Vector Quantization

Jenn-Jier James Lien (連 震 杰)
Professor

Computer Science and Information Engineering
National Cheng Kung University

(O) (06) 2757575 ext. 62540
jjlien@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Major Issues

1. **Supervised learning (PCA, LDA) Vs.**
   Unsupervised learning (VQ)

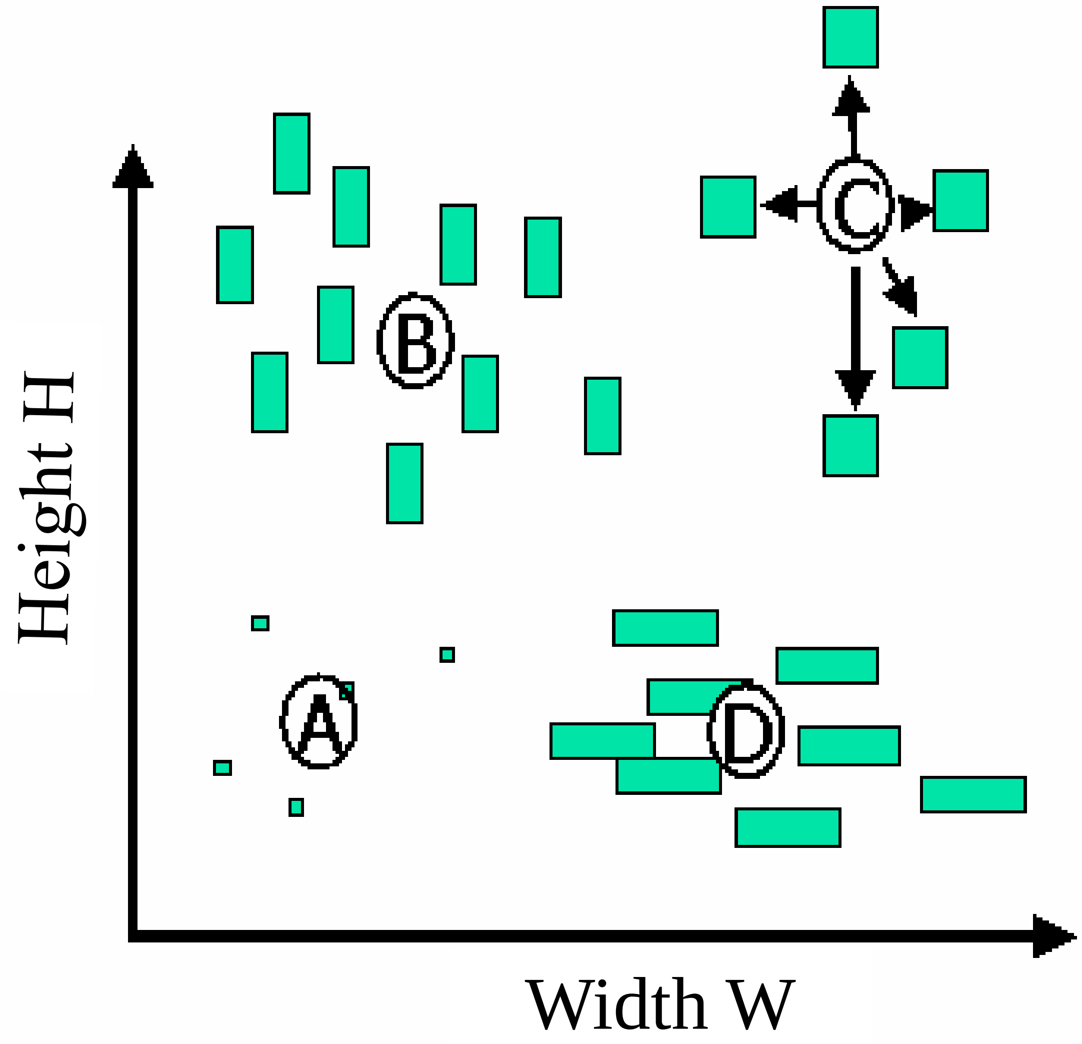
2. **Clustering or classification:**
   K-means (C-means), VQ

3. **k-NN (nearest neighbor) and nearest neighbor rule.**

4. **Codeword, codebook**

Vector Quantization: Example

$$sample \;\; x_i(W_i, H_i)$$

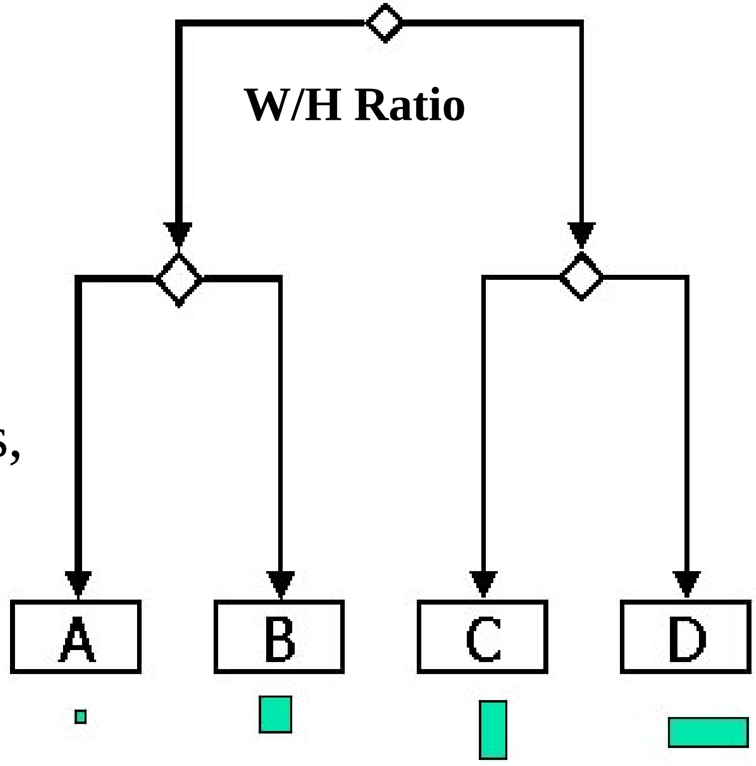


Codebook Size M = 4,
Codeword: A, B, C, D
or 0, 1, 2, 3

Types of Clustering Algorithms

- **The various clustering concepts can be grouped into two broad categories :**

1. **Hierarchical methods –** Minimal Spanning Tree Method (as Figure – problem? Initial error can not be recovered => improved by Wavelets, (FFT: sin@, cos@), (finite elements))
2. **Non-hierarchical methods –K-means Algorithm**



Codebook Size M = 4,
Codeword: A, B, C, D
or 0, 1, 2, 3

Hierarchical/Pyramid/Multi-resolution

Vector Quantization (VQ)

- **Vector Quantizers = Block Quantizers = Block Source Codes**

Ex. after PCA

- **The purpose of designing an M-level vector quantizer (called a codebook with size M) is to partition all k-dimensional training feature vectors into *M* clusters and associate each cluster C^i, whose centroid is the k-dimensional vector c^i, with a quantized value named codeword (symbol) o^i.**

=>codewords

Far away from each cluster

- **While VQ will reduce data redundancy and get rid of small noise, it will inevitably cause a quantization error between each training feature vector x and c^i.**
- **As the size of the codebook increases, the quantization error decreases, and required storage for the codebook entries increases. It is very difficult to find a trade-off among these three factors.**

Consideration of Codebook Design

- **To minimize quantization error, two primary issues are considerable for the design of the codebook:**
  1. Codebook creation (the size of codebook)
  2. Distortion measurement (local Vs. global optimization)

1. **Defining the size of codebook is still an open problem when we use the VQ technique.**
   - According to our experimental result (in speech), the codebook size is at least 1/50 less than the number of all k-dimensional training feature vectors

Make sure each cluster has sufficient samples

2. **For the distortion measurement, there are two main considerations for optimizing the VQ:**

Local optimization, not global optimization

(i) The quantizer must satisfy the nearest neighbor rule.

$$x = [x_1, x_2, \ldots, x_k] \in C^i \qquad if \ \left\| x - c^i \right\| < \left\| x - c^j \right\|$$

weight: fuzzy, probability…

$$where \ \boxed{\left\| x - c^i \right\| = \sum_{h=1}^{k} (x_h - c_h^i)^2} \ and \ i \neq j, \quad i,j = 0,1,\ldots,M\text{-}1$$

Similarity Measure

$$\boxed{\left\| x - c^i \right\| = \sum_{h=1}^{k} w_h (x_h - c_h^i)^2}$$

$$and \ \ q(x) = o^i \qquad where \ \ 0 \le o^i \le M - 1$$

$q(.)$ is the quantization operator

(ii) Each cluster center c^i must minimize not only the local distortion D^i in cluster C^i but also total/global quantization errors D.

$$D = \sum_{i=0}^{M-1} D^i \qquad where \ \ D^i = \sum_{n=1}^{N} \left\| x_n^i - c^i \right\| = \sum_{n=1}^{N} \sum_{h=1}^{k} (x_{n,h}^i - c_h^i)^2$$

Using the overall distortion measurement, it is hard to guarantee global minimization.

- Only sum each local distortion. Improved by, ex., Fisher Discriminant (within/between or intra/inter)

- Global optimization can be approximated by iterative computation of local optimization.

K-Nearest Neighbor (K-NN)

- **Winner/majority takes all**
  1. K samples
     - Not limit the distance/range
  2. Within the distance k

The K-Means Clustering

The K-Means Clustering or
The C-Means Clustering

(AAM or GMM+EM)

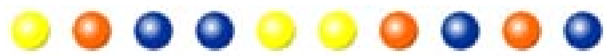
What Are Clustering Algorithms?

What is clustering ?

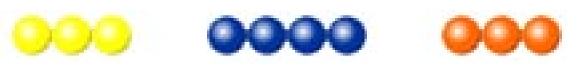
Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.

Example:

The balls of same color are clustered into a group as shown below :



Thus, we see clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.



definition

Better example: different shape (circle, rectangle…) and color

Classification by color, shape …

What Is a Clustering Algorithm ?

- **A clustering algorithm attempts to find natural groups of components (or data) based on some similarities.**

- **The clustering algorithm also finds the *centroid* of a group of data sets.**

- **The centroid of a cluster is a point (one or high dimensional vector) whose parameter values are the mean of the parameter values of all the points in the clusters.**

What Is the Common Metric for Clustering Techniques ?

- **Generally, the** distance between two points **is taken as a common metric to assess the** similarity **among the components of a population. The most commonly used distance measure is the** Euclidean metric **which defines the distance between two points $p = (p_1, p_2, \ldots, p_k)$ and $q = (q_1, q_2, \ldots, q_k)$ as :**

high dimensional points

$$d = \sqrt{\sum_{i=1}^{k} (p_i - q_i)^2}$$

Other distance / similarity measured metric:

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} \left| x_i - \hat{x}_i \right|^2$$

Norm L^v,

v=1/2, 1, or 2

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} \left| x_i - \hat{x}_i \right|^v = \left\| x - \hat{x} \right\|_v^v$$

$$d(x,\hat{x}) = \left\{ \sum_{i=0}^{k-1} \left| x_i - \hat{x}_i \right|^v \right\}^{1/v} \triangleq \left\| x - \hat{x} \right\|_v$$

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} w_i \left| x_i - \hat{x}_i \right|^2$$

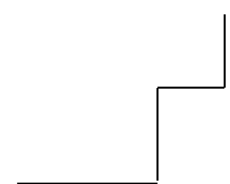
$$d(x,\hat{x}) = \max_{0 \le i \le k-1} \left| x_i - \hat{x}_i \right|$$

$$d(x,\hat{x}) \le d(x,y) + d(y,\hat{x})$$

$\underline{P(c_i|x)}$

$\underline{P(c_j|x)}$

- Hausdorff distance
- Gaussian measure
- Posterior prob. / Likelihood prob.
- Earth mover method

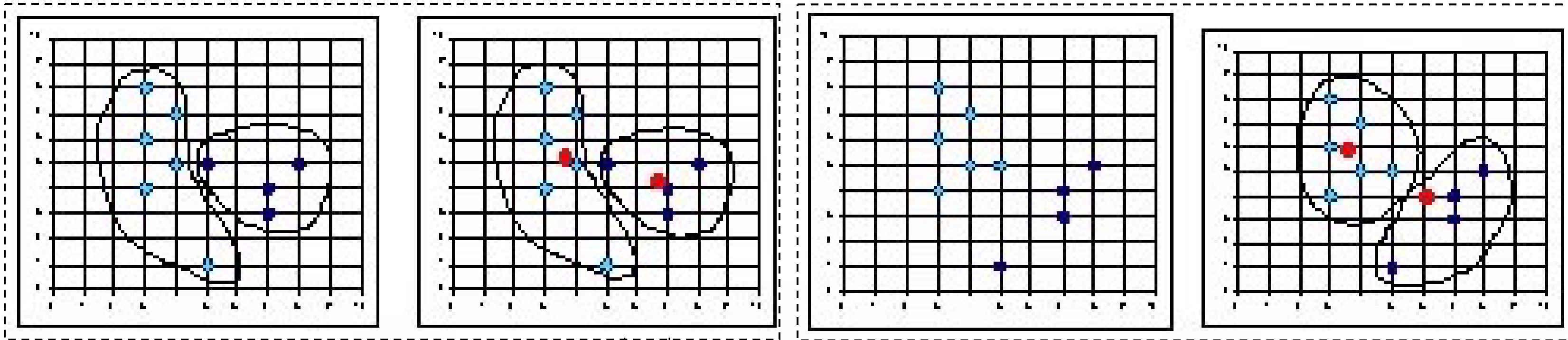


Manhattan distance

Uses of Clustering Algorithms

- **Engineering sciences:**
  - Pattern recognition, artificial intelligence, cybernetics, multimedia, compression, information security, etc.
  - Typical examples to which clustering has been applied include handwritten characters, samples of speech, fingerprints, and pictures.
- **Life sciences:**
  - Biology, botany, zoology, entomology, cytology, microbiology.
  - The objects of analysis are life forms such as plants, animals, and insects.
- **Information, policy and decision sciences:**
  - The various applications of clustering analysis to documents include votes on political issues, survey of markets, survey of products, survey of sales programs, and R & D.

The K-Means (Clustering) Algorithm



new center +
new clustering

Definition:

This non-hierarchical method initially takes the number of components of the population equal to the final required number of clusters. In this step itself, the final required number of clusters is chosen such that the points are mutually farthest apart.

Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance.

The centroid's position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

The K-means algorithm: (corresponding to the 12 pts example)

Step 1: Initialization - Define the codebook size to be M and choose M initial (1st iteration) k-dimensional cluster centers $c^0(1), c^1(1), \ldots, c^{M-1}(1)$ corresponding to each cluster C^i where $0 \le i \le M-1$.

Step 2: Classification - At the lth iteration, according to the nearest neighbor rule, classify each k-dimensional sample x of training feature vectors into one of the clusters C^i.

Local optimization

$$x \in C^i(l) \quad \text{if} \quad \left\| x - c^i(l) \right\| < \left\| x - c^j(l) \right\| \text{ where } i \neq j,\ i, j = 0,1,\ldots, M-1$$

Step 3: Codebook Updating - Update the codeword (symbol) o^i of each cluster C^i by computing new cluster centers $c^i(l+1)$ where $i = 0,1,\ldots,M-1$ at the $l+1$th iteration.

$$c^i(l+1) = \frac{1}{N}\sum_{n=1}^{N} x_n^i \qquad \text{where} \quad x^i \in C^i(l+1)$$

N is the number of feature vectors in cluster $C^i(l+1)$ at the $l+1$th iteration, and

$$q(x) = o^i \qquad \text{where} \quad 0 \le o^i \le M-1$$

where $q(.)$ is the quantization operator.

Step 4: Termination - If the decrease in the overall distortion at the current iteration $l+1$ compared with that of the previous iteration l is below a selected threshold, then stop; otherwise goes back to Step 2.

Local optimization

$$\begin{cases} \text{if } \left| D(l+1) - D(l) \right| < \text{threshold, then Stop} \\ \text{if } \left| D(l+1) - D(l) \right| \ge \text{threshold, then Goes to Step } 2 \end{cases}$$

Convergent condition

(D(l+1)-D(l))/D(l)

Direct k-means clustering algorithm:

k cluster centers

function Direct-k-means()

Initialize k prototypes $(w_1, \ldots, w_k)$ such that $w_j = i_l,\ j \in \{1, \ldots, k\},\ l \in \{1, \ldots, n\}$

n samples

Each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_l, where $l \in \{1, \ldots, n\}$, *do*

Assign i_l to the cluster C_{j*} with nearest prototype w_{j*} (i.e., $| i_l - w_{j*} | \leq | i_l - w_j |,\ j \in \{1, \ldots, k\}$)

for each cluster C_j, where $j \in \{1, \ldots, k\}$, *do*

Update the prototype w_j to be the centroid of all samples currently in C_j, so that $w_j = \sum_{i_l \in C_j} i_l / | C_j |$

Compute the error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} | i_l - w_j |^2$$

Until E does not change significantly or cluster membership no longer changes

Convergent condition

The Parameters and Options for the K-means Algorithm

1) **Initialization**: Different init Methods
2) **Distance Measure**: There are different distance measures that can be used. (Manhattan distance & Euclidean distance).
3) **Termination/convergence**: k-means should terminate when no more pixels are changing classes.
4) **Quality/total cost or error (entropy)**: the quality of the results provided by k-means classification
5) **Parallelism**: There are several ways to parallelism the k-means algorithm (other comparable methods)
6) **What to do with dead classes**: A class is "dead" if no pixels belong to it.
7) **Split or combine if necessary ?**
8) **Variants**: One pass on-the-fly calculation of means
9) **Number of classes**: Number of classes is usually given as an input variable. (M is given in the beginning)

Comments on the K-means Methods

Strength of the K-means:

• Relatively efficient: O(lMn), where **n** is the number of objects, **M** is the number of clusters, and **l** is number of iterations. Normally, M,l << n.

Make sure each cluster has sufficient samples

• Often terminates/convergence at a local optimum.

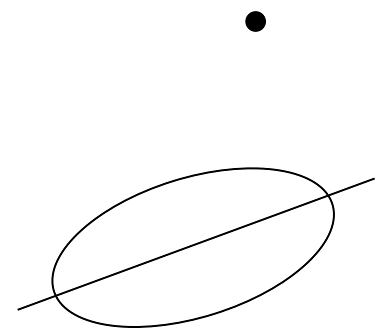
That is, the K-means algorithm is an iterative algorithm which can guarantee a local minimum, and works well in practice.

Weakness of the k-means:

• Applicable only when mean is defined, then what about categorical data ? (need to specify cluster centers in the beginning)

• Need to specify **M**, the number of clusters, in advance.

•That is, the behavior of the K-means algorithm is affected by the number of clusters specified and the choice of initial cluster centers.



outlier

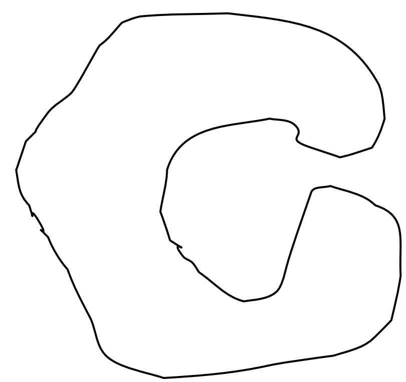
- Unable to handle noisy data and outlines.

  (conflicting with previous statement ?

  Unable to handle noisy training data.

  Good to deal with small noisy testing data.)

- Not suitable to discover clusters with non-convex shapes.
- The K-means algorithm can only converge to a local optimum, not global optimum.



A Simple Example of K-means

(0) Initialization: $N = 4$, $k = 2$, $\epsilon = .001$, $n = 12$.

M   length of each vector   # of vectors

Training Sequence:

$\underset{\sim}{x}_1 = (-.37449, .98719)$	$\underset{\sim}{x}_7 = (-.59161, .17968)$
$\underset{\sim}{x}_2 = (.63919, -.11875)$	$\underset{\sim}{x}_8 = (.14093, 1.76413)$
$\underset{\sim}{x}_3 = (-.83293, .60645)$	$\underset{\sim}{x}_9 = (.70898, -.35017)$
$\underset{\sim}{x}_4 = (-.70534, -1.21856)$	$\underset{\sim}{x}_{10} = (.30038, .79836)$
$\underset{\sim}{x}_5 = (-.28952, -.94821)$	$\underset{\sim}{x}_{11} = (.30165, 1.06552)$
$\underset{\sim}{x}_6 = (1.09924, .516)$	$\underset{\sim}{x}_{12} = (-.37801, -.32708)$

$$\hat{A}_0 = [(2,2), (2,-2), (-2,2), (-2,-2)]$$

$$= [y_1, y_2, y_3, y_4]$$

$D_{-1} = 9.99E + 62$ (∞ on a microcomputer)

Set $m = 0$.

$\underline{\underline{m=0}}$ (1) Find $P(\hat{A}_0) = \{S_1, S_2, S_3, S_4\}$:

$\underset{\sim}{x}_j \in S_1$ if $\boxed{d(\underset{\sim}{x}_j, \underset{\sim}{y}_1) \le d(\underset{\sim}{x}_j, \underset{\sim}{y}_m)}$, all m.

$S_1 = \{\underset{\sim}{x}_6, \underset{\sim}{x}_8, \underset{\sim}{x}_{10}, \underset{\sim}{x}_{11}\}$

$S_2 = \{\underset{\sim}{x}_2, \underset{\sim}{x}_9\}$

$S_3 = \{\underset{\sim}{x}_1, \underset{\sim}{x}_3, \underset{\sim}{x}_7\}$

$S_4 = \{\underset{\sim}{x}_4, \underset{\sim}{x}_5, \underset{\sim}{x}_{12}\}$

Compute D_0:

$$D_0 = \frac{1}{12} \sum_{j=1}^{12} \min_{\underset{\sim}{y} \in \hat{A}_0} d(\underset{\sim}{x}_j, \underset{\sim}{y}) = 2.0172 .$$

(2) $(D_{-1} - D_0)/D_0 > .001$, continue.

(3) Find the optimal reproduction alphabet $\hat{A}_1 \triangleq \hat{\underset{\sim}{x}}(P(\hat{A}_0)) = \{\hat{\underset{\sim}{x}}(S_i), i=1,\ldots,4\}$:

$$\hat{\underset{\sim}{x}}(S_1) = (\underset{\sim}{x}_6 + \underset{\sim}{x}_8 + \underset{\sim}{x}_{10} + \underset{\sim}{x}_{11})/4 = (.46055,\ 1.036)$$

$$\hat{\underset{\sim}{x}}(S_2) = (\underset{\sim}{x}_2 + \underset{\sim}{x}_9)/2 = (.674085,\ -.23446)$$

$$\hat{\underset{\sim}{x}}(S_3) = (\underset{\sim}{x}_1 + \underset{\sim}{x}_3 + \underset{\sim}{x}_7)/3 = (-.599676,\ .591106)$$

$$\hat{\underset{\sim}{x}}(S_4) = (\underset{\sim}{x}_4 + \underset{\sim}{x}_5 + \underset{\sim}{x}_{12})/3 = (-.457623,\ -.831283)$$

Set $m = 1$. Go to (1).

$\underline{\underline{m=1}}$ (1) Find $\mathcal{P}(\hat{A}_1)$:

Evaluating distortions shows $\mathcal{P}(\hat{A}_1) = \mathcal{P}(\hat{A}_0)$ (no change in partition)

Compute D_1:

$$D_1 = \frac{1}{12} \sum_{j=1}^{12} \min_{\underset{\sim}{y} \in \hat{A}_1} d(\underset{\sim}{x}_j, \underset{\sim}{y}) = .0997306 .$$

$$\boxed{(2) \quad (D_0 - D_1)/D_1 \;\tilde{=}\; 19 > .001}$$

(3) $\hat{A}_2 \triangleq \hat{\underset{\sim}{x}}(\mathcal{P}(\hat{A}_1)) = \hat{A}_1$, since $\mathcal{P}(\hat{A}_1) = \mathcal{P}(\hat{A}_0)$ and hence

$\hat{\underset{\sim}{x}}(\mathcal{P}(\hat{A}_1)) = \hat{\underset{\sim}{x}}(\mathcal{P}(\hat{A}_0)) = \hat{A}_1$. Thus $\hat{A}_1$ is a fixed point. Set $m=2$. Go to (1).

$\underline{\underline{m=2}}$ (1) $\mathcal{P}(\hat{A}_1) = \mathcal{P}(\hat{A}_0)$ and hence $D_2 = D_1$ and hence $(D_1 - D_2)/D_2 = 0 < .001$.

Halt with final quantizer described by $\{\hat{A}_1, \mathcal{P}(\hat{A}_1)\}$.

Fuzzy K-Means Algorithm

The Vector Quantization (VQ) Algorithm

- **Is an extended algorithm of K-means, but unlike K-means which initializes each cluster center in the beginning.**
- **This VQ algorithm uses iterative methods, splits the training vectors from assuming whole data to be one cluster to 2,4,8,…,M (M's size is power of 2) clusters, and determines the centroid for each cluster. The centroid of each cluster is refined iteratively by K-means clustering.**

problem ? Split or Combine ?

The Vector Quantization Algorithm

Step 1: Initialization - Assume all N k-dimensional training vectors to be one cluster C^0, *i.e.*, codebook size M = 1 and codeword o^0 = 0, and find its k-dimensional cluster centroid $c^0(1)$ where 1 is the initial iteration.

$$c^0(1) = \frac{1}{N}\sum_{n=1}^{N} x_n^0$$

where x is one sample of all N k-dimensional feature vectors at cluster C^0.

Step 2: Splitting - Double the size M of the codebook by splitting each cluster into two. The current codebook size M is split into $2M$. Set $M = 2M$ by

$$\begin{cases} c_+^i(l) = c^i(l) + \varepsilon \\ c_-^i(l) = c^i(l) - \varepsilon \end{cases} \qquad where \;\; 0 \le i \le M-1$$

c^i is the centroid of the ith cluster C^i, M is the size of current codebook, ε is a k-dimensional splitting parameter vector and is value 0.0001 for each dimension in our study. *1* is the initial iteration.

Step 3: Classification - At the lth iteration, according to the nearest neighbor rule, classify each k-dimensional sample x of training feature vectors into one of the clusters C^i.

$$x \in C^i(l) \quad if \ \left\| x - c^i(l) \right\| < \left\| x - c^j(l) \right\| \ \text{where } i \neq j, \ i, j = 0,1,\ldots, M-1$$

Step 4: Codebook Updating - Update the codeword (symbol) o^i of each cluster C^i by computing new cluster centers $c^i(l+1)$ where $i = 0,1,\ldots,M-1$ at the $l+1$th iteration.

$$c^i(l+1) = \frac{1}{N} \sum_{n=1}^{N} x_n^i \ \text{ where } \mathrm{x}^{\mathrm{i}} \in C^i(l+1)$$

N is the number of feature vectors in cluster $C^i(l+1)$ at the $l+1$th iteration. And

$$q(x) = o^i \ \text{ where } 0 \leq o^i \leq M-1$$

where $q(.)$ is the quantization operator.

Step 5: Termination 1 - If the difference between the current overall distortion $D(l+1)$ and that of the previous iteration $D(l)$ is below a selected threshold, proceed to Step 6; otherwise goes back to Step 3.

$$\begin{cases} if \ \left| D(l+1) - D(l) \right| \ < \ threshold, \ then \ Goes \ to \ Step \ 6 \\ if \ \left| D(l+1) - D(l) \right| \ \geq \ threshold, \ then \ Goes \ to \ Step \ 3 \end{cases}$$

Convergent condition

$(D(l+1)-D(l))/D(l)$

(where *threshold* is 0.0001 in our study.)

How to improve it to be heuristic ?

Step 6: Termination 2 -

Is the codebook size M equal to the VQ codebook size required ?

$$\begin{cases} if \ Yes, \ then \ Stop \\ if \ No, \ then \ Goes \ to \ Step \ 2 \end{cases}$$

The VQ Algorithm = Codebook Creation

- **Step 1: Initialization: M=1**
- **Step 2: Splitting: M=2M**
- **Step 3: Classification: the nearest neighbor rule**
- **Step 4: Codebook Updating: the new cluster center computation**
- **Step 5: Termination 1: overall distortion (D) -**

  If $|D_{current} - D_{previous}| \leq$ threshold, then Step 6

  If $|D_{current} - D_{previous}| >$ threshold, then Step 3

  Convergent condition

  $(D(l+1)-D(l))/D(l)$

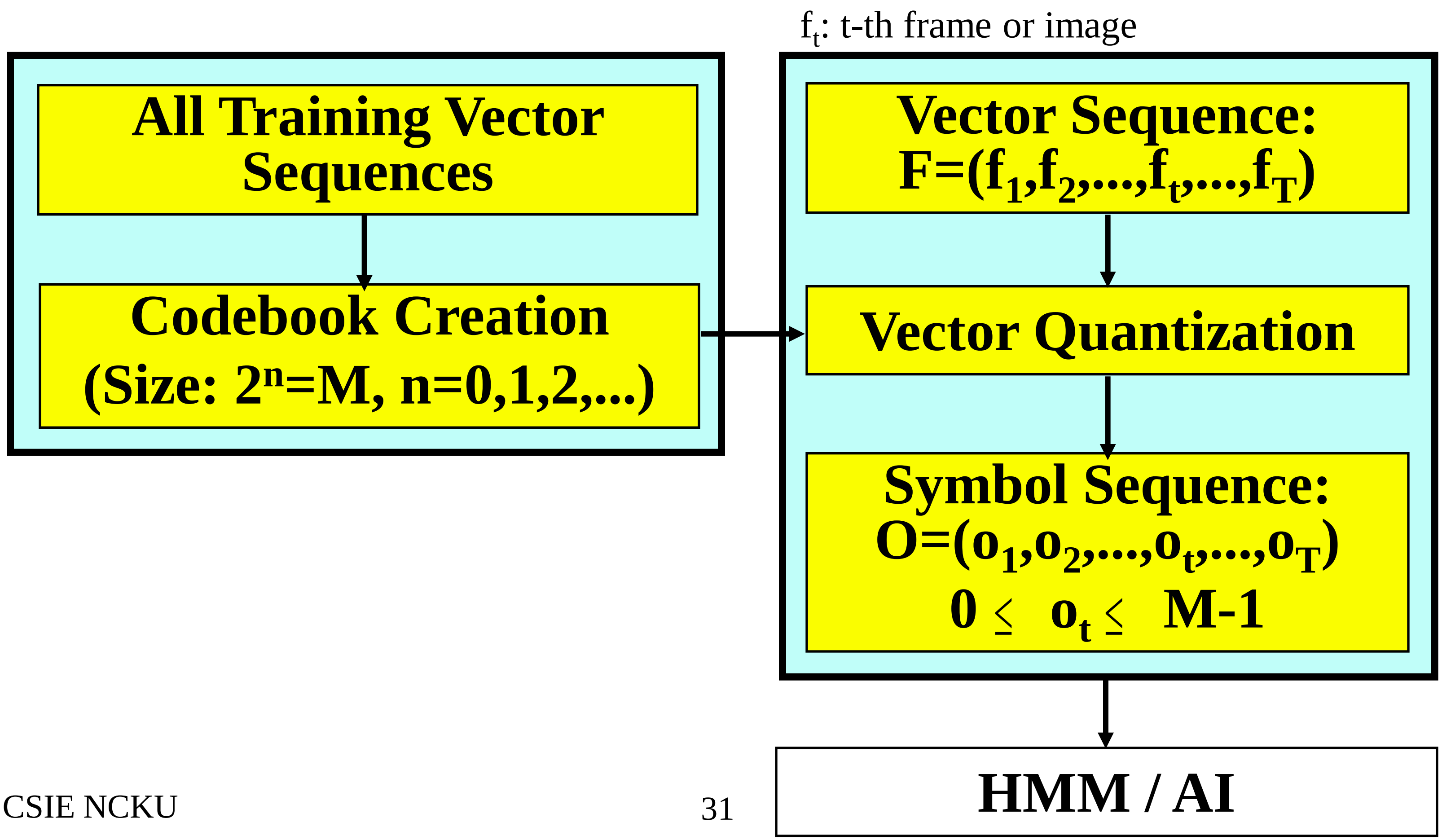
- **Step 6: Termination 2: M = the VQ codebook size ?**

  If 'Yes', then Stop   If 'No', then Step 2

- Once the final codebook is obtained, according to all training vectors by using this VQ algorithm, it is used to vector quantize each training and test feature (or motion) vector into a symbol value (codeword) for the preprocessing of the discrete HMM recognition process or Gaussian mixture model

Preprocessing of Hidden Markov Model: Vector Quantization

Vector quantization for encoding any vector sequence to a symbol sequence based on the codebook.



Preprocessing of GMM (+EM)

References

1. **Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantizer Design,"IEEE Transactions on Communications, Vol. Com-28, No. 1, January 1980.**