ASRSV – curs 10

Metode si tehnici de recunoastere a vorbirii (2)

- **Sisteme şi metode utilizate în recunoaştere**

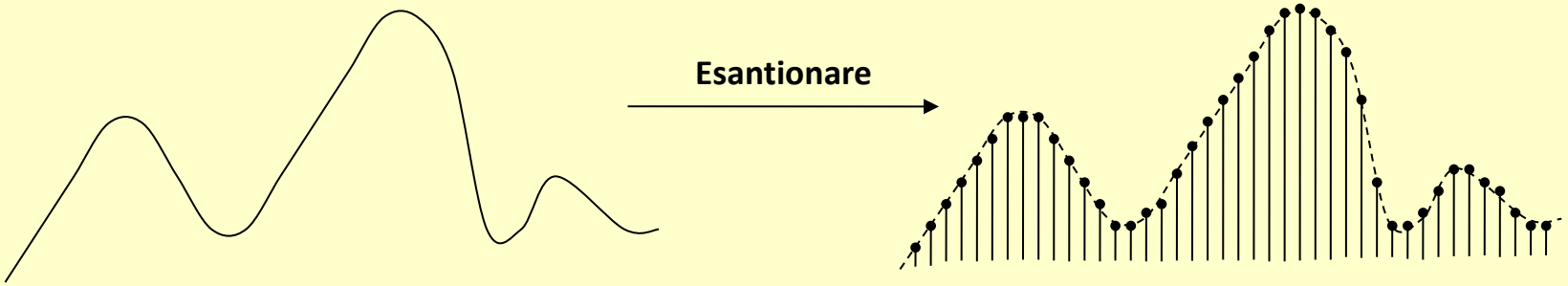
- Abordările de recunoaştere a vorbirii/vorbitorului în raport cu metodele şi tehnicile folosite se pot grupa în mai multe familii :

  - sisteme bazate pe spectrograme
  - sisteme bazate pe metodele programării dinamice (DTW)
  - sisteme care folosesc cuantizarea vectorială (VQ)
  - sisteme bazate pe modele Markov ascunse (MMA) - HMM
  - sisteme bazate pe mixturi Gaussiene GMM
  - sisteme utilizând reţelele neuronale NN
  - sisteme bazate pe metoda TESPAR FANN
  - sisteme folosind metode algebrice/statistice
  - combinaţii ale lor.
  - Abordari IA

Cuantizarea vectorială (VQ)

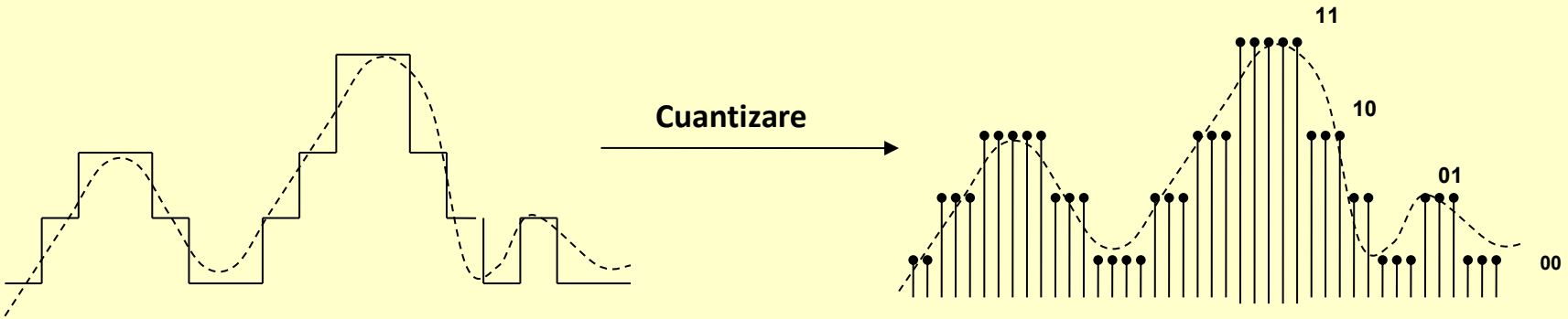
Introducere

- Majoritatea semnalelor in natura sunt analogice



Amplitudine-continua, Timp-continuu

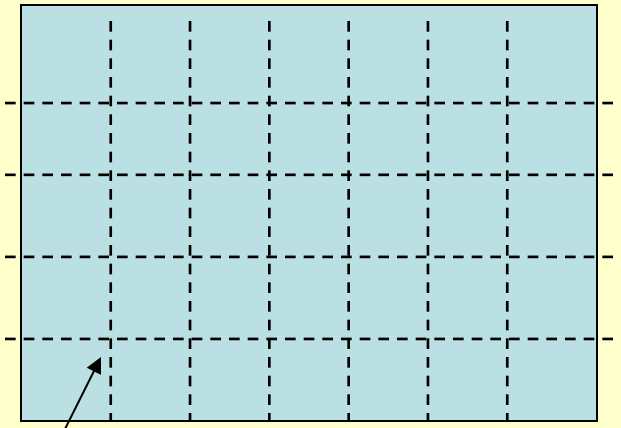
Amplitudine-continua, Timp-discret



Amplitudine- discreta, Timp-discret

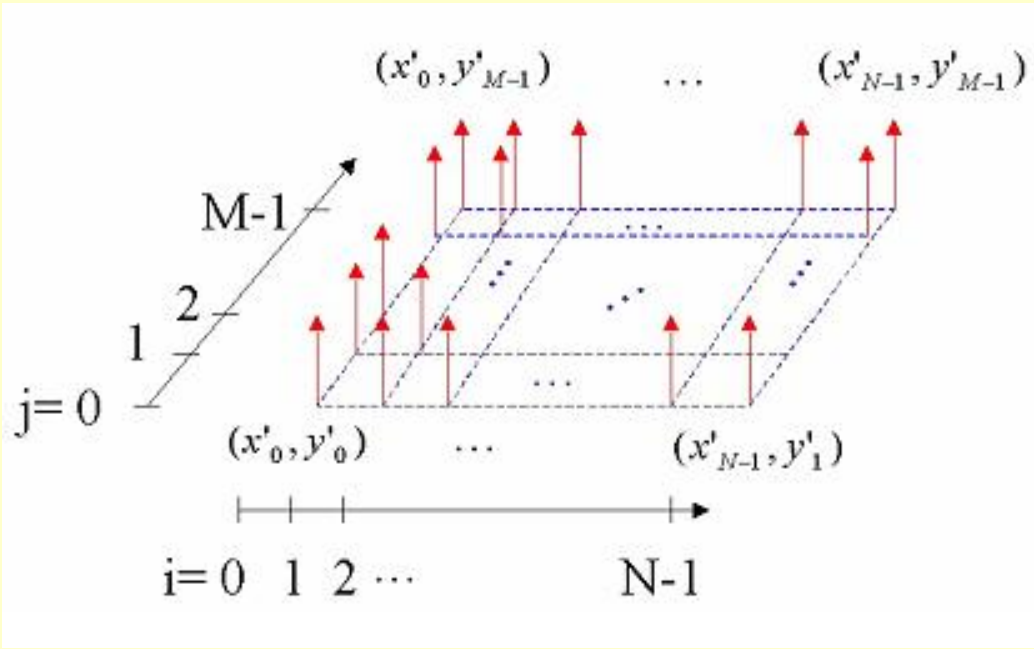
Exemplu pe imagine.

Esantionarea

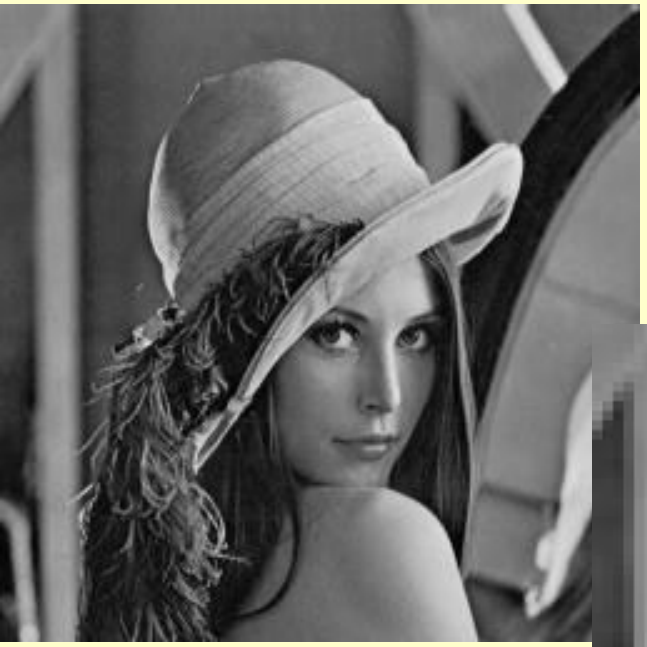


• Se esantioneaza valorile imaginii in nodurile unei grile regulate pe planul imaginii.

• Un pixel (element de imagine) la coordonatele (i, j) este valoarea intensitatii imaginii in punctul grilei indexat cu număr întreg de coordonate (i, j).

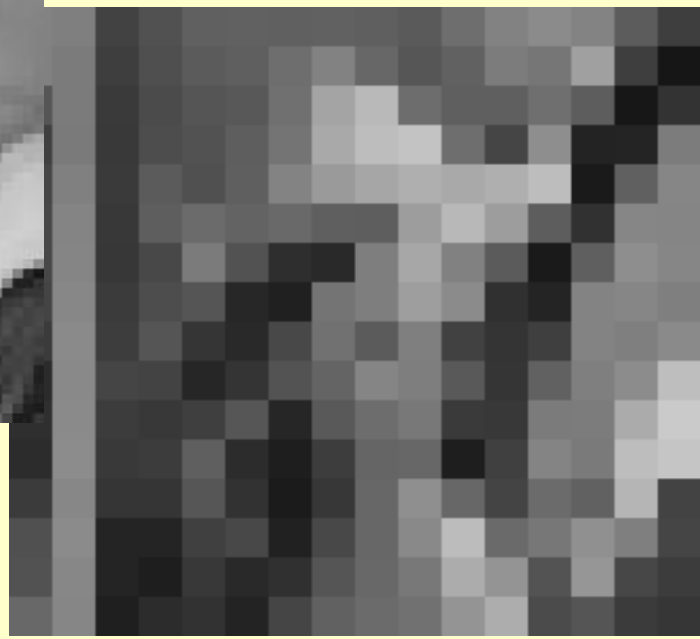


Exemple de esantionare



256x256

64x64

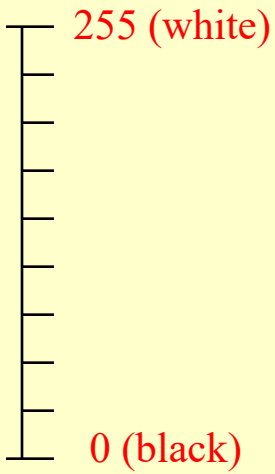


16x16

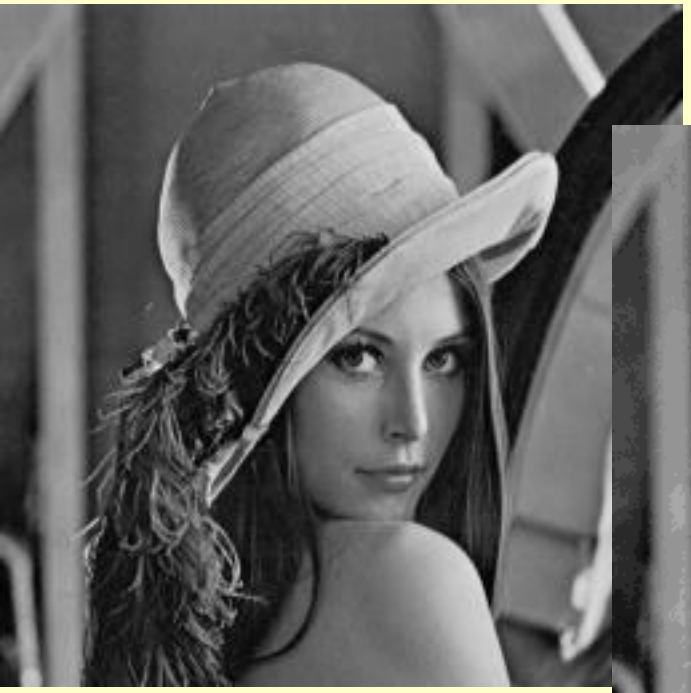
Cuantizarea scalara (SQ)

- Este un proces de transformare a unei imagini reale esantionate luand numai un numar finit de valori distincte.

- Fiecare valoare esantionata a imaginii pe o scara de 256 nivele de gri este reprezentata pe 8 biti.



Exemple de cuantizare



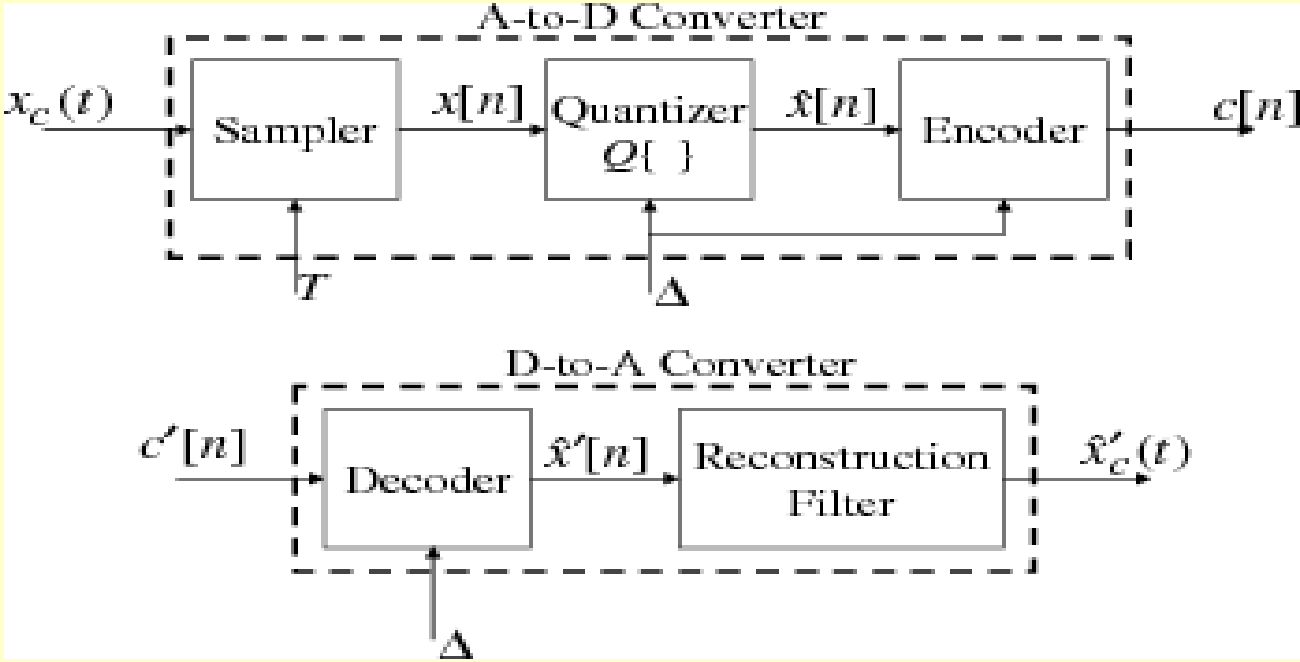
8 bits / pixel

4 bits / pixel

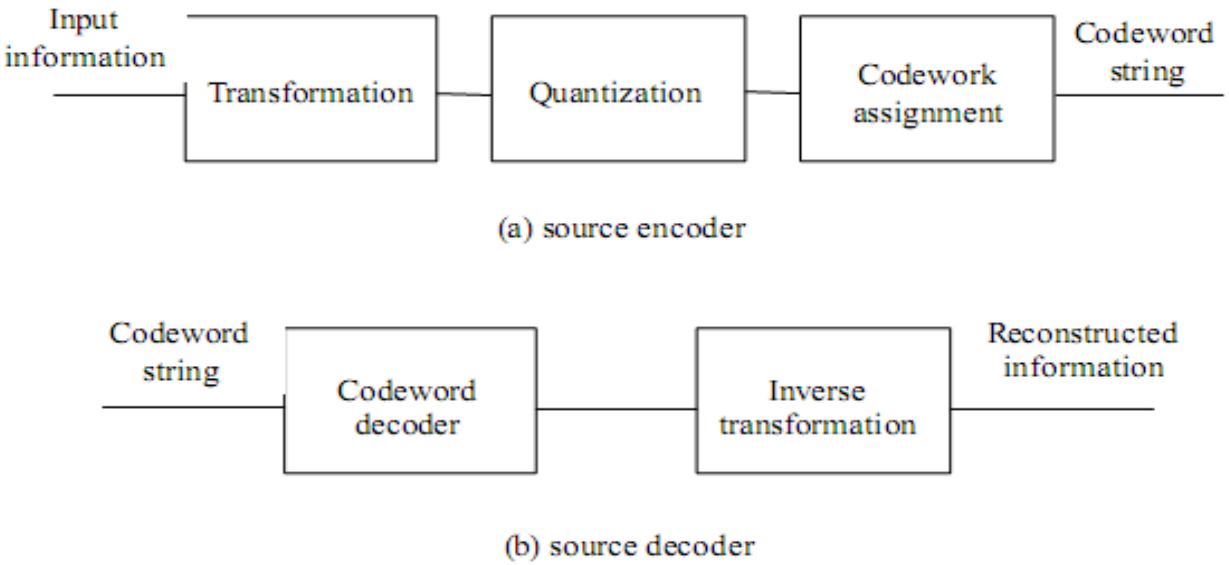
2 bits / pixel

Cuantizarea in sistemele de prelucrarea semnalelor

- **Conversia A-D:**



- **Codarea**



- **Problemele cuantizarii:**
  - Cuantizarea este un proces ireversibil
  - Cuantizarea este o sursa de pierdere a informatiei (erori)

- Cuantizarea este un nivel critic in compresia semnalelor, cu efecte asupra:
  - Distorsiunii semnalelor reconstruite
  - Debitului binar al codorului

- Obiectivul – cuantizor optimal sau de inalta calitate:
  - Distorsiuni medii cat mai mici (D).
  - debit cat mai redus (R).

- Cuantizarea: a)Uniforma

  b) Neuniforma

  c) Adaptiva

  d) Predictiva

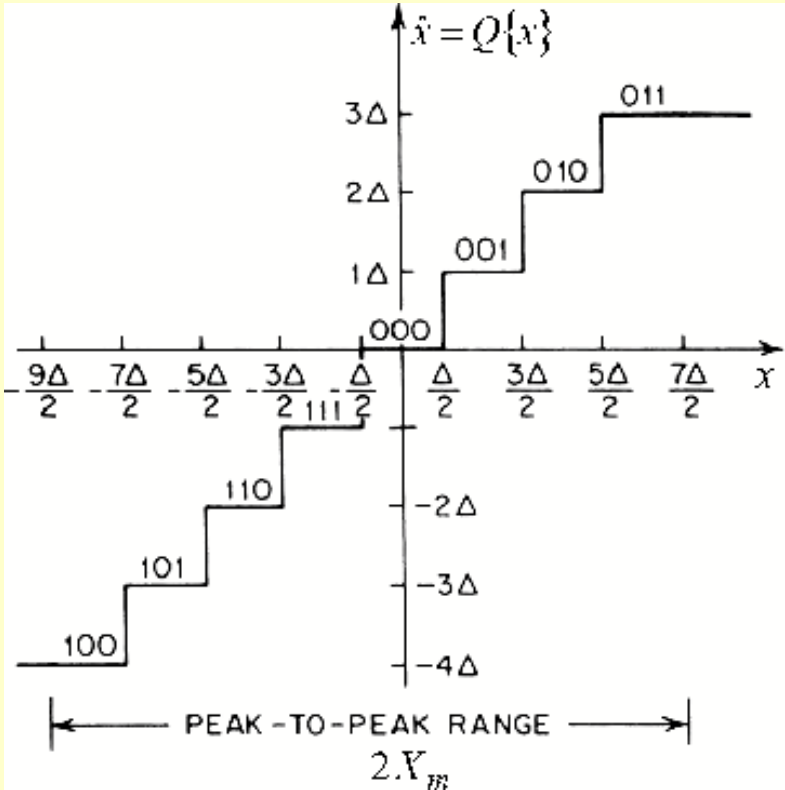
• a) Cuantizarea uniforma

- Toate regiunile de decizie a cuantizarii sunt de dimensiune egala :

$$\Delta = 2X_m / 2^B = 2X_m / N$$

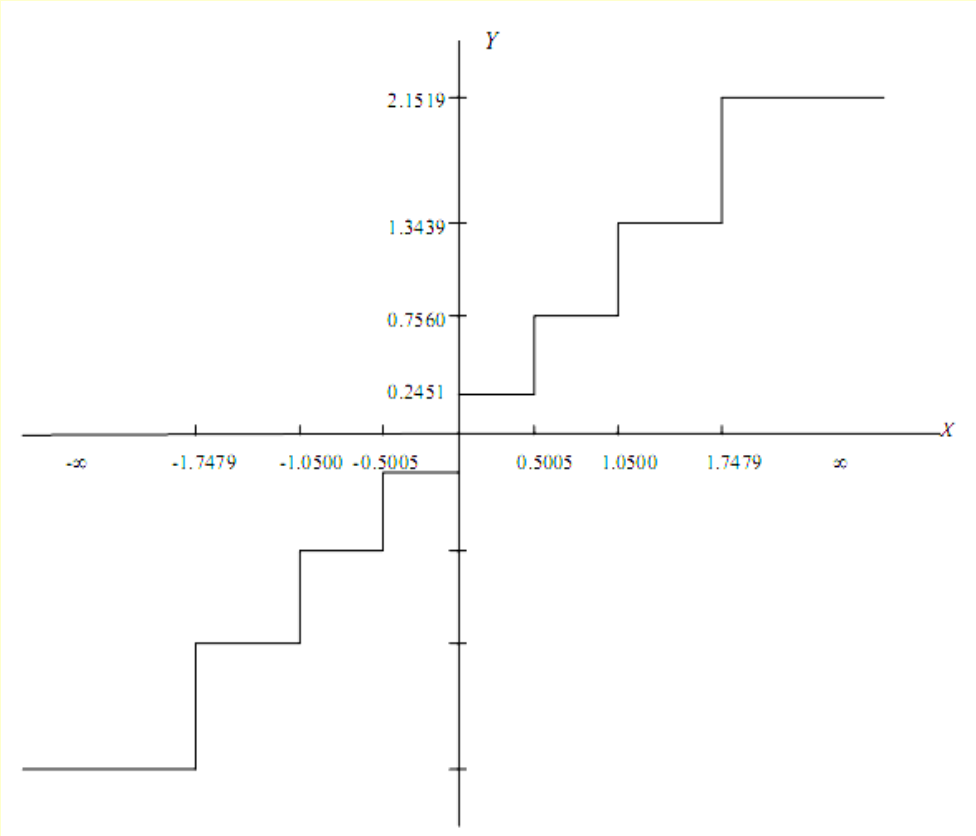
- Nivelele de cuantizare sunt la jumatatea nivelelor de decizie

- La N regiuni de cuantizare, se folosesc B=$\log_2$(N) biti pentru a reprezenta fiecare valoare cuantizata.



b) Cuantizare Neuniforma

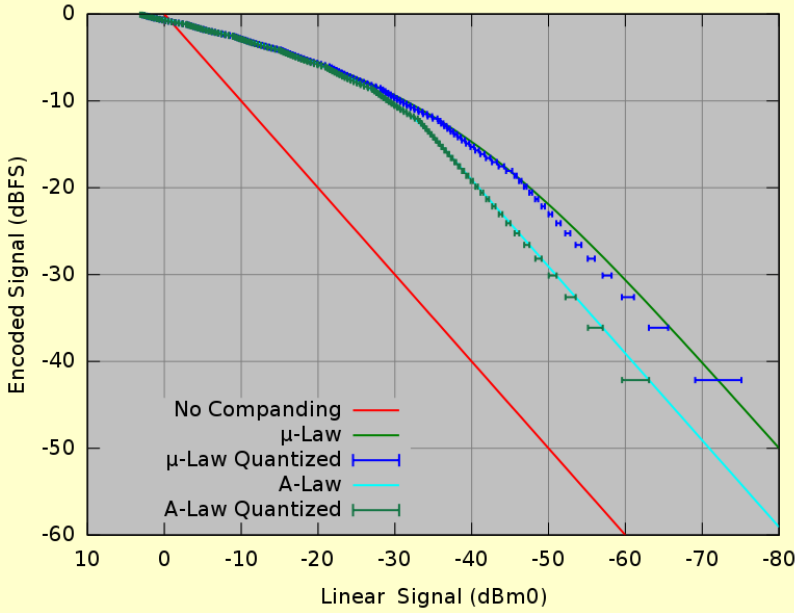
- Semnalele cu Fct Densit Probab. de tip (Gaussian, Laplacian etc.).
- Semnalele de intrare de amplitudine mica trebuie cuantizate mai precis
- Semnalele de intrare de amplitudine mare se cuantizeaza mai grosier
- Algoritmul Lloyd-Max: algoritm numeric iterativ ptr. cuantizorul optimal

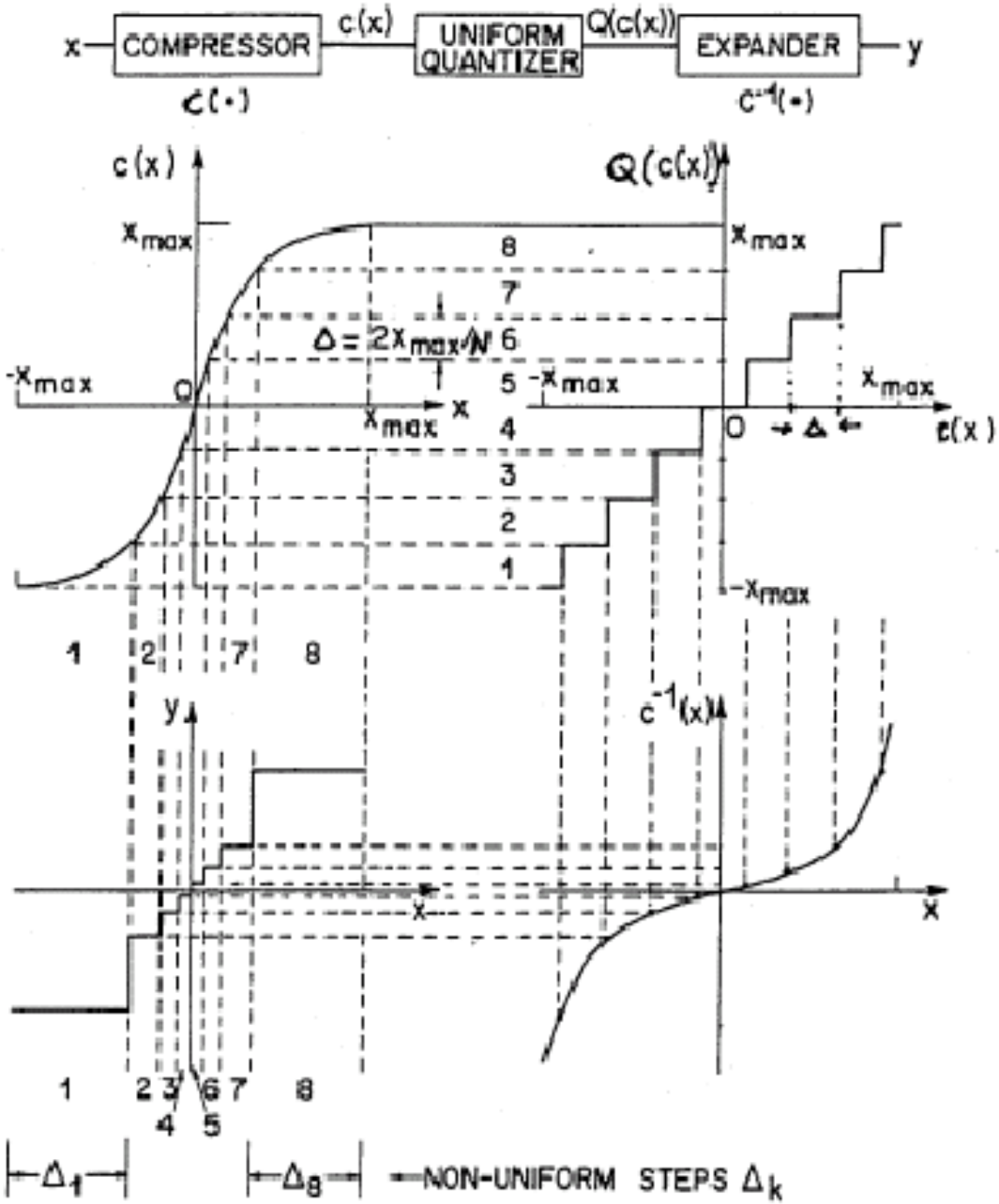


Exemplu – G.711

- G.711 - standard ptr. compandare audio, un codor al formei de unda a vocii, cunoscut si ca PCM logaritmic.
- Frecventa de esantionare 8 kHz
- PCM uniform foloseste 12-16 bits/sample
- G.711 foloseste cuantizare neuniforma cu 8 bits/sample; compander (a-law/μ-law) + cuantizor uniform
- debit binar 64 kbit/s (8 kHz x 8 bits/sample).
- Intarzierea tipica algoritmica este 0.125 ms

Exemplu – G.711





(From Jayant & Noll)

c) Cuantizarea Adaptiva

- Cuantizarea Uniforma : regiunile de cuantizare fixe
- Cuantizarea neuniforma : regiunile de cuantizare variabile, dar constante in timp
- Cuantizarea Adaptiva :
  - Regiunile de cuantizare variaza in timp
  - Cuantizorul se adapteaza la modificarile statisticii semnalului de intrare
  - Furnizează distorsiune mai mică - performanţă mai bună
  - Costul: intarziere prin procesare suplimentara si cerinte suplimentare de stocare

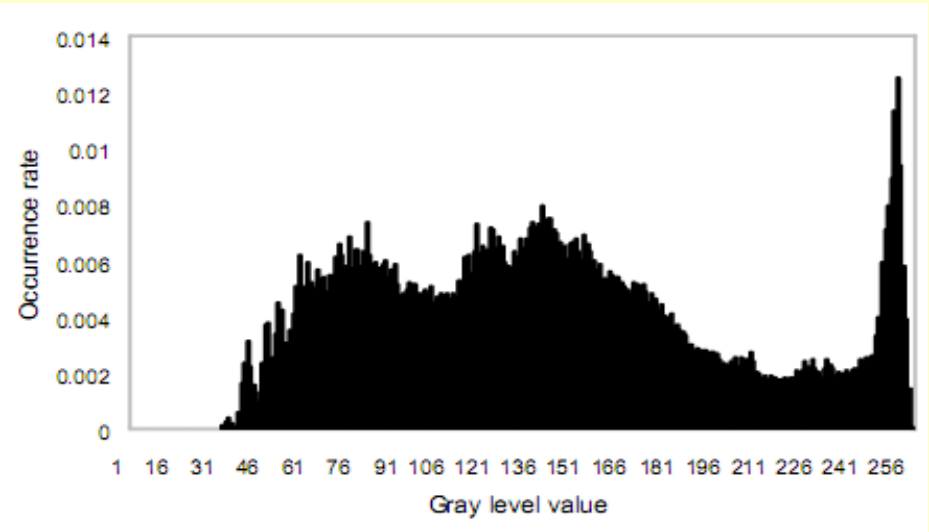
d) Cuantizarea Predictiva

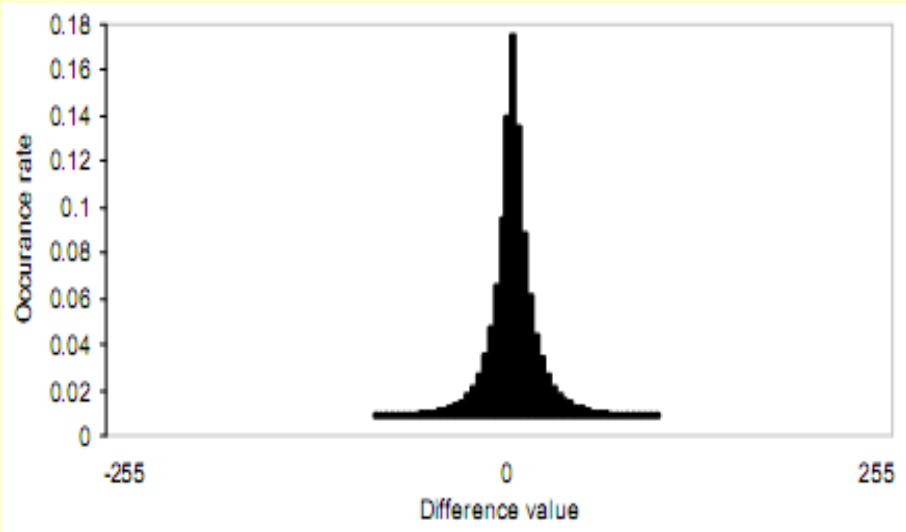
- în locul codarii directe a semnalului, se codifică diferenţa dintre semnalul însasi şi cel prezis prin tehnica de codare diferenţiala;

- este cunoscut si sub numele de codor de predicţie

- Cuantizorul necesita gama de cuantizare mai redusa

- Ex: histograma imagine în comparaţie cu o histogramă a imaginii diferenţa





Cuantizare Vectoriala

- Este o operaţie care generalizează cuantizarea scalară şi constă în reprezentarea unui vector X, ale cărui M componente sunt valori reale şi continue, printr-un vector $(x \in R^M)$, aparţinând unei mulţimi finite:

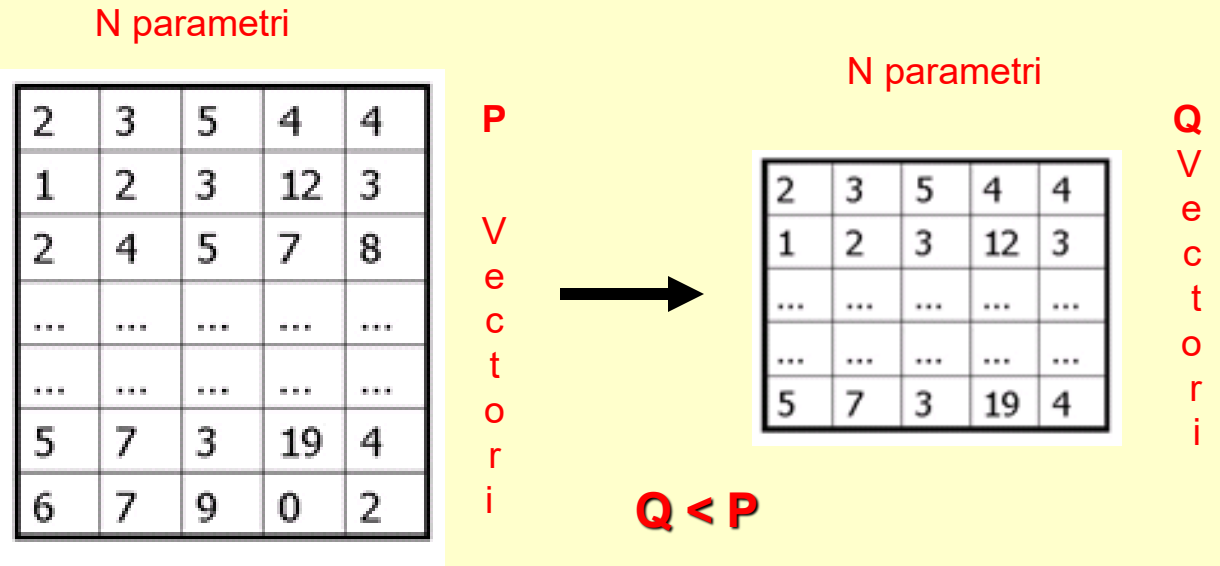
$$y = \{y_i \in R^M, i = 1,2,\ldots K\}$$ de vectori care alcătuiesc dicţionarul.

$$y = q(x), \qquad y \in Y$$

y_i - cuvânt de cod (centroid, codeword) ,

k - mărimea dicţionarului (numărul de nivele de cuantizare)

- Scopul cuantizarii vectoriale poate fi ***reducerea dimensiunii bazei de date***



2	3	5	4	4
1	2	3	12	3
2	4	5	7	8
...	...	...	...	...
...	...	...	...	...
5	7	3	19	4
6	7	9	0	2

2	3	5	4	4
1	2	3	12	3
...	...	...	...	...
...	...	...	...	...
5	7	3	19	4

- Alegerea unui dictionar constă în partitionarea spatiului M - dimensional al vectorului x în k regiuni $\{C_i\}$, $i = \overline{1,k}$, e clase (clustere), fiecărei clase fiindu-i asociat câte un cuvant de cod y_i .
Cuantizorul asignează cuvântul de cod y_i dacă x se află în regiunea C_i :

$$q(x) = y_i$$

- La realizarea cuantizării x→y apare o măsură de distorsiune (distanţă) d(x,y) care se defineste între x si y si care de obicei se măsoară prin distanta euclidiană:

$$d(x,y) = \sum_{i=1}^{M} (x_i - y_i)^2$$

Distorsiunea totală medie se calculează prin relaţia:

$$D = \lim_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} d(x(n), y(n))$$

Un cuantizor este *optim dacă distorsiunea totală este minimizată* pe toate cele k nivele de cuantizare.

•Sunt două condiţii pentru optim :

1)cuantizorul să fie realizat pe baza distorsiunii minime sau a regulii de selectie a celui mai apropiat vecin:

$$q(x)=y_i \text{ ddacã } d(x,y_i)\leq d(x,y_j)\ ,\ \forall\ j\neq i\ ,\ j=1,...k$$

2) fiecare cuvânt de cod yi se alege astfel încât să se minimizeze distorsiunea medie în regiunea Ci; centroidul pentru fiecare regiune depinde de modul de definire a măsurii de distorsiune:

$$y_i=\frac{1}{n_i}\sum_{x_i\in c_i} x_i\ ,\qquad i=1,...k$$

•Scopul urmărit la un sistem de codare este:

- minimizarea distorsiunii medii pentru un debit dat, sau
- reducerea debitului pentru o distorsiune acceptată;
- În cazul recunoaşterii vorbirii, obiectivul urmărit este reducerea volumului de calcul printr-o compresie de date.

• Problema majora a VQ este complexitatea computationala ridicata.

Solutia : algoritm eficient de cautare corecta a centroidului pentru vectorul de intrare.

• Rezolutia Cuantizorului:

$$R = \frac{\log_2 N}{k}\left[bits/sample\right]; N = 2^{kR}$$

$$\mathrm{K}\uparrow \Rightarrow \mathrm{R}\downarrow \Rightarrow \mathrm{D}\uparrow \text{ si}$$

$$\mathrm{K}\uparrow \Rightarrow \mathrm{R}\uparrow \Rightarrow \mathrm{D}\downarrow$$

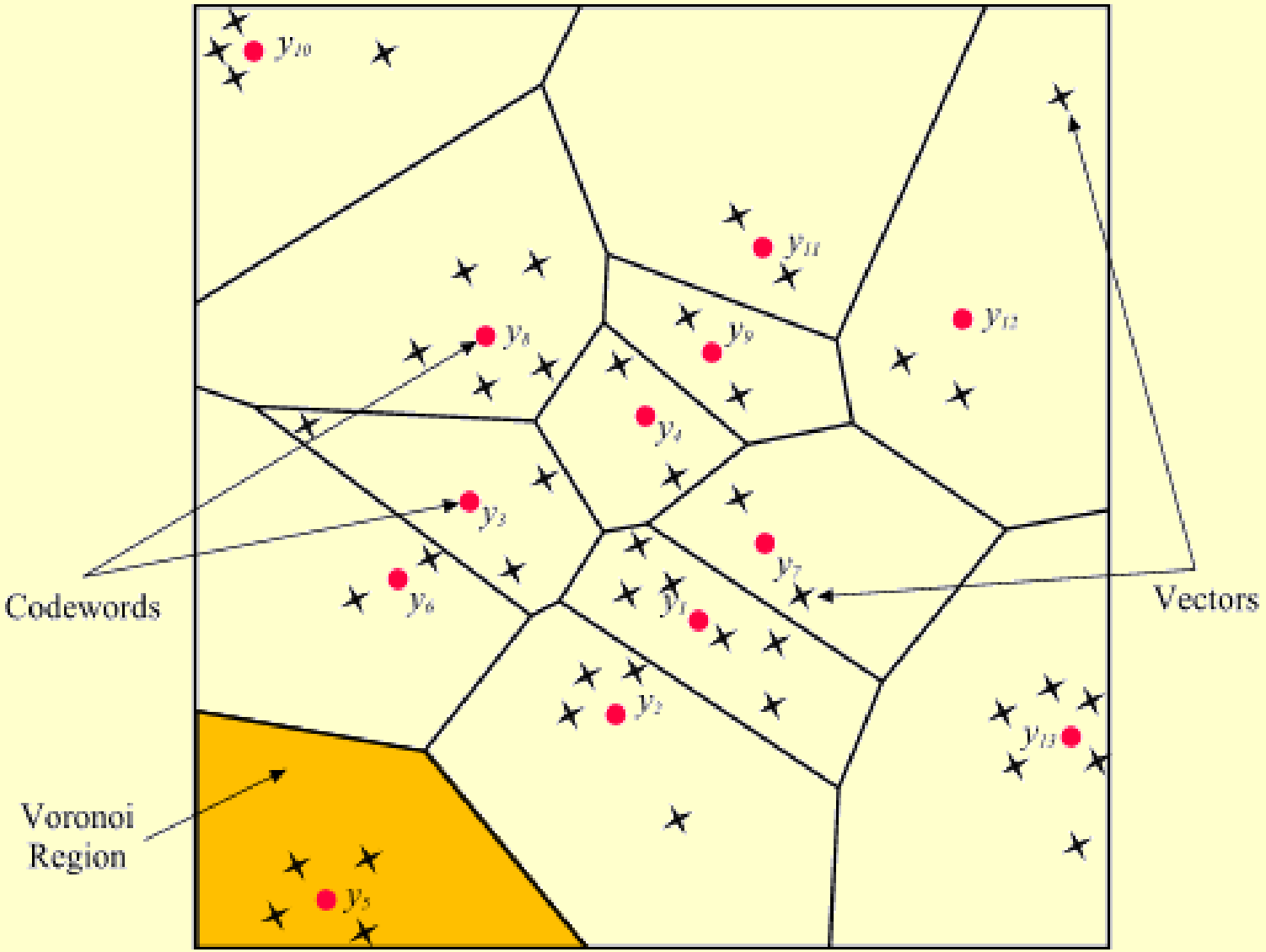
unde K- dimensiunea vectorilor, N- dimensiunea dictionarului, D- distorsiunea

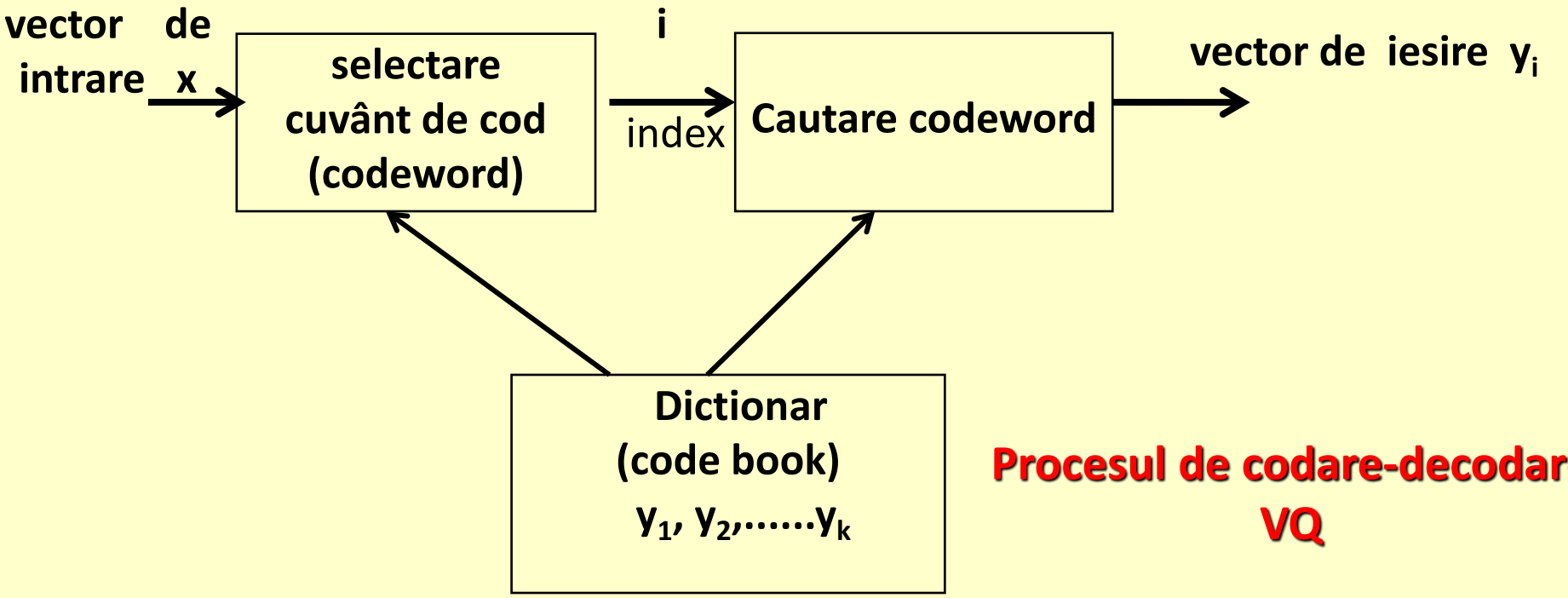
• Asociat fiecarui cuvant de cod, y_i, este regiunea celor mai apropiati vecini numita regiune Voronoi, definita de :

$$V_i = \left\{x \in R^k : \left\|x - y_i\right\| \le \left\|x - y_j\right\|, \forall\ j \ne i\right\}$$

• Setul de regiuni Voronoi partitioneaza intreg spatiul R^k

Ex. de regiuni Voronoi bidimensionale (k=2) ptr. o sursa cu FDP neuniforma: Vectorii de intrare sunt marcati cu x, iar cuvintele de cod (codewords) cu cercuri rosii; regiunile Voronoi sunt separate de liniile de delimitare.





Procesul de codare-decodare VQ

Cod vector

Imagine

100	100	110	110
80	80	90	90
110	110		
100	100		

X=(100,100,80,80)

Codebook

index				
1	0	0	10	20
2	10	10	10	10
:			:	
k	100	100	90	90
:			:	
Nc	255	255	200	200

⇨ Transmis k

Problema Determinarii Dictionarului de VQ

- Pentru o secventa de antrenament compusa din T= *M >10*N* de vectori sursa, de dimensiune *k* (*M* suficient de mare pentru a captura proprietatile statistice ale sursei) :

$$\mathrm{T} = \{x_i\}_{i=1}^{M}$$

- dat fiind numarul de vectori de cod: $N = 2^{kR}$

- Sa se afle un dictionar $Y = \{y_i\}_{i=1}^{N}$ si o partitie, $V = \{v_i\}_{i=1}^{N}$

- Pentru care sa rezulte distorsiunea medie minima:

$$\mathrm{D_{avg}} = \frac{1}{\mathrm{Mk}} \sum_{\mathrm{m=1}}^{\mathrm{M}} \left\| \mathrm{x_m} - \mathrm{Q(x_m)} \right\|^2 ; \mathrm{Q(x_m)} = \mathrm{y_n} \text{ daca } \mathrm{x_m} \in \mathrm{v_n}$$

Algoritmi pentru obţinerea dicţionarului

- Algoritmul "cu prag“
- Algoritmul LLOYD (k-means)
- Algoritmul LBG (Linde- Buzo-Gray)

Algoritmul “cu prag”

- este un algoritm simplu şi rapid care nu dă un număr fixat de clase .
- fie distanța "d" pe care o folosim şi $\{x_1, x_2 ..... x_k\}$ mulțimea vectorilor de antrenare
- Primul punct, x_1, formează singur o clasă i el va fi reprezentantul clasei, $C_1 = x_1$
- Pe parcursul funcționării clasarea punctului x_i va avea următorul efect:
- dacă distanțele $d(x_i, c_j)$ între x_i şi clasele deja formate c_j (între x_i şi reprezentantul c_i, centroidul său) sunt toate superioare unui prag dat “p” se creează o nouă clasă al cărui singur element $c_i = x_i$, dacă x_i se atribuie clasei celei mai apropiate se reactualizează eventual reprezentantul său.
- Algoritmul se opreşte când i=N.

```
Begin
     {x1}→C1 , x1→y1 , k=1
     do   i=2,N
          j←Argmin 1≤ k ≤ K  d(xi,yk)
      if d(xi,Cj)>p  atunci k=k+1, {xi}→Ck   yk←xi ,
          else xi se adaugă Cj , recalculez. Yj
     end if
     end do
End
```

Dezavantaje:
- sensibil la ordinea de sosire a datelor, distanţa folosita şi pragul joacă un rol primordial
- numărul de clase nu este controlabil cu precizie, fiind dependent de prag

Avantaje:
- simplu
- permite tratarea unui volum mare de date în timp de calcul rezonabil
-este posibilă întreruperea antrenării şi obţinerea unui dicţionar şi apoi refolosirea şi mărirea lui în cazul în care apar noi date

Reprezentantul unei clase poate fi actualizat sau nu, pentru aceasta existând două strategii:
a) reprezentantul unei clase este un element care a servit la crearea ei (această tehnică garantează distorsiunea minimă, dar poate duce la clase dispersate)
b) reprezentantul se actualizează cu fiecare nou vector adăugat la clasă - poate fi de exemplu centrul de gravitaţie al clasei - dar există riscul creării de clase alungite care se pot limita printr-un prag relativ redus.

Algoritmul LLOYD (k-means)

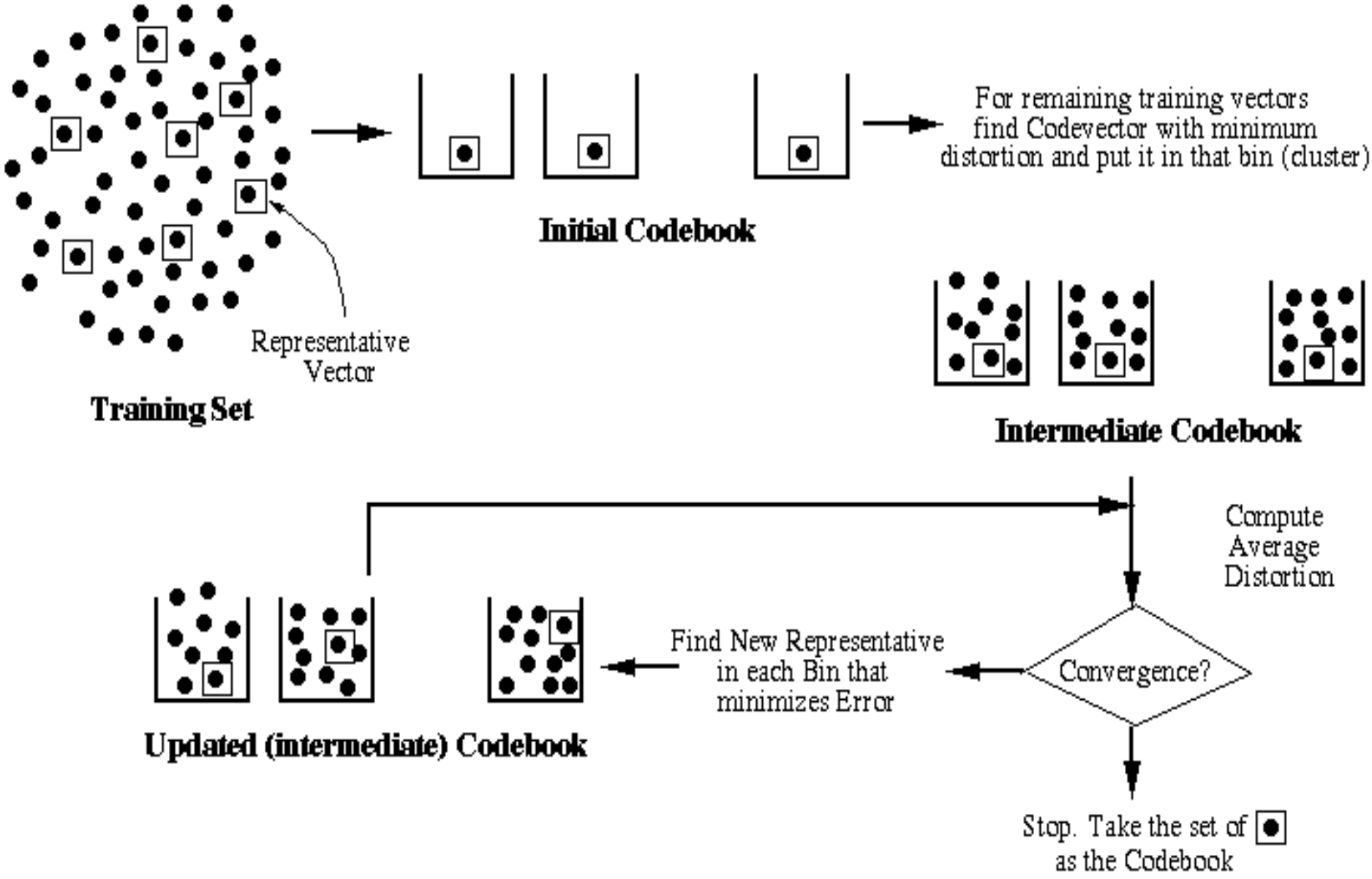
• Acest tip de algoritm a fost studiat şi generalizat pentru *clasificare automată şi recunoaştere a formelor sub numele "k-means*" şi sub o formă mult mai generalizată ca metodă numită "nori dinamici".

• Plecând de la un dicţionar dat a priori, se construiesc o serie de dictionare având acelasi număr de centroizi şi diminuând distorsiunea globală la fiecare etapă, iar algoritmul se opreste când se consideră că am atins o convergentă dorită.

• Nimic nu asigură că optimul găsit este global, un alt dicţionar de plecare putând duce la o convergentă diferită.

PROPRIETATI:

- Dictionarul se modifica numai dupa prezentarea intregului set de date
- MSE scade la fiecare iteratie
- Riscul de a se bloca intr-un minim local este ridicat
- Dictionarul final depinde de cel initial


Representative Vector
Training Set
Initial Codebook
For remaining training vectors find Codevector with minimum distortion and put it in that bin (cluster)
Intermediate Codebook
Compute Average Distortion
Convergence?
Find New Representative in each Bin that minimizes Error
Updated (intermediate) Codebook
Stop. Take the set of as the Codebook

a) Initializarea : m=0, m - indicele de buclare

$\{x(n)\}$, n=1,...,N - set antrenare .Se alege dictionarul initial de dimensiunea k:

$$\left\{\mathbf{Y}_1^{(o)}, \mathbf{Y}_2^{(o)}, \ldots\ldots, \mathbf{Y}_k^{(o)}\right\}$$

b) Clasificarea

- se clasifică setul de vectori de antrenare $\{x(n)\}$, $1 \leq n \leq N$ în clase $\{C_i(m)\}$ pe baza regulii proximităţii maxime (a vecinului cel mai apropiat)

$$\mathbf{x \in C_i\,(m)\ \ ddacă\ d[x,Y_i(m)] \leq d[x,Y_j(m)] \quad \forall\ j \neq i} \qquad (1)$$

unde d reprezintă o distantă adecvată

c) Determinarea centroizilor

m=m+1 şi se actualizează centroizii în fiecare cluster (clasă) prin recalcularea lor din vectorii de antrenare : $Y_i(m)$ = centroiod $(C_i(m))$, i=1,...,k

$$Y_i^{(m)} = \frac{1}{card\left\{c_i^{(m)}\right\}} \sum_{x \in c_i^{(m)}} x \qquad (2)$$

d) **Calculul erorii globale:**

$$D(m)= \frac{1}{N}\sum_{j=1}^{M}\sum_{x\in c_j(m)} d(x, Y_j) \qquad (3)$$

e) **Verificarea condiţiei de sfârşit :**

Dacă $\frac{D(m) - D(m-1)}{D(m)} \leq \varepsilon$ (4) sau $\mathbf{m=m_0}$ atunci este gata,

dacă nu, atunci se revine la b).

Pentru determinarea centroidului se pot folosi si alte relatii în locul lui(2).
Fiecare clasă poate fi reprezentată de axa de inertie Δc_i :

$$Y_i(m) = inf\, d(x,y)_{y\in\Delta\varpi} \qquad (5)$$

sau o clasă se poate reprezenta prin p puncte (y_1, y_2, ..., y_p) care o minimizează:

(6)

$$y_i(m) = \frac{1}{p} \sum_{p\in[1,p]} d(x, y_p)$$

Cum se initializeaza algoritmul Lloyd?

1. Aleator in spatiul vectorilor de intrare
2. Primii Q vectori de date x^i - Se folosesc primii 2^R vectori (ca vectori centroid) in secventa de antrenare ca si dictionar initial.
3. Q vectori x^i selectati aleator din secventa de antrenament si distantati intre ei. (uneori aceasta abordare este numita generarea aleatoare a codurilor)
4. Cresterea setului initial – aplicam algoritmul cu prag pina ajungem la dimensiunea dictionarului
5. Vecinii pereche cei mai apropiati: primul dictionar e format din toate punctele $\mathbf{x}^i$; – cei mai apropiati 2 centroizi se contopesc (centru de greutate); se repeta procedura pana ajugem la dimensiunea dorita

Algoritmul LBG (Linde- Buzo-Gray)

- Algoritmul Lloyd nu genereaza decât un dictionar optimal local, alegerea centroizilor initiali fiind importantă.

-Algoritmul LBG propus ne plasează în metrica euclidiană şi se creează un dictionar de $N=2^p$.

-Pasii algoritmului sunt urmatorii:

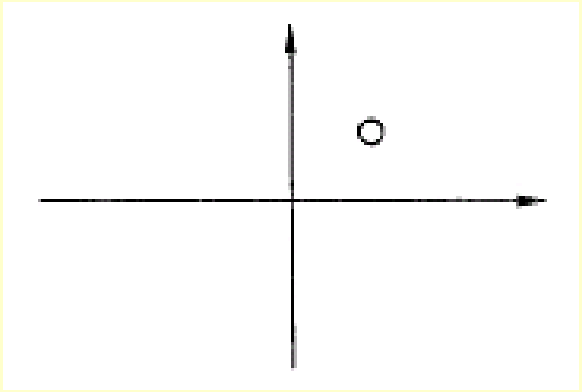
1. **Iniţializarea**
Se alege un singur centru de gravitaþie al mulþimii de antrenament, Y_0, k=0

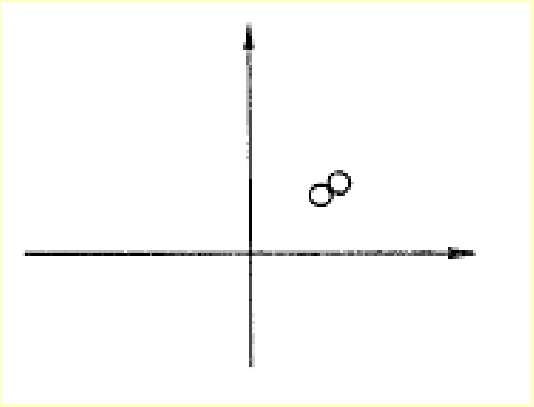
2. **Perturbarea centroizilor** Y_j (2^k) cu o micã valoare, transformând fiecare centroid Y_j în $Y_j+\varepsilon$ şi $Y_j-\varepsilon$, unde ε este un vector dat de normã micã

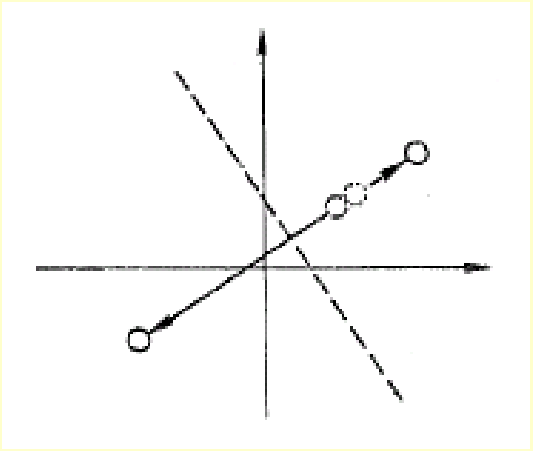
3. **Convergenţa** - se aplicã algoritmul Lloyd cu cei 2^{k+1} centroizi astfel constituiţi ca dicţionar de plecare

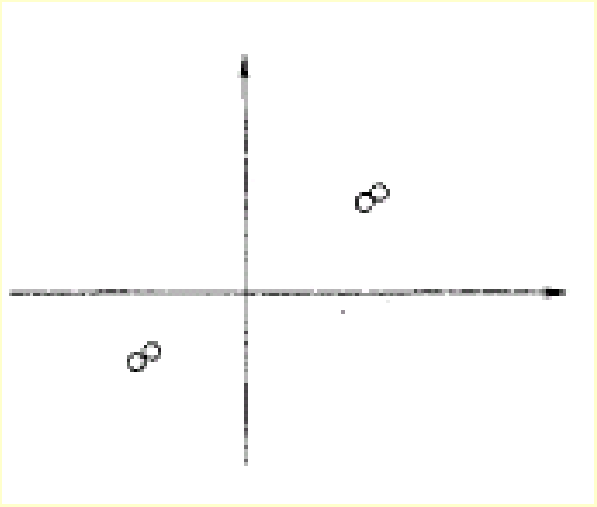
4. **Condiţie sfârşit** - k=k+1 , dacã k>k0 : stop, dacã nu, se revine la 2.

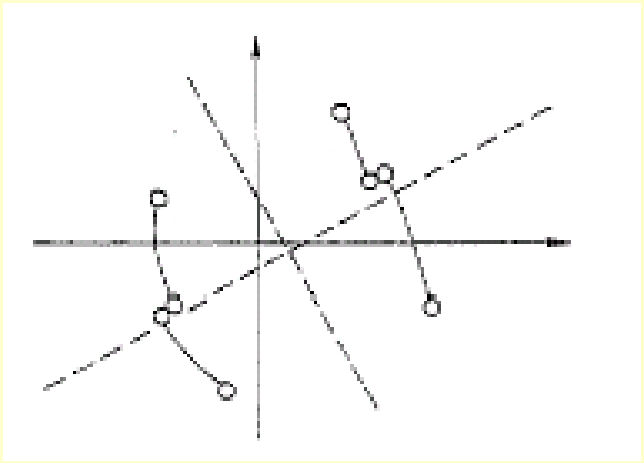
- **LBG VQ** (http://www.data-compression.com/vqanim.html)











Metoda LBG consolidată (Enhanced LBG)

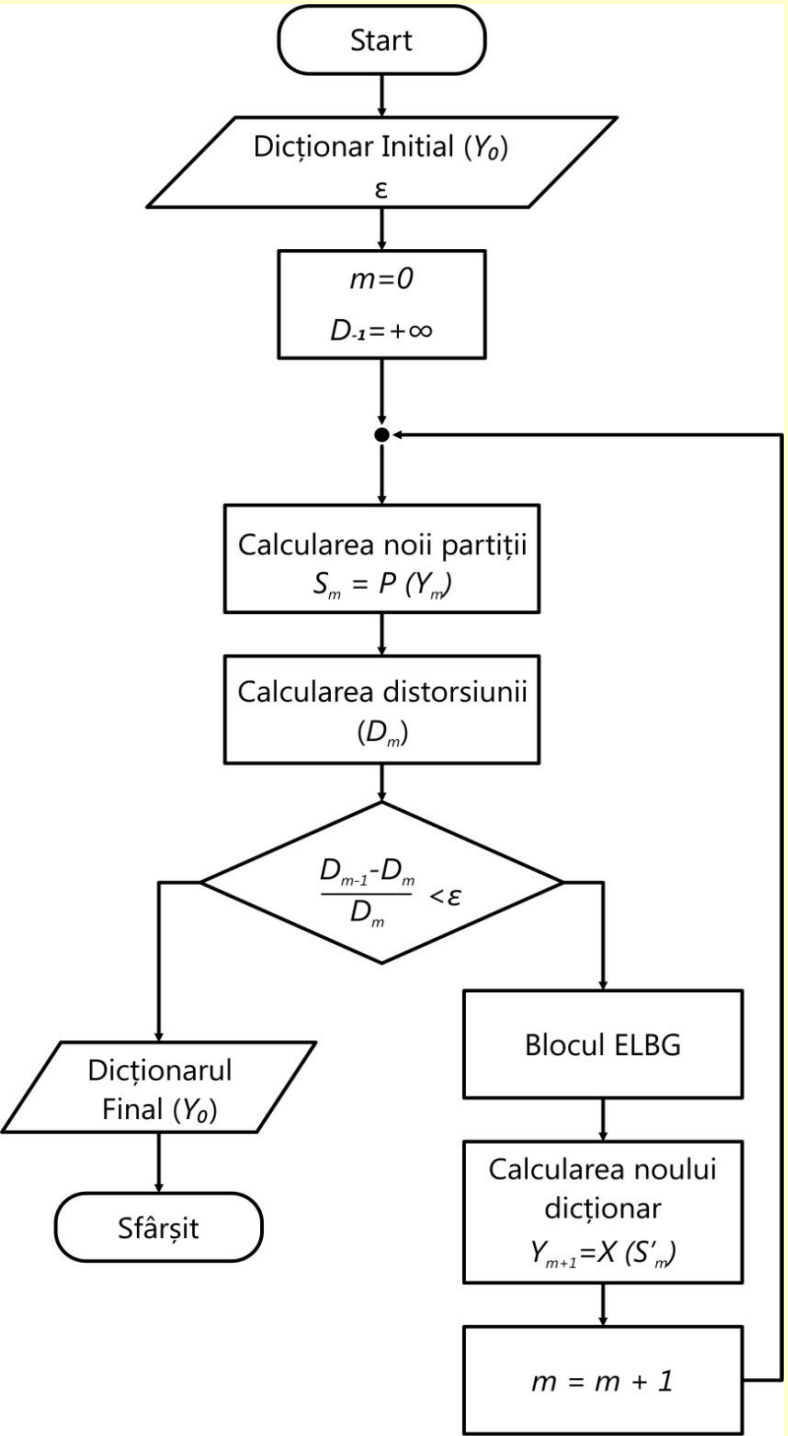
-are scopul de a soluţiona dezavantajele metodei clasice LBG
-Se introduce o nouă noţiune, care poartă numele de “utilitatea unui cuvânt de cod”
-După evaluarea utilităţii cuvintelor de cod, se pot identifica cele cu o utilitate scăzută.
-Această informaţie este utilă pentru pasul următor, deplasarea cuvintelor de cod. Se încearcă deplasarea tuturor cuvintelor de cod de utilitate scăzută, care se află în preajma celor de utilitate crescută.
- Pentru un cuantizor vectorial optim, conform teoremei lui Gersho, fiecare celulă are aceiaşi contribuţie la distorsiunea totală şi astfel indicele de utilitate este 1 pentru toate celulele.
- Partea principală a algoritmului ELBG constă în secvenţa de executare a tentativelor de deplasare a cuvintelor de cod (SoCA’s – Shifting of Codewords Attempts)

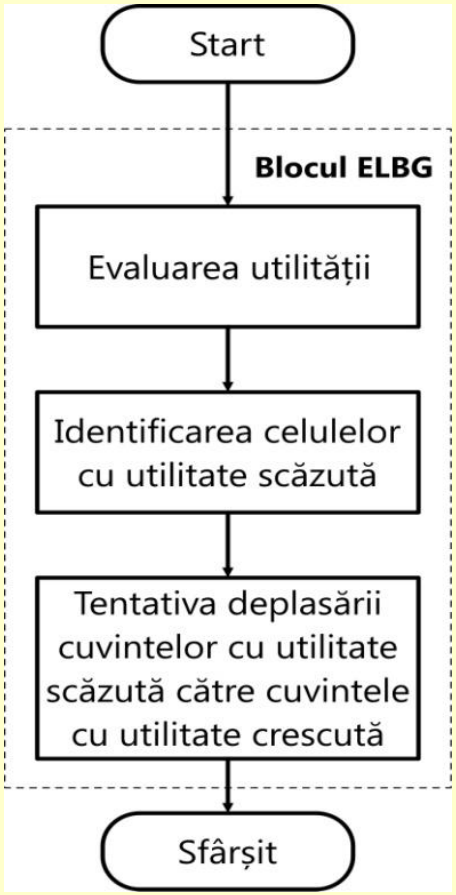
- Pentru început, se calculează distorsiunea medie pentru toate celulele:

$$D_m = \frac{1}{N}\sum_{i=1}^{N} D_i$$

- Se defineşte o măsură a utilităţii , care depinde de distorsiunea celulei i, D_i şi normalizată cu valoarea medie D_m .

$$U_i = \frac{D_i}{D_m} \quad , \quad i=1,N$$





Organigrama algoritmului ELBG

Modele MARKOV Ascunse (HMM)

- Bazele teoretice ale lanturilor Markov se cunosc de peste 100 de ani

- Metoda DTW se făcea o comparatie de tip formă-formă, fiind nevoie de o normalizare neliniară a esantioanelor (dilatare-contractare), pentru calculul distantei dintre forma de test si model ;
- la metodele stochastice ***distanta este înlocuită cu probabilităti*** calculate în etapa de antrenare;

- Forma semnalului vocal este considerată ca un ***semnal continuu , observabil în timp în anumite puncte de observatie;***

- Modelul descrie aceste stări cu ajutorul probabilităților de tranzitie dintr-o stare în alta si de probabilitatea de observare din fiecare stare;

- Compararea constă în căutarea în acest graf de stări a drumului cel mai probabil corespunzător unei succesiuni de evenimente observate în lantul de intrare ;

- Aceste metode sunt robuste si fiabile datorită existentei unor buni algoritmi de antrenare;

- Recunoasterea este rapidă pentru că metodele au în general putine stări si volumul de calcul este redus.

MODELE STOCHASTICE

- Un model stochastic este un proces aleator care poate schimba starea s_i, i =1, ... N, aleator în momentele t=1,2,... T.
- In fiecare stare este o variabilă aleatoare $X(t)= s_i$, $t \in [1,T]$, care ia valori într-o multime de observatie care se face în timpul analizei fenomenului
- Evolutia sistemului este o succesiune de tranzitii de stare plecând de la $X(1)=S_1$: $\mathbf{s_1 \rightarrow s_2 \rightarrow \ldots\ldots\ldots \rightarrow s_T}$

- Legea de evolutie a sistemului constă în a avea lantul de tranzitii $s_1, s_2, \ldots s_m$, $\forall m \leq T$ cu probabilitatea $P(s_1,s_2,\ldots s_T)$, care se defineste din aproape în aproape, astfel :

$$P(s1,s2,\ldots sT) = P(s1,s2,\ldots sT\text{-}1) \bullet P(sT \mid s1,s2,\ldots sT\text{-}1) =$$
$$= P(s1,s2,\ldots sT\text{-}2) \bullet P(sT\text{-}1 \mid s1,s2,\ldots sT\text{-}2) \bullet P(sT \mid s1,s2,\ldots sT\text{-}1) =$$
$$= P(s1) \bullet P(s2 \mid s1) \bullet P(s3 \mid s2s1) \bullet \ldots\ldots P(sT \mid s1,s2,\ldots sT\text{-}1) . \qquad (1)$$

- Un proces stochastic verifică proprietatea Markov dacă :

$$\forall t \quad P(X(t)=si \mid X(t-1)=sj, X(t-2)=sk, ...) = P(X(t)=si \mid X(t-1)=sj) \quad (2)$$

- Dacă notăm $X(t)=q_t$, formula devine :

$$\forall t \qquad P(qt=si \mid qt-1=sj, qt-2=sk, ...) = P(qt=si \mid qt-1=sj) \qquad (3)$$

de unde $P(q1,q2,.....qT) = P(q1) \bullet P(q2 \mid q1) \bullet ..... P(qT \mid qT-1)$.

-Evolutia este total determinată de probabilitatea initială si de cele de tranzitie succesivă

- un proces Markov astfel definit este de ordinul întâi

- Un model Markov este stationar dacă :

$$\forall t,k \qquad P(qt=si \mid qt-1=sj) = P(qt+k=si \mid qt+k-1=sj) \qquad (4)$$

- se defineste matricea de probabilitate de tranzitie $A= [a_{ij}]$:

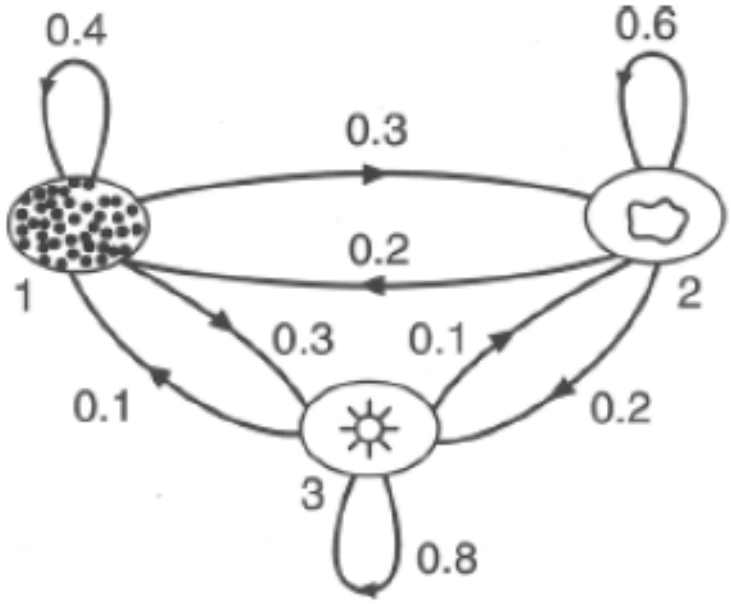
$$aij = P(qt=sj \mid qt-1=si) \quad , \quad i,j = 1, ... ,N \qquad (5)$$

si $\forall\, i,j \;,\; aij \geq 0 \;,\; \forall i \quad \sum_{j=1}^{N} a_{ij} = 1$.

Problema:

Considerand ca azi vremea e insorita (S3), care este probabilitatea (conforma modelului ca timpul in urmatoarele 7 zile sa fie “soare-soare-ploaie-ploaie-soare-noros–soare” ?

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$



Solutie: Definim secventa de observatii, O, ca:
O = { S3, S3, S3, S1, S1, S3, S2, S3 }
Si calculam ca P(O|Model).

$P(O|\text{Model}) = \text{P[S3, S3, S3,S1, S1, S3, S2, S3|Model]}$

$$P(O \mid \text{Model}) = P\left[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid \text{Model}\right]$$

$$= P\left[S_3\right] P\left[S_3 \mid S_3\right]^2 P\left[S_1 \mid S_3\right] P\left[S_1 \mid S_1\right]$$

$$\cdot P\left[S_3 \mid S_1\right] P\left[S_2 \mid S_3\right] P\left[S_3 \mid S_2\right]$$

$$= \pi_3 \left(a_{33}\right)^2 a_{31} a_{11} a_{13} a_{32} a_{23}$$

$$= 1(0.8)^2 (0.1)(0.4)(0.3)(0.1)(0.2)$$

$$= 1.536 \cdot 10^{-04}$$

$$\boxed{\pi_i = P\left[q_1 = S_i\right], \quad 1 \le i \le N}$$

Modele Markov ascunse
(MMA = HMM - Hidden Markov Models)

- MMA este un model stochastic particular si prezintă o formă (semnal, fenomen...) prin două siruri de variabile aleatoare : unul ascuns si celălalt observabil.

Sirul ascuns ***corespunde succesiunii de stări*** $q_1,q_2,.....q_T$, Q(1:T), unde q_i iau valori prin multimea de stări $\{s_1,s_2,....s_N\}$

Sirul observabil ***corespunde succesiunii observatiilor*** $O_1,O_2,....O_T$ notate cu O(1:T), unde O_i sunt functie de timp si se găsesc intr-o multime de M simboluri observabile $\{v_1,v_2,...v_M\}$.

- Diferenta între un model Markov observabil si unul ascuns se poate sesiza în ex. următor, de lansare a unei monede si explicarea succesiunii de observatii fată /stemă

Modelul aruncarii monedei

-Se efectueaza o serie de experimente de aruncare a monedei;

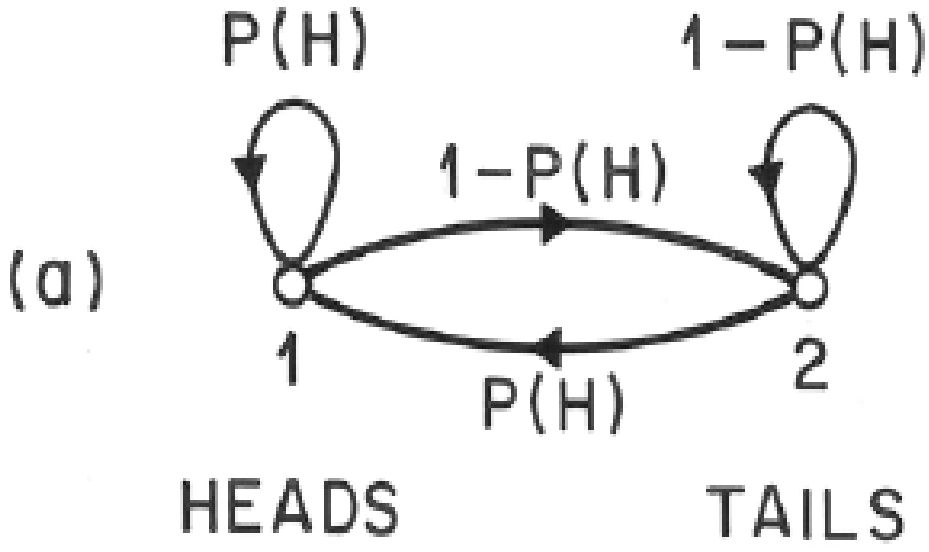
-Numarul de monede este necunoscut, se stiu doar rezultatele experimentelor. Seria de observatii este:

$$O = O_1 O_2 O_3 \ldots O_T = HHTTTHTTH \ldots H$$

- **Problema: sa se determine un model MMA** care sa explice secventa de observatii.

Cerinte:

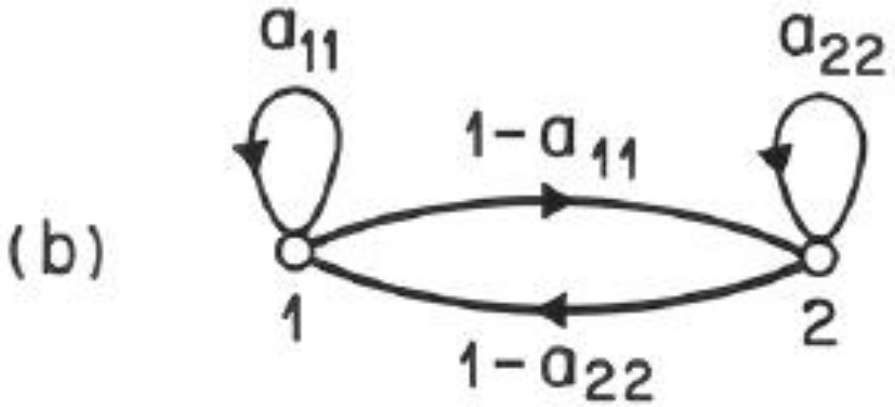
1. Care sunt starile modelului?
2. Cate stari sunt necesare?
3. Care sunt probabilitatile de tranzitie a starilor?



Modelul cu o moneda

MM observabil

O = H H T T H T H H T T H . . .
S = 1 1 2 2 1 2 1 1 2 2 1 . . .



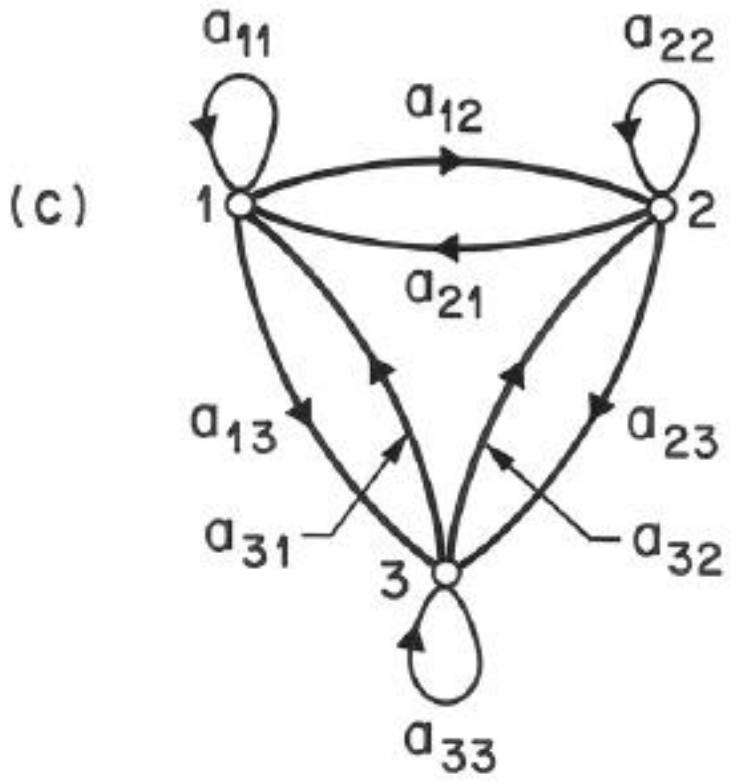
$P(H) = P_1$ $P(H) = P_2$

$P(T) = 1-P_1$ $P(T) = 1-P_2$

Modelul cu 2 monede

MM Ascuns

O = H H T T H T H H T T H . . .
S = 2 1 1 2 2 2 1 2 2 1 2 . . .



	STATE 1	2	3
P(H)	P_1	P_2	P_3
P(T)	$1-P_1$	$1-P_2$	$1-P_3$

Modelul cu 3 monede

MM Ascuns

O = H H T T H T H H T T H . . .

S = 3 1 2 3 3 1 1 2 3 1 3 . . .

EX. Consideram o reprezentare a experimentului aruncarii monedei printr-un model MMA (HMM) (model λ) – varianta cu 3 monede avand probabilitatile de tranzitie si cele initiale egale cu 1/3:

	Stare 1	Stare 2	Stare 3
P(H)	0.5	0.75	0.25
P(T)	0.5	0.25	0.75

Observam secventa : **O=H H H H T H T T T T**

a. Care este secventa de stari cea mai probabila? Care este probabilitatea de observare a secvenţei şi această succesiune de stari, cea mai probabila?

b. Care este probabilitatea ca secvenţa de observare sa provina în întregime din starea 1 (numai moneda 1)?

c. Considerand secventa de observatii: **O1= HTTHTHHTTH,** cum se modifica cerintele a, b?

SOLUTIE.

a) Avand sirul de obs. **O=HHHHTHTTTT**,cea mai probabila secventa de stari este aceea ptr. care probabilitatea obs. individuale este maxima.
Astfel ptr. fiecare H, cea mai probabila stare este S2 si pentru fiecare T cea mai probabila este starea S3.
Astfel secventa de stari cea mai probabila este:
S= S2 S2 S2 S2 S3 S2 S3 S3 S3 S3

	Stare 1	Stare 2	Stare 3
P(H)	0.5	0.75	0.25
P(T)	0.5	0.25	0.75

Probabilitatea de O si S (date de model) este:

$$P(O,S \mid \lambda) = (0.75)^{10}\left(\frac{1}{3}\right)^{10}$$

b) Probabilitatea ca observatiile sa provina numai din S1 adica:

S= S1 S1 S1 S1 S1 S1 S1 S1 S1 S1

$$P(O,\hat{S} \mid \lambda) = (0.50)^{10}\left(\frac{1}{3}\right)^{10}$$

	Stare 1	Stare 2	Stare 3
P(H)	0.5	0.75	0.25
P(T)	0.5	0.25	0.75

$$R = \frac{P(O,S \mid \lambda)}{P(O,\hat{S} \mid \lambda)} = \left(\frac{3}{2}\right)^{10} = 57.67$$

c) Fiind data secventa de obs. **O1=HTTHTHHTTH** avand acelasi nr. de stari ca si anterior H=T=5, rezultatele nu se schimba.

TEMA. Daca probabilitatile de tranzitie intre stari se modifica astfel:

$$a_{11} = 0.9, \quad a_{21} = 0.45, \quad a_{31} = 0.45$$
$$a_{12} = 0.05, \quad a_{22} = 0.1, \quad a_{32} = 0.45$$
$$a_{13} = 0.05, \quad a_{23} = 0.45, \quad a_{33} = 0.1$$

Care sunt noile rezultate pentru cerintele a-c ale problemei ?

Elementele unui MMA: $\lambda = (A,B,\Pi)$; se definesc următoarele elemente :

N - numărul de stări reprezentate prin $S = \{s_1, s_2, \ldots . s_N\}$, o stare la un moment dat este notată cu q_t ($q_t \in S$).

M - numărul de simboluri observabile în fiecare stare. Multimea de observatii este $V = \{v1, v2, \ldots . vM\}$

Un element O_t din V desemneaza un simbol observat la momentul t.

A - o matrice de probabilităti de tranzitie între stările modelului: a_{ij} - reprezentând probabilitatea cu care modelul evoluează din starea i spre j, în cazul general :

$a_{ij} = A(i,j) = P(q_{t+1}=s_j \mid q_t = s_i) \quad \forall i,j \in [1,N] , \forall t \in [1,T] , a_{ij} \geq 0$ si $\forall i,j , \sum a_{ij} = 1$.

B – o matrice de probabilităti de observare a simbolurilor în fiecare stare a modelului : $b_j(k)$ care reprezintă probabilitatea ca să observăm simbolul v_k când modelul se află în starea j :

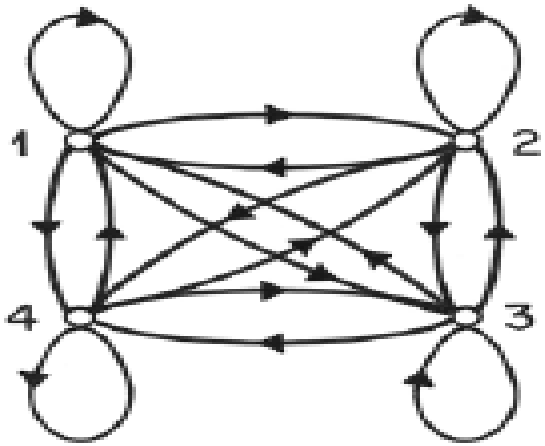
$$b_j(k) = P(O_t = v_k \mid q_t = s_j), \quad 1 \leq j \leq N \quad 1 \leq k \leq M$$

$$b_j(k) \geq 0 , \forall j,k \text{ si } \sum b_j(k) = 1$$

Π - o multime a densitătii de probabilitate initiale : $\Pi = \{\pi_i\}$, i= $\overline{1}$,... N , unde π_i reprezintă probabilitatea ca starea de pornire a modelului să fie i, astfel :

$$\pi_i = P(q_1 = s_i) \quad i=1,..N \text{ si } \pi_i \geq 0 , \sum \pi_i = 1.$$

Tipuri de MMA

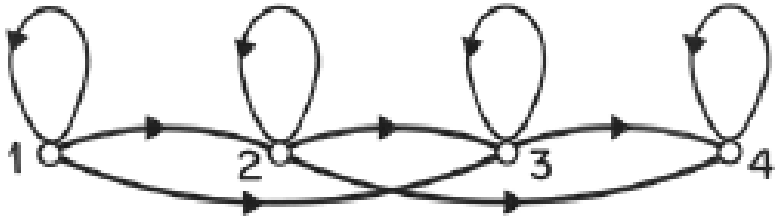


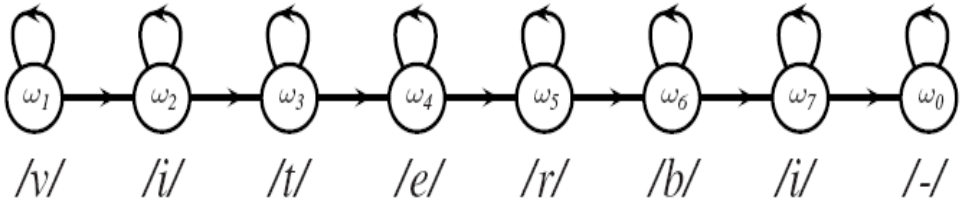
(a)

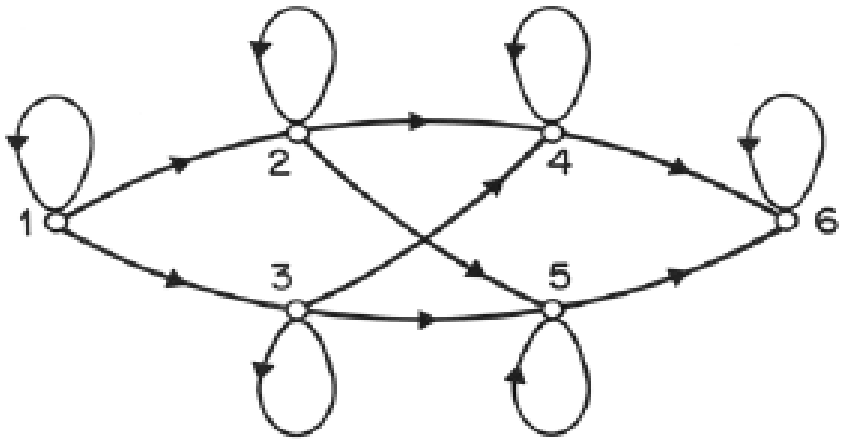
a) Ergodic

b) Stanga-dreapta

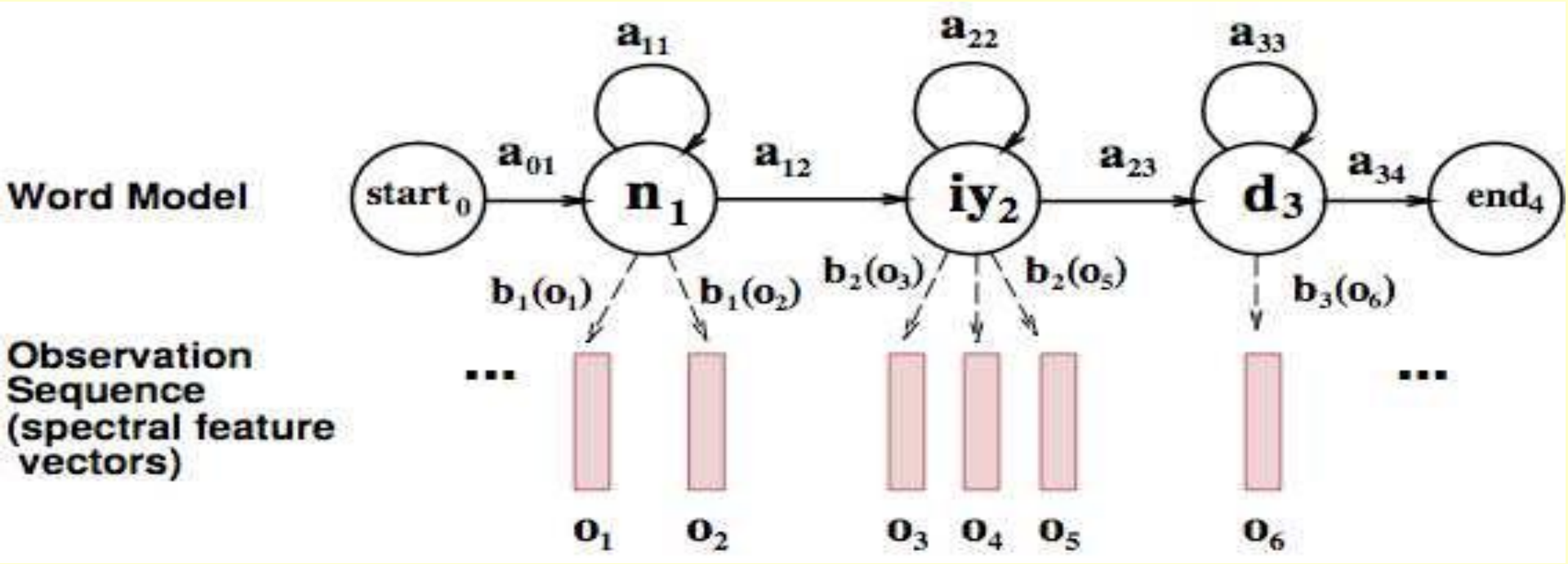
c) Mixt







(c)


Word Model
Observation Sequence (spectral feature vectors)
start 0
n 1
iy 2
d 3
end 4
a 01
a 11
a 12
a 22
a 23
a 33
a 34
b 1(o 1)
b 1(o 2)
b 2(o 3)
b 2(o 5)
b 3(o 6)
o 1
o 2
o 3
o 4
o 5
o 6

Procedură de generare a observatiilor

Un MMA se poate folosi ca o procedură de generare a observatiilor, iterativă cu patru etape :

1. Alegerea stării de plecare : t=1, se alege starea initială q1=si cu π_i.

2. Observatia în starea selectată : se alege observatia Ot=v_k cu $b_i(k)$;

3. Selectia stării următoare : se trece la starea următoare q_{t+1}=sj cu ajutorul lui a_{ij}.

4. Schimbarea efectivă de stare :
   Schimbăm pasul: t=t+1, dacă t<T se trece la 2 , dacă nu se termină.

Problemele de rezolvat la MMA

Sunt trei probleme de bazã ale unui MMA care trebuiesc solutionate :

1. Evaluarea probabilitãtilor de observare

Fiind data succesiunea de observatii: $O = O_1, O_2, ...O_T$ si modelul $\lambda=(A,B,\Pi)$, se pune problema determinării probabilitãtilor $P(O|\lambda)$, ca succesiunea observatiilor sã fie generatã de model.

-Solutia - **Algoritmul Forward-backward**

2. Problema descoperirii secventei de stãri ascunse

- Fiind datã succesiunea observatiilor $O=O_1,O_2,....O_T$ si modelul λ : sã se determine secventa de stãri, $Q= q_1,q_2,....q_T$, cea mai probabilã din care au rezultat observatiile O_t.

-Se obtin informatii legate de partea ascunsă a modelului (succesiunea de stãri).

-Se foloseste un criteriu optimal pentru descoperirea secventei de stãri

-Solutia – **Algoritmul Viterbi**

3. Problema antrenãrii

-Cum trebuie modificati parametrii modelului $\lambda=(A,B,\Pi)$ pentru a maximiza $P(O|\lambda)$ - reprezintã problema antrenãrii modelului.

-Solutia – **algoritmul Baum-Welch**

REFERINTE

•L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Vol.77, No.2, pp.257--286, 1989. Available from http://ieeexplore.ieee.org/iel5/5/698/00018626.pdf?arnumber=18626

- Hidden Markov Model Toolkit (HTK)
  http://htk.eng.cam.ac.uk/

- Hidden Markov Model (HMM) Toolbox for Matlab
  http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html