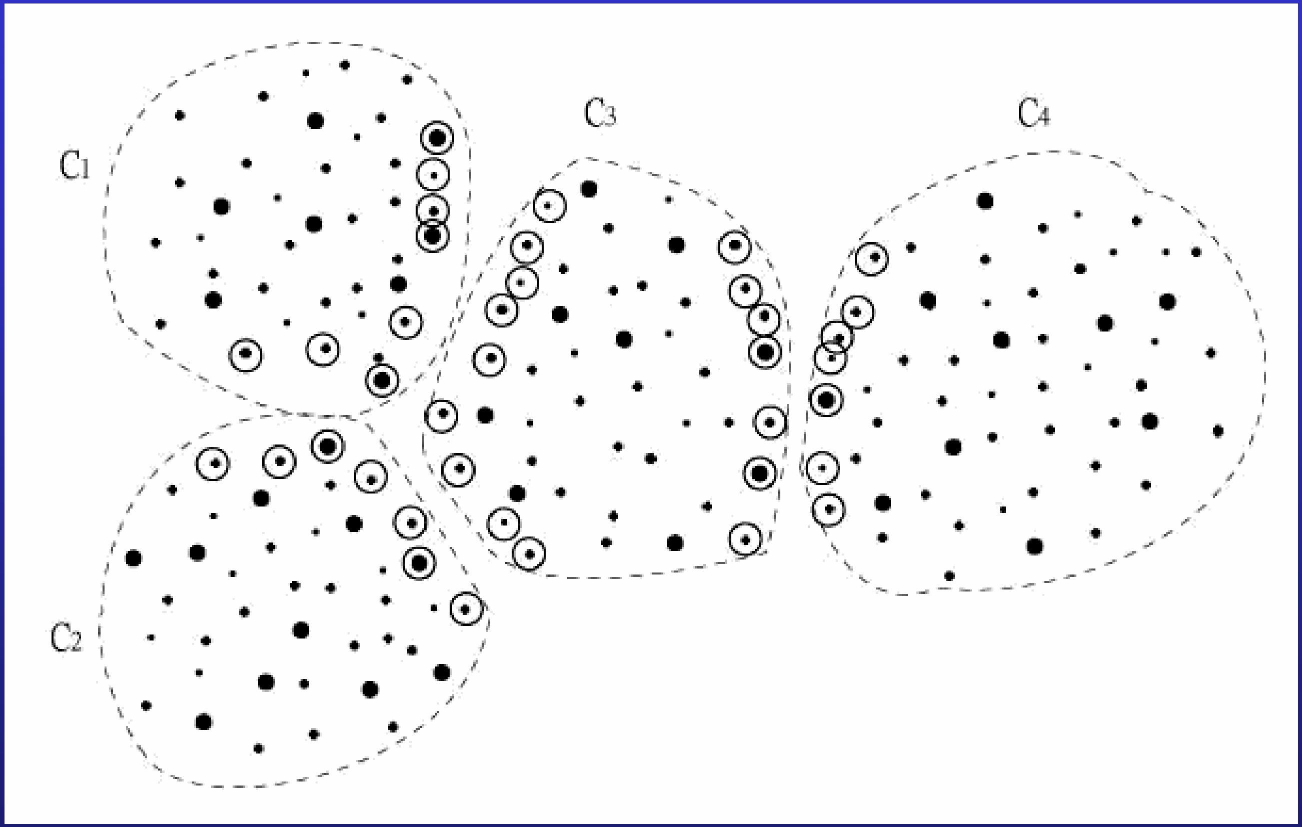
- How to find the classification points in the node?



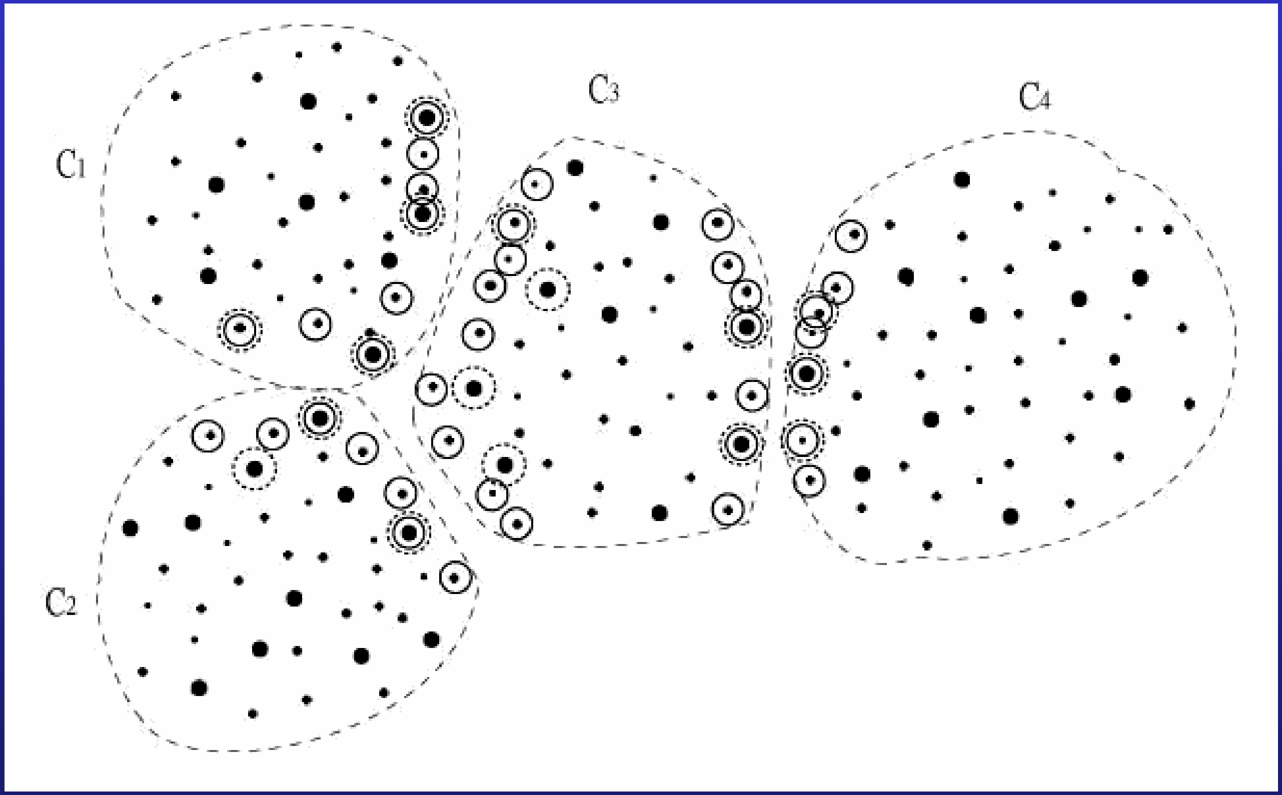
Node X



Step 1. Find the nearest-neighbor points.



Step 2. Find the classification points in each cluster using the genetic algorithm.



Conclusions

- Improvement for both of the variable-branch decision tree and variable-branch vector quantizer.

References

- Shiueng-Bien Yang, "General-tree-structured vector quantizer for image progressive coding using the smooth side-match method", *IEEE Transactions on Circuits and Systems for Video Technology* , vol. 13, pp. 193-202, 2003.

- Shiueng-Bien Yang, "Variable-branch tree-structured vector Quantization", *IEEE Transactions on Image Processing*, vol. 13, pp. 1275-1285, 2004.