Cours Polytech'Orleans – 02 Décembre, 2005 :

L'Algorithme de Viterbi

Laurent Depersin – Philips Mobile Phones
Mise à jour: 02 Décembre, 2005

Table des matières

I. Introduction

Bref Historique

L'algorithme de Viterbi (VA) trouve ses origines dans un article parut en 1967 par Andrew J. Viterbi. Celui-ci proposait alors une solution optimale pour le décodage de *code convolutionnel*. Depuis ce temps, les applications de cet algorithme n'ont cessées de se développer. En témoigne aujourd'hui l'utilisation importante de cet algorithme dans le domaine des télécommunications.

Principe

Comme le filtre de Kalman, le VA traque de manière récursive les états d'un processus stochastique, cependant le processus sous-jacent n'est pas un processus Gaussien mais un processus de Markov. Ainsi, de la même façon que le filtre de Kalman offre une solution à l'estimation *analogique* d 'un système, le VA quant à lui se révèle particulièrement adapté à l'estimation numérique.

I. Introduction (2)

Définition

« L'algorithme de Viterbi (VA) est une solution optimale au sens du maximum de vraisemblance pour l'estimation d'une séquence d'états d'un processus de Markov à temps discrets et nombres d'états finis observés dans un bruit sans mémoire. »

Estimation d'un modèle statistique

Lorsque pour un phénomène donné il n'existe pas de modèle analytique ou numérique satisfaisant, une solution alternative consiste à construire un modèle statistique aussi proche que possible du modèle physique directement à partir d'observations.

Notons x la donnée mesurée, m le modèle général conjecturé (e.g. polynôme, modèle Gaussien, Laplacien, …) et φ l'ensemble des paramètres du modèle. L'objectif est donc d'identifier les valeurs des paramètres φ qui vont permettre d'expliquer au mieux les données observées x .

Une façon de trouver ces coefficients revient à chercher les valeurs de φ les plus probables sachant le modèle choisis et les données mesurées. Ce qui signifie que l'on cherche les valeurs de φ qui maximise $p(\varphi \,|x,m)$.

Estimation d'un modèle statistique (suite)

En utilisant la règle de Bayes sur la probabilités des causes, on réécrit le problème de la façon suivante, dite Maximisation A-Posteriori (MAP) :

$$\max_{\varphi} p(\varphi \mid \mathrm{x},\mathrm{m}) = \max_{\varphi} \frac{p(\mathrm{x} \mid \varphi,\mathrm{m})p(\varphi \mid \mathrm{m})}{p(\mathrm{x} \mid \mathrm{m})} = \max_{\varphi} \frac{p(\mathrm{x} \mid \varphi,\mathrm{m})p(\varphi \mid \mathrm{m})}{\int_{\varphi} p(\mathrm{x} \mid \varphi,\mathrm{m})p(\varphi \mid \mathrm{m})\mathrm{d}\varphi}$$

$$= \max_{\varphi} \frac{vraisemblance \times a\ priori}{évidence}$$

On voit que la probabilité à maximiser peut être factorisée en 3 termes de nature indépendante. La *vraisemblance* mesure la correspondance entre les données et le modèle prédit, modèle dont les coefficients sont calculés à partir d'un calcul d'erreur. L'*a priori* contient la connaissance a priori que l'on peut avoir sur la probabilité d'un ensemble de paramètres donnés. Enfin, l'*évidence* est la somme d'événements mutuellement exclusifs et collectivement exhaustifs qui sont censés représentés x.

Estimation par maximum de vraisemblance

Maintenant, si on suppose une distribution uniforme des paramètres (i.e. $p(\varphi \mid m)=const.$) la maximisation a posteriori est équivalente au Maximum de vraisemblance.

Soit:

$$\max_{\varphi} p(\varphi \mid \mathrm{x}) = \max_{\varphi} p(\mathrm{x} \mid \varphi)$$

Jusqu'en 1920 les statisticiens utilisaient comme méthode d'estimation la méthode dite des moments. Cette méthode est à la fois la plus immédiate et la plus intuitive puisqu'elle consiste simplement à prendre comme estimation du moment d'une population le moment correspondant de l'échantillon. Ainsi, on estime la moyenne μ d'une population par la moyenne d'échantillon que l'on note E(X) (i.e. espérance de X), et la variance σ^2 d'une population par la variance d'échantillon s^2 notée encore E(X-E(X)). Bien que dans beaucoup de cas simple cette méthode soit suffisante, elle s'avère tout à fait inadaptée pour des cas plus complexes. C'est donc pour résoudre ces derniers que R. Fisher introduit la notion d'estimation par maximum de vraisemblance qui outre sa capacité à résoudre de nouveau problème fournit un cadre beaucoup plus général à l'estimation statistique.

III. Modèle de Markov (1)

Définition: Processus (ou modèle) de Markov

Un processus de Markov est un processus doublement aléatoire composé d'un certain nombre de variables aléatoires $(q_1, q_2, q_3, ...)$ ayant une probabilité intrinsèque dite d'émission, et pour lesquelles on définit en outre une probabilité dite de transition. En plus, de ces deux probabilités on ajoute une probabilité d'être dans un état donné à l'instant $t=0$, on parle donc de probabilité initiale.

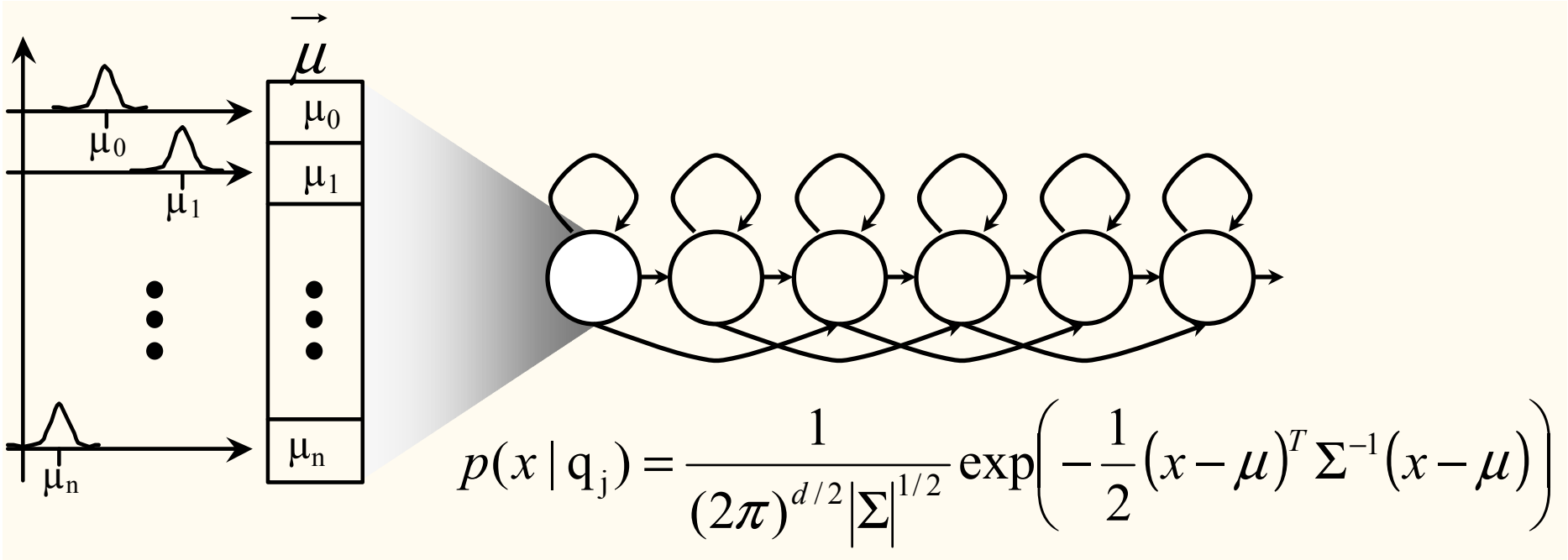
Par ailleurs, dans un PM la probabilité d'occurrence d'une variable dépend uniquement des variables précédentes. Un cas particulier important est celui des processus de Markov d'ordre 1, pour ceux-ci la probabilité d'occurrence d'une variable ne dépend que de la variable précédente.

Andrei Andreyevich Markov,
14 Juin 1856 (Ryazan), RU -
20 July 1922 (Petrograd), RU

Définition: Chaîne de Markov

C'est un processus de Markov pour lequel les variables aléatoires sont discrètes (par opposition à continues). On parle dans ce cas d'états.

Exemple: Chaîne de Markov suivant le modèle de Bakis – couramment utilisé en reconnaissance vocale. μ est un vecteur de paramètres représentant une tranche de signal stationnaire



III. Modèle de Markov (3)

Propriétés :

- **États observables :** $1, 2, 3, \ldots, N$
- **Séquence observée :** $q_1, q_2, q_3, \ldots, q_t, \ldots, q_T$
- **Hypothèse chaîne de Markov d'ordre 1 :**

$$P(q_t=j|q_{t-1}=i\ |q_{t-2}=k, \ldots) = P(q_t=j|q_{t-1}=i)$$

- **Stationnarité :**

$$P(q_t=j|q_{t-1}=i) = P(q_{t+l}=j|q_{t+l-1}=i)$$

III. Modèle de Markov (4)

Propriétés (suite) :

- **Matrice de transition :**

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2N} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{iN} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nj} & \cdots & a_{NN} \end{bmatrix}$$

avec :

$$a_{ij} = P(q_t = j | q_{t-1} = i) \qquad 1 \leq i, j, \leq N$$

$$a_{ij} \geq 0, \qquad \forall i, j$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall i$$

III. Modèle de Markov (5)

Exemple :

- **Etats:**
  1. Pluvieux (R)
  2. Nuageux (C)
  3. Ensoleillé (S)

- **Matrice de transition:**

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

- **Calculer la probabilité d ’observer la séquence SSSRRSCS sachant qu’aujourd’hui il fait ensoleillé (i.e. q_1=S).**

III. Modèle de Markov (6)

Exemple (suite) :

- **Formule de la probabilité conditionnelle**

$$P(A,B) = P(A|B)P(B)$$

- **Chaîne de Markov d 'ordre 1:**

$$\begin{aligned} & P(q_1, q_2, \cdots, q_T) \\ &= P(q_T|q_1, q_2, \cdots, q_{T-1})P(q_1, q_2, \cdots, q_{T-1}) \\ &= P(q_T|q_{T-1})P(q_1, q_2, \cdots, q_{T-1}) \\ &= P(q_T|q_{T-1})P(q_{T-1}|q_{T-2})P(q_1, q_2, \cdots, q_{T-2}) \\ &= P(q_T|q_{T-1})P(q_{T-1}|q_{T-2}) \cdots P(q_2|q_1)P(q_1) \end{aligned}$$

Exemple (suite 2) :

- Séquence d ’observations: $O = (S, S, S, R, R, S, C, S)$

- Donc pour Chaîne de Markov d ’ordre 1:

$$
\begin{aligned}
& P(O|model) \\
&= P(S, S, S, R, R, S, C, S|model) \\
&= P(S)P(S|S)P(S|S)P(R|S)P(R|R) \times \\
&\quad P(S|R)P(C|S)P(S|C) \\
&= \pi_3 a_{33} a_{33} a_{31} a_{11} a_{13} a_{32} a_{23} \\
&= (1)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}
$$

- En ayant supposé que: $\pi_i = P(q_1 = i)$

IV. Chaîne de Markov cachée

Définition: Chaîne de Markov cachée

C'est une CM qui contient deux ensembles d'états :

- Les états cachés, qui sont les états vrai du système considéré non observable (séquence $q=(q_1,q_2,\ldots,q_T)$).
- Les états observables, qui représentent les états physiquement visibles en sortie du système (n.b. pour éviter la confusion on parlera d'observations et la séquence associée sera notée $O=(o_1,o_2,\ldots,o_T)$).

Ces derniers sont liés de façon probabiliste aux états cachés.

IV. Chaîne de Markov cachée (2)

Modèle « Urne et balles »

- N urnes contenant des balles colorées
- M couleurs de boules distinctes
- Chaque urne peut avoir une distribution de couleurs différente
- Algorithme de génération d ’une séquence

  1. Choisir de façon aléatoire une des urnes,
  2. Tirer une boule de façon aléatoire, puis la replacée dans l ’urne,
  3. Sélectionner aléatoirement une nouvelle urne,
  4. Répéter les étapes 2 et 3 jusqu’à obtenir une séquence de la longueur souhaitée.

IV. Chaîne de Markov cachée (3)

Propriétés

Éléments de la chaîne de Markov:

- N : Nombre d 'états cachés
- Q : Ensemble des états $\boldsymbol{Q = \{1, 2, \ldots, N\}}$
- M : Nombre de symboles
- V : Ensemble des symboles $\boldsymbol{V = \{1, 2, \ldots, M\}}$
- A : La matrice de transition d 'états

$$\boldsymbol{a_{ij} = P(q_{t+1} = j | q_t = i) \quad 1 \leq i, j, \leq N}$$

- B : La distribution des observations

$$\boldsymbol{B_j(k) = P(o_t = k | q_t = j) \quad 1 \leq k \leq M}$$

- π : La distribution de l'état initial

$$\boldsymbol{\pi_i = P(q_1 = i) \quad 1 \leq i \leq N}$$

- λ : Le modèle complet $\boldsymbol{\lambda = (A, B, \pi)}$

IV. Chaîne de Markov cachée (4)

Les 3 problèmes de bases

1. Sachant la séquence d 'observation $O=(o_1,o_2,\ldots,o_T)$ et le modèle $\lambda=(A,B,\pi)$, calculer de manière optimale $P(O|\lambda)$

- *Le fait que les états soit cachés complique l'évaluation.*
- *Dans le cas où l'on dispose de 2 modèles, ceci peut être utilisé pour choisir le « meilleur ».*

2. Sachant la séquence d 'observation $O=(o_1,o_2,\ldots,o_T)$ et le modèle $\lambda=(A,B,\pi)$, trouver la séquence d'états optimale $q=(q_1,q_2,\ldots,q_T)$

- *On doit décider d 'un critère d 'optimalité (e.g. maximum de vraisemblance).*
- *La séquence cherchée est censée expliquer au mieux les données*

3. Sachant la séquence d 'observation $O=(o_1,o_2,\ldots,o_T)$, estimer les paramètres du modèle $\lambda=(A,B,\pi)$ qui maximise $P(O|\lambda)$.

<La solution existe mais elle ne sera pas abordée dans ce cours>

Solution au 1er problème

- Problème calculer $P(o_1,o_2,...,o_T|\lambda)$
- Algorithme:

Soit $q=(q_1,q_2,...,q_T)$ la séquence d 'états.

En supposant les observations indépendantes, on a:

$$P(O|q,\lambda) = \prod_{i=1}^{T} P(o_t|q_t,\lambda)$$
$$= b_{q_1}(o_1)b_{q_2}(o_2)\cdots b_{q_T}(o_T)$$

La probabilité d 'une séquence donnée est:

$$P(q|\lambda) = \pi_{q1}a_{q1q2}a_{q2q3}\cdots a_{q_{T-1}q_T}$$

Soit encore: $P(O,q|\lambda) = P(O|q,\lambda)P(q|\lambda)$

Donc en énumérant tous les chemins et en sommant leurs

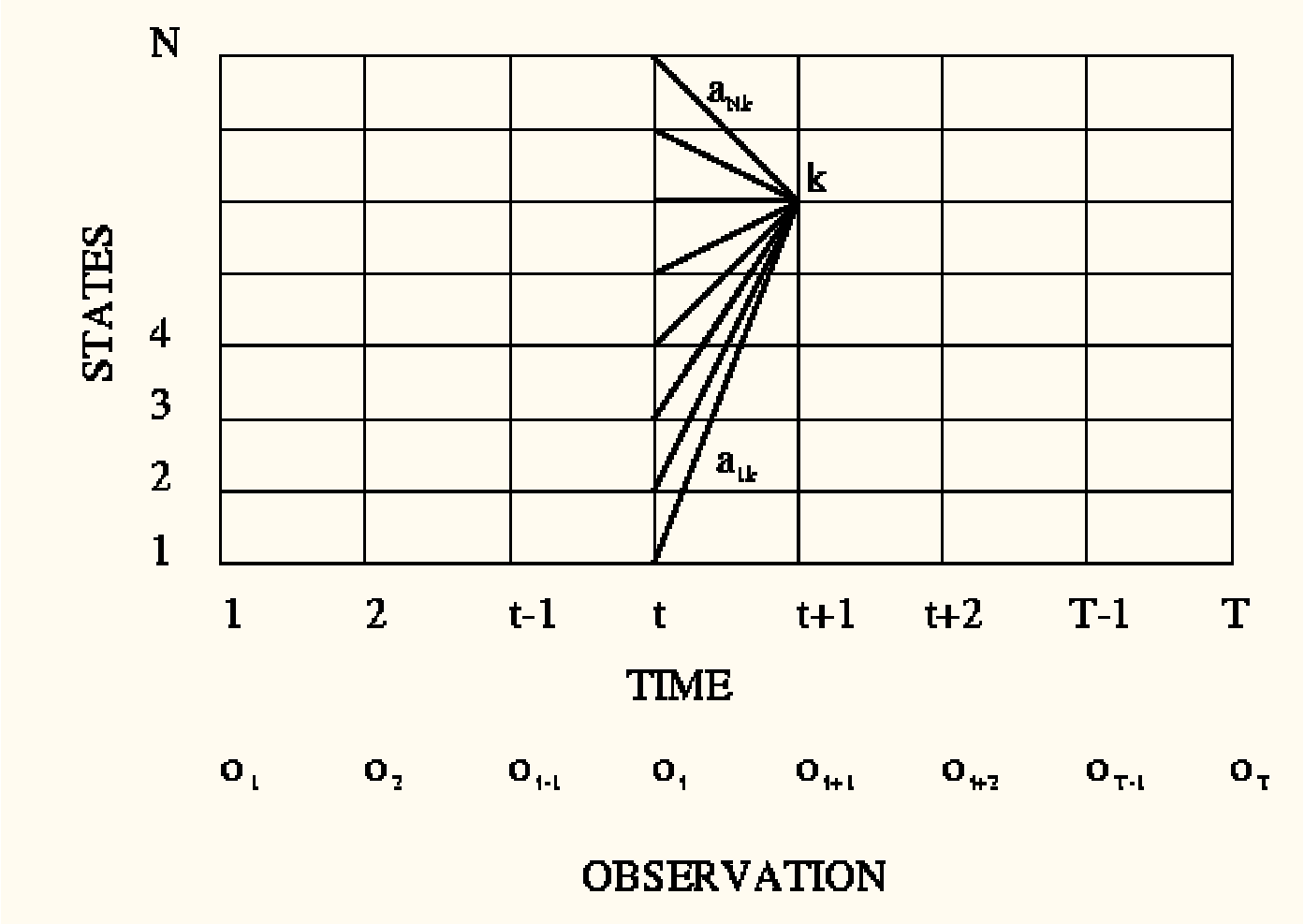
Probabilités: $P(O|\lambda) = \sum_{q} P(O|q,\lambda)P(q|\lambda)$

Remarque: La complexité de calculs est en TxN^T !!!

V. Algorithme récursif Méthode Avant

Solution au 1er problème par la méthode récursive avant

- Intuitions :



V. Algorithme récursif Méthode Avant (2)

Solution au 1er problème : Méthode récursive avant (suite)

• On définit la variable:

• $\alpha_t(i)$ est la probabilité d'observer la séquence partielle $(o_1, o_2, \ldots, o_t)$ telle que l'état q_t est i.

• Algorithme:

1. Initialisation: $\alpha_1(i) = \pi_i b_i(o_1)$
2. Induction: $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1})$
3. Rassemblement: $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

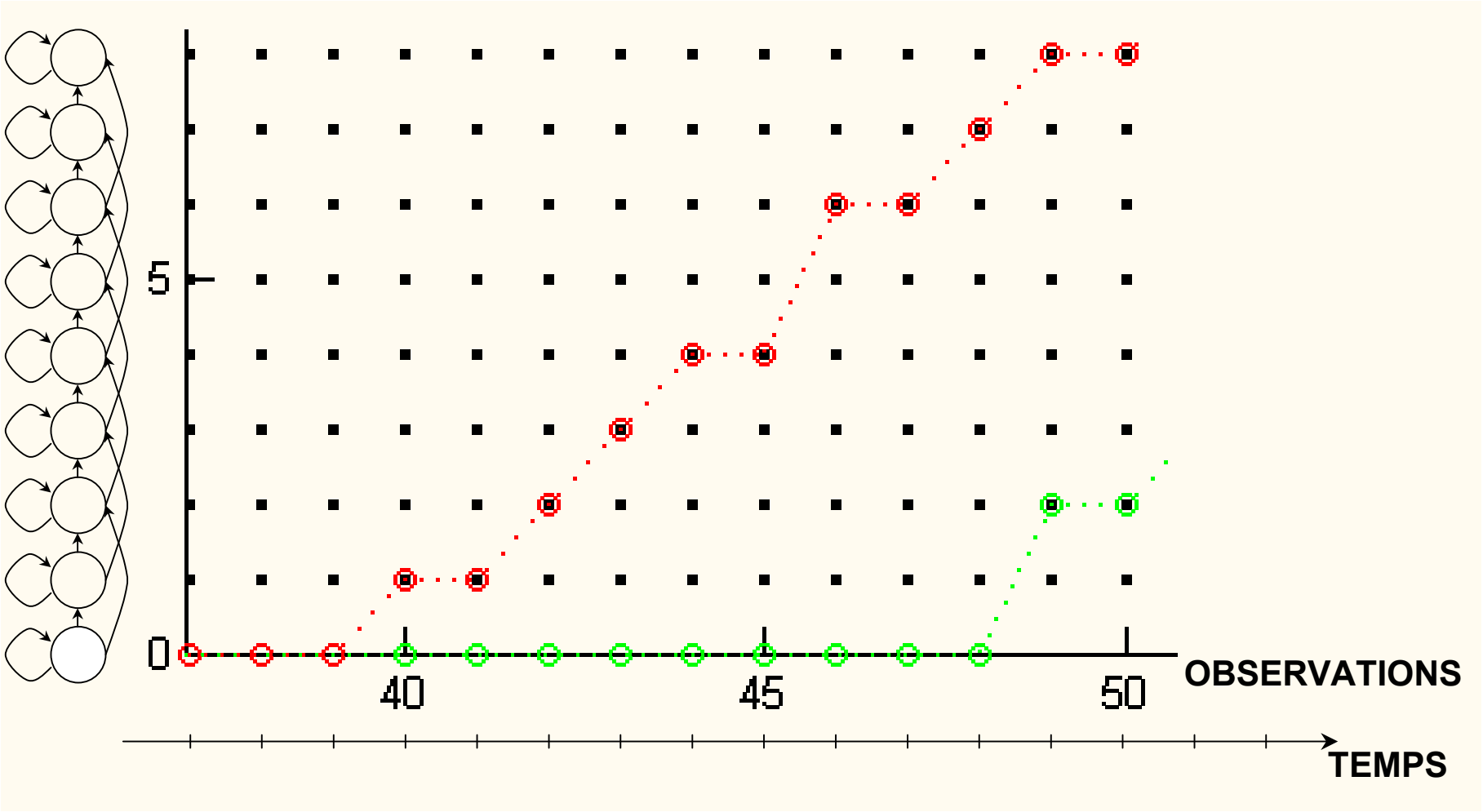
Remarque: La complexité est ramenée en N^2T.

Solution au 2nd problème par Viterbi

- On rappelle que l 'on cherche le chemin le plus probable.
- On cherche donc le chemin $(q_1,q_2,...,q_T)$ pour lequel la vraisemblance est maximale, c 'est à dire: $P(q_1, q_2, \ldots, q_T|O, \lambda)$
- Solution en utilisant la programmation dynamique
- On définit: $\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P(q_1, q_2, \ldots, q_t = i, o_1, o_2, \ldots, o_t|\lambda)$
- $\delta_t(i)$ est la probabilité maximale des chemins se terminant à l'état i.
- Par induction on obtient: $\delta_{t+1}(j) = \max_i[\delta_t(i)a_{ij}] \cdot b_j(o_{t+1})$

VI. Algorithme Viterbi (2)

Représentation en grille d'un meilleur chemin



VI. Algorithme Viterbi (3)

Description de l'algorithme

- Initialisation: $\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$

$$\psi_1(i) = 0$$

- Récursion $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

$$2 \leq t \leq T, 1 \leq j \leq N$$

- Évaluation du meilleur score (i.e. chemin plus probable)

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

- « Backtracking » des chemins

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1$$

VI. Algorithme Viterbi (4)

Implémentation de l'algorithme

Dans la pratique on prends le log de chacune des probabilités misent en jeu, ainsi, tous les produits de probabilités sont remplacés par des sommes. Donc, on définit:

$$\phi_t(i) = \max_{q_1, q_2, \cdots, q_t} \{\log P[q_1\, q_2 \cdots q_t, O_1\, O_2 \cdots O_t | \lambda]\}$$

D'ou les modifications suivantes:

- Initialisation: $\phi_1(i) = \log(\pi_i) + \log[b_i(O_1)]$
- Récursion: $\phi_t(j) = \max_{1 \le i \le N} [\phi_{t-1}(i) + \log a_{ij}] + \log[b_j(O_t)]$
- Évaluation du meilleur score (i.e. chemin plus probable):

$$\log P^* = \max_{1 \le i \le N} [\phi_T(i)]$$

Dans la pratique les valeurs $log(a_{ij})$ sont pré-calculées et stockées en ROM.

VII. Applications du VA

3 applications majeures

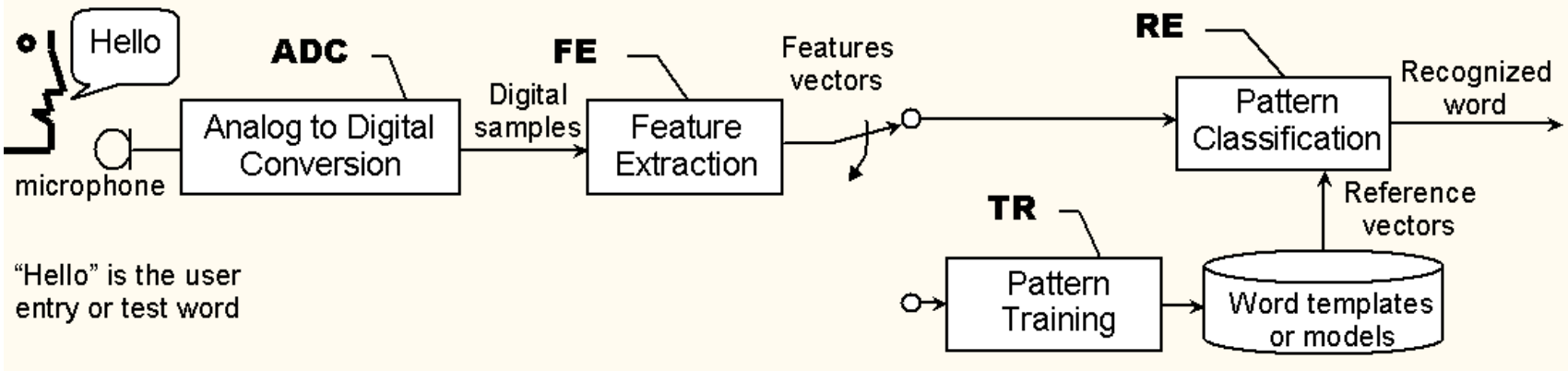
- La reconnaissance vocale (ASR), ainsi que d'autres applications de classifications telle que la reconnaissance d'écriture.
- Le contrôle d'erreur par codage/décodage canal à l'aide de code convolutionnel.
- L'égalisation de canal, utiliser par exemple dans le cas de la modulation GSM-EDGE.

Mais aussi:

- La démodulation « FSK phase continue »,
- Interférence Entre Symbole (ISI, Inter-Symbol Interference),
- Le traitement du langage naturel.
- Etc.

VIII. Exemple : Reconnaissance Vocale (ASR)

Schéma classique d'un système ASR



Sur le schéma ci-dessus les 2 modes sont représentés i.e. entraînement et reconnaissance. La sélection de l'un ou l'autre est effectuée au travers du commutateur.

Description des blocks:

• ADC: Convertit le signal analogique à l'entrée du microphone en échantillons numériques afin qu'ils puissent être traités par un microprocesseur (µContrôleur ou DSP).

• FE: Plusieurs échantillons sont groupés afin d'être traités et convertis pour obtenir des paramètres plus adaptés. Ce block est souvent appelé « front-end » dans la littérature.

• RE: Les vecteurs de caractéristiques liés au mot prononcé et, les vecteur de références représentant tous les modèles de mots sont comparés. Le mot le plus probable est sorti.

• TR: Les vecteurs de caractéristiques liés au mot prononcé sont utilisés pour construire un nouveau modèle pour ce mot ou, en adapté un déjà existant. Le modèle est composé par ce que l'on nomme les vecteurs de référence.

Exemple de « Front-end » basé sur la FFT

Signal d'entrée: 20ms @ f_S = 8 kHz,
256 échantillons audio (16 bits) prélevés avec un « overlap » 96.

$$H_{pre} = k \times (1 - a.z^{-1}) \quad , avec : a \in [0.4, 1]$$

Pré-emphase du signal

$$y[i] = x[i] \times (0.54 - 0.46.\cos(2\pi i / N))$$

Fenêtrage de Hamming

$$Y(m) = \sum_{n=0}^{N-1} y(n).e^{-j\frac{2\pi nm}{N}} \quad (Cooley - Tuckey)$$

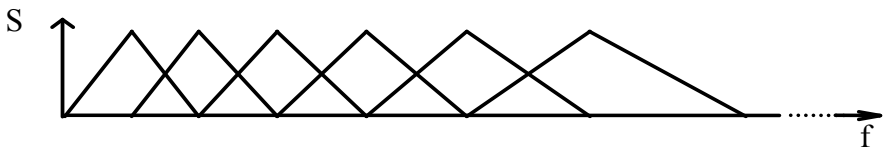
FFT

On utilise la méthode de Winograd avec une FFT 128 points complexes. En utilisant les différentes symètries on récupère le spectre complexe [0, PI[.

Extraction de la partie Réelle

$$Y[i] = real(X[i]) \times real(X[i]) + imag(X[i]) \times imag(X[i])$$

Estimation de la DSP



Intégration en bandes Mel

Calcul du Logarithme des puissances par bande

On fait une DCT inverse DCT^{-1}

Calculs des coeff. Cepstraux par DCT

Vecteur de caractéristiques $\vec{\mu}$
(16 valeurs)

VIII. Exemple : Reconnaissance Vocale (ASR) (3)

Le VA en reconnaissance vocale

Montrons comment s'intègre concrètement le VA au travers de l'exemple suivant. On pose le modèle λ suivant:

Une probabilité initiale : $\pi(i) = 1 \quad \Leftrightarrow \quad \log(\pi(i)) = 0$

Un ensemble de 3 transitions notées *LOOP*, *NEXT*, *SKIP,* telles que:

$$T(i,j) = \log(a_{ij}) = T(k) = \begin{cases} LOOP = T(i,i) = T(0) \\ NEXT = T(i,i+1) = T(1) \\ SKIP = T(i,i+2) = T(2) \end{cases} , \quad \forall i, \forall j = i + k, \forall k \in [0,3[$$

Maintenant, pour la probabilité d'émission partons de la version multidimensionnelle générale de la Gaussienne:

$$\log(b_j(o_t)) = \log(p(o_t \mid q_j)) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)\right)$$

Le VA en Reconnaissance vocale (suite)

Cette dernière équation se simplifie une première fois si les composantes du vecteur x associées a l'observation o_t sont décorellées, puisque dans ce cas, la matrice de covariance est diagonale:

$$\log(b_j(o_t)) = \log(p(o_t \mid q_j)) = \log\left(\prod_{k=1}^{d} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma_k^2}\right)\right)$$

Une simplification supplémentaire est possible si on considère une variance commune au différentes classes:

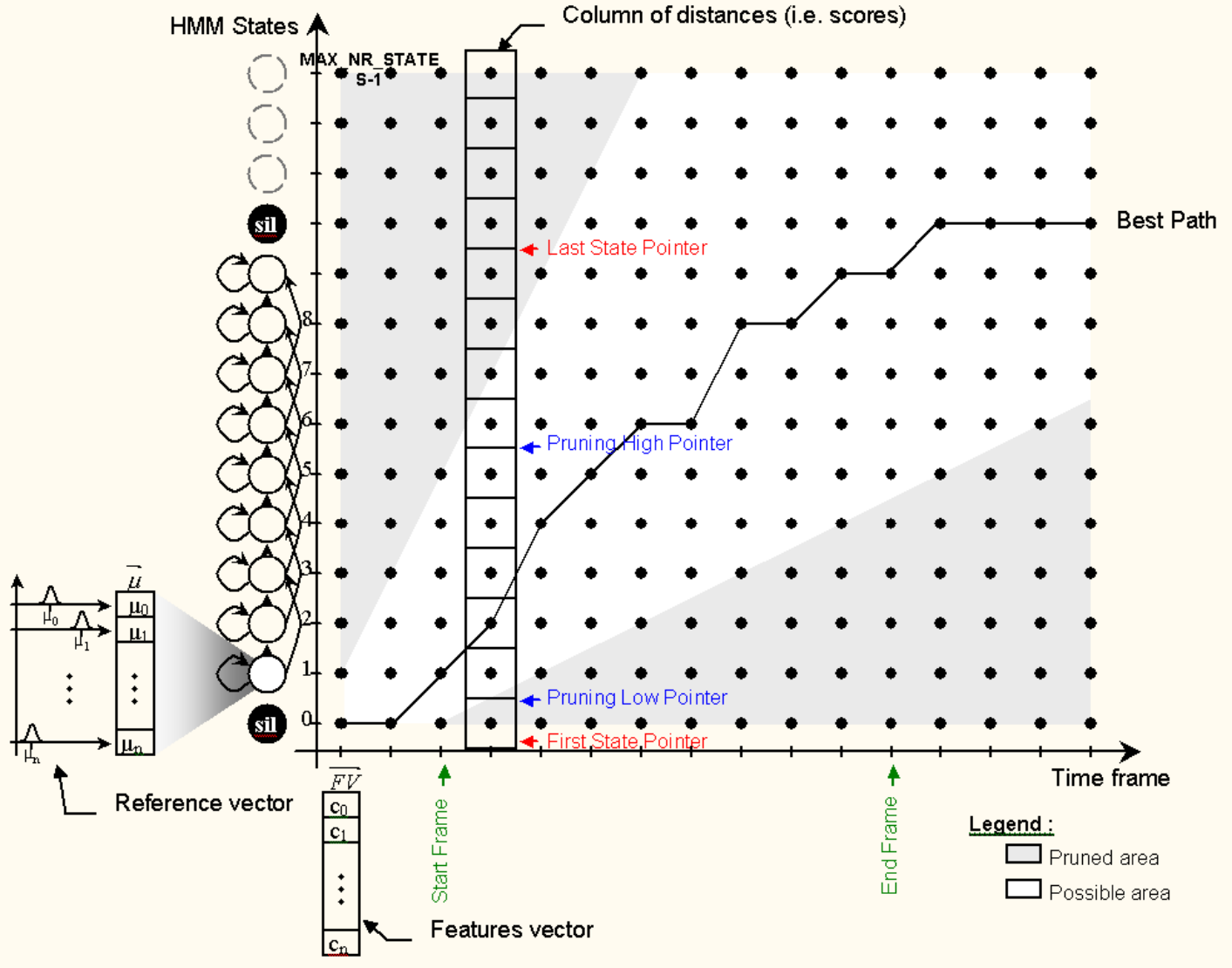
$$\log(b_j(o_t)) = \log(p(o_t \mid q_j)) = \log\left(\prod_{k=1}^{d} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)\right)$$

$$= \sum_{k=1}^{d} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x_i-\mu_k)^2}{\sigma^2}\right)\right)$$

$$= K(\sigma,d) - \frac{1}{2\sigma^2}(x_i-\mu_k)^2 = K(\sigma,d) - \frac{1}{2\sigma^2} d_{L_2}^2(x_i,\mu_k)$$

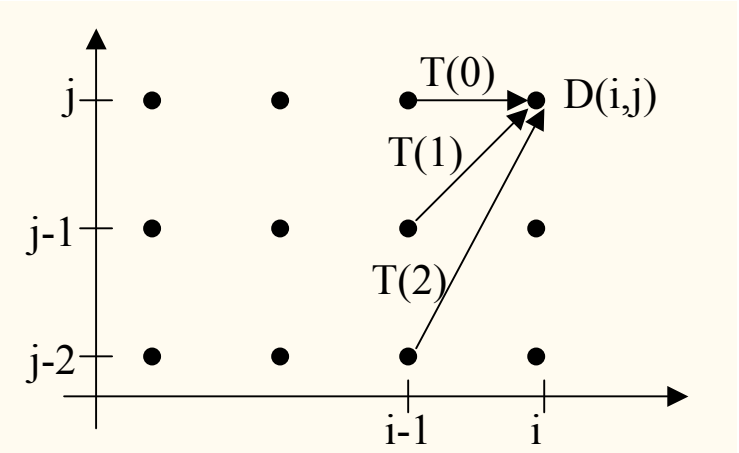
VIII. Exemple : Reconnaissance Vocale (ASR) (5)

Le VA en reconnaissance vocale (suite 2) – Grille complète



Le VA en Reconnaissance vocale (suite 3)

On aboutit à la représentation suivante:



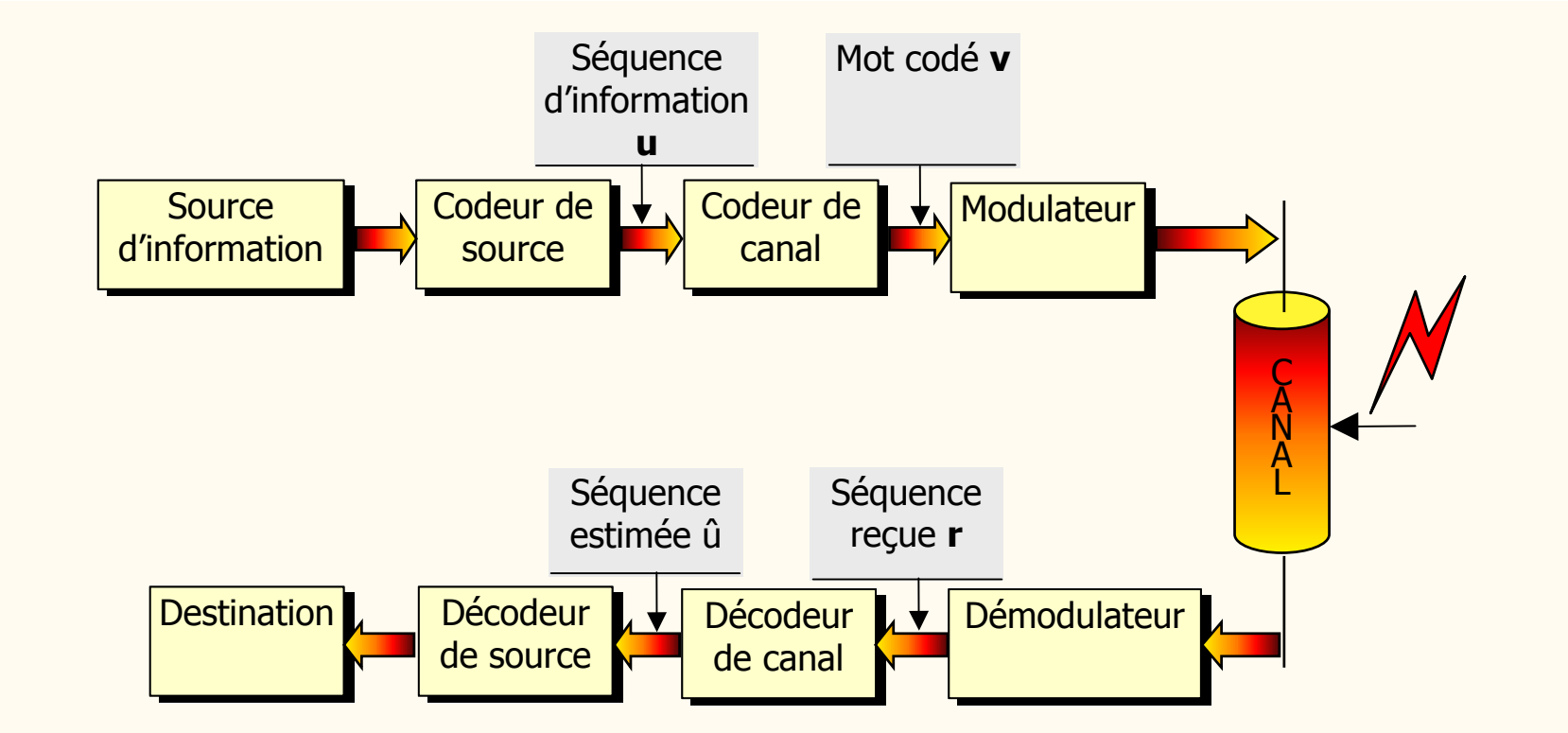
La formule de récursion devient donc:

$$D(i,j) = \min_{\substack{0 \le i < N \\ 0 \le k < 3}} \left[D(i-1, j-k) + T(k) \right] + d(i,j) \quad , \quad avec : d(i,j) = d_{L^2}(x_i, \mu_k)$$

On remarque que l'on est passé d'une maximisation de proba. à une minimisation de la distance globale $D(i,j)$. En reconnaissance vocale, on parlera souvent de *score* pour l'opposée de $D(i,j)$ $(i.e.\ -D(i,j))$, de *pénalités* pour $T(k)$, et de *distance locale* pour $d(i,j)$.
Pendant la phase de reconnaissance, un score est calculé pour chaque mot du dictionnaire, le mot reconnu étant celui qui obtient le plus grand score.

IX. Exemple : Code convolutionnels (1)

Schéma d'une transmission numérique:



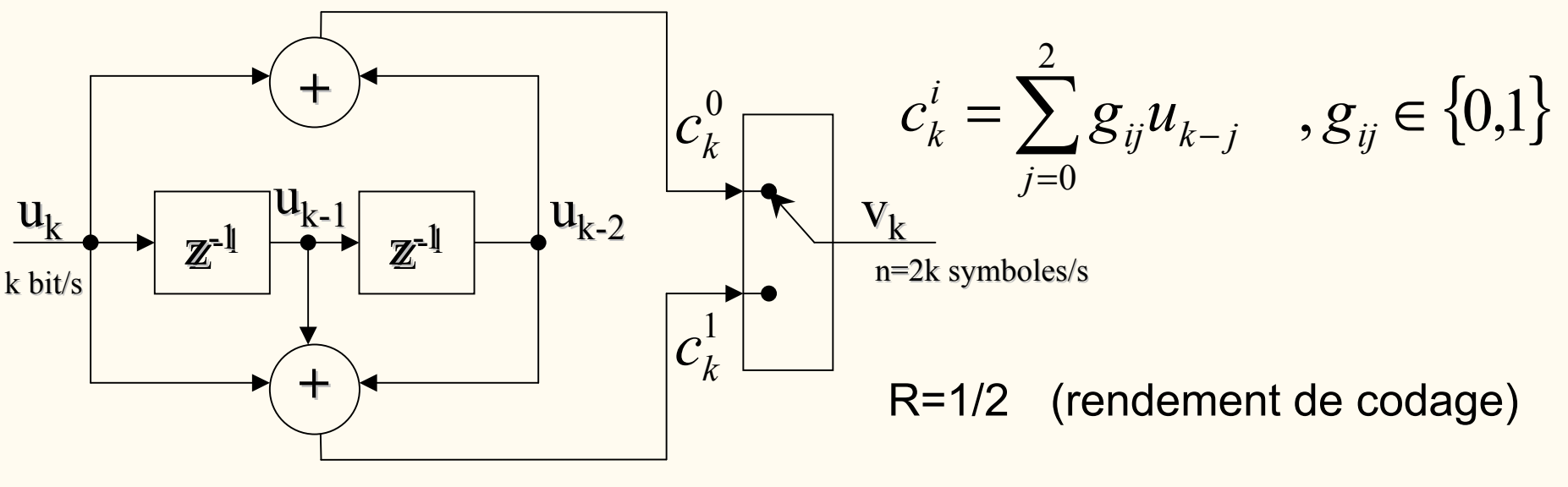
Le codage canal est effectué le plus souvent à l'aide un codeur de type convolutionnel. Il permet d'ajouter de la redondance au signal afin de pouvoir, au moment du décodage, annuler ou, tout du moins, réduire les erreurs introduites lors de la transmission.

IX. Exemple : Code convolutionnels (2)

Le codage convolutionnel:

Le codage convolutionnel est réalisé à l'aide d'un registre à décalage et d'additionneurs modulo 2. Ces derniers sont implémentés simplement en utilisant une porte « OU exclusif ».

Exemple: Codeur dit 3,5,7 (K=3 longueur de contrainte)



$g_i = [g_{i0}, g_{i1}, g_{i2}]$ avec i=1,2. En général on exprime les séquences génératrices en octal, ainsi pour le codeur ci-dessus, on a :

$g_0=[1,0,1]=5$ (octal) et $g_1=[1,1,1]=7$ (octal)

IX. Exemple : Code convolutionnels (3)

Codage convolutionnel – position du problème:

Dans le cas du décodage convolutionnel les observations o_k dépendent de façon probabiliste de la sortie v_k, elle même étant liée à la séquence d'entrée u_k.
La séquence v_k correspond à la séquence sans bruit et non observable q vue précédemment. C'est elle qui sera estimée lors du décodage.

Dans le cas le plus général, on a donc un processus de Markov m-aire d'ordre v pour lequel on définit:

1. L'état : $q_k \equiv (u_{k-1},\ldots,u_{k-v})$

2. La transition : $\xi_k \equiv (u_k,u_{k-1},\ldots,u_{k-v})$

Dans notre exemple on a $v=2$, soit :

$$q_k \equiv (u_{k-1},\ldots,u_{k-2}) \qquad \xi_k \equiv (u_k,u_{k-1},\ldots,u_{k-2})$$

Et m=2 ce qui nous donne $m^v=2^2=4$ états et $m^{v+1}=2^3=8$ transitions.

IX. Exemple : Code convolutionnels (4)

Codage convolutionnel – tables de transitions et de sorties :

Ci-dessous sont présentées les tables du codeur convolutionnel précédent :

Table de transitions d'états :

	État suivant, si	
État courant	**$u_k = 0$**	**$u_k = 1$**
00	00	10
01	00	10
10	01	11
11	01	11

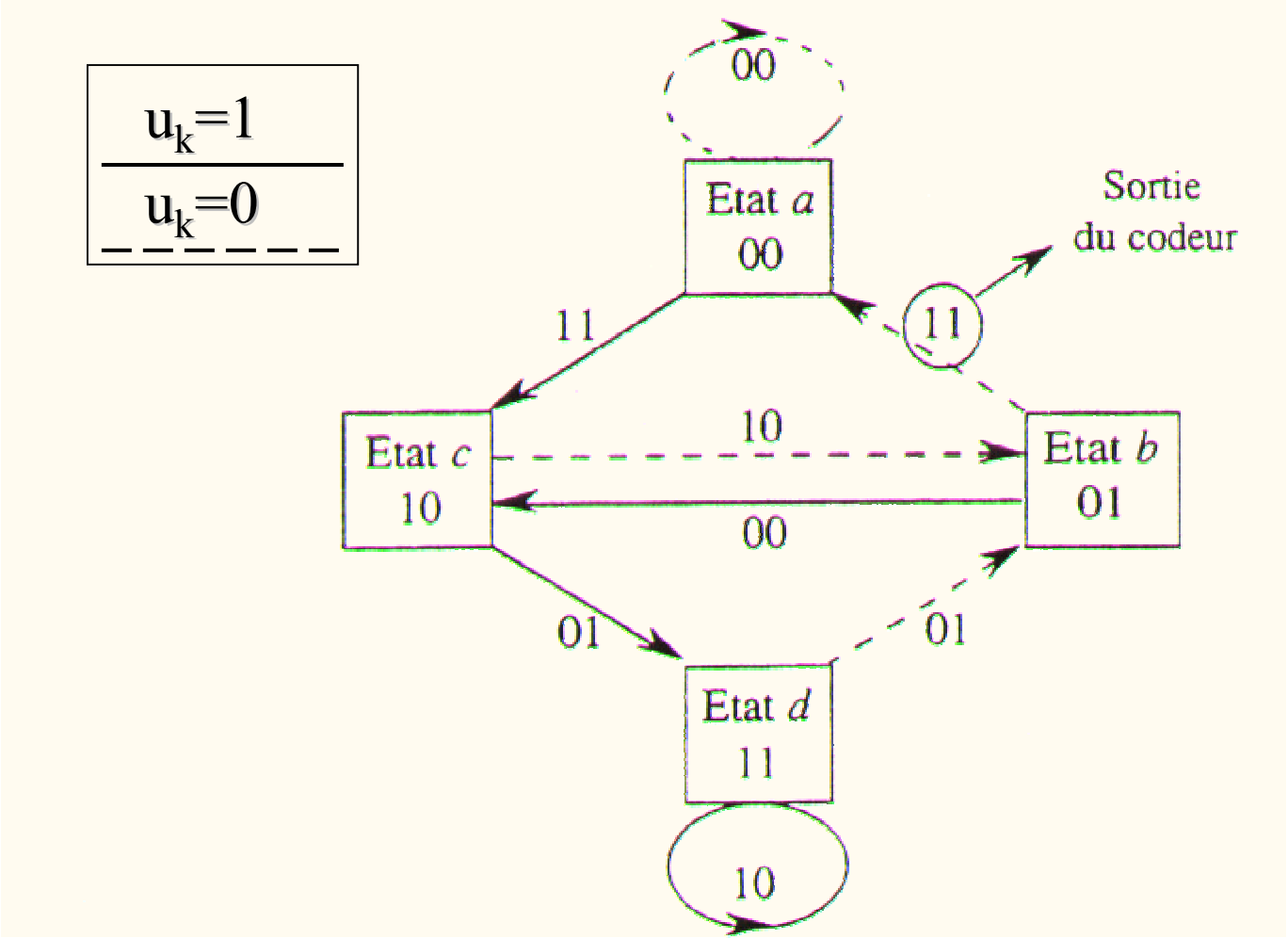
Table de sorties :

	Symbole v_k en sortie, si	
État courant	**$u_k = 0$**	**$u_k = 1$**
00	00	11
01	11	00
10	10	01
11	01	10

IX. Exemple : Code convolutionnels (5)

Codage convolutionnel – diagramme d'états :

Diagramme d'états du codeur convolutionnel pris en exemple :



IX. Exemple : Code convolutionnels (6)

Décodage de code convolutionnel par Viterbi – Exemple :

Comme énoncé auparavant, l'algorithme de Viterbi correspond à une recherche séquentielle à travers un treillis visant déterminer la séquence ayant le maximum de vraisemblance.

Exemple, en reprenant le codeur précédent on suppose que l'on a :

Entrée du codeur	1 0 0 1
Sortie du codeur	11 10 11 11
Entrée du décodeur	11 00 11 11 (erreur en position 3)

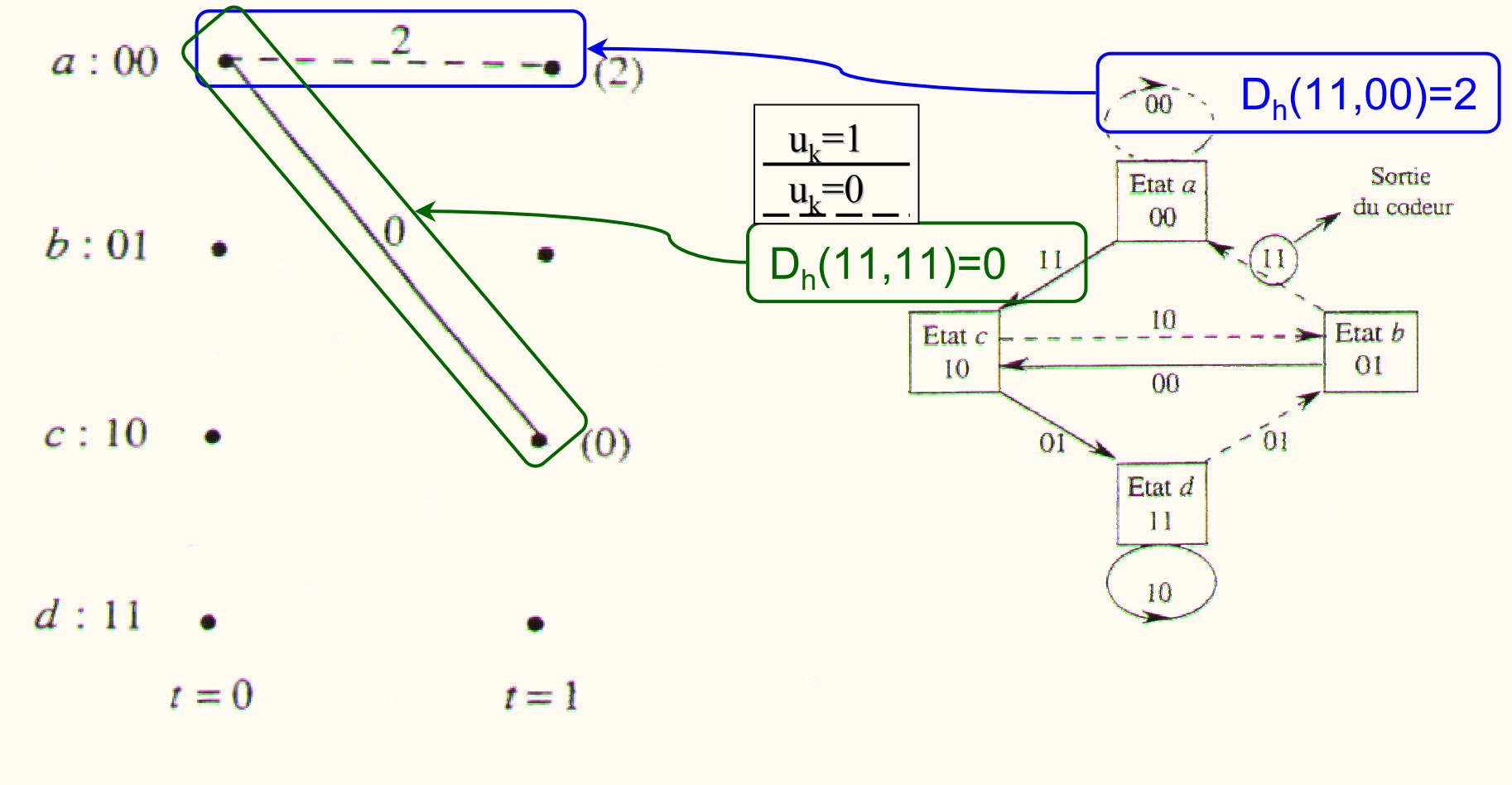
Par la suite on va étudier comment malgré l'erreur reçue, le VA nous permet de retrouver la séquence émise.

On va utiliser la distance dite de Hamming qui corresponds aux nombre de composantes (dans notre exemple ce sont les bits) qui diffèrent entre deux mots code.

IX. Exemple : Code convolutionnels (7)

Décodage de code convolutionnel par Viterbi – Exemple suite:

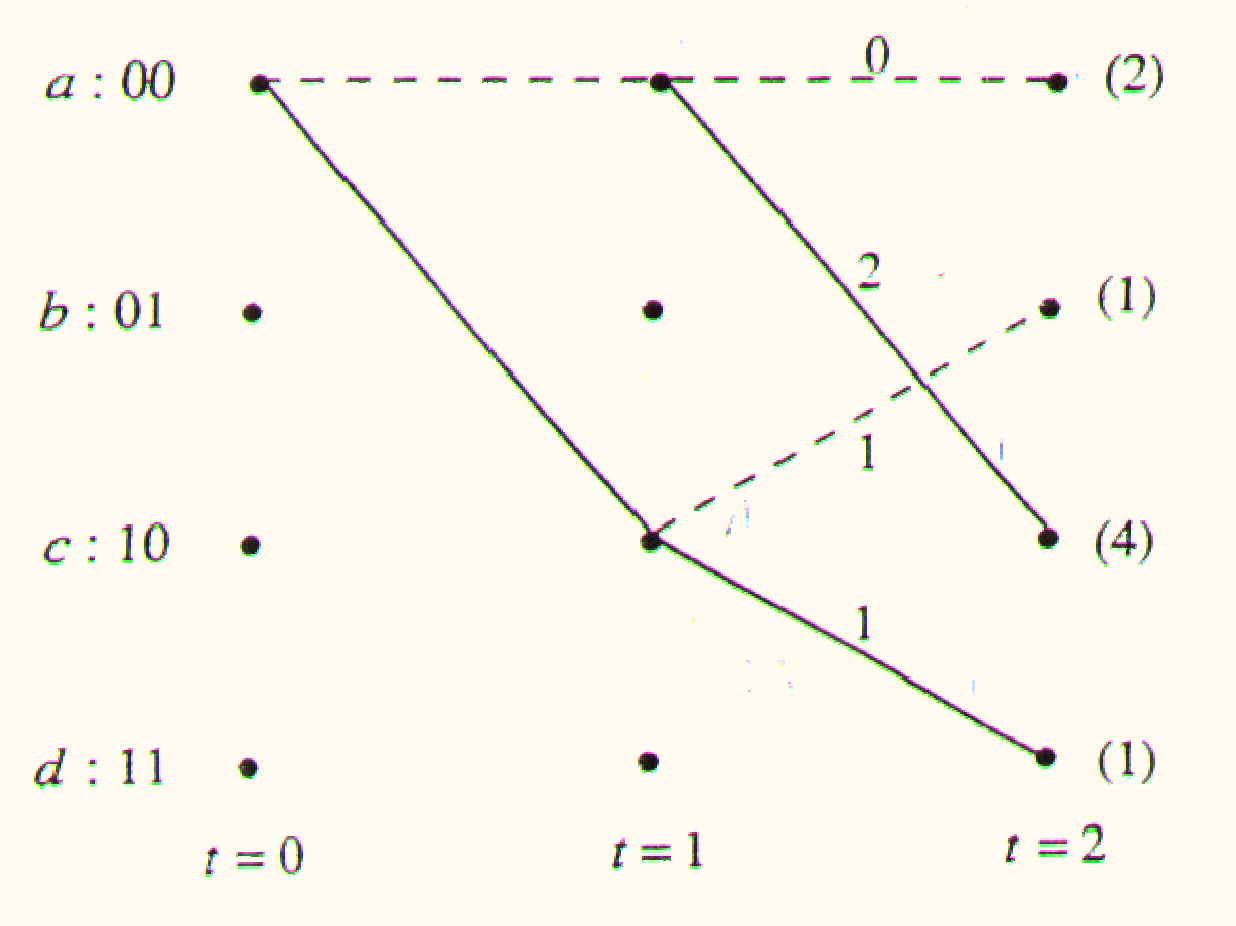
Symbole en entrée du décodeur: 11



IX. Exemple : Code convolutionnels (8)

Décodage de code convolutionnel par VA – Exemple suite 2:

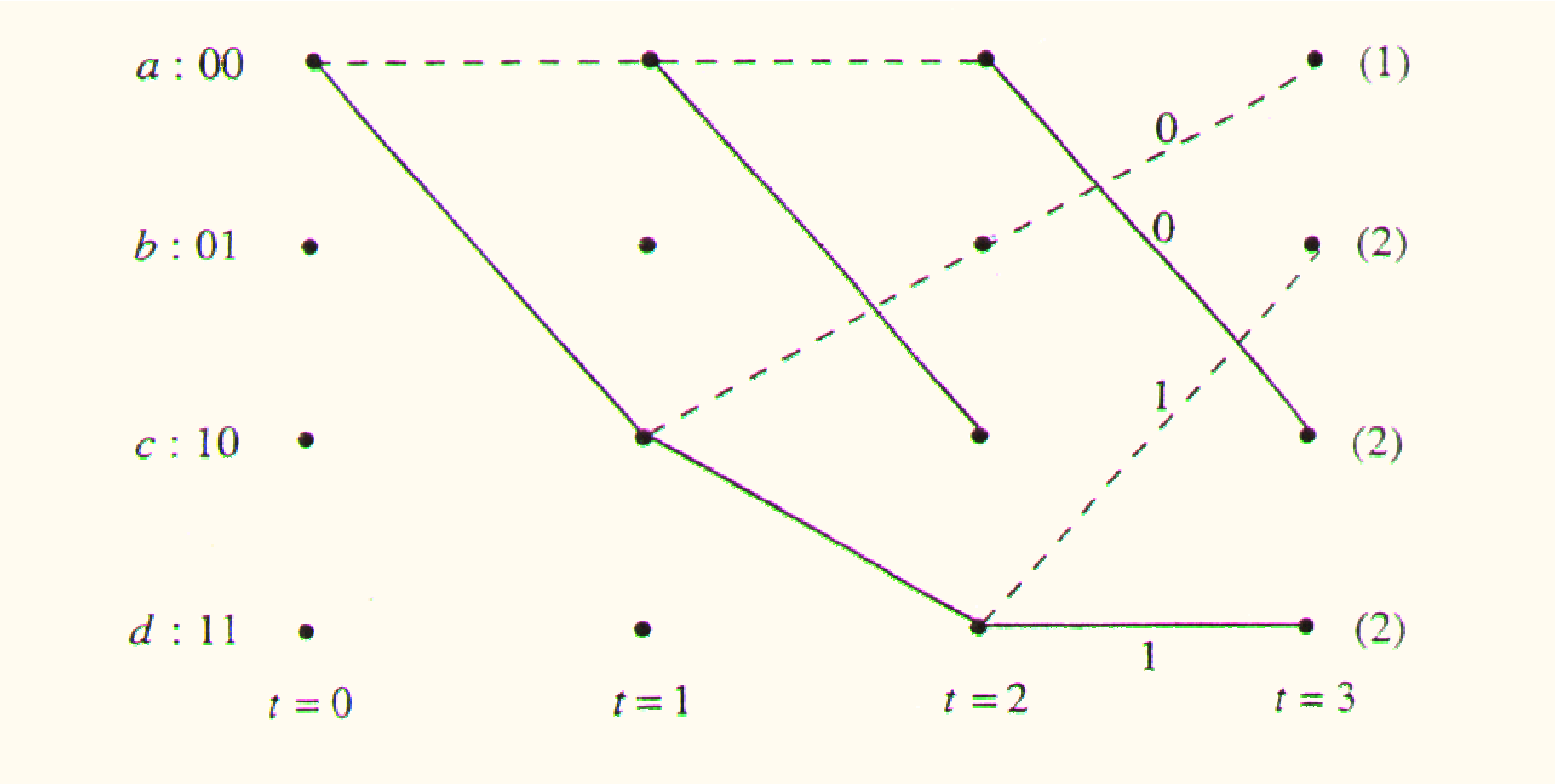
Symbole en entrée du décodeur: 00



IX. Exemple : Code convolutionnels (9)

Décodage de code convolutionnel par VA – Exemple suite 3:

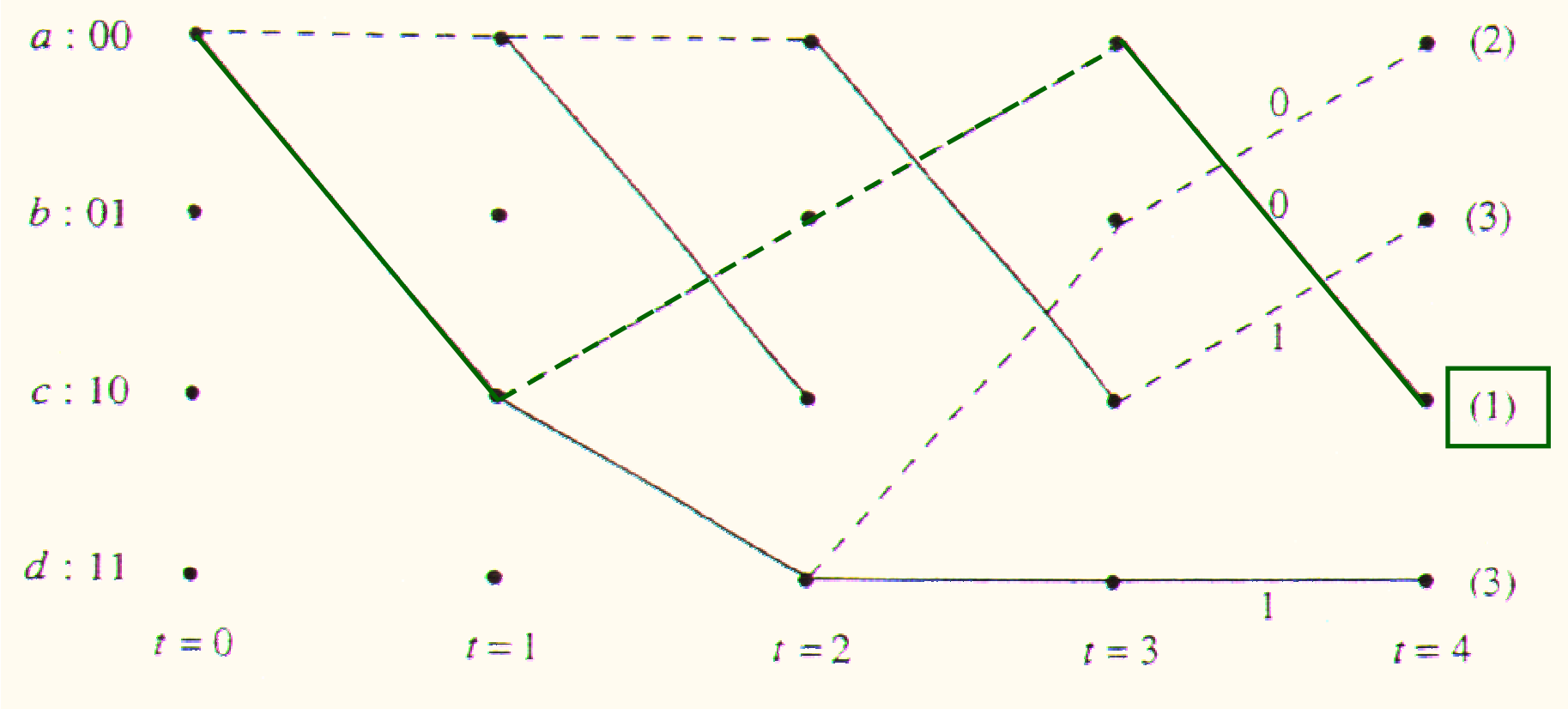
Symbole en entrée du décodeur: 11



IX. Exemple : Code convolutionnels (10)

Décodage de code convolutionnel par VA – Exemple suite 4:

Symbole en entrée du décodeur: 11



En remontant le meilleur chemin (i.e. celui avec la métrique cumulée la plus petite, en vert sur le schéma) on retrouve la séquence d'entrée: 1001

Merci …

Pour plus d'information:

Laurent Depersin
Multimedia Expert

PHILIPS

Philips Consumer Electronics
BLC Mobile Phones - BG Mobile Infotainment
Route d'Angers - 72000 Le Mans cedex 9 - France

Tel : +33 (0)2 43 41 1696 – Fax ext.: 1222
E-mail : laurent.depersin@philips.com