Modèles Bayesiens de séquences

Plan (2007-2008)

Introduction à l'analyse de séquences (Parole, Ecrit, Analyse textuelle)
- Modélisation de séquences : Première approche
- distance d'édition, Programmation dynamique, limitations

Modèles de Markov Cachés Discrets
- Définitions : Modèles discrets
- Algorithme de viterbi et programmation dynamique
- Algorithme Forward
- Apprentissage : Algorithme de Baum-Welch

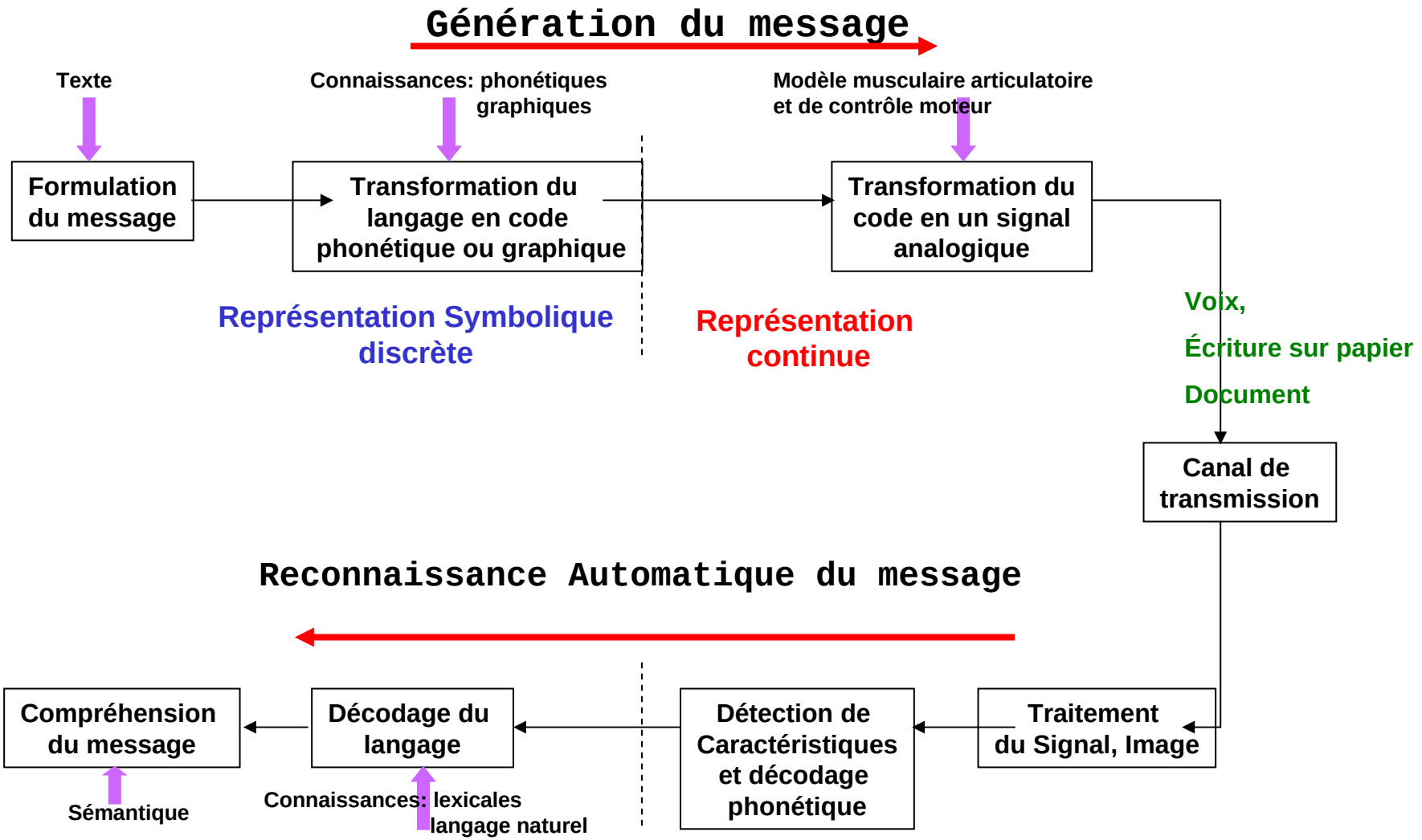
Modèles de Mélanges et algorithme EM

Modèles de Markov Continus
- Applications à la reconnaissance de séquences : Parole et Ecrit
- à l'extraction d'information : Taggers Syntaxiques, Fouille de texte

Modèles de Champs de Markov Cachés
- Définitions générales
- Champs Markoviens
- Champs de Gibbs
- Théorème de Hammersley Cliffors
- Calcul de la probabilité locale d'un état
- Méthode de simulation : Echantillonneur de Gibbs
- Méthode d'optimisation : algorithme de recuit simulé, algorithme ICM

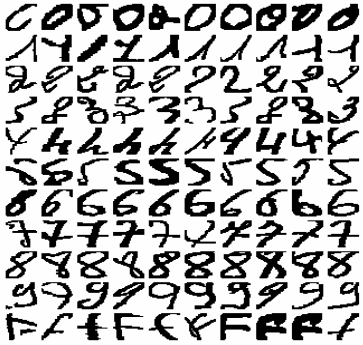
http://www.univ-rouen.fr/psi/paquet/

Production et perception de la parole, de l'écrit et des documents



Les sources de variabilité ou de bruit dans la chaîne de communication

A l'émission : le producteur produit un message personnel



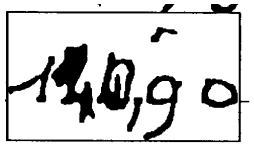
voix : intonation, prosodie

écriture : style d'écriture, inclinaison, taille

mise en page du document : simple/double colonne, polices de caractères…

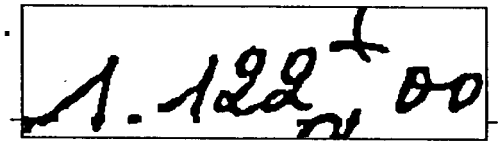
on veut des systèmes omni-locuteurs, omni-scripteurs, omni-fontes…

Dégradation due au canal et au milieu



parole en milieu ambiant,

conditions d'écriture : stylo, papiers, ambiance (chèque, brouillon,…



documents faxés, archives anciennes…

L'analyse automatique doit s'affranchir de cette variabilité

il faut des systèmes de décodage robustes

Le Traitement Automatique

Les systèmes de décodage doivent tirer partie des connaissances du domaine pour être plus robustes

- phonétiques
- graphiques
- mise en page
- lexicales
- langage

Ils doivent si possible pouvoir s'adapter à des environnements variables

adaptabilité : nouveaux utilisateurs (reco vocale)
nouvelle mise en page
nouveau lexique
nouvelle langue

L'apprentissage automatique est un moyen d'adapter les systèmes à de nouvelles conditions d'utilisation en limitant l'intervention humaine au maximum

Modèles de Séquences

Modèles de connaissances adaptés au langage

mot = séquence de caractères
phrase = séquence de mots

Modèles Bayesiens de séquences

robustes
adaptables par apprentissage

Les plus performants pour l'analyse automatique de la parole, de l'écriture, de la langue naturelle
…et pour beaucoup d'autres problèmes

Reconnaissance de Formes (RdF)

Techniques Informatiques de représentation et de décision permettant de donner aux machines des capacités de perception et d'interprétation

Représentation : comment décrire l'information mesurée par les capteurs? quelles caractéristiques?

si on a les bonnes caractéristiques la décision est facile

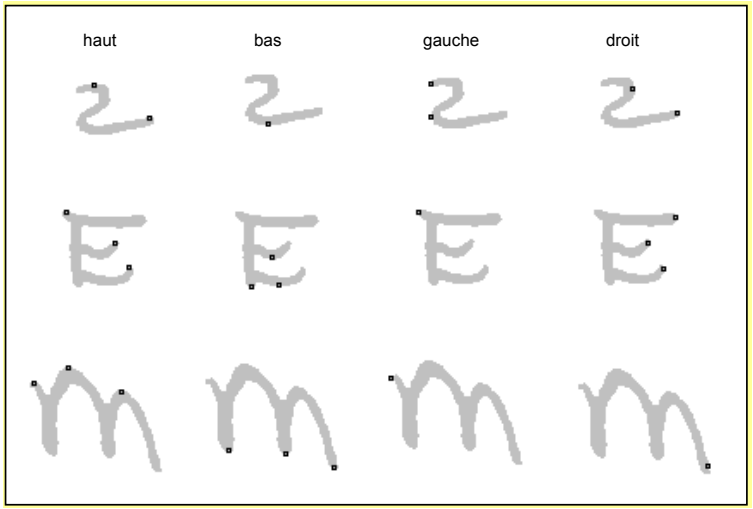
Décision : comment prendre une décision d'affectation d'une représentation à un modèle ou une classe?

quand il n'y a pas de caractéristiques évidentes il faut savoir décider

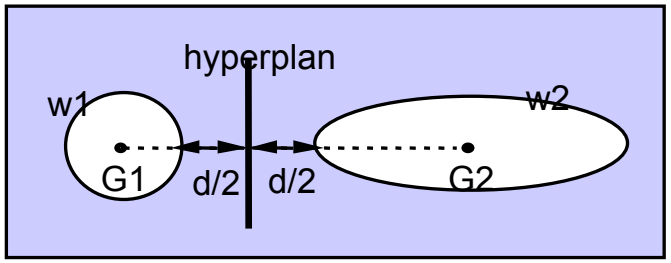
Principe :
construire une distance, une probabilité… qui permet de décider du plus proche au plus éloigné, du plus probable au moins probable…en minimisant les mauvaises décisions

RdF : Exemple

Caractéristiques



Décision

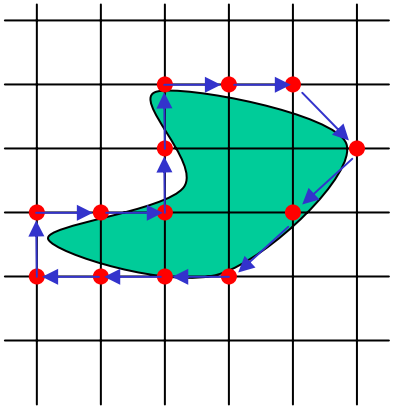


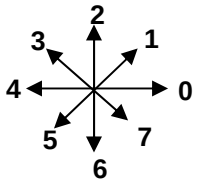
1. Modèle de séquences et distance d'édition

Soit $X=\{a,b,...\}$ **un alphabet de formes élémentaires**

Une phrase, ou une séquence, est définie sur X **par une concaténation ordonnée d'éléments de** X

Exemple : **représentation des contours dans une image à l'aide des directions de Freeman**

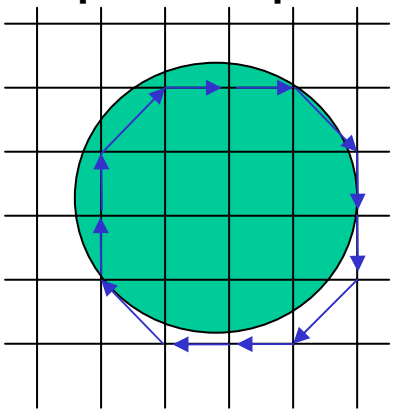




S = 0 0 7 5 5 4 4 4 2 0 0 2 2

Modèle de séquences et distance d'édition

Un modèle de séquence peut facilement être construit pour chaque type de forme que l'on souhaite distinguer



Rond : 0 0 7 6 6 5 4 4 3 2 2 1

Carré : 0 0 0 0 6 6 6 6 4 4 4 4 2 2 2 2

Comment calculer la distance d'une forme inconnue à chacun des modèles?

On calcule une distance d'édition entre chaînes

Distance d'édition entre chaînes

Levenstein (1966), Wagner et Fisher (1974)

On définit trois opérations élémentaires possibles entre 2 symboles de l'alphabet considéré. Chacune est associée à un coût.

La subsitution	$a \longrightarrow b$	a est transformé en b
L'insertion	$\lambda \longrightarrow a$	le symbole vide est transformé en a
La destruction	$a \longrightarrow \lambda$	le symbole a est transformé en le symbole vide

Il faut définir la matrice des coûts élémentaires

exemple de matrice de coûts pour la correction de frappes au clavier

$\gamma(x,y)$	**a**	**b**	λ
a	0	2	1
b	2	0	1
λ	1	1	0

Distance de Levenstein

A l'aide des transformations élémentaires, toute chaîne peut être transformée en une autre

X = ab peut se transformer en Y = bac par la suite de transformations

ab, (insertion de c) cab, (suppression de b) ca, (subs de c en b) ba, (insertion de c) bac

…mais il y a une infinité de transformations possibles.

Distance de Levenstein

C'est le coût de la transformation la moins coûteuse (optimale) de X en Y

On note $\delta(X,Y)$

On impose : $\delta(a,b) = \gamma(a,b)$

Ce qui revient à imposer l'inégalité triangulaire : $\gamma(a,c)+\gamma(c,b) \geq \gamma(a,b)$

γ doit être une vraie distance!

Algorithme de Wagner et Fisher

Soient les deux chaînes $X=a_1 \ldots a_n$ **et** $Y=b_1 \ldots b_m$

Soit X(i) la sous-chaîne de longueur i construite à partir de X

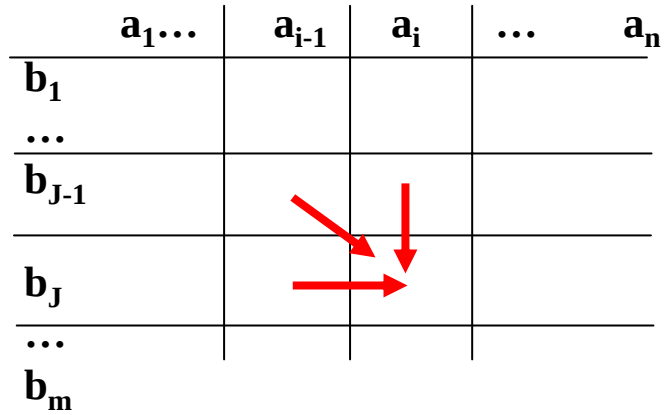
$$X(i)=a_1 \ldots a_i$$

On note la distance de Levenstein entre les sous-chaînes X(i) et Y(j) par

$$\delta(X(i),Y(j)) = D(i,j) \qquad \delta(X,Y)=D(n,m)$$

alors la distance de coût minimal entre les deux sous-chaîne X(i) et Y(j) est

$$D(i,j)=\min\begin{cases} D(i-1,j-1)+\gamma(a_i,b_j) \\ D(i-1,j)+\gamma(a_i,\lambda) \\ D(i,j-1)+\gamma(\lambda,b_j) \end{cases}$$



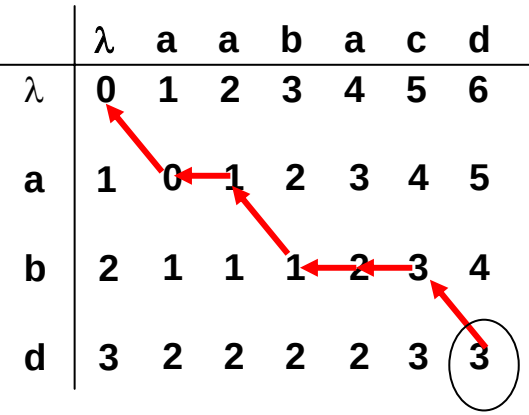
Principe d'optimalité de Bellman (1955) mis en œuvre par la Programmation Dynamique pour les problèmes d'optimisation décomposables avec une fonction de coût monotone non décroissante

Algorithme de Wagner et Fisher

```
1)   Prec(0,0)=0;                    /*initilisation
2)   D(0,0)=0;
3)   Pour i=1 à n
     faire D(i,0)=D(i-1,0)+γ(a_i,λ);
3)   Pour j=1 à m
     faire
          D(0,j)=D(0,j-1)+γ(λ,b_j);
4)   Pour i=1 à n                    /*boucle principale
     Pour j=1 à m
     faire
          m_1 = D(i-1,j-1)+γ(a_i,b_j);
          m_2 = D(i-1,j)+γ(a_i,λ);
          m_3 = D(i,j-1)+γ(λ,b_j);
          D(i,j)=Min(m_1 m_2 m_3);
          Prec(i,j)=Argmin_{i=1,2,3}(m_i); /*mémorisation du meilleur chemin
5)   δ(X,Y)=D(n,m);
6)   i=n;j=m;c=0;                    /* calcul du chemin optimal
7)   tantque Prec(i,j) ≠ 0           /* par backtrack
     faire
          chemin(c) = prec(i,j);
          c=c+1;
          selon prec(i,j)
          faire
               cas 1: i=i-1;j=j-1;
               cas 2: i=i-1;
               cas 3: j=j-1;
```

Exemple

	λ	a	a	b	a	c	d
λ	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
b	2	1	1	1	2	3	4
d	3	2	2	2	2	3	3



$$\delta(aabacd, abd) = 3$$

Les éléments en correspondance sont les substitutions marquées

d – d
b – b
a – a

Généralisation:
Méthodes de comparaison dynamique

Dynamic Time Warping (DTW), très utilisé en parole avant 1985

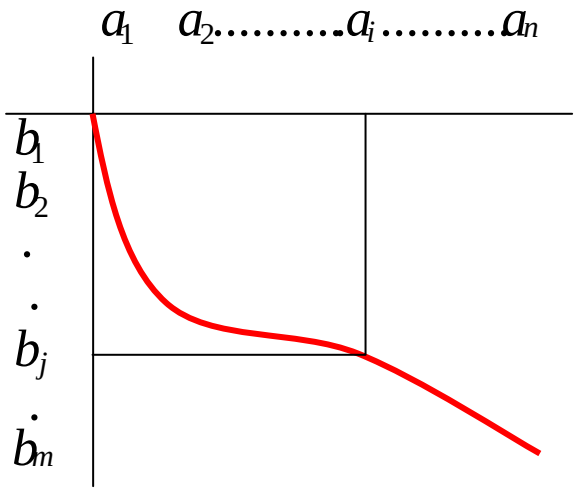
Les deux séquences à comparer ne sont pas constituées de caractères

$d(a_i, b_j)$ **est la distance élémentaire entre l'échantillon i de X et j de Y**

On supprime l'élément vide mais en revanche on autorise qu'un échantillon d'un chaîne puisse correspondre à plusieurs de l'autre en ayant des déplacements asymétriques sur les deux séquences

$$D(i,j) = \min \begin{cases} D(i-1, j-1) + 2d(a_i, b_j) \\ D(i-1, j) + d(a_i, b_j) \\ D(i, j-1) + d(a_i, b_j) \end{cases}$$

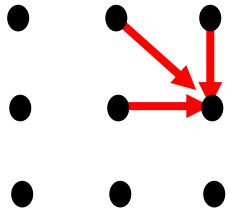
$$\delta(X,Y) = \frac{D(n,m)}{n+m}$$



Le meilleur chemin est une courbe de déformation (Warping) entre les deux chaînes.

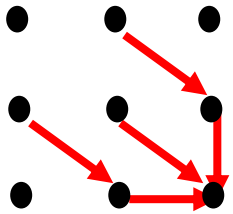
Extensions

Dans la formule de base le chemin recherché ne comporte que les 3 sauts élémentaires

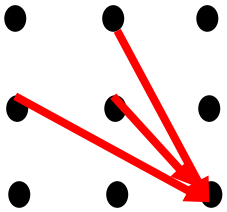


$$D(i,j)=\min\begin{cases} D(i-1,j-1)+2d(a_i,b_j) \\ D(i-1,j)+d(a_i,b_j) \\ D(i,j-1)+d(a_i,b_j) \end{cases}$$

On étend le principe en ajoutant d'autres règles locales



$$D(i,j)=\min\begin{cases} D(i-2,j-1)+\frac{1}{2}\left[d(a_{i-1},b_j)+d(a_i,b_j,)\right] \\ D(i-1,j)+d(a_i,b_j) \\ D(i-1,j-2)+\frac{1}{2}\left[d(a_i,b_{j-1})+d(a_i,b_j,)\right] \end{cases}$$



$$D(i,j)=\min\begin{cases} D(i-2,j-1)+3d(a_i,b_j,) \\ D(i-1,j-1)+2d(a_i,b_j) \\ D(i-2,j-1)+3d(a_i,b_j,) \end{cases}$$

Discussion

Quelle distance locale $d(a_i, b_j)$?

Comment l'adapter au problème étudié (adaptabilité)?

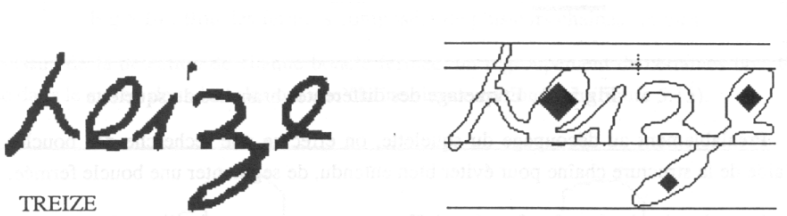
Quelles règles choisir?

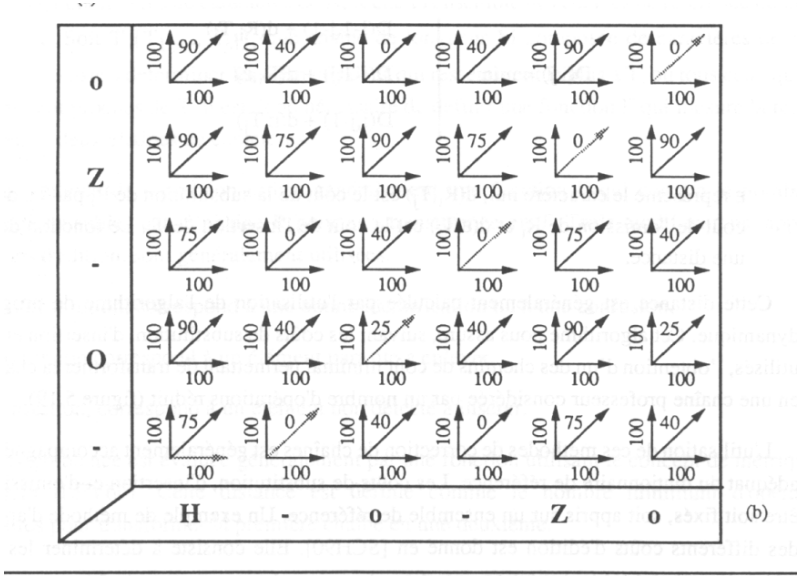
Avec quelle pondération?

Exemple: Système de reconnaissance de l'écriture

Alphabet de 21 symboles

H hampe
L boucle zone supérieure
B jambage
O grande boucle zone médiane
o boucle zone médiane
b boucle à droite d'une hampe
d boucle à gauche d'une hampe
- rien
etc…





2. Modélisation statistique de séquences

On souhaite dépasser les limitations des approches par comparaison dynamique

- Comment déterminer les modèles de séquences? Apprentissage?

- Ne pas imposer le choix d'un distance locale?

- Conserver l'idée d'aligner une séquence sur un modèle

- Distance d'une séquence à un modèle

→ **probabilité d'émission de la séquence par le modèle**
Cadre des modèles génératifs

2.1 Modèle Markovien de séquences

On souhaite construire un modèle statistique qui permette de calculer la probabilité de la séquence $O=o_1 o_2 o_3 \ldots o_T$

Soit à calculer $P(O)=P(o_1 o_2 o_3 \ldots o_T)$

La loi des probabilités conditionnelles permet d'écrire

$$P(o_1 o_2 o_3 \ldots o_T)=P(o_T | o_1 o_2 \ldots o_{T-1}) P(o_1 o_2 o_3 \ldots o_{T-1})$$

L'hypothèse Markovienne d'ordre 1 $P(o_T | o_1 o_2 \ldots o_{T-1})=P(o_T | o_{T-1})$ (dépendance à court terme)

permet d'écrire $P(o_1 o_2 o_3 \ldots o_T)=P(o_T | o_{T-1}) P(o_1 o_2 o_3 \ldots o_{T-1})$

En itérant cette décomposition pour toutes les sous-séquences on obtient

$$P(o_1 o_2 o_3 \ldots o_T)=P(o_T | o_{T-1}) P(o_{T-1} | o_{T-2}) \times \ldots \times P(o_2 | o_1) P(o_1)$$

$$P(o_1 o_2 o_3 \ldots o_T)=P(o_1) \prod_{t=2}^{T} P(o_t | o_{t-1})$$

Paramètres d'un Modèle de Markov

Soit $S=\{s_1,s_2,\ldots,s_N\}$ **l'alphabet de symboles utilisé**

Le modèle Markovien est complètement déterminé par

La matrice A **des probabilités de transition, ou matrice de transition**

$$A=\begin{pmatrix} P(s_1|s_1) & \ldots & P(s_N|s_1) \\ . & P(s_i|s_j) & . \\ P(s_1|s_N) & \ldots & P(s_N|s_N) \end{pmatrix}=\begin{pmatrix} a_{11}\ldots a_{1N} \\ . \; a_{ij} \; . \\ a_{N1}\ldots s_{NN} \end{pmatrix}$$

- Le vecteur Π **des probabilités initiales**

$$\Pi=\begin{pmatrix} P(s_1=s_{init}) \\ \\ P(s_N=s_{init}) \end{pmatrix}$$

Au total il y a $N+N^2$ **paramètres que l'on peut estimer par comptage sur une base d'exemples**

Exemple: modèles de n-grams

Tirés de (Shannon 1948)

Générateur de caractères selon un modèle du premier ordre (caractères tirés selon les fréquences de l'anglais).

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

Générateur de caractère selon un modèle tri-gram de l'anglais

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES
OF THE REPTAGIN IS REGOACTIONA OF CRE

Générateur de mots selon un modèle bi-gram de l'anglais

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER
OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT
THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Convient à la modélisation des connaissances linguistiques

Mais pas assez robuste pour modéliser la variabilité et le bruit en parole, écrit ou langage naturelle

2.2. Modèles de Markov Cachés Discrets (MMC) ou Hidden Markov Model (HMM)

On considère que la séquence observée $O=o_1 o_2 o_3 \ldots o_T$
est trop bruitée pour être modélisée par un simple modèle de Markov.

Les symboles observés appartiennent à un alphabet V **de M symboles**

$V=\{v_1, v_2, \ldots, v_M\}$ **et** $o_t \in V$

On introduit un ensemble de N états non observables (cachés) $S=\{s_1, s_2, \ldots, s_N\}$
qui vont jouer le même rôle que les symboles du modèle non caché
i.e. le modèle est Markovien sur les états :

A **: matrice de transition et entre les états**

Π **: matrice de probabilités des états initiaux**

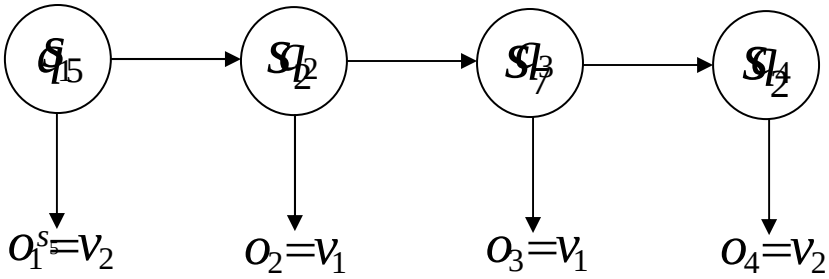
On fait le lien entre les états cachés et les symboles observés en introduisant les probabilités conditionnelles d'émission des observations en fonction des états

$$B=\left(P(v_k|s_j)\right)=\begin{pmatrix} P(v_1|s_1) & P(v_2|s_1) & \ldots & P(v_M|s_1) \\ \vdots & \vdots & \ldots & \vdots \\ P(v_1|s_N) & P(v_2|s_N) & \ldots & P(v_M|s_N) \end{pmatrix}$$

B : matrice d'émission

Représentation graphique dans le temps

La séquence Q n'est pas connue, mais la séquence O est mesurée



C'est un modèle génératif : les états génèrent les observations

Il permet:

- **de calculer la probabilité d'émission de la séquence observée**
- **de déterminer la meilleure séquence d'états correspondant à la séquence d'observation: $q_1=s_5$, $q_2=s_2$ etc…**

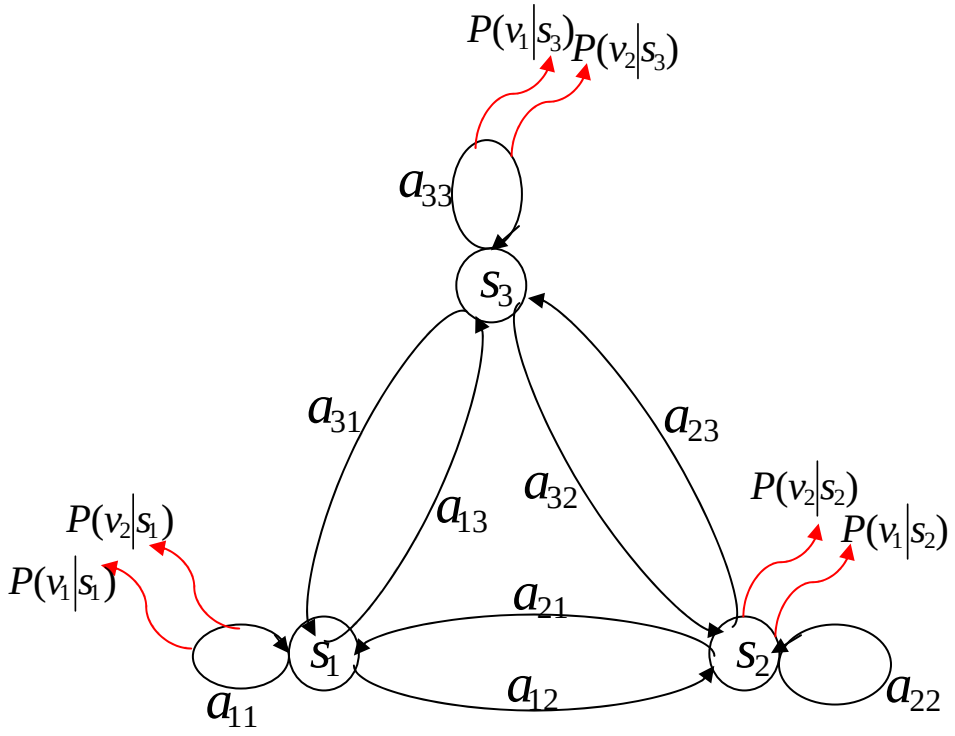
De plus il existe des techniques d'apprentissage des paramètres à partir de séquences d'exemples. Adaptabilité du modèle!

Représentation Graphique des dépendances probabilistes

C'est un automate d'états fini qui émet une séquence d'observations O à partir d'une séquence d'états Q que l'on ne connaît pas (cachés)

$$O = o_1 o_2 o_3 \ldots o_T$$

$$Q = q_1 q_2 q_3 \ldots q_T$$



$$S = \{s_1, s_2, s_3\}$$

$$V = \{v_1, v_2\}$$

$$o_t \in V$$

$$q_t \in S$$

Paramètres d'un MMC

Un Modèle de Markov Caché est défini par trois matrices de probabilités

Probabilités initiales

$$\Pi = (\pi_i) = (P(q_1 = s_i))_{N,1}$$

Matrice de transitions

$$A = \begin{pmatrix} P(s_1|s_1) & \dots & P(s_N|s_1) \\ . & P(s_i|s_j) & . \\ P(s_1|s_N) & \dots & P(s_N|s_N) \end{pmatrix} = \begin{pmatrix} a_{11} \dots a_{1N} \\ . \; a_{ij} \; . \\ a_{N1} \dots s_{NN} \end{pmatrix}_{(N,N)}$$

Matrice d'émission

$$B = (P(v_k|s_j)) = \begin{pmatrix} P(v_1|s_1) \, P(v_2|s_1) \dots P(v_M|s_1) \\ . \quad . \quad \dots \quad . \\ P(v_1|s_N) P(v_2|s_N) \dots P(v_M|s_N) \end{pmatrix}_{(N,M)}$$

Probabilité d'émission d'une séquence

On veut calculer la probabilité d'émission de la séquence $O=o_1 o_2 o_3 \ldots o_T$
par le modèle M **soit:** $P(O|M)$

1. Supposons connue la séquence d'états cachée $Q=q_1 q_2 q_3 \ldots q_T$

Alors $$P(O,Q|M) = P(O|Q,M)P(Q)$$

modèle d'attache aux données **modèle de connaissances**

en faisant l'hypothèse d'indépendance conditionnelle des observations entre elles et avec les états passés

$$P(O|Q,M) = p(o_1|q_1)\,p(o_2|q_2)\ldots p(o_T|q_T)$$

L'hypothèse Markovienne

$$P(Q) = \prod P(q_t|q_{t-1})$$

Finalement

$$P(O,Q|M) = \pi_{q_1}\, p(o_1|q_1)\,p(q_2|q_1)\,p(o_2|q_2)\ldots p(q_T|q_{T-1})\,p(o_T|q_T)$$

Probabilité d'émission d'une séquence

2. Mais la séquence d'observations est émise quelle que soit la séquence d'états

hors pour une séquence de longueur T il y a N^T séquences d'états possibles.

Donc $$P(O)=P(O|M)=\sum_{i=1}^{N^T} P(O,Q_i|M)$$ **mais complexe à calculer!!**

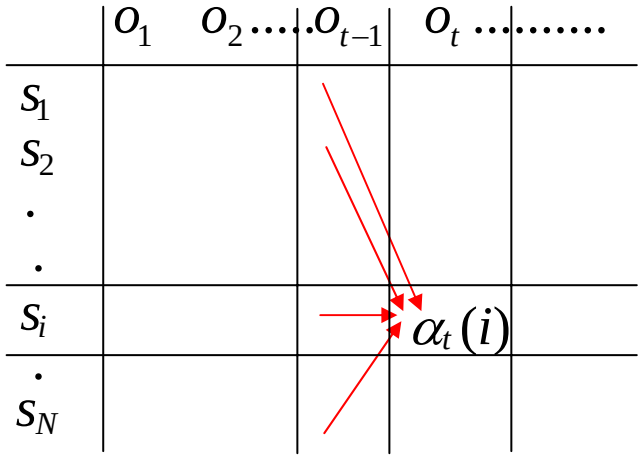
Algorithme forward

Il y a N^T chemins possibles, mais on peut factoriser les sous-chemins communs

On définit la variable forward par $\alpha_t(i) = P(o_1 o_2 o_3 \ldots o_t, q_t = i | M)$

C'est la probabilité d'être dans l'état i à l'instant t en ayant observé les premières observations jusqu'à t-1 quel que soit l'état qui leur ait associé

Alors



$$\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} b_i(o_t)$$

Algorithme forward

Finalement:

$$P(O|M)=\sum_{i=1}^{N}P(o_1 o_2 ... o_T, q_T=i)=\sum_{i=1}^{N}\alpha_T(i)$$

Algorithme Forward (en avant)

Initialisation

$$\alpha_1(i)=\pi_i$$

Récursion pour t=2:N

$$\alpha_t(i)=\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji}b_i(o_t)$$

Terminaison

$$P(O|M)=\sum_{i=1}^{N}\alpha_T(i)$$

. Probabilité que le modèle émette la séquence observée
. Distance au modèle

Quelle est la séquence d'états cachés?

On veut trouver la séquence Q optimale qui « explique » le mieux les données.

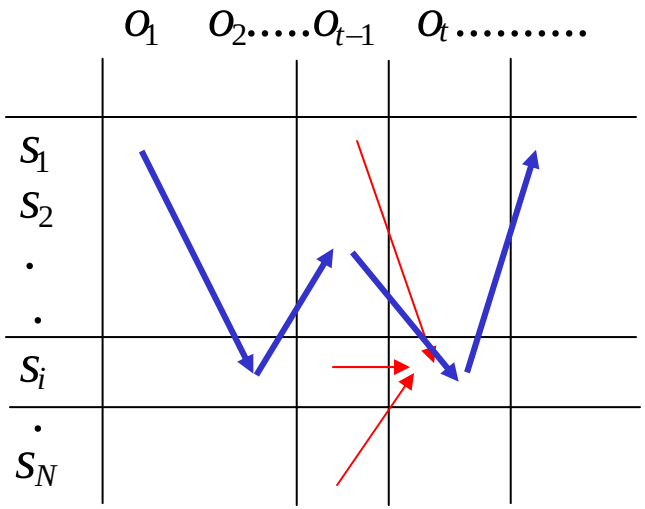
Parmi les N^T séquences possibles on veut déterminer la plus probable

$$Q^* = \underset{i=1,..,N^T}{\operatorname{argmax}}\left(P(Q_i \mid O,M)\right) = \underset{i=1,..,N^T}{\operatorname{argmax}}\left(\frac{P(O,Q_i \mid M)}{P(O)}\right) = \underset{i=1,..,N^T}{\operatorname{argmax}}\left(P(O,Q_i \mid M)\right)$$

On a intérêt à factoriser les sous-chemins.

Même principe que forward mais en transformant la somme en max et en mémorisant le chemin pris (programmation dynamique)

Algorithme de viterbi (1967)



Recherche du meilleur chemin dans le treillis d'états possibles

Chaque pas est affecté d'une probabilité

Les probabilités se multiplient entre elles, leurs logarithmes s'ajoutent

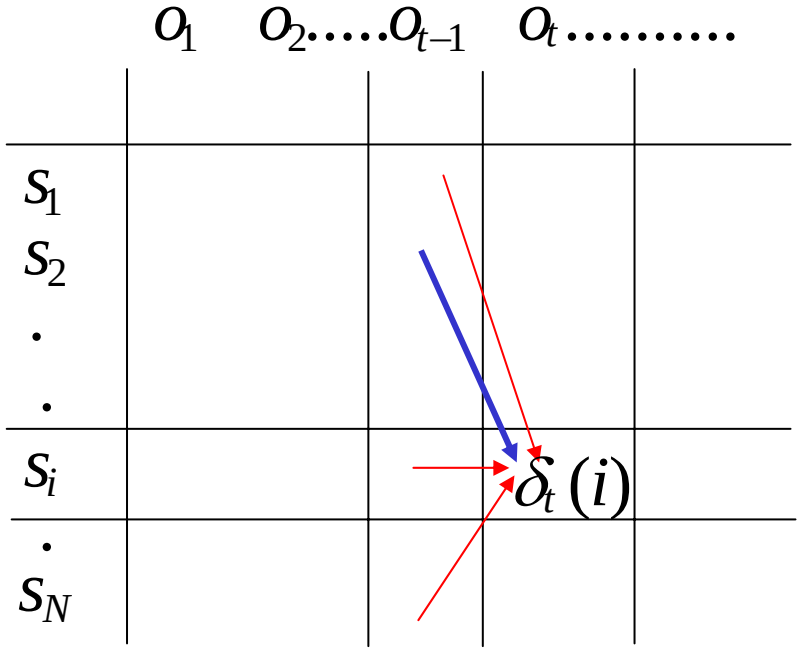
Le problème est décomposable et à fonction de coût monotone

Se résout par programmation dynamique

Algorithme de viterbi (1967)

On introduit la variable $\delta_t(i) = \max_{q_1 q_2 \ldots q_{t-1}} \left(P(o_1 o_2 o_3 \ldots o_t, q_1 q_2 q_3 \ldots q_t = i | M) \right)$

probabilité du meilleur sous-chemin conduisant à l'état i à l'instant t quels que soient les chemins précédents



Alors $\delta_t(i) = \max_j \left(\delta_{t-1}(j) a_{ji} \right) b_i(o_t)$

Algorithme de viterbi (1967)

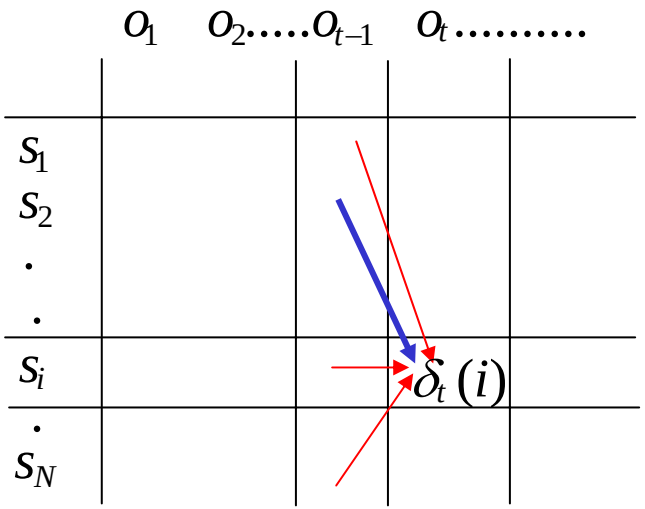
Finalement

$$P^* = \max_j \left(\delta_T(j) \right)$$

Pour retrouver la séquence d'états optimale

on introduit une variable qui mémorise le chemin localement optimal emprunté

$$\psi_t(i) = \arg\max_j \left(\delta_{t-1}(j) a_{ji} \right)$$



Algorithme de Viterbi

Initialisation

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \le i \le N$$

$$\psi_1(i) = 0$$

Récursion

$$\delta_t(i) = \max_j \left(\delta_{t-1}(j) a_{ji} \right) b_i(o_t) \qquad 1 \le i \le N \qquad 2 \le t \le T$$

$$\psi_t(i) = \operatorname{argmax}_j \left(\delta_{t-1}(j) a_{ji} \right) \qquad 1 \le i \le N \qquad 2 \le t \le T$$

Terminaison

$$P^* = \max_j \left(\delta_T(j) \right)$$

$$q_T^* = \operatorname{argmax}_i \left(\delta_T(i) \right)$$

Calcul du meilleur chemin (backtracking)

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

Version beaucoup plus efficace en passant en log de proba car produits deviennent sommes

Algorithme de Viterbi: Exemple

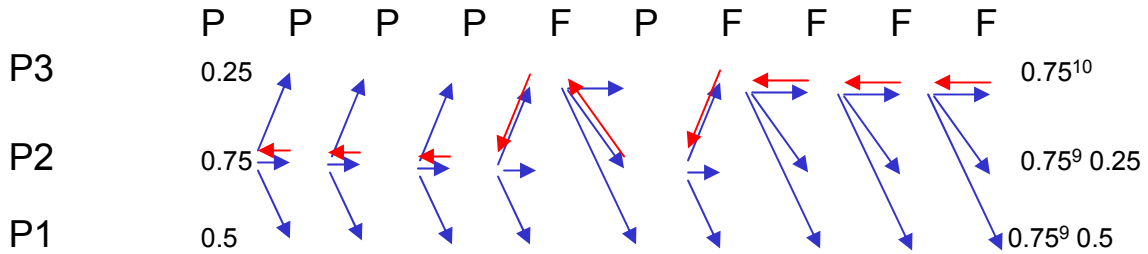
On joue à Pile ou Face avec trois pièces biaisées

	P1	P2	P3
Pile	0.5	0.75	0.25
Face	0.5	0.25	0.75

Après chaque lancé de pièce le choix de la nouvelle pièce est équi-probable (1/3)
Les probabilités initiales des pièces sont identiques (1/3)

On observe la séquence O = PPPPFPFFFF

Quelles pièces a-t-on utilisé à chaque tirage?



en bleu les meilleurs chemins locaux

La séquence de pièces la + probable est Q* = 2222323333

Apprentissage d'un MMC (1)

Comment déterminer les paramètres d'un MMC?

A partir d'un ensemble d'exemples de séquences d'observations on va entraîner le modèle à générer de mieux en mieux les exemples.

Mesure de l'aptitude du modèle à générer une séquence:

$P(O|M)$ vraisemblance de la séquence conditionnellement au modèle

Critère d'apprentissage : maximiser la vraisemblance des exemples

Pour un modèle non caché avec n séquences exemples:

$$V=\sum_{i=1}^{n} P(O^i|M)=\sum_{i=1}^{n}\left(P(o_1^i)\prod_{t=2}^{T} P(o_t^i|o_{t-1}^i)\right)$$

La solution du problème est évidente:

$$a_{ij}=\frac{n_{ij}}{n_i}=\frac{nb\ couple\ ij}{nb\ symbole\ i}$$

$$\pi_i=\frac{n_i}{n}=\frac{nb\ symbole\ i\ débute}{n}$$

Solution 1 : Apprentissage par Viterbi

Problème: on ne connaît pas les états cachés $O=o_1 o_2 o_3 \ldots o_T$

les données sont incomplètes

Solution : estimer les probabilités à partir des meilleures séquences d'états que l'on peut associer à chaque séquence d'observation grâce à viterbi.

Algorithme EM (Expectation-Maximisation)

Etape E: Estimation des données manquantes

$O=o_1 o_2 o_3 \ldots o_T$

$Q^*=q_1^* q_2^* q_3^* \ldots q_T^*$

Etape M: calcule des paramètres qui maximisent la vraisemblance des observations (se ramène au cas non caché)

itérer E puis M tant que la vraisemblance des données augmente

L'algorithme converge vers un optimum local fortement dépendant du modèle initial

Solution 2 : Algorithme de Baum-Welch (1)

Même idée que précédemment mais on prend en compte tous les chemins, pas uniquement le meilleur

Algorithme EM (Expectation-Maximisation)

Etape E: d'estimation des données manquantes à partir d'un premier modèle

$$O = o_1 o_2 o_3 \ldots o_T$$

$$P(q_t = i \mid M)$$

Etape M: calcule des paramètres qui maximisent la vraisemblance des observations en tenant compte de tous les chemins

$$a_{ij} = \frac{Estimation\ \ probabilité\ \ couple\ \ ij}{Estimation\ \ probabilité\ \ symbole\ \ i}$$

$$b_i(k) = \frac{Estimation\ \ probabilité\ \ observation\ \ k\ \ dans\ l'état\ \ j}{Estimation\ \ probabilité\ \ état\ \ j}$$

$$\pi_i = \frac{Estimation\ \ nbfois\ \ état\ \ i\ \ débute}{n}$$

itérer E puis M tant que la vraisemblance augmente

L'algorithme converge vers un optimum local moins dépendant du modèle initial

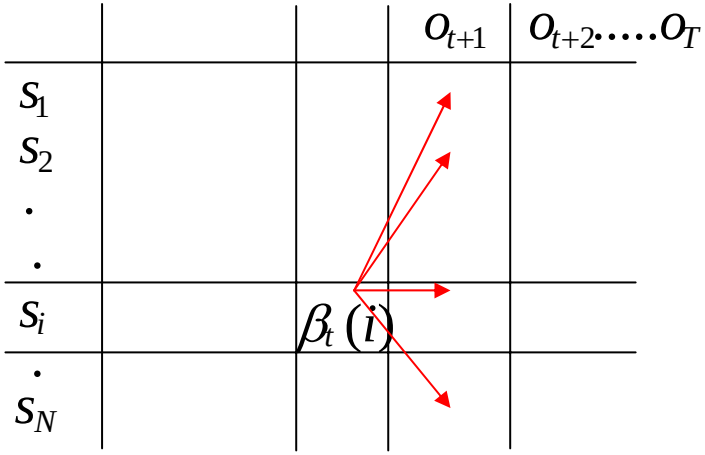
Algorithme de Baum-Welch (2)

On introduit la variable Backward

$$\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T \mid q_t = i, M)$$

Alors $\beta_t(j) = \sum_{i=1}^{N} a_{ji} b_i(o_{t+1}) \beta_{t+1}(i)$ $t = T-1, T-2, \ldots, 1$

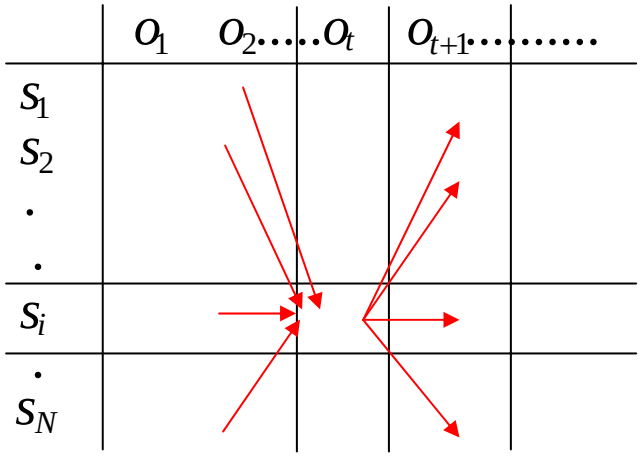
se calcule en revenant en arrière…(« backward »)



On a alors les formules suivantes:

$$P(O, q_t = i \mid M) = \alpha_t(i) \beta_t(i)$$

Probabilité d'être dans l'état i à l'instant t

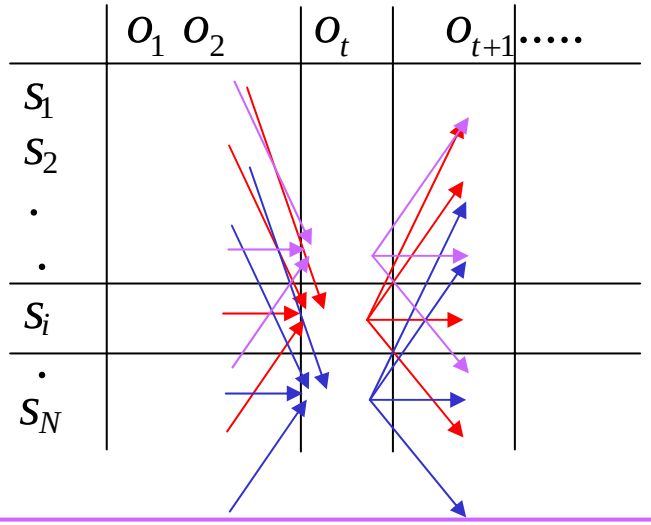


Algorithme de Baum-Welch (2)

Finalement

En sommant sur tous les états possible à l'instant t on obtient

$$P(O|M)=\sum_{i=1}^{N}\alpha_T(i)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)$$



Algorithme de Baum-Welch (3)

on introduit les variables intermédiaires suivantes

$$\gamma_t(i) = P(q_t = i | O, M) = \frac{P(q_t = i, O | M)}{P(O | M)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

et

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, M) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | M)}$$

alors

$$\sum_{t=1}^{T-1} \gamma_t(i) = \textit{Estimation de la probabilité de transition à partir de i}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \textit{Estimation de la probabilité de transition de i vers j}$$

Algorithme de Baum-Welch (4)

Alors les formules de ré-estimation s'écrivent:

$$\pi_i = \gamma_1(i)$$

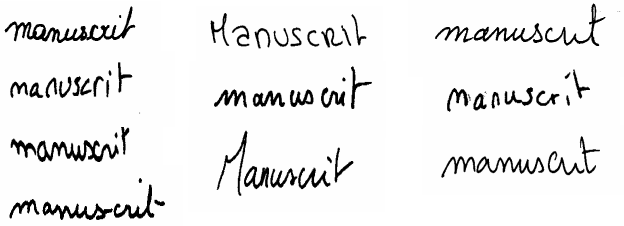
$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}$$

$$b_j(k) = \frac{\sum_{t=1, o_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Apprentissage sur plusieurs observations

Dans bien des situations, une observation comporte peu de données :
mot, phrase,…

Or l'apprentissage nécessite de disposer d'un nombre important d'exemples (plusieurs centaines en général)



Soit $O=[O^1,O^2,\ldots,O^K]$ **l'ensemble des K séquences d'observation**
L'estimation des paramètres du modèle se fait en cumulant les variables sur tous les exemples

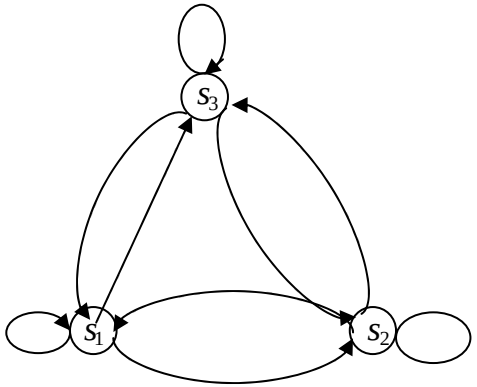
$$\pi_i = \frac{1}{K}\sum_{k=1}^{K}\gamma_1^k(i)$$

$$a_{ij} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\xi_t^k(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\gamma_t^k(i)} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\alpha_t^k(i)a_{ij}b_j(o_{t+1}^k)\beta_{t+1}^k(j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\alpha_t^k(i)\beta_t^k(i)}$$

$$b_j(l) = \frac{\sum_{k=1}^{K}\sum_{t=1,o_t=v_l}^{T_k-1}\gamma_t^k(j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k}\gamma_t^k(j)}$$

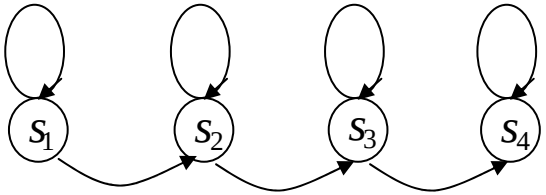
Topologies d'un MMC

Ergodique



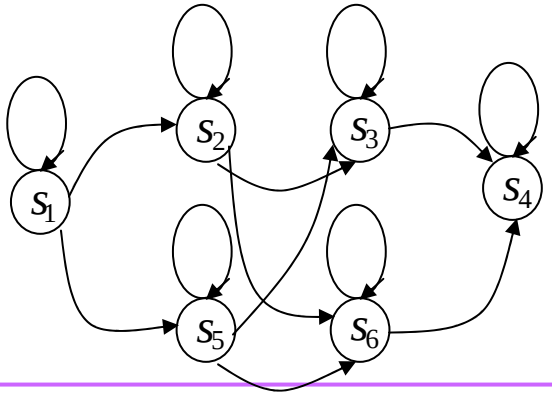
$$\Pi=\begin{pmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{pmatrix} \qquad A=\begin{pmatrix}a_{11}a_{12}a_{13}a_{14}\\ a_{21}a_{22}a_{23}a_{24}\\ a_{31}a_{32}a_{33}a_{34}\\ a_{41}a_{42}a_{43}a_{44}\end{pmatrix}$$

Gauche droite



$$\Pi=\begin{pmatrix}\pi_1\\ 0\\ 0\\ 0\end{pmatrix} \qquad A=\begin{pmatrix}a_{11}a_{12} & 0 & 0\\ 0 & a_{22}a_{23} & 0\\ 0 & 0 & a_{33}a_{34}\\ 0 & 0 & 0 & a_{44}\end{pmatrix}$$

Parallèle



Modalités d'Apprentissage d'un MMC (1)

Le critère d'apprentissage d'un MMC est la maximisation de la vraisemblance des données d'apprentissage

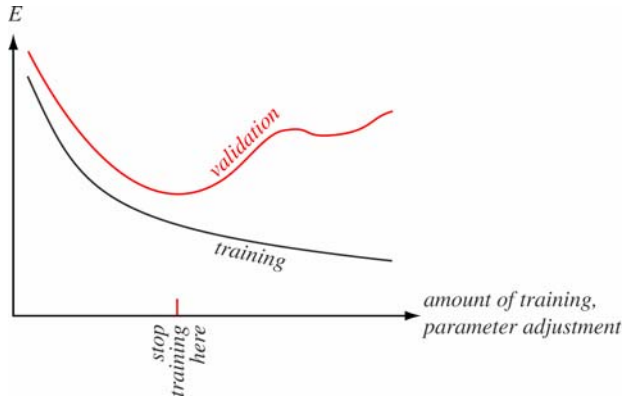
Sur-Apprentissage :
le modèle se spécialise sur l'ensemble d'apprentissage au risque de dégrader son comportement sur un ensemble de données qu'il n'a jamais vu

Cross-validation :
On cherche à éviter se phénomène en construisant deux sous-ensembles distincts à partir de l'ensemble D d'apprentissage:

D = A + V = Apprentissage + Validation

Apprentissage sur A

Critère d'arrêt :
minimum de la vraisemblance sur V



Extension :
Répéter plusieurs cross-validations sur plusieurs sous-ensembles A_i + V_i

Modalités d'Apprentissage d'un MMC (2)

Conditions initiales:

La qualité du modèle appris dépend du modèle initial choisi

- A et π influencent peu l'apprentissage

- B est déterminante

- répéter plusieurs initialisations aléatoires pour A et π

- initialisation par l'utilisateur de B ou à l'aide d'une procédure spécifique

Complexité du modèle:

Nombre de paramètres libres à estimer

- minimiser la complexité du modèle pour une meilleure estimation

solution 1: peu d'états, petit alphabet

solution 2: introduire des zéros structurels
(connaissances imposées par le concepteur)

solution 3: introduire des états liés qui partagent la même matrice d'émission des observations

solution 4: apprentissage de la structure

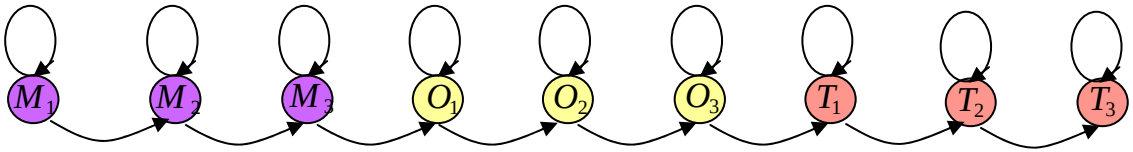
Applications à l'écriture

Modélisation analytique

- 26 modèles de caractères suffisent pour travailler sur un lexique quelconque

- 1 HMM par caractère, généralement gauche -droite

- le modèle d'un mot du lexique est construit en concaténant les modèles de caractères



$$A = \begin{pmatrix} m_{11} m_{12} & & & & & & & & \\ & m_{22} m_{23} & & & & & & & \\ & & m_{33} mo & & & & & & \\ & & & o_{11} o_{12} & & & & & \\ & & & & o_{22} o_{23} & & & & \\ & & & & & o_{33} ot & & & \\ & & & & & & t_{11} t_{12} & & \\ & & & & & & & t_{22} t_{23} & \\ & & & & & & & & t_{33} \end{pmatrix}$$

- la reconnaissance met en compétition tous les modèles de mots, le modèle ayant la plus grande vraisemblance "gagne"

$$M^* = \arg\max_i \left(P(M_i / O) \right) = \arg\max_i \left(\frac{P(M_i, O)}{P(O)} \right)$$

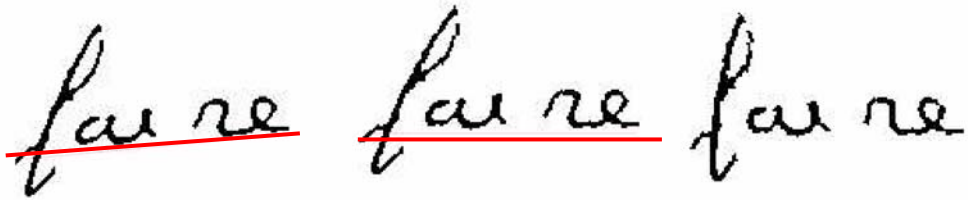
MAP :maximum *a posteriori*

$$M^* = \arg\max_i \left(\frac{P(O|M_i) P(M_i)}{P(O)} \right) \propto \arg\max_i \left(P(O|M_i) \right)$$

MV : **maximum de vraisemblance**
si mots équi-probables

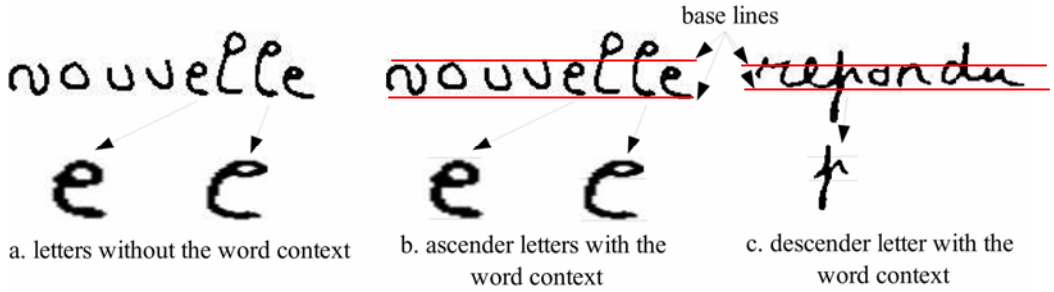
Pré-traitements

Normalisation de l'écriture



redressement horizontal de la ligne d'écriture
puis vertical des caractères

Contextualisation de l'information



a. letters without the word context

b. ascender letters with the word context

c. descender letter with the word context

Segmentation en graphèmes

Surtout utilisé en reconnaissance **Hors-ligne** pour aider la mise en correspondance d'une représentation 2D avec un modèle 1D

trame = graphème

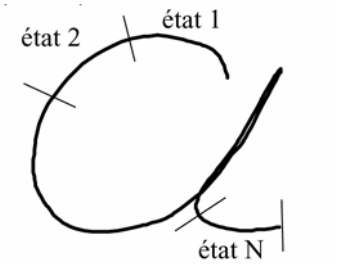
Reconnaissance en-ligne

Le signal est la position du stylo au cours du temps: (x_i,y_i,t_i)
on enregistre l'ordre du tracé

Stylo électronique = tracé dynamique

Pen Based Computing:
Interfaces Mobiles, PDA (personnal dedicated assistant)

La reconnaissance recherche la meilleure mise en correspondance d'une représentation mono-dimensionnelle avec un model 1D



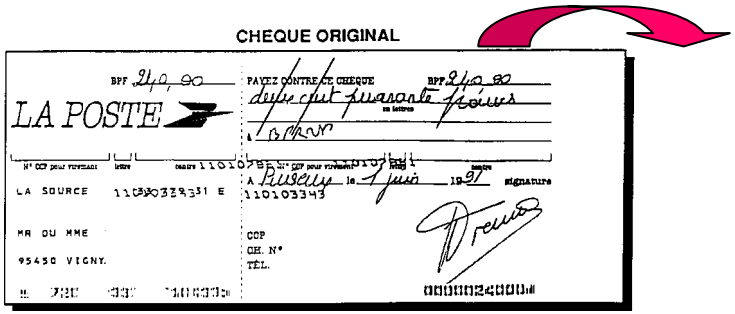
Caractéristiques utilisées en chaque point (trame):
pente, pentes des échantillons voisins, déplacement relatif au précédent, au suivant, courbure dans le voisinage…

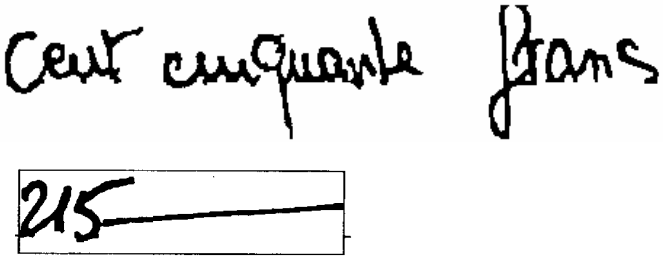
Reconnaissance Hors-Ligne

Le signal est une image: papier électronique = images document
on a perdu l'ordre du tracé → superposition de traits

Applications: Courrier, Chèques, Formulaires

Pré-traitements pour
Localiser l'information à reconnaître





Caractéristiques utilisées

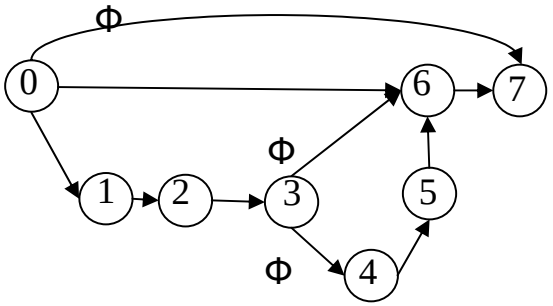
sans-segmentation:
chaque trame est caractérisée par point particulier du squelette ou du contour. Point triple, fin de trait haut, bas, boucle…

avec segmentation:
chaque graphème est caractérisé par la présence d'une caractéristique particulière dans un certain contexte. Hampe, jambage, occlusion…

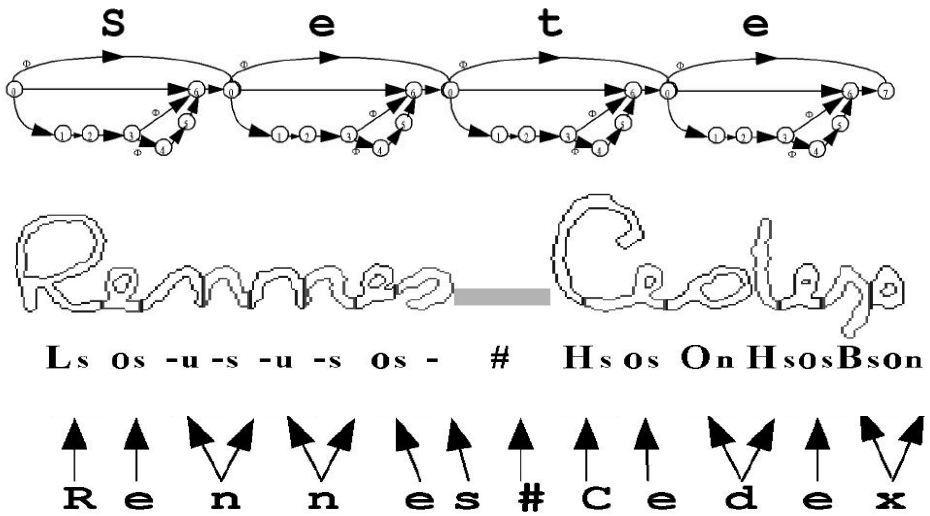
Reconnaissance Hors-Ligne avec Segmentation (El-Yacoubi 96)

Modèle de caractère

émission sur les transitions
1 caractère au plus sur trois segments consécutifs
T06 caractère sur 1 segment, T67 pt de segmentation
T01, T23 et T56, 1er, 2nd et 3em segment
T12 et T45 points de segmentation entre les segments
T07 le caractère est absent



Modèle de mots et de phrases



Alphabet de 21 symboles

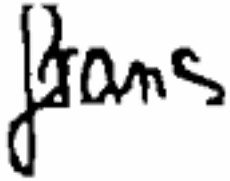
H hampe
L boucle zone supérieure
B jambage
O grande boucle zone médiane
o boucle zone médiane
b boucle à droite d'une hampe
d boucle à gauche d'une hampe
- rien
etc…

L'apprentissage de sous-unités

Les modèles analytiques de l'écriture s'appuient sur les modèles des 26 caractères.

Il faut donc apprendre les 26 modèles de caractères

mais la constitution de bases de caractères est fastidieuse, parfois délicate.
Si l'on ne retient que les situations qui ne posent pas de pb alors la base n'est plus représentative



"Embedded Training" = " apprentissage embarqué "

Apprendre les modèles de caractères dans des mots

```
Tantque vraisemblance augmente faire
début
            pour chaque mot de la base d'apprentissage faire
            début
                        - construire le HMM du mot en concaténant les modèles caractères
                        - calculer les variables forward et backward
                        - cumuler la vraisemblance de l'exemple
                        - mettre à jour les variables impliquées par les caractères du modèles
                                    Σ_{t=1}^{T_k-1} ξ_t^k(i,j)   et   Σ_{t=1}^{T_k-1} γ_t^k(i)
            fin
            - ré-estimer les modèles de chaque caractère
fin
```

L'algorithme converge vers l'apprentissage d'un modèle "moyen " de caractère qui tient compte du bruit et des situations ambiguës

Détermination des observations

Les observations constituent un alphabets de symboles déterminé

- Soit par le concepteur (voir système de El Yacoubi)

-Soit à l'aide d'une procédure de classification non supervisée des observations sur lesquelles on effectue un ensemble de mesures

soit $O = o_1 o_2 o_3 \ldots o_T$ une séquence d'observations (des graphèmes par exemple) sur chaque observation élémentaire on effectue une ensemble de **n** mesures
o_i est représentée d'ans un espace à **n** dimensions
on choisit une distance (euclidienne par exemple)

On détermine le meilleur alphabet de N symboles

algorithme des k-moyennes par exemple

```
Initialiser aléatoirement N vecteurs centroïdes μ1,μ2 ,…, μM
répéter
            affecter chaque vecteur oi au centroïde le plus proche
            calculer les nouveaux centres des groupes obtenus μ1,μ2 ,…, μM
Tantque les centres sont modifiés
```

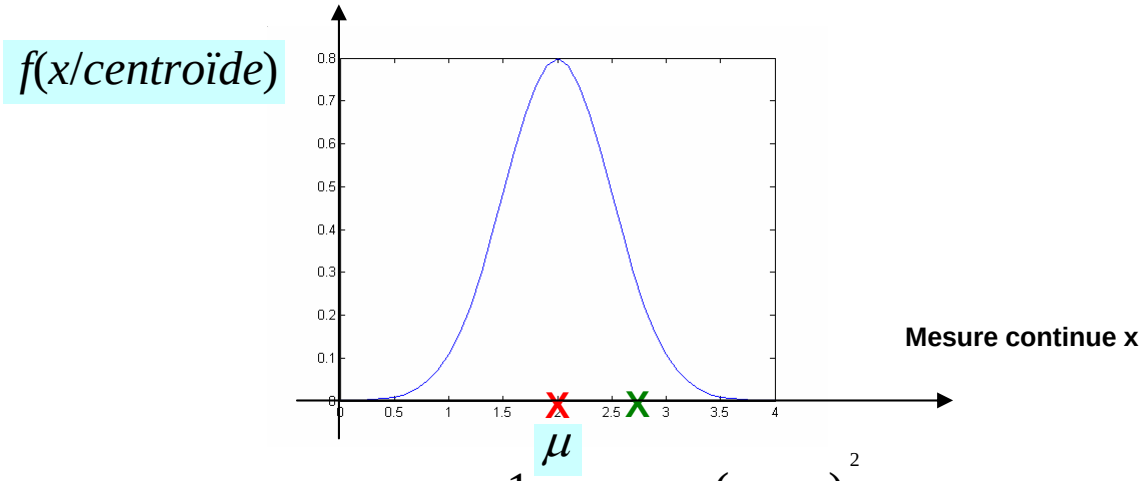
Inconvénient: la procédure est indépendante de l'apprentissage du HMM

Modélisation des observations continues par Modèles de Mélanges

On souhaite modéliser plus finement les données en tenant compte de la distance de la mesure au centroïde le plus proche.

Exemple sur 1 dimension:

Plus la mesure X est proche du centroïde X plus elle est vraisemblable



Vraisemblance $f(x/centroïde) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(x-\mu)^2}{2\sigma^2}$

nulle si distance grande
grande si distance nulle

Moyenne = position du centroïde μ

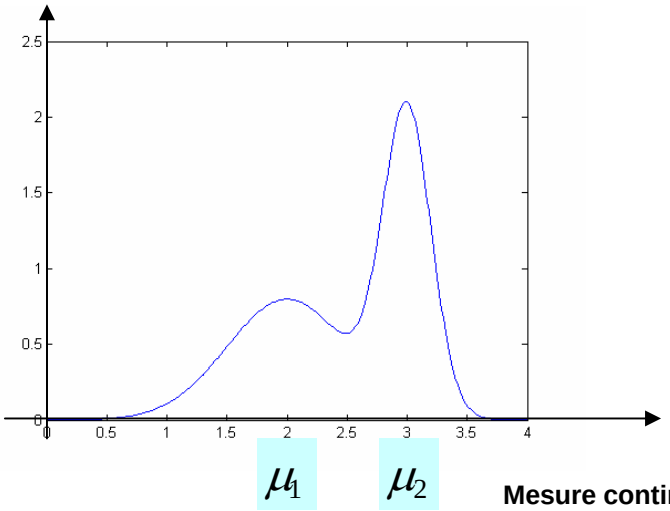
Variance = dispersion autour du centroïde σ

Modélisation des observations continues par Modèles de Mélanges

Dans le cas non Gaussien, on approche la distribution des données par un mélange de Gaussiennes

Exemple à 2 Gaussiennes

$f(x/\text{modèle})$



Vraisemblance grande si données proche d'un centroïde
On somme les vraisemblances des Gaussiennes

$$f(x/\text{modèle}) = \frac{1}{2} f(x/\text{centroïde}1) + \frac{1}{2} f(x/\text{centroïde}2)$$

Si les centroïdes ne sont pas équiprobables on pondère par les probabilités *a priori* des centroïdes

$$f(x/\text{modèle}) = P(C_1) f(x/C_1) + P(C_2) f(x/C_2)$$

Modèles de Mélanges: Position du Problème

On dispose d'un ensemble de N mesures
On veut le modéliser par un modèle de Mélange de M Gaussiennes

$$f(x) = \sum_j f(x/C_j)P(C_j) = \sum_{j=1}^{M} P(C_j)\frac{1}{\sigma_j\sqrt{2\pi}}\exp\left(\frac{-(x-\mu_j)^2}{2\sigma_j^2}\right)$$

Quel est le meilleur modèle?
Comment déterminer les paramètres du meilleur modèle?

Il y a 3M paramètres $\mu_{1\le j\le M}$ $\sigma_{1\le j\le M}$ $P(C_{1\le j\le M})$

Stratégie de résolution:
Le meilleure modèle est celui pour lequel les données ont la plus grande vraisemblance

Critère du maximum de vraisemblance

Modèle de Mélanges: Un modèle génératif

Le modèle de mélange modélise la source qui a généré les mesures
On peut considérer qu'il y a M sources Gaussiennes

Pour générer une mesure x on procède en deux étapes

1- on choisit au hasard l'une des composantes en respectant les $P(C_j)$

2- on génère la mesure x en utilisant la densité $f(x/C_j)$ **sélectionnée**

Inversement, si l'on a la mesure x on peut calculer la source qui a le plus probablement généré la mesure

$$C_k = \max_{C_j}\left(P(C_j/x)\right)$$

Avec

$$P(C_j/x) = \frac{f(x,C_j)}{f(x)} = \frac{f(x/C_j)P(C_j)}{\sum_{j=1}^{M} f(x/C_j)P(C_j)}$$

probabilité *a posteriori* de chaque source

Recherche du Meilleur Modèle

On cherche le modèle qui maximise la vraisemblance des N mesures $x_{1 \le i \le N}$

C'est le produit des vraisemblances des mesures: $Critère = \prod_{i=1}^{N} f(x_i)$

De manière équivalente on prend le log de ce critère (log-vraisemblance)

$$\mathbf{LV} = \sum_{i=1}^{N} \log(f(x_i))$$

A MAXIMISER

Cas des données complètes

Il correspond au cas où l'on connaît pour chaque mesure $x_{1 \le i \le N}$

La source Gaussienne C_j **qui l'a généré**

Soit $I_j = \{x_i / x_i \in C_j\}$ **l'ensemble des mesures générées par la source j**

Alors le calcul des paramètres de la source qui maximise la vraisemblance est immédiat

On a:

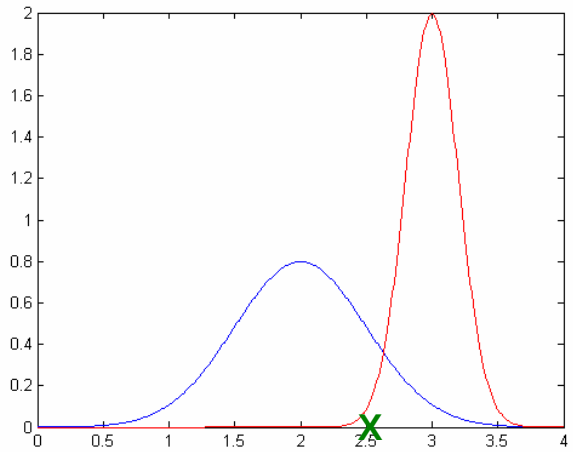
$$P(C_j) = \frac{Card(I_j)}{N}$$

$$\mu_j = \frac{1}{Card(I_j)} \sum_{x_i \in I_j} x_i$$

$$\sigma_j^2 = \frac{1}{Card(I_j)} \sum_{x_i \in I_j} (x_i - \mu_j)^2$$

Cas des données incomplètes

C'est le cas général où on ne sait pas quelle source Gaussienne a pu générer chaque mesure $x_{1 \le i \le N}$



On essaie de se ramener au cas des données complètes
On ajoute une variable supplémentaire associée à chaque mesure (x_i, y_i)

$y_i = j \in [1, M]$ **signifie que la donnée est générée par la source n° j**

Cas des données incomplètes

On note l'ensemble des paramètres que l'on cherche $\Theta = \{P(C_j), \mu_j, \sigma_j\}_{1 \le j \le M}$

On note l'ensemble des données mesurées $X = \{x_{1 \le i \le N}\}$

On note l'ensemble des données inconnues $Y = \{y_{1 \le i \le N}\}$

Si les données manquantes étaient connues le critère de vraisemblance s'écrirait

$$\mathbf{LV} = \log(f(X,Y/\Theta) = \log\left(\prod_{1 \le i \le N} f(x_i, y_i/\Theta)\right)$$

$$\mathbf{LV} = \sum_{1 \le i \le N} \log\left(f(x_i/y_i)P(C_{y_i})\right)$$

On pourrait alors optimiser ce critère et trouver les paramètres qui le Maximisent

On s'y ramène en introduisant une étape d'Estimation des données manquantes

C'est le principe de l'algorithme EM!!

Algorithme EM

On note l'ensemble des paramètres que l'on cherche $\Theta=\{P(C_j),\mu_j,\sigma_j\}_{1\le j\le M}$

On note l'ensemble des données complètes $Z=(X,Y)$

On ne connaît ni Y ni Θ donc on ne peut pas calculer $f(Z/\Theta)$

Supposons que l'on ait un modèle approchant le modèle que l'on cherche

On le note Θ^g

On va estimer les données manquantes avec le modèle approché
RAPPEL : Estimation conditionnelle
Soit Y une variable aléatoire de densité $f_Y(y)$

L'estimation d'une fonction $h(Y)$ **est** $E(h(Y))=\int h(y)f_Y(y)dy$

Si on connaît la distribution de Y conditionnellement à X
L'estimation conditionnelle de $h(Y)$ **que l'on note** $E(h(Y)/X)$

est $E(h(Y)/X)=\int h(y)f_{Y/X}(y/x)dy$

Algorithme EM

Application: On va estimer la vraisemblance des données complètes conditionnellement au modèle Θ^g

soit $E(\mathbf{LV}/\Theta^g)=E\left[\log\left(f(Z/\Theta)\right)/\Theta^g\right]=\mathbf{Q}(\Theta/\Theta^g)$

qui s'écrit $\mathbf{Q}(\Theta/\Theta^g)=\int\log\left(f(X,y/\Theta)\right)f_Y\left(y/\Theta^g\right)dy$

On doit d'abord Estimer $f_Y\left(y/\Theta^g\right)$ **les données manquantes** **ETAPE E**

On doit ensuite trouver un modèle qui Maximise le critère **ETAPE M**

Puis on retourne à l'étape E avec $\Theta^g=\Theta^{g+1}$

Tant que l'accroissement du critère est significatif

EM appliqué au modèle de Mélange

On suppose connu un modèle initial $\Theta^g=\left\{P^g(C_j),\mu_j^g,\sigma_j^g\right\}$

Alors on peut estimer pour chaque mesure les données manquantes qui sont les probabilités *a posteriori* de chaque source connaissant la mesure

C'est l'étape E

$$P(y_i=j)=P(C_j/x_i)=\frac{f(x_i,C_j)}{f(x_i)}=\frac{f(x_i/C_j)P(C_j)}{\sum_{j=1}^{M} f(x_i/C_j)P(C_j)}$$

On détermine ensuite le meilleur modèle

C'est l'étape M: on montre en dérivant le critère $\mathbf{Q}(\Theta/\Theta^g)$ **par rapport aux paramètres que h**

$$\mu_j=\frac{\sum_{i=1}^{N} x_i P(y_i=j)}{\sum_{i=1}^{N} P(y_i=j)}$$

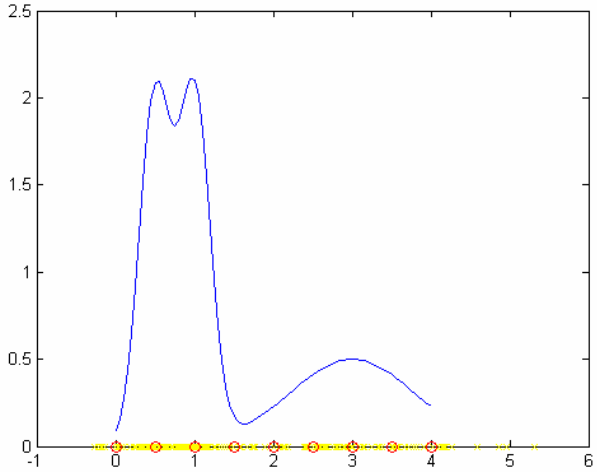
$$\sigma_j=\frac{\sum_{i=1}^{N} (x_i-\mu_j)^2 P(y_i=j)}{\sum_{i=1}^{N} P(y_i=j)}$$

EM : illustration 1D

Modèle à 3 Gaussiennes

centres = 3 0.5 1

variance = 0.8 0.2 0.2



$$f(x/C_j)$$

Vraisemblance de chaque Gaussienne

0.0016	0.4775	0.0732
0.0090	0.8921	0.4775
0.0366	0.4775	0.8921
0.1093	0.0732	0.4775
0.2387	0.0032	0.0732
0.3815	0.0000	0.0032
0.4460	0.0000	0.0000
0.3815	0.0000	0.0000
0.2387	0.0000	0.0000

$$f(x)=\sum_j f(x/C_j)P(C_j)$$

Vraisemblance

0.1841
0.4595
0.4687
0.2200
0.1051
0.1283
0.1487
0.1272
0.0796

$$P(C_j/x)=\frac{f(x/C_j)P(C_j)}{f(x)}$$

Probabilités *a posteriori*

0.0029	0.8645	0.1326
0.0065	0.6471	0.3464
0.0260	0.3396	0.6344
0.1656	0.1109	0.7234
0.7575	0.0102	0.2323
0.9915	0.0001	0.0084
0.9999	0.0000	0.0001
1.0000	0.0000	0.0000
1.0000	0.0000	0.0000

EM : illustration 1D

Modèle à 3 Gaussiennes

centres = 3 0.5 1

variances = 0.8 0.2 0.2

Etapes EM à partir des 100 exemples représentés en rouge

Modèle initial

Centres = 1.3198 -0.9094 -2.3056

Variances = 1 1 1

Après 5 itérations

Centres = 1.7475 0.5684 0.2153

Variances = 1.2779 0.1734 0.0312

Après 100 itérations

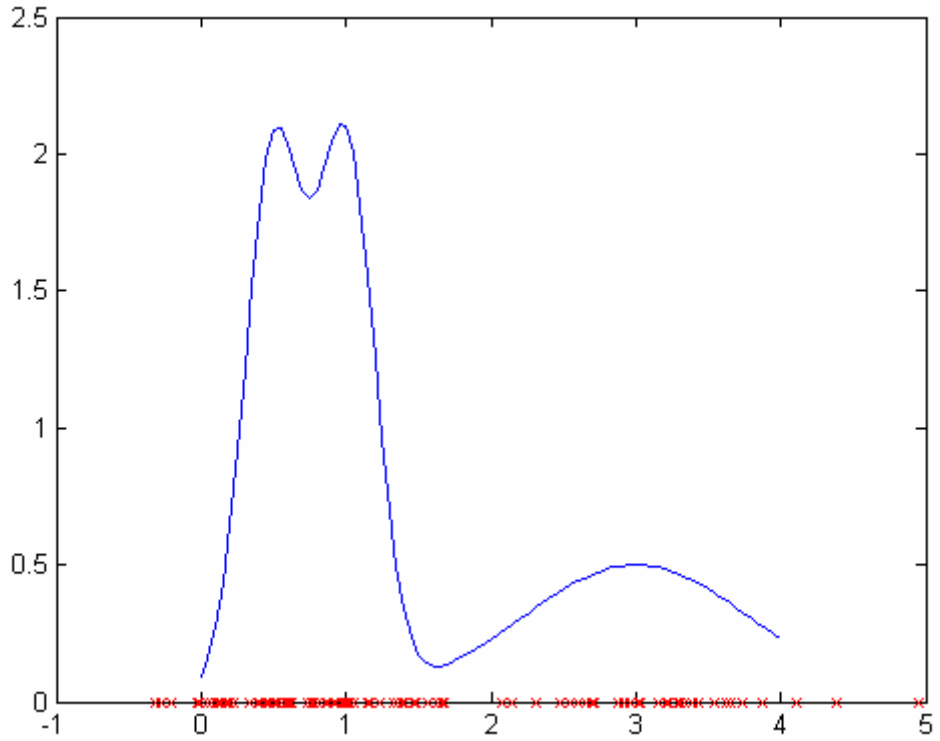
Centres = 0.6859 2.6300 0.7202

Variances = 0.5411 0.6302 0.1312

Après 200 itérations

Centres = 1.2268 3.1538 0.4279

Variances = 0.3572 0.3672 0.1997



Modèle de Mélange en dimension d

Chaque Gaussienne est de la forme

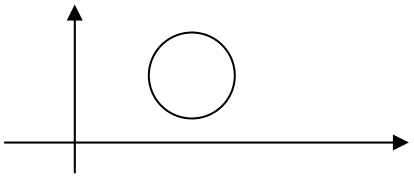
$$f(x/C_j) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma_j)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_j)^t \Sigma_j^{-1}(x-\mu_j)\right)$$

μ_j **vecteur moyen ou centre de la composante**

Σ_j **matrice de covariance de la composante**

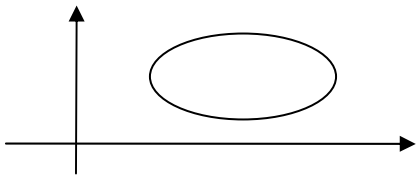
Modèle sphérique

$$\Sigma_j = \begin{pmatrix} \sigma^j & 0 \\ 0 & \sigma^j \end{pmatrix}$$



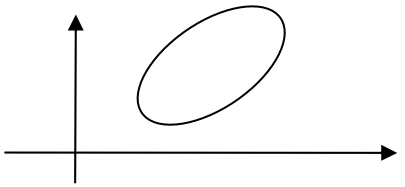
Diagonal

$$\Sigma_j = \begin{pmatrix} \sigma_{11}^j & 0 \\ 0 & \sigma_{22}^j \end{pmatrix}$$



Complet

$$\Sigma_j = \begin{pmatrix} \sigma_{11}^j & \sigma_{12}^j \\ \sigma_{21}^j & \sigma_{22}^j \end{pmatrix}$$



EM en dimension d

Le mélange est toujours de la forme

$$f(x)=\sum_j f(x/C_j)P(C_j)$$

Formules de ré-estimation par EM à partir du modèle Θ^g

$$P(j) = \frac{1}{N}\sum_{i=1}^{N} P(j / x_i, \Theta^g)$$

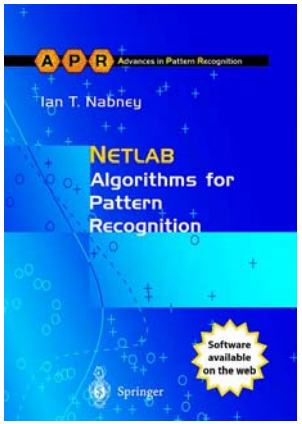
$$\mu_j = \frac{\sum_{i=1}^{N} x_i P(j / x_i, \Theta^g)}{\sum_{i=1}^{N} P(j / x_i, \Theta^g)}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} P(j / x_i, \Theta^g)(x_i - \mu_j)(x_i - \mu_j)^t}{\sum_{i=1}^{N} P(j / x_i, \Theta^g)}$$

Boîte à outils

I. T. Nabney. Netlab: Algorithms for Pattern Recognition, Springer Verlag, London, 2002. 410pp. Reprinted 2003, 2004.

Télécharger la boîte à outils
http://www.ncrg.aston.ac.uk/netlab/



HMM à observations continues

Plutôt que de projeter les vecteurs de mesures continues sur les centroïdes puis d'effectuer l'apprentissage du HMM discret

On cherche à faire les deux ensembles

Il n'y a plus d'alphabet d'observations discrètes

$O = o_1 o_2 o_3 \ldots o_T$ **est une séquence de vecteurs de mesures**

On modélise la vraisemblance de l'observation o_t dans l'état s_j par un modèle de mélange de Gaussiennes

$$b_j(o_t) = p(o_t \mid s_j) = \sum_{k=1}^{M} c_{jk} N(o_t, \mu_{jk}, \Sigma_{jk})$$

Remplace la Matrice B du modèle discret

Ré-estimation

A chaque itération de la procédure d'apprentissage on ré-estime les paramètres de chaque gaussienne et les coefficients des mélanges

$$c_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{l=1}^{M} \gamma_t(j,l)}$$

$$\mu_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).o_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\Sigma_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).(o_t - \mu_{jk})(o_t - \mu_{jk})^t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

Avec

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}N(o_t,\mu_{jk},\Sigma_{jk})}{\sum_{m=1}^{M}c_{jm}N(o_t,\mu_{jm},\Sigma_{jm})}\right]$$

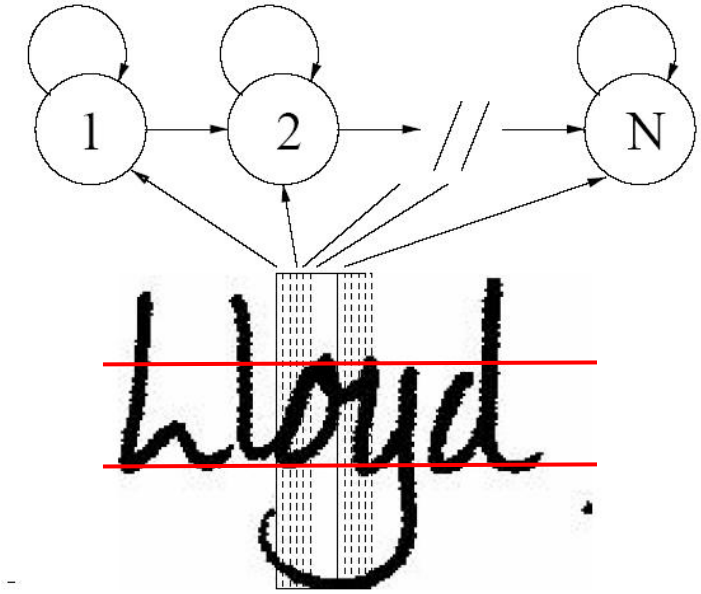
La matrice de transition est estimée comme dans le cas discret

$$a_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} = \frac{\sum_{t=1}^{T-1}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1}\alpha_t(i)\beta_t(i)}$$

Reconnaissance Hors-Ligne sans segmentation

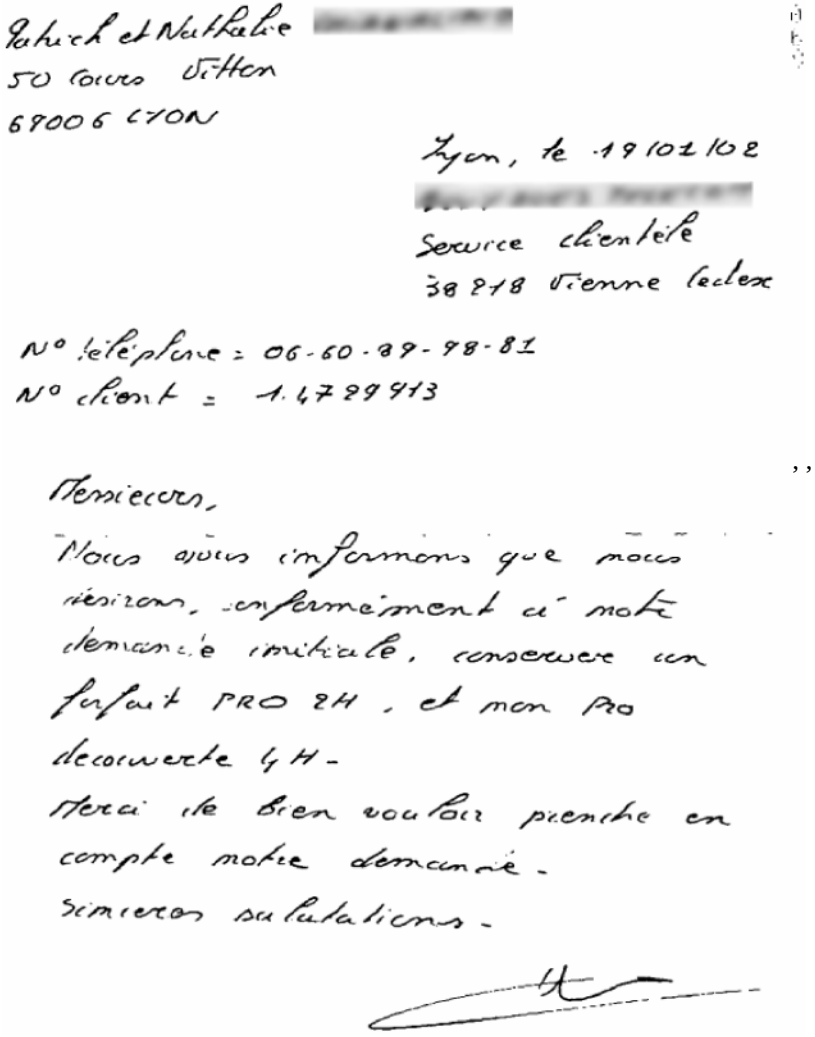
Vinciarelli & Luettin 2000

HMM gauche Droite + Fenêtre coulissante



Caractéristiques: % de pixels noirs dans chaque zone de la fenêtre glissante

HMM pour l'extraction d'information (1)



Patrick et Nathalie [redacted]
50 Cours Vitton
69006 LYON

Lyon, le 19/02/02
[redacted]
Service clientèle
38278 Vienne Cedex

N° téléphone : 06-60-89-98-81
N° client = 1.4729413

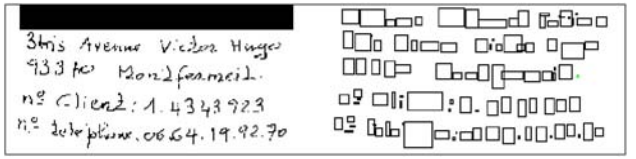
Mesdames,

Nous vous informons que nous résilions, conformément à notre demande initiale, conservée en fichier PRO 24, et non Pro découverte 4 H.

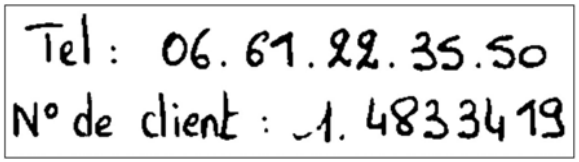
Merci de bien vouloir prendre en compte notre demande.

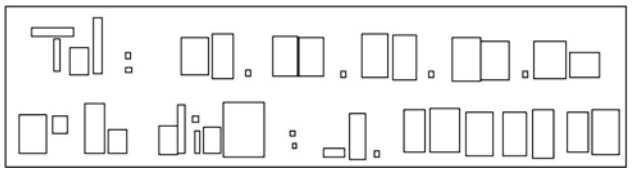
Sincères salutations.

Localiser le champs numérique sans le reconnaître en étant guidé par sa syntaxe



3bis Avenue Victor Hugo
933 [illegible] Montfermeil
n° Client : 1.4343923
n° téléphone : 06.64.19.92.70





Caractéristiques

$$f_1=\frac{H_{C-1}}{H_C} \qquad f_4=\frac{W_{C+1}}{W_C} \qquad f_7=\frac{H_C}{W_C}$$

$$f_2=\frac{H_{C+1}}{H_C} \qquad f_5=\frac{|G_C-G_{C-1}|}{W_C}$$

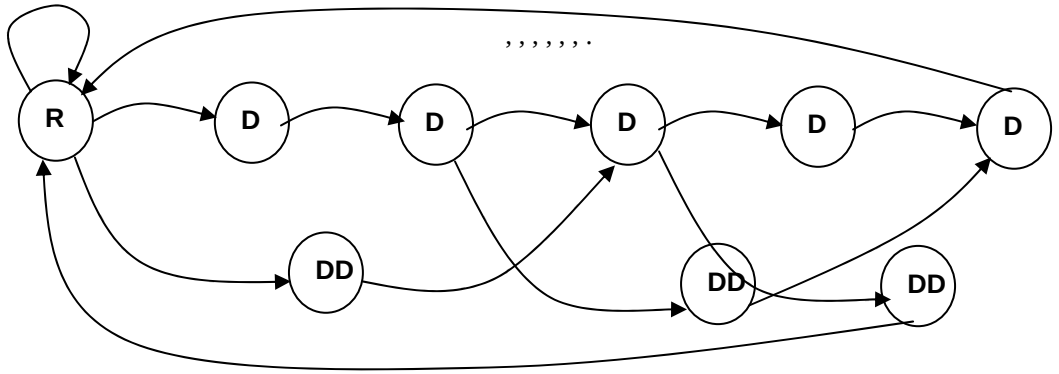
$$f_3=\frac{W_{C-1}}{W_C} \qquad f_6=\frac{|G_C-G_{C+1}|}{W_C}$$

HMM pour l'extraction d'information (2)

Une ligne d'écriture comprend

des informations non numériques à rejeter: R
des nombre (digits) : D
des signes séparateurs : S
des chiffre collés (doubles digits): DD

Syntaxe d'une ligne comportant un code postal : apprentissage automatique très délicat



Extraction des champs numériques par l'algorithme de Viterbi

Champs Markoviens

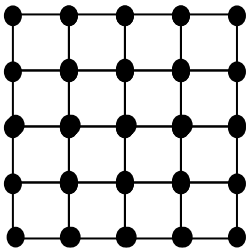
- Les champs Markoviens reprennent les idées déjà présentées dans le cas 1D pour modéliser des dépendances locales entre états
- Les états ne sont pas observables
- On accède indirectement aux états à travers des observations qui sont effectuées sur le signal ou sur l'image
- On cherche le plus souvent à trouver la meilleure configuration d'état expliquant l'image

- Les observations ne sont plus séquentielles
- Elles sont mesurées sur un maillage

Champ Markovien : Définitions

- **Sites et maillage**

  Un champ de Markov caché est un processus doublement stochastique formé de deux champs aléatoires X et Y dont les variables aléatoires sont indexées par un ensemble S de sites ou positions s dans l'image. S est défini par un maillage le plus souvent carré ou rectangulaire.



$$S = \{s(i, j) / i = 1,..., L, j = 1,..., H\}$$

Les sites peuvent correspondre aux pixels, ou à un sous échantillonnage de l'image selon la taille de la maille élémentaire

Champ Markovien : Définitions

- **Voisinage d'un site**
  L'ensemble des sites est muni d'un système de voisinage v défini de la façon suivante :

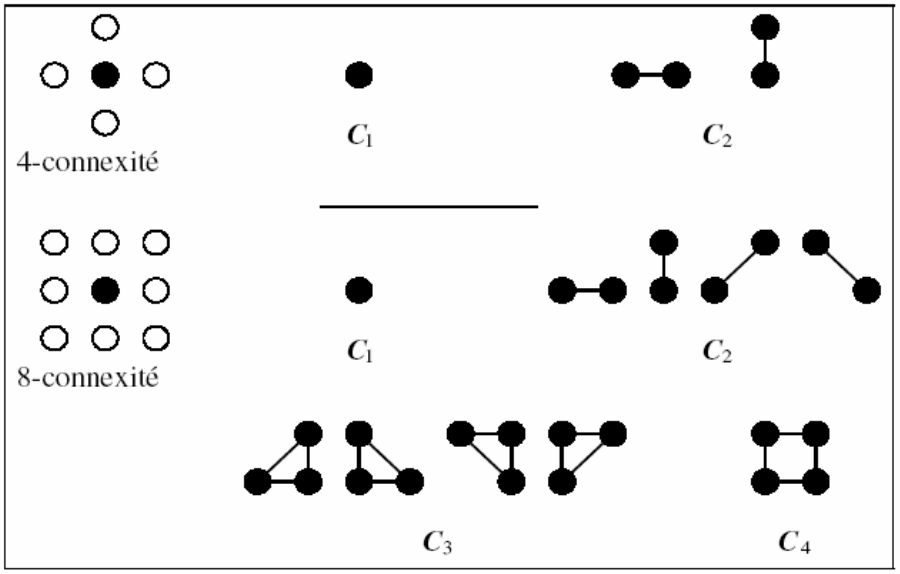
$$\forall s \in S \quad v_s = \left\{ t / t \in v_s \Rightarrow s \in v_t , s \notin v_s \right\}$$

Le relation de voisinage est symétrique

Champ Markovien : Définitions

- **Voisinage et cliques**
  A partir d'un système de voisinage, un système de cliques est défini.

  Une clique est constituée soit d'un site singleton, soit de sites tous voisins entre eux au sens du voisinage



Champ Markovien : Définitions

- **Champ des observations**
  Le champ Y est le champ des observations et se note $Y = \left(Y_s\right)_{s \in S}$

  Une réalisation de ce champ est notée $Y = y$

  En chaque site, la valeur du champ notée y_s est la réalisation de la variable aléatoire Y_s et on note $Y_s = y_s$

Les observations correspondent à des mesures réelles effectuées sur l'image.
Ces mesures peuvent être scalaires

ex. niveau de gris du pixel ou niveau de gris moyen sur le site

ou vectorielles

ex. valeur RVB des pixels ou 3 valeurs RVB moyennes sur le site

Champ Markovien : Définitions

- **Champ des états cachés**
  Le champ X est le champs caché des étiquettes et se note $X = (x_s)_{s \in S}$

  Une réalisation de ce champs est notée $X = x$

  En chaque site, la valeur du champs notée x_s est la réalisation de

  la variable aléatoire X_s et on note $X_s = x_s$

Les variables aléatoires de ce champs prennent leur valeur dans un ensemble Q d'étiquettes ou états fini $L = \{l_1, l_2, ..., l_q\}$

L'ensemble des réalisations possibles du champs X se note $L^{|S|}$

Exemple :

pour une image 512×512 **et 5 états différents il y a** 5^{36} **configurations d'états différentes**

Champ Markovien : Définition

- **Modèle de dépendance statistique**

  Le Modèle statistique d'un champ Markovien permet d'accéder au champs caché par la connaissance des mesures effectuées en chaque site de la façon suivante

$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)} \propto P(Y/X)P(X)$$

$P(Y/X)$ est le modèle d'attache aux données

$P(X)$ est le modèle a priori du champ considéré

C'est un modèle génératif 2D

Champ Markoviens : Définition

- **Modèle d'attache aux données**

  On fait l'hypothèse que les observations en chaque site sont indépendantes entre elles et qu'elles ne dépendent que du site considéré

$$p(Y = y / X = x) = \prod_{s \in S} p(Y_s = y_s / X_s = x_s)$$

Comme dans le cas des modèles 1D les probabilités d'émission sont modélisées par des mélanges Gaussiens

$$p(y_s / x_x) = \sum_{j=1}^{M} P(C_j) \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(\frac{-(x - \mu_j)^2}{2\sigma_j^2} \right)$$

Champ Markovien : Définition

- **Modèle a priori du champ caché**

  Il s'agit de calculer la probabilité d'une configuration du champ d'états c'est-à-dire calculer $P(X = x)$

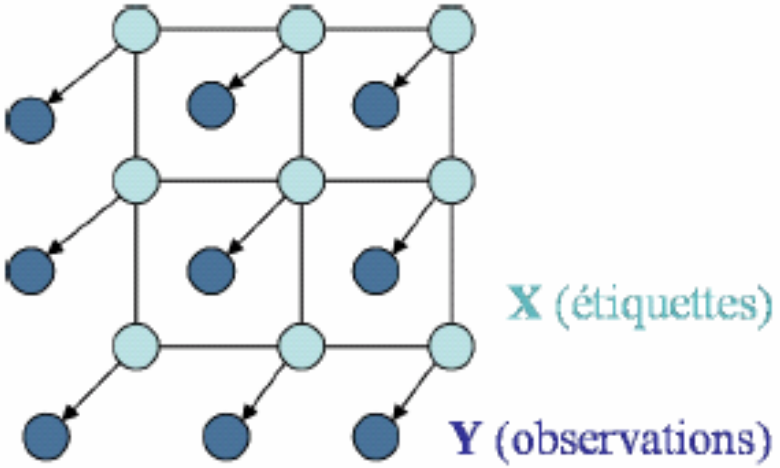
  Sous l'hypothèse Markovienne, la probabilité locale en chaque site ne dépend que des sites voisins

$$P(X_s = x_s / X - X_s) = P(X_s = x_s / X_t = x_t, t \in v_s)$$

C'est l'hypothèse classique Markovienne de dépendance locale

Champ Markovien

- **Modèle graphique d'un champ Markovien**



Champ de Gibbs

Définition: Mesure de Gibbs

La mesure de Gibbs de fonction d'énergie U est la probabilité P définie sur l'espace des configurations du champ par

$$P(X = x) = \frac{1}{Z}\exp\left(-U(x)\right)$$

Avec $U(x) = \sum_{c \in C} U_c(x)$

On somme les énergies de toutes les cliques du champ des états

et $Z = \sum_{x \in \Omega} U(x)$ la somme de toutes les énergies de toutes les configurations possibles. C'est la fonction de partition qui sert de normalisation pour donner une probabilité (incalculable en pratique)

Exemple : Calcul de l'énergie dans le cas d'une 4-connexité

$$U(x) = \sum_{c=s \in C_1} U_c(x_s) + \sum_{c=(s,t) \in C_2} U_c(x_s, x_t)$$

Champ de Gibbs

Définition: Champ de Gibbs

Un champ de Gibbs est un champ aléatoire dont la probabilité est une mesure de Gibbs. c.a.d. quelle provient d'un potentiel

Théorème de Hammersley-Clifford (Besag 74)

- X est un champ de Markov relativement au système de voisinage v et de probabilité $P(X = x) > 0$ si et seulement si X est un champs de Gibbs de potentiel associé au même voisinage v

Grâce à ce théorème on va pouvoir passer de la probabilité globale d'une configuration du champ aux probabilités locales des sites conditionnellement à leurs voisins grâce aux énergies des cliques

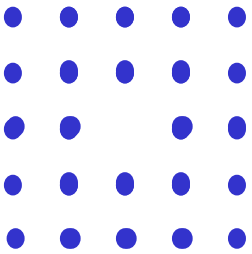
Probabilité locale d'un état

- On veut déterminer la probabilité locale de l'état du champ au site s connaissant la configuration globale du champs $X = x$

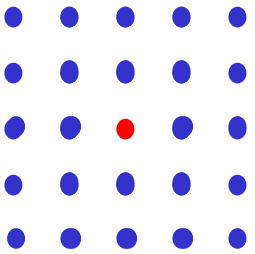
  c.a.d. calculer $P(X_s = x_s / X^s = x^s)$

$X^s = X - X_s$ Désigne le champs total des variables aléatoires sans celle en s

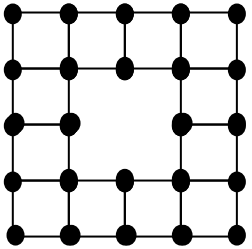
$x^s = x - x_s$



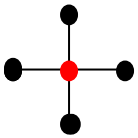
$x = x^s + x_s$



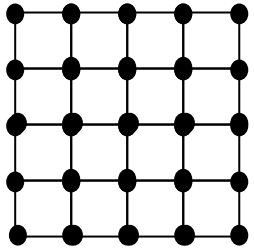
l'ensemble des cliques ne contenant pas s



l'ensemble des cliques contenant s



l'ensemble des cliques de X



Probabilité locale d'un état

- On veut déterminer la probabilité locale de l'état du champ au site s connaissant la configuration globale du champ $X^s = x^s$
  c.a.d. calculer $P(X_s = x_s / X^s = x^s)$
- Par la loi de Bayes on a $$P(X_s = x_s / X^s = x^s) = \frac{P(X = x)}{P(X^s = x^s)}$$

- où $$P(X = x) = \frac{1}{Z}\exp\left(-U(x)\right)$$
- et $$U(x) = \sum_{c \in C / s \in c} U_c(x) + \sum_{c \in C / s \notin c} U_c(x) = U_s(x_s / v_s) + \sum_{c \in C / s \notin c} U_c(x)$$

les cliques contenant s — les autres

$U_s(x_s / v_s)$ est la contribution du site s à l'énergie globale

Probabilité locale d'un état

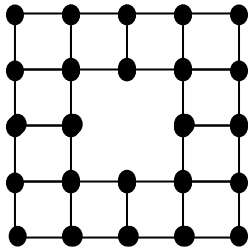
- Le numérateur s'écrit donc

$$P(X = x) = \frac{1}{Z}\exp\left(-U_s(x_s / v_s) - \sum_{c \in C, s \notin c} U_c(x)\right)$$

- Le dénominateur s'écrit

$$P(X^s = x^s) = \sum_{\xi} P(X^s = x^s, X_s = \xi)$$

$$= \sum_{\xi} \frac{1}{Z}\exp(-U(x^s, X_s = \xi)$$

x^s  **et** $X_s = \xi_1, \ldots, \xi_N$

toutes les valeurs possibles en s

Probabilité locale d'un état

- Alors la probabilité locale en s conditionnellement au champ total s'écrit

$$P(X_s = x_s / X^s = x^s) = \frac{\exp(-U(x^s, x_s))}{\sum_{\xi} \exp(-U(x^s, \xi))}$$

- en faisant apparaître l'énergie en dehors du site s on a

$$P(X_s = x_s / X^s = x^s) = \frac{\exp(-\sum_{c \in C, s \notin c} U_c(x) - U_s(x_s / v_s))}{\sum_{\xi} \exp(-\sum_{c \in C, s \notin c} U_c(x) - U_s(X_s = \xi / v_s))}$$

- Finalement les énergies sur les cliques ne contenant pas s se compensent et il reste

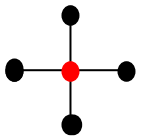
$$P(X_s = x_s / X^s = x^s) = \frac{\exp(-U_s(x_s / v_s))}{\sum_{\xi} \exp(-U_s(X_s = \xi / v_s))}$$

Probabilité locale d'un état

- On retrouve bien l'hypothèse Markovienne

$$P(X_s = x_s / X^s = x^s) = \frac{\exp(-U_s(x_s / v_s))}{\sum_{\xi} \exp(-U_s(X_s = \xi / v_s))}$$

- Bien qu'il ne soit pas possible de calculer la probabilité d'une configuration totale du champ à cause de la constante de normalisation, on parvient à calculer une probabilité locale en chaque site
- Le calcul ne fait intervenir que les cliques contenant s



- Les algorithmes de segmentation, de simulation ne feront intervenir que des calculs sur les voisinages

Echantillonnage de champs de Markov

- **Connaissant les paramètres du champ, comment générer une image se conformant à ses paramètres?**
  - Dans le cas 1D on tire aléatoirement l'état suivant en respectant les probabilités de transition entre états (matrice A)
  - La difficulté dans le cas 2D est liée à la non existence d'un parcours causal dans l'image
  - On met en œuvre des techniques par tirage aléatoire des sites et remise à jour de l'état en fonction des valeurs des voisins (on utilise plutôt les énergies des cliques que les probabilités de transition)

L'Echantillonneur de Gibbs (Geman & Geman 1984)

- **Connaissant les paramètres du champ, comment générer une image se conformant à ses paramètres?**

  - Etape n=0 :
    - Initialiser une image noté A_n
  - Etape n = n+1
    - Choisir un site s au hasard
    - Calculer la probabilité de chaque état possible de s

      $$P(X_s = \xi / X^s = x^s)_{\xi=1,\ldots,N}$$

    - Affecter à s la valeur $x_s = \xi$ obtenue par tirage aléatoire respectant $P(X_s = \xi / X^s = x^s)$
  - Continuer tant que des sites changent d'état

- L'algorithme converge vers une configuration de sites obéissant aux paramètres du champ (Winkler 1995)

Recherche de la configuration optimale

- **On cherche à déterminer la meilleure configuration d'états**
  - Cette fois encore on est gêné par la non existence d'un parcours causal dans l'image
  - On doit donc mettre en œuvre des techniques de simulation (dite encore techniques de relaxation)

Algorithme de recuit simulé

- **Définition:** Distribution de Gibbs avec température

  - **C'est une probabilité définie par le biais d'une fonction d'énergie et d'un paramètre de température**

$$P_T(X = x) = \frac{1}{Z(T)} \exp\left(\frac{-U(x)}{T}\right)$$

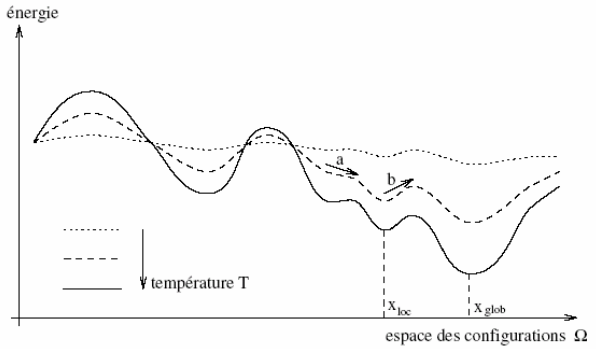
  - **Avec** $Z(T) = \sum_x \exp\left(\frac{-U(x)}{T}\right)$

  - **Si** $T \to \infty$ **alors** $P_T(.)$ **converge vers une probabilité uniforme**
  - **Si** $T \to 0$ **alors** $P_T(.)$ **converge vers une loi uniforme sur toute les configuration d'énergie minimale**

Algorithme de recuit simulé

- **Initialisation n=0 :**
  - Choisir une température initiale élevée T_0
  - Choisir une configuration initiale du champs $x(0)$
- Etape n = n+1 :
  - Obtenir l'image $x(n)$ par simulation (de Gibbs) selon la loi d'énergie $\frac{U(x)}{T_n}$
  - Diminuer lentement la température $T_n = \frac{c}{\log(1+n)}$
  - Continuer tant que l'image est modifiée

Converge lentement vers l'optimum globale car des configurations sous-optimales sont acceptées qui permettent de sortir des minimum locaux



a : $\Delta U < 0$, transition acceptée
b : $\Delta U > 0$, transition selon une probabilité $p = \exp(-\Delta U/T)$

Algorithme ICM :
Iterated Conditional Modes (Besag 1986)

- **Reprend l'idée du recuit mais déterministe**
- **Initialisation n=0 :**
  - Choisir une configuration initiale du champs $x(0)$
- Etape n = n+1 :
  - Pour chaque site s de l'image faire
    - Calculer toutes les probabilités conditionnelles $P(X_s = \xi / X^s = x^s)_{\xi=1,\ldots,N}$
    - Mettre à jour le site avec l'état ξ le plus probable (déterminisme)
  - Continuer un nouveau parcours tant que des modifications sont effectuées

S'apparente à un recuit simulé à température nulle: converge vers un minimum local

Beaucoup plus rapide que le recuit (dizaine d'itérations)

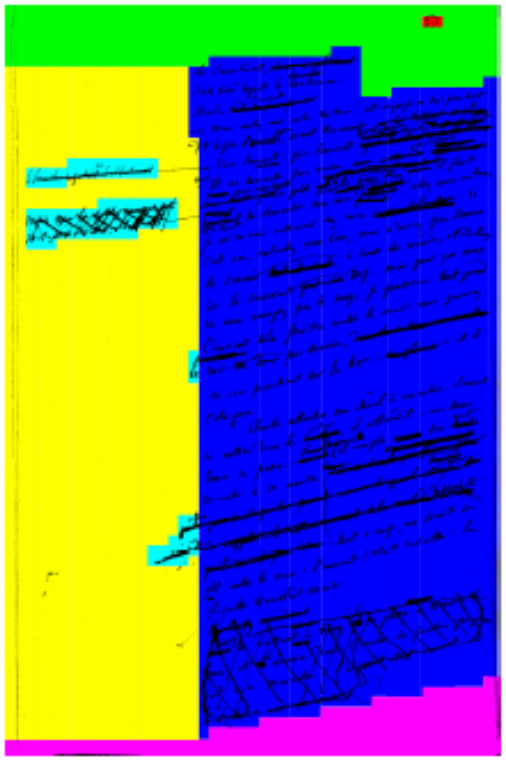
Exemple : restauration d'images

Exemple : segmentation d'images

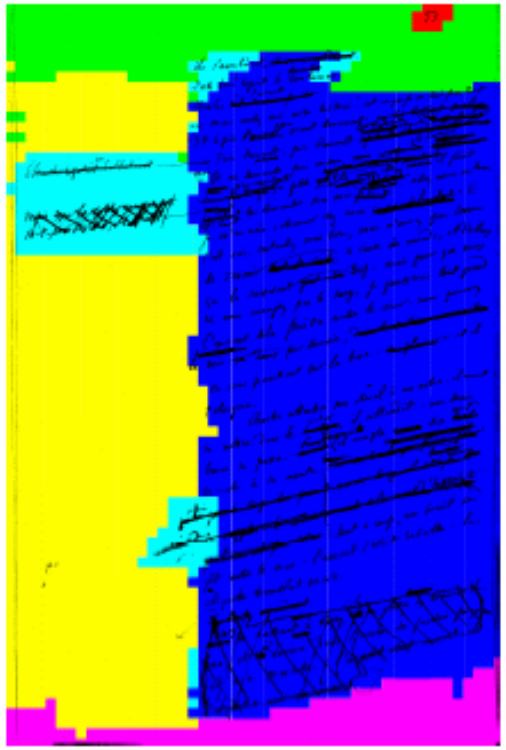
a. Image radar originale ©ERS-1

b. Image segmentée en régions

Exemple : segmentation d'images de documents

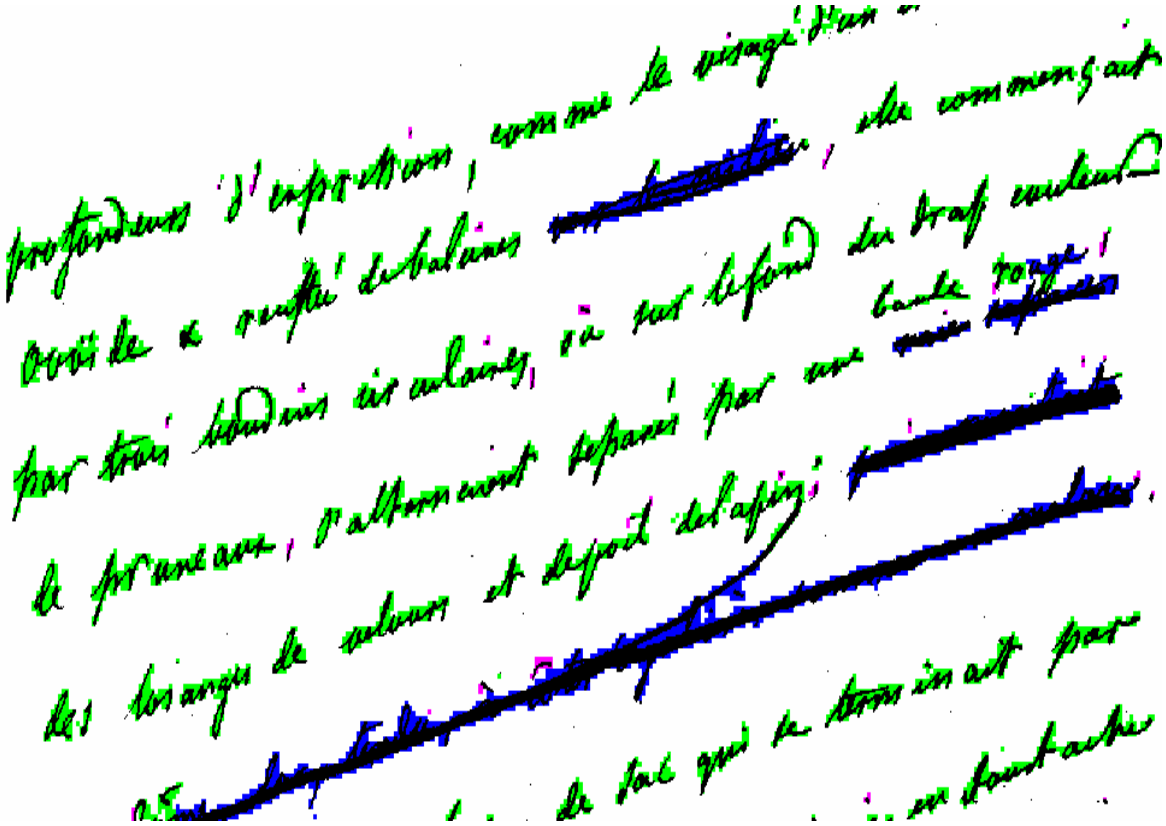


(a)

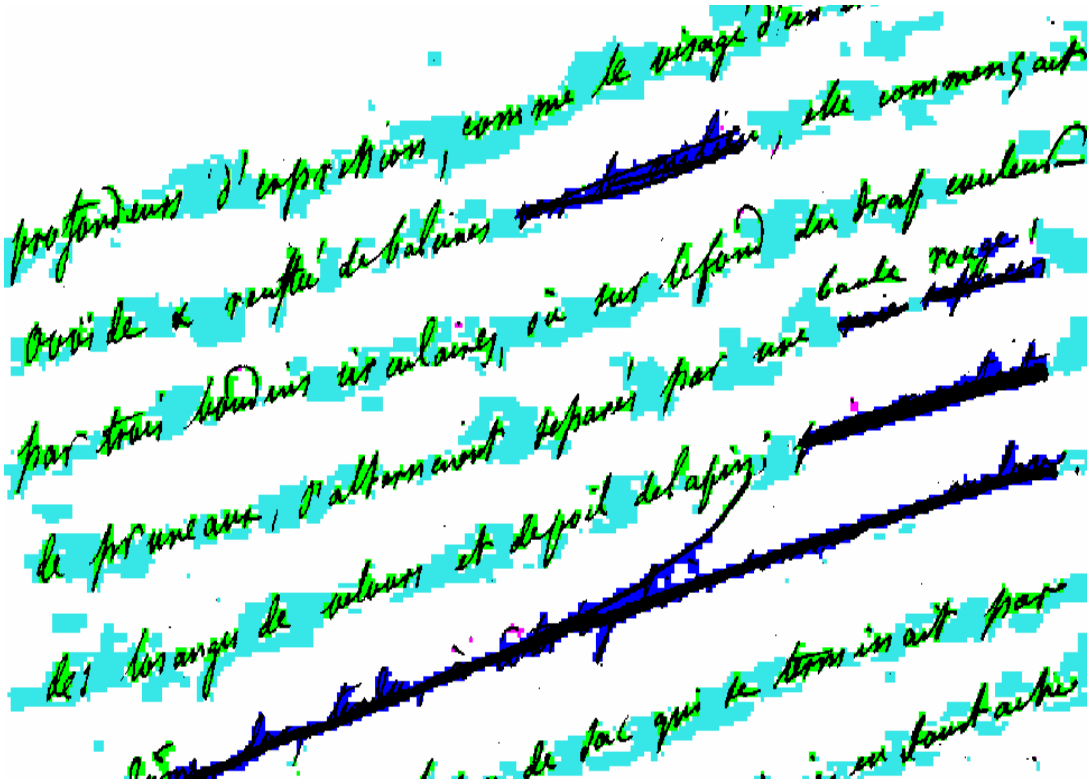


(b)

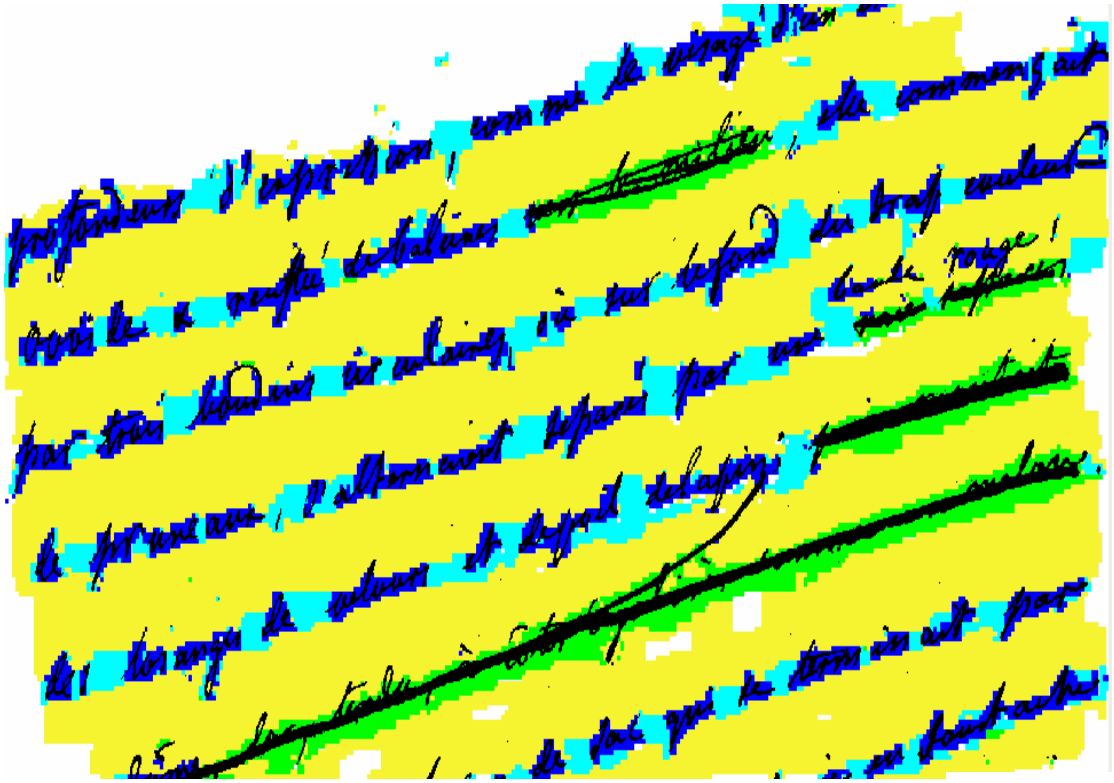
Exemple : Segmentation d'images de documents

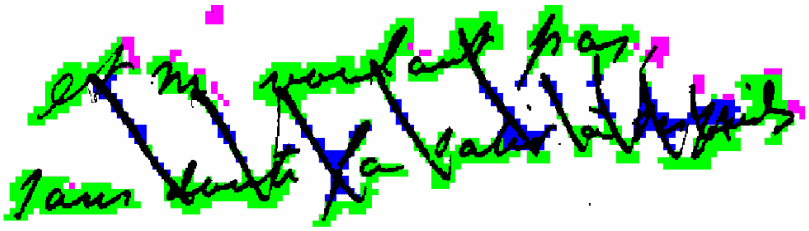


Exemple : Segmentation d'images de documents

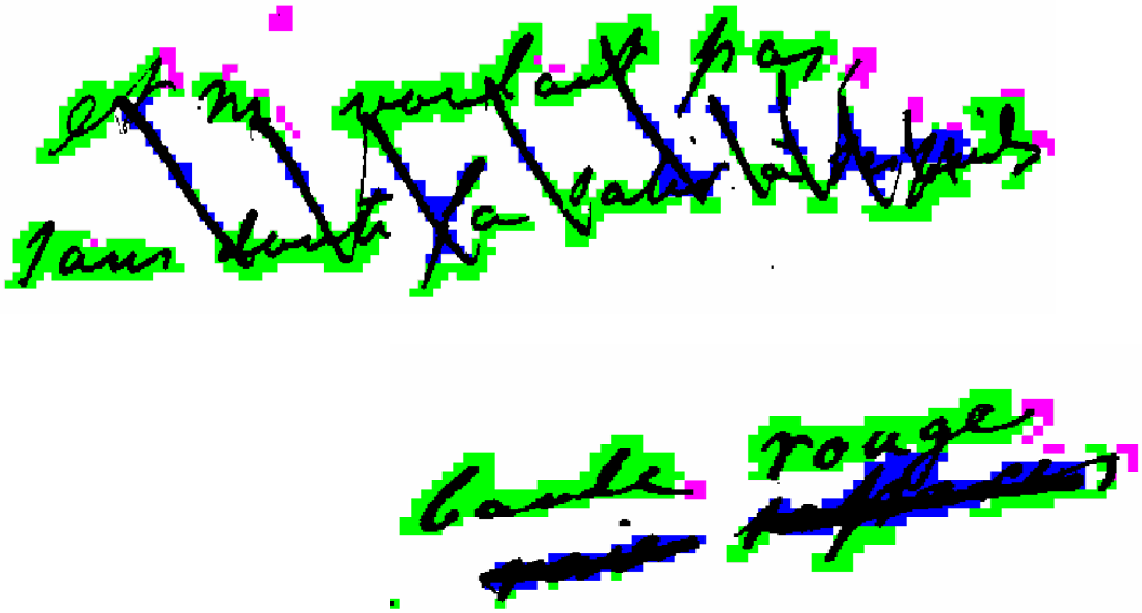


Exemple : Segmentation d'images de documents



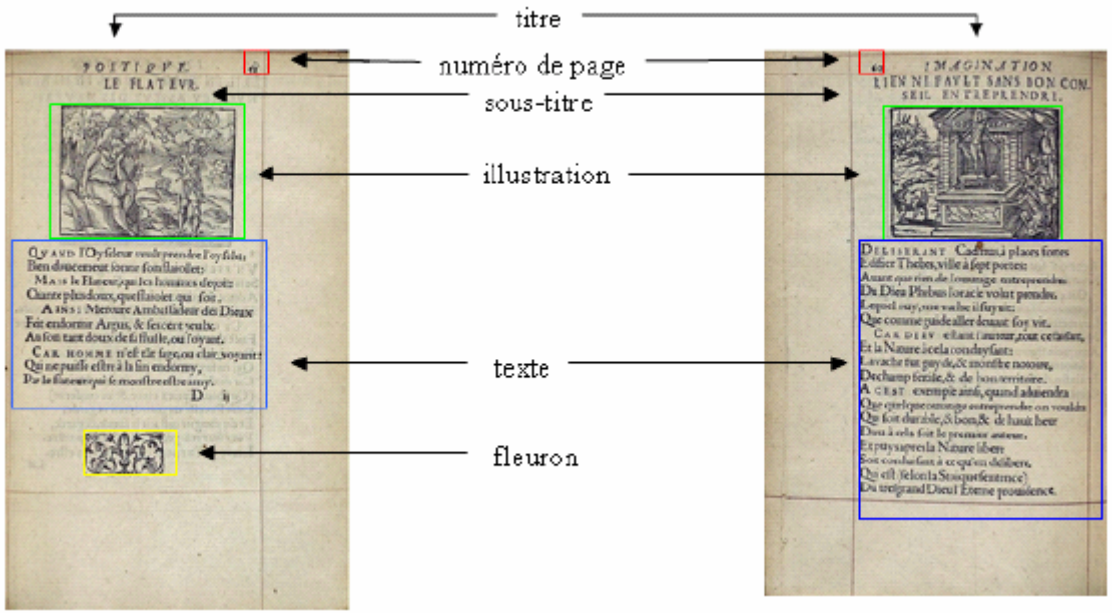


Exemple : Segmentation d'images de documents

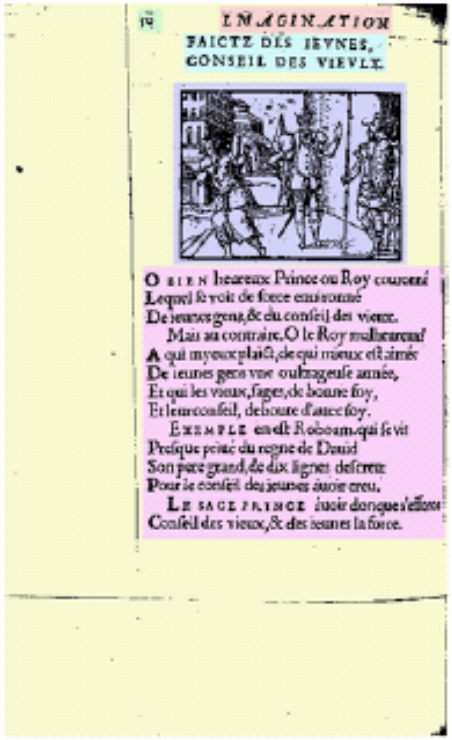


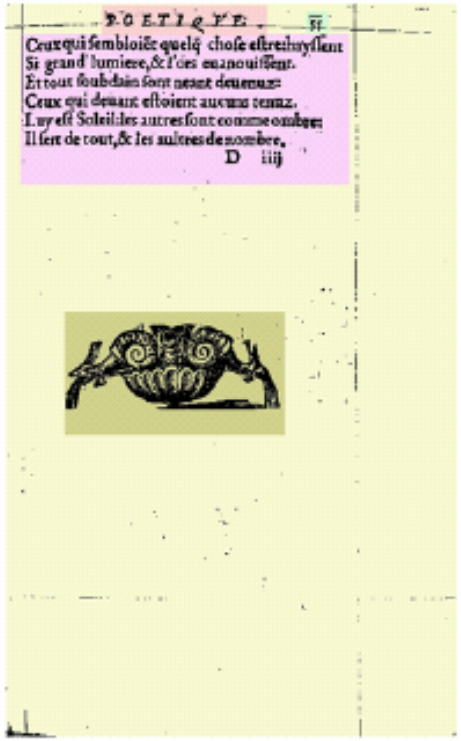
Exemple : Segmentation d'images de documents

Exemple : Segmentation d'images de documents



Exemple : Segmentation d'images de documents

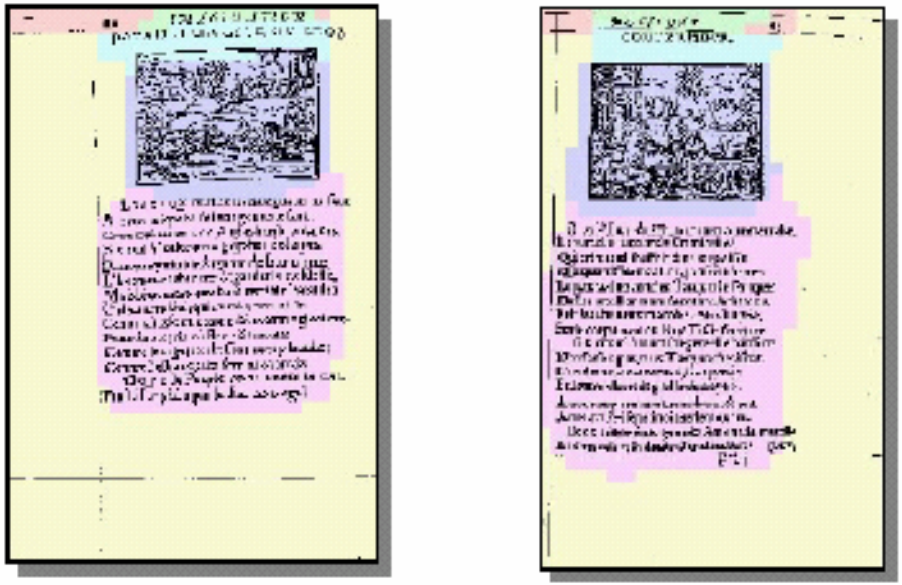




Base Poétique	
Etiquettes	Couleur associée
Numéro de page	
Image	
Titre	
Corps de Texte	
Fond	
Sous-Titre	
Fleuret	

Exemples Étiquetés

Exemple : Segmentation d'images de documents



Exemples Étiquetés

Bibliographie

Ouvrages (Bibliothèque UFR Sciences Rouen)

L. Miclet, Méthodes structurelles pour la reconnaissance des formes, Eyrolles 1984.
A. Cornuéjols, L. Miclet, Apprentissage artificiel : Concepts et algorithmes, Eyrolles, 2002
A.Belaïd, Y. Belaïd, Reconnaissance des formes, méthodes et applications, InterEditions, 1992
La parole et son traitement automatique, Calliope, éd. Masson , 1989.
R. Boite et al., Traitement de la parole, Presses Polytechniques et Universitaires Romandes, 2000.
R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, 2nd Edition, 2001.
L.R. Rabiner & B.H. Juang, Fundamentals of Speech Recognition, Englewood Cliffs, NJ:Prenctice Hall 1993.
C. D. Manning & H. Schütze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999
H. Maître, **Le traitement des images (Traité IC2, série Traitement du signal et de l'image) Hermes,** 2003.

Articles de synthèse

L. R. Rabiner, A tutorial on Hidden Markov Model and selected applications in speech recognition, IEEE proceedings, Vol. 77, pp. 257-286, 1989.
J. A. Bilmes, A gentle Tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov Models, Department of Electrical Engineering and Computer Science, Berkeley, 1998.
J. McMahon et al., A review of statistical language processing techniques, the Queen's University of Belfast, 1995.
Eugene Charniak, Statistical Techniques for Natural Language Parsing, 1997.
D. Afentenos, On Grammars: The Chomsky Hierarchy and Probabilistic Grammars, 2001.
Marcus et al., The Penn Treebank: Annotating predicate argument structure, Univ. Of Pennsylvania,1994.
Notes on probabilistic context free Grammars, Language and statistics, March, 1999.

Étude Bibliographique

H. Bourlard, S. Dupont,C. Ris, Multi-Stream Speech Recognition, IDIAP, Rapport Interne, 1996.
A. Vinciarelli, S. Bengio, Writer Adaptation Techniques in HMM based Off-Line Cursive Script Recognition, Pattern Recognition Letters, 2002.
D. Freitag, A. McCallum, Information Extraction with HMM and Shrinkage, 1999.
F. Pla, A. Molina, N. Prieto, Tagging and Chunking with bi-grams, Universita Polytechnica Valencia, 2000.
F. Pla, A. Molina, et al., Language Understanding using two level stochastic Models with POS and semantic Units, 2001.
T. Scheffer, S. Wrobel, Active Learning of Partially Hidden Markov Models, University of Magdeburg, 2001.