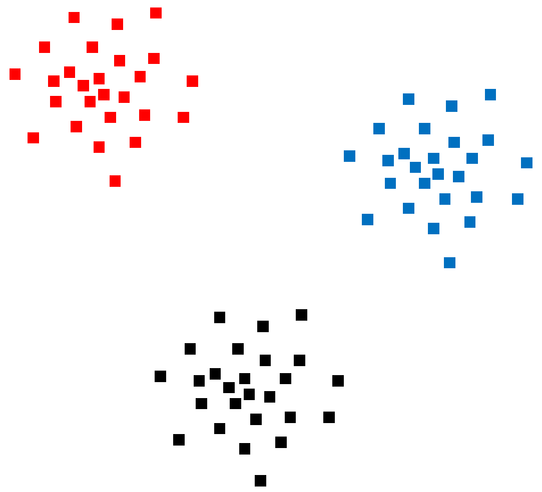
CHAPITRE IV

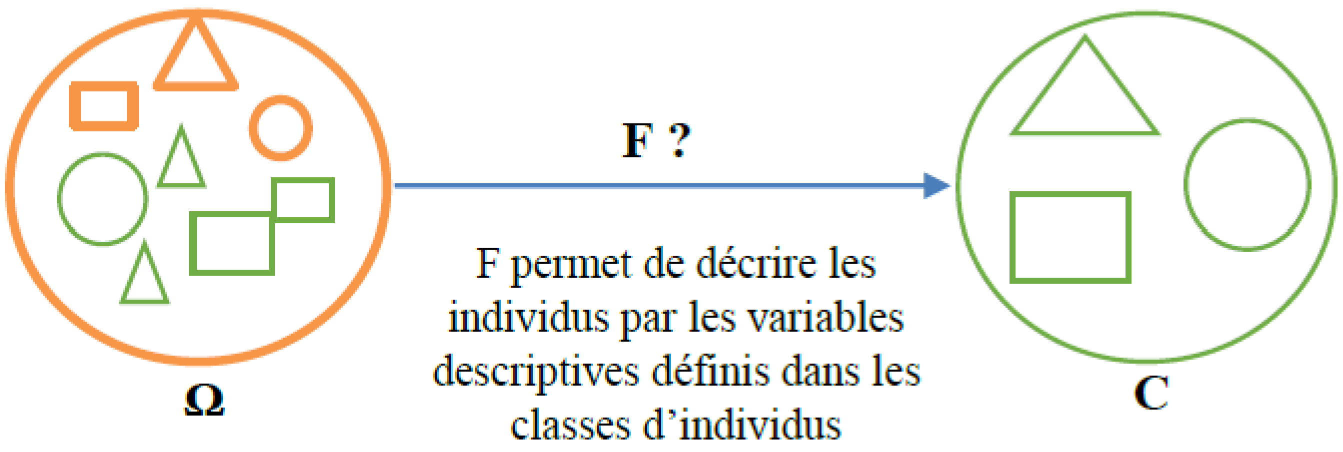
Classification automatique



Classification automatique

Définition

La **classification automatique** est la catégorisation algorithmique d'objets. Elle consiste à attribuer une classe ou catégorie à chaque objet (ou individu) à classer, en se basant sur des données statistiques. Elle fait couramment appel à l'apprentissage **automatique** et est largement utilisée en reconnaissance de formes. wiki



Deux grands types d'approches

- Non-paramétriques :

Plus deux individus sont proches, plus ils ont de chances de faire partie de la même classe

  - Méthodes hiérarchiques
  - Centres mobiles

- Probabilistes :

Utilisent une hypothèse sur la distribution des individus à classifier. à quelle classe les individus ont le plus de chances d'appartenir

  - Méthodes statistiques bayésiennes
  - Méthodes stochastiques

Méthodes bayésiennes

- **Principe**

− Un problème de classification peut être assimilé dans certains cas à un problème de diagnostic qui consiste à construire une décision à partir de certains paramètres.

− Dans le domaine médical par exemple, établir un diagnostic signifie être capable d'associer le nom d'une maladie à un certain nombre de symptômes présentés par des malades. On repère dans ce problème trois objets essentiels : les malades, les maladies et les symptômes.

- Population = Malades
- Classes = Maladies
- Descriptions = Symptômes (variables descriptives)

→ Associer une maladie à une liste de symptômes

Méthodes bayésiennes

- **Formalisation**

Ω : la population

D : l'ensemble des descriptions

C : l'ensemble des classes

X : Ω → D est la fonction qui associe une description à tout élément de la population

Y : Ω → C est une fonction qui associe une classe à tout élément de la population

F : D → C est la fonction de classification.

$$\begin{tikzcd} \Omega \arrow[rr, "X"] \arrow[dr, "Y"'] & & D \arrow[dl, "F"] \\ & C & \end{tikzcd}$$

→ Classer revient à trouver F **?**

Méthodes bayésiennes

- **Formalisation : Probabilités**

Pour simplifier les choses on supposera que l'ensemble Ω est probabilisé et que l'ensemble D est discret.

Soit P la probabilité définit sur la population Ω, on peut définir les probabilités suivantes :

P(d) : probabilité qu'un élément de Ω possède **d** pour description.

P(k) : probabilité qu'un élément de Ω soit de la classe **k**.

P(d/k) : probabilité qu'un élément de la classe **k** possède **d** pour description.

P(k/d) : probabilité qu'un élément ayant **d** pour description soit de la classe **k**.

- Formule de Bayés :

$$p(k/d) = \frac{p(d/k) \times p(k)}{p(d)}$$

Cette formule suppose qu'on peut évaluer les probabilités P(d/k), P(k) et P(d).

Méthodes bayésiennes

Exemple : soit Ω la population d'un pays, on dispose d'un échantillon représentatif de la population de ce pays.

On décrit les individus par un attribut logique « Smartphone » qui vaut 'vrai' si l'individu possède un Smartphone et 'faux' sinon.

L'espace de descriptions est donc D={Smartphone, Pas de Smartphone}.

On souhaite classer les individus en deux classes, 'Aisé' pour les individus ayant un revenu supérieur à la moyenne et 'Non aisé' pour les autres.

Méthodes bayésiennes

On dispose des informations suivantes :

- 40% de la population dispose d'un revenu supérieur à la moyenne.
- 80% des personnes aisés ont un Smartphone, alors que 45% de la population restante dispose d'un Smartphone, ce qui peut être résumé par le tableau suivant :

Classe K	Aisé	Non aisé
P(k)	0.4	0.6
P(Smartphone/k)	0.8	0.45
P(¬ Smartphone/k)	0.2	0.55

Méthodes bayésiennes

- **Choix de la fonction de classement F :**

Classe K	Aisé	Non aisé
P(k)	0.4	0.6
P(Smartphone/k)	0.8	0.45
P(┐ Smartphone/k)	0.2	0.55

- **Première règle (Classe majoritaire):** Attribuer à chaque description la classe majoritaire (i.e) pour laquelle **P(k)** est maximale. La fonction **F** qui est appelée dans ce cas F_{maj}, va attribuer à tout individu, qu'il possède un Smartphone ou pas la classe majoritaire (**non aisé**) qui a la probabilité **0.6**.

  - **Inconvénient :** Le défaut principal de cette règle est qu'elle ne fait jouer aucun rôle à la description.

Méthodes bayésiennes

- **Choix de la fonction de classement F :**

Classe K	Aisé	Non aisé
P(k)	0.4	0.6
P(Smartphone/k)	0.8	0.45
P(┐ Smartphone/k)	0.2	0.55

- **Deuxième règle (Maximum de vraisemblance) :** Si on observe ‘d’, on choisit la classe pour laquelle cette observation est la plus probable, i.e, celle pour laquelle P(d/k) est maximale. Cette règle est appelée règle de maximum de vraisemblance.

  La fonction de classement F ($\mathbf{F_{vraisemblance}}$) va attribuer la classe (**aisé : 0.8**) à tout individu possédant un Smartphone et la classe (**non aisé**) à tous les autres.

- On voit que cette fonction de classement est plus fine que la précédente et qu’elle correspond d’avantage à ce que l’on attend intuitivement.

Méthodes bayésiennes

Classe K	Aisé	Non aisé
P(k)	0.4	0.6
P(Smartphone/k)	0.8	0.45
P(┐ Smartphone/k)	0.2	0.55

- **Choix de la fonction de classement F :**

- **Deuxième règle (Maximum de vraisemblance) :**

- **Inconvénient :** Le défaut de cette fonction de classement apparait dans l'exemple suivant :

  Supposons que l'ensemble des classes C soit composé de trois classes (employé de poste, médecin, ouvrier) et supposons que la probabilité pour qu'un employé de poste ait un Smartphone est égale à 1.

  La règle de maximum de vraisemblance va associer alors la classe 'employé de poste' à tout individu possédant un Smartphone et ceci son tenir compte des proportions des différentes classes à l'intérieur de la population.

Classification automatique

Méthodes bayésiennes

Classe K	Aisé	Non aisé
P(k)	0.4	0.6
P(Smartphone/k)	0.8	0.45
P(┐ Smartphone/k)	0.2	0.55

- **Choix de la fonction de classement F :**

- **Troisième règle (Fonction de Bayés) :** Cette règle consiste à attribuer à une description 'd' la classe k qui maximise la probabilité P(k/d) en utilisant la formule de Bayés et en remarquant que P(d) est constante. P(d) = P(d/k) . P(k) + P(d/k') . P(k')
- Il suffit donc de choisir la classe k qui maximise le produit [P(d/k).P(k)].
  - P(Smartphone/aisé) × P(aisé) = 0.8 × 0.4 = 0.32
  - P(Pas de Smartphone/aisé) × P(aisé) = 0.2 × 0.4 = 0.08
  - P(Smartphone/non aisé) × P(non aisé) = 0.45 × 0.6 = 0.27
  - P(Pas de Smartphone/non aisé) × P(non aisé) = 0.55 × 0.6 = 0.33

La fonction $F_{Bayés}$ va associer à toute personne possédant un Smartphone la classe '**aisé**' et à toute personne ne possédant pas de Smartphone la classe '**non aisé**'. ($F_{bayés} = F_{vraisemblance}$ mais ce n'est pas toujours le cas)

Méthodes bayésiennes

Exemple 2 : On considère deux attributs pour déterminer la nationalité d'un individu. L'attribut taille qui peut prendre les valeurs grand ou petit , l'attribut couleur des cheveux qui peut prendre les valeurs brun ou blond . Les nationalités possibles sont français et suédois.

On suppose que les populations française et suédoise se répartissent selon le tableau suivant :

Classe / Description	Petit,Brun	Petit,Blond	Grand,Brun	Grand,Blond
Suédois	10	20	30	40
Français	25	25	25	25

- **Dans une assemblée comprenant 60% de suédois et 40% de français, décrire :**

a. la règle de décision majoritaire ?

b. la règle du maximum de vraisemblance ?

c. la règle de Bayes ?

Classification automatique

Méthodes bayésiennes

Classe / Description	Petit,Brun	Petit,Blond	Grand,Brun	Grand,Blond
Suédois	10	20	30	40
Français	25	25	25	25

60% de suédois et 40% de français

- **La règle de décision majoritaire :**

On affecte chaque individu quelque soit sa taille et la couleur de ses cheveux à la classe « Suédois » qui est majoritaire (60% de la population).

Classification automatique

Méthodes bayésiennes

Classe / Description	Petit,Brun	Petit,Blond	Grand,Brun	Grand,Blond
Suédois	10	20	30	40
Français	25	25	25	25

60% de suédois et 40% de français

- **Le maximum de vraisemblance:**

On affecte un individu ayant une description d(taille, couleur) à la nationalité pour laquelle cette description est la plus probable, c'est à dire P(d/k) est maximum. Ainsi tout individu ayant :

- (petit, brun) sera affecté à Français,
- (petit, blond) sera affecté à Français
- (grand, brun) sera affecté à Suédois
- (grand, blond) sera affecté à Suédois.

Méthodes bayésiennes

Classe / Description	Petit,Brun	Petit,Blond	Grand,Brun	Grand,Blond
Suédois	10	20	30	40
Français	25	25	25	25

60% de suédois et 40% de français

- **Règle de Bayés :**
- P(petit, brun/suédois).P(suédois)=0.10x0.6=0.06
- P(petit, brun/français).P(français)=0.25x0.4=<u>0.10</u>
- P(petit, blond/suédois).P(suédois)=0.2x0.6=<u>0.12</u>
- P(petit, blond/français).P(français)=0.25x0.4=0.1
- P(grand, brun/suédois).P(suédois)=0.3x0.6=<u>0.18</u>
- P(grand, brun/français).P(français)=0.25x0.4=0.1
- P(grand, blond/suédois).P(suédois)=0.40x0.6=<u>0.24</u>
- P(grand, blond/français).P(français)=0.25x0.4=0.1

Donc, tout individu ayant :

(petit, brun) sera affecté à Français, (petit, blond) sera affecté à Suédois

(grand, brun) sera affecté à Suédois (grand, blond) sera affecté à Suédois.

Modèles de Markov Cachés

Hidden Markov Models (HMM)

HMM : Origines

- Les modèles de Markov cachés (Hidden Markov Models : HMM) ont étés introduits par BAUM dans les années 70s, ce modèle s'inspire des automates probabilistes.

- Un automate probabiliste est défini par une structure composée d'états et de transitions et par un ensemble de distribution de probabilités sur les transitions. A chaque transition est associé un symbole d'un alphabet fini. Ce symbole est généré à chaque fois que la transition est empruntée.

HMM : Définition

- Un HMM se défini également par une structure composée d'états et de transitions et par un ensemble de distributions de probabilités sur les transitions.

- La différence essentielle avec les automates probabiliste est que la génération des symboles s'effectue sur les états et non sur les transitions. De plus, on associe à chaque état non pas un symbole mais une distribution de probabilités sur les symboles de l'alphabet.

- Les HMM sont utilisés pour modéliser les séquences d'observations, ces observations peuvent être de nature **discrète** ou **continue**.

HMM : Applications

- Les HMM sont utilisés dans les domaines suivants :
  - La reconnaissance de la parole
  - La reconnaissance des textes manuscrits
  - La reconnaissance des séquences d'ADN
  - L'extraction d'informations…etc.

Processus Markovien

- On peut voir un processus markovien comme la génération d'un chemin à travers un espace de valeurs possibles au cours du temps et d'une manière aléatoire.
- La caractéristique des processus markoviens est que le générateur de chemin possède une faible mémoire du parcours emprunté. Souvent, il ne se souvient que de la dernière étape.

Processus stochastique

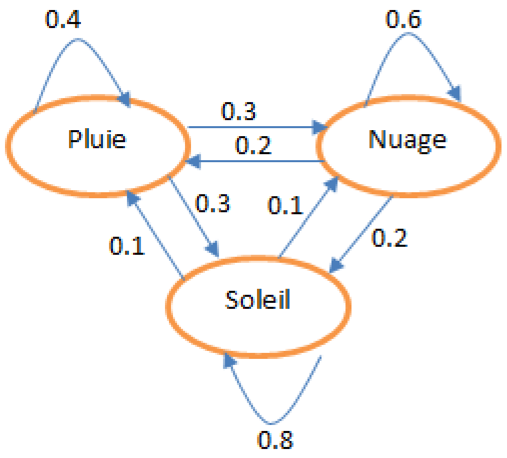
- Un processus stochastique est une suite éventuellement infini ou même continu et dont la structure est temporelle et aléatoire.

- Notant **E** l'espace dans lequel la suite prend ses valeurs, on l'appelle l'espace des états **E = {états}**. Et **T** l'espace qui paramétrise la suite (**le temps**).

- Un processus stochastique est noté : $\{q(t) \ / \ t \in T, \ q \in E\}$.

Chaine de Markov : Exemple

- La chaine de Markov suivante est organisée au tour de 3 états (observations) qui décrivent des conditions météorologiques d'une journée (pluie, nuage, soleil)



Chaine de Markov

	Pluie	Nuage	Soleil
Pluie	0.4	0.3	0.3
Nuage	0.2	0.6	0.2
Soleil	0.1	0.1	0.8

Matrice de transition

- Ce modèle de Markov nous permet de faire quelques inférences (déductions).
- Supposant que le temps d'aujourd'hui (t=1) est ensoleillé, répondez aux questions suivantes :

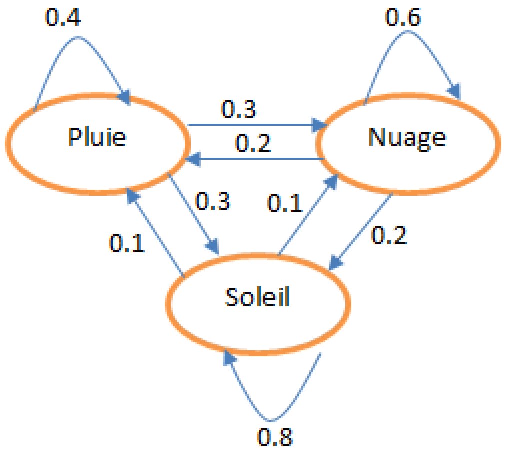
- Quel temps va-t-il faire demain (t=2) **?**

- Quelle est la probabilité que le temps sera pluvieux après-demain **?**

- Quelle est la probabilité pour que le temps de la semaine d'avenir soit :

(soleil-soleil-pluie-pluie-soleil-nuage-soleil) **?**

Chaine de Markov : Exemple



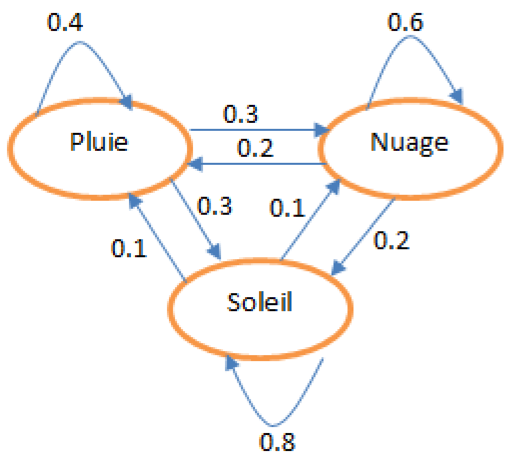
	Pluie	Nuage	Soleil
Pluie	0.4	0.3	0.3
Nuage	0.2	0.6	0.2
Soleil	0.1	0.1	0.8

- q(1) = soleil

- Quel temps va-t-il faire demain (t=2) **?**

  - **Il y a 80% de chance que la journée de demain soit ensoleillée**

Chaine de Markov : Exemple



	Pluie	Nuage	Soleil
Pluie	0.4	0.3	0.3
Nuage	0.2	0.6	0.2
Soleil	0.1	0.1	0.8

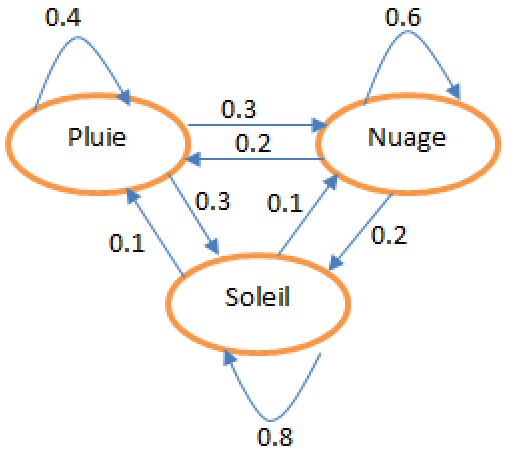
- q(1) = soleil

- Quelle est la probabilité que le temps sera pluvieux après-demain **?**

	Demain	Après-demain	
Cas 1	Soleil	Pluie	0.8 x 0.1 = 0.08
Cas 2	Nuage	Pluie	0.1 x 0.2 = 0.02
Cas 3	Pluie	Pluie	0.1 x 0.4 = 0.04

➢ **Il y a 14% de chance que la journée d'après-demain soit pluvieuse**

Chaine de Markov : Exemple



	Pluie	Nuage	Soleil
Pluie	0.4	0.3	0.3
Nuage	0.2	0.6	0.2
Soleil	0.1	0.1	0.8

- q(1) = soleil

\- Quelle est la probabilité pour que le temps de la semaine d'avenir soit : (soleil-soleil-pluie-pluie-soleil-nuage-soleil) **?**

P(semaine)= $0.8 \times 0.8 \times 0.1 \times 0.4 \times 0.3 \times 0.1 \times 0.2 = 0.0001536 = 1.536 \times 10^{-4}$

➢ **Donc il est très peu probable que le temps de la semaine prochaine sera (soleil-soleil-pluie-pluie-soleil-nuage-soleil).**

Modèle de Markov Caché

- Un HMM est un processus doublement stochastique dans le sens où il est constitué d'un processus stochastique sous-jacent qui n'est pas observable directement (il est caché), il ne peut être observé qu'à travers un autre ensemble de processus stochastiques qui produisent la séquence de symboles observés (i.e) qu'il y a deux types de probabilités :
  - Une probabilité de changement d'états
  - Une probabilité d'émission de symboles

Formalisation d'un HMM

Formellement, un HMM H est défini par un quadruplet (S, ∑, T, G)

- Un HMM H=(S, ∑, T, G)
- S : est un ensemble de N états, il contient deux états spéciaux qui sont : Start et End, qui servent respectivement à débuter et accomplir une séquence.
- ∑ : est un Alphabet composé de M symboles.
- T : c'est une matrice qui indique les probabilités de transition d'un état à un autre
  - $T = S\text{-}\{start\} \times S\text{-}\{end\} \rightarrow [0,1]$
- G : c'est une matrice qui indique la probabilité de génération associée aux états.
  - $G : S\text{-}\{start,end\} \times \sum \rightarrow [0,1]$

Formalisation d'un HMM

- On note **P(o/s)**, la probabilité de générer le symbole **o** à partir de l'état **s.**
- A chaque état **s** est associé :
  - Une distribution de probabilités de transition d'où :

$$\sum_{s\prime \epsilon S} P(s \rightarrow s') = 1$$

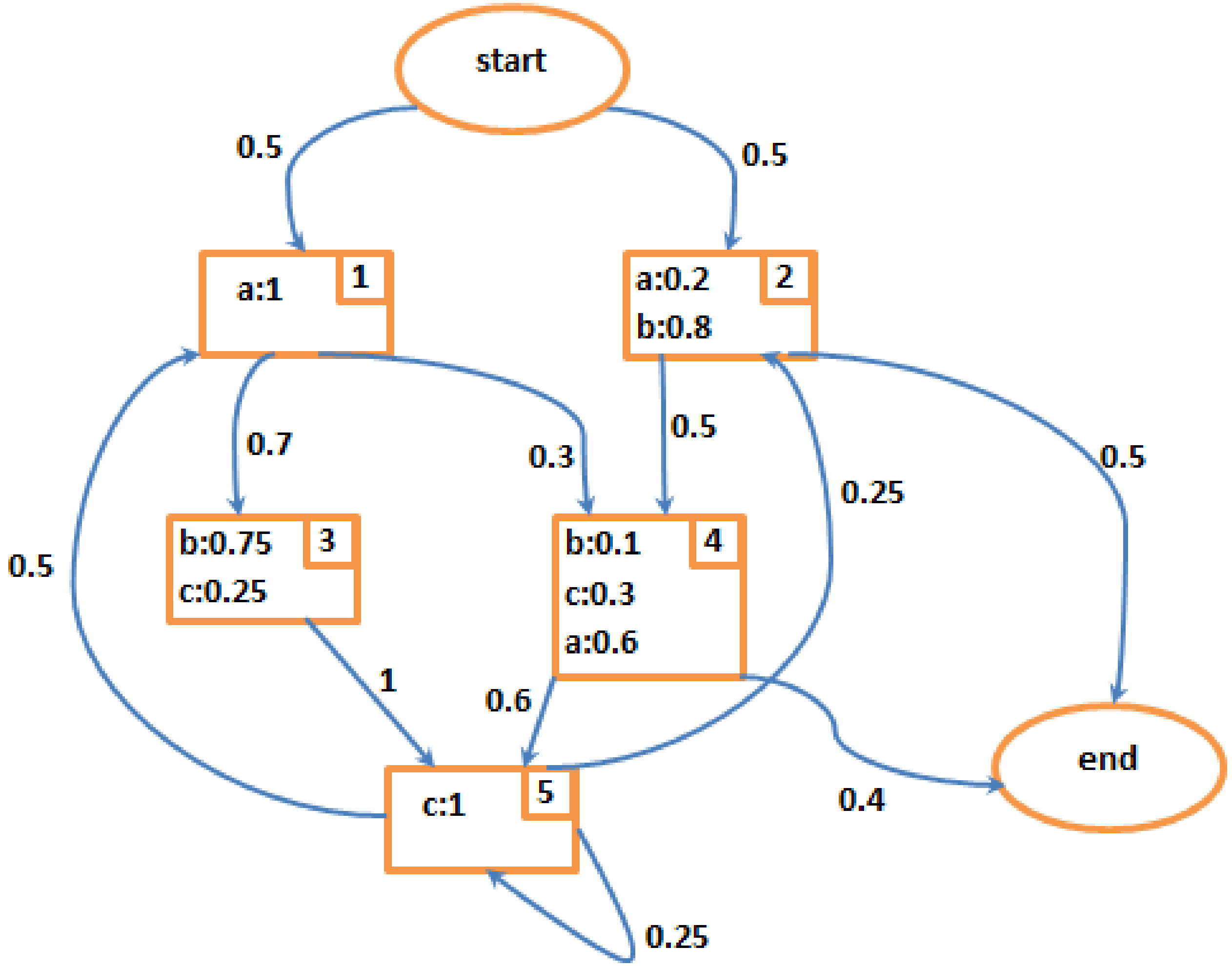
  - Une distribution de probabilité d'émission d'où :

$$\sum_{o\prime \in \Sigma} P(o' \;/\; s) = 1$$

Dans la plupart des cas on se limite à un alphabet fini (∑ fini).
Par exemple, dans les applications de reconnaissance de la parole, l'ensemble des phonèmes est fini.

Exemple d'un HMM

- La figure suivante représente un exemple d'un HMM avec 7 états et 11 transitions :



Exemple d'un HMM

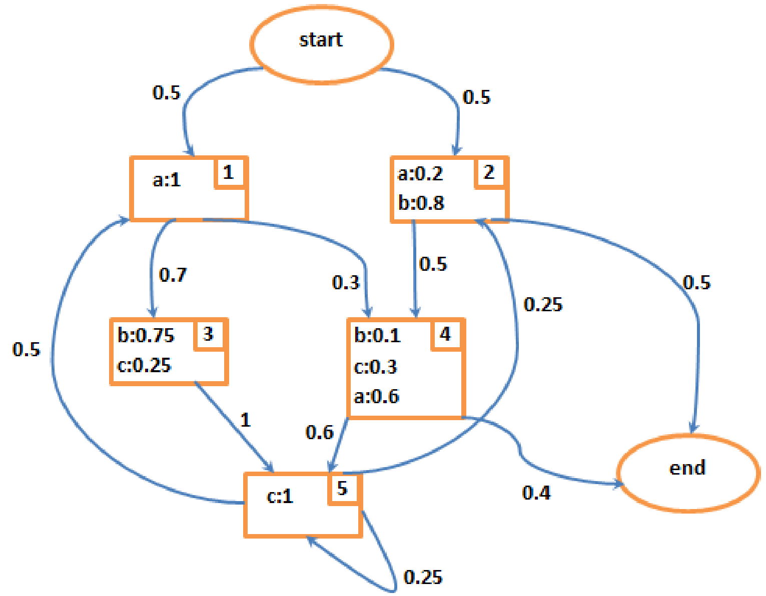
- S={start,1,2,3,4,5,end}
- ∑={a,c,b}

- T : Matrice de transition

	1	2	3	4	5	end
start	0.5	0.5				
1			0.7	0.3		
2				0.5		0.5
3					1	
4					0.6	0.4
5	0.5	0.25			0.25	

- G : Matrice de génération

	a	b	c
1	1		
2	0.2	0.8	
3		0.75	0.25
4	0.6	0.1	0.3
5			1



Exemple d'un HMM

Cet exemple d'HMM permet d'obtenir les séquences observables suivantes :

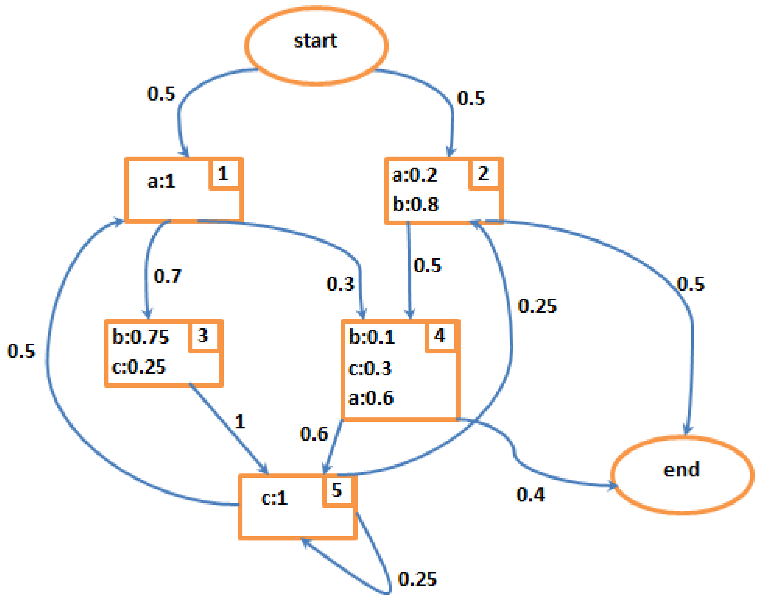
abca, aacb, ab,…etc.

A ces séquences correspondent les séquences cachées suivantes :

1-3-5-2, 1-4-5-2, 2-4

La procédure de génération d'une séquence de symboles à l'aide d'un HMM consiste à se déplacer à partir de l'état ***start***, d'état en état suivant les probabilités de transition et de générer un symbole sur chacun des états rencontrés en utilisant la distribution de probabilités de génération associée à l'état.

La procédure est répétée jusqu'à atteindre l'état ***end***.



Exemple d’un HMM

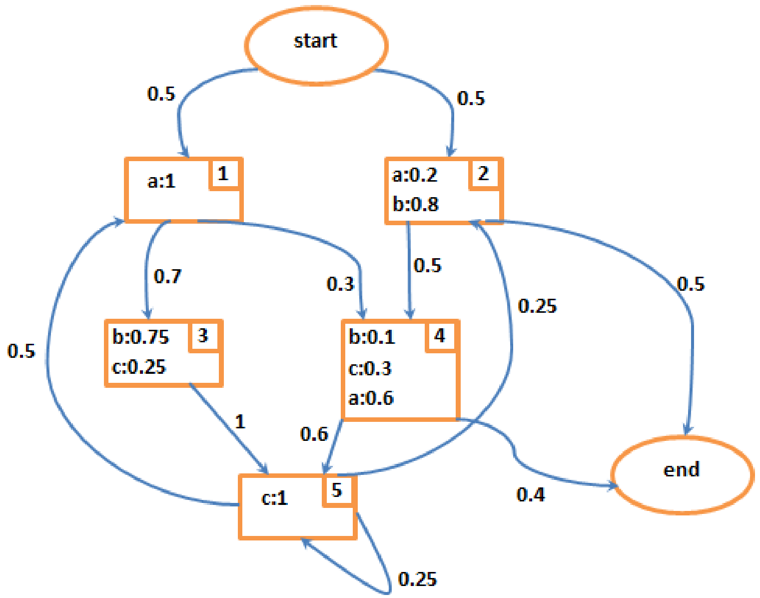
Par exemple, la séquence : abccb peut être générée en partant de l’état *start* ensuite 1, 3, 5,5 ensuite 2 et *end*.

Notant que cette séquence de symboles peut être générée suivant d’autres chemins : 1-4-5-5-2 ou bien : 2-4-5-5-2

Chemin 1 : start-1-3-5-5-2-end

Chemin 2 : start-1-4-5-5-2-end

Chemin 3 : start-2-4-5-5-2-end



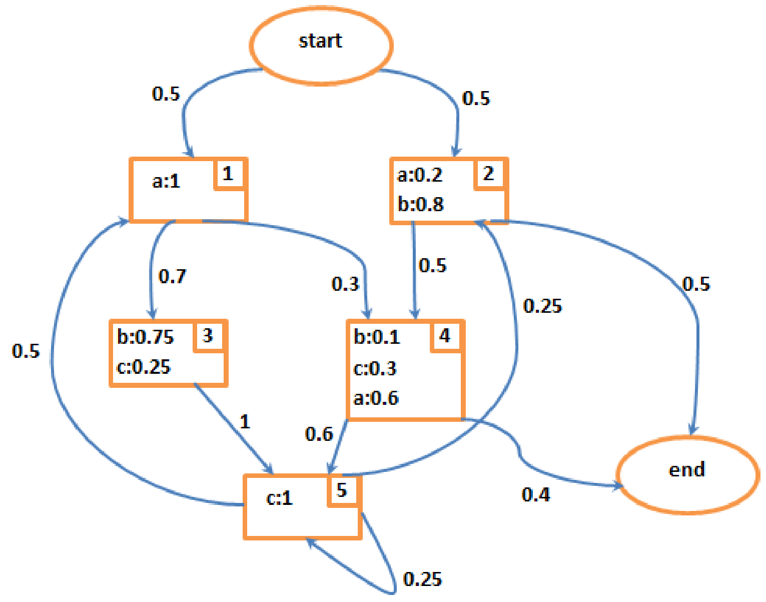
Exemple d'un HMM

La séquence : abccb

Chemin 1 : start-1-3-5-5-2-end

Chemin 2 : start-1-4-5-5-2-end

Chemin 3 : start-2-4-5-5-2-end



La probabilité de génération suivant chaque chemin est calculée comme suit :

P(chemin 1) = (0.5 x 1) x (0.7 x 0.75) x (1 x 1) x (0.25 x 1) x (0.25 x 0.8) x (0.5) = 6.5 x 10^{-3}

P(chemin 2) = (0.5 x 1) x (0.3 x 0.1) x (0.6 x 1) x (0.25 x 1) x (0.25 x 0.8) x (0.5) = 2.2 x 10^{-3}

P(chemin 3) = (0.5 x 0.2) x (0.5 x 0.1) x (0.6 x 1) x (0.25 x 1)x(0.25 x 0.8) x (0.5) = 0.75 x 10^{-3}

La probabilité de génération de la séquence abccb par le HMM est donc :

P(abccb) = (6.5 + 2.2 + 0.75) x 10^{-3} = 9.45 x 10^{-3}

Caractéristiques des HMM

- Les HMM définissent donc un processus **stochastique non déterministe**, puisqu'une même séquence de symbole peut être générée de plusieurs manières différentes, ce qui explique le nom donné à ce modèle.
- Le processus de génération est un processus **Markovien**, puisque la probabilité de transition vers un état ne dépond que de l'état actuelle et non pas des états rencontrées précédemment.
- De plus il est **caché** car on ne connait pas le processus suivi pour la génération d'une séquence de symboles donnée.

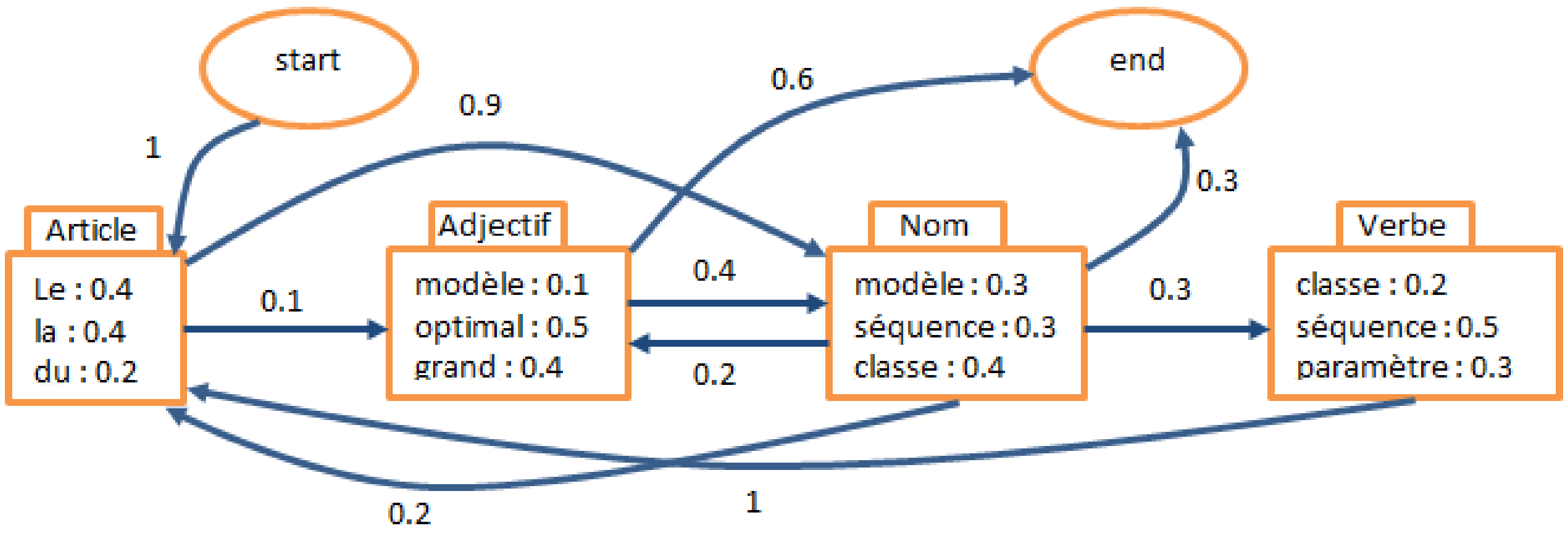
Principales applications des HMM

- On peut classer les principales applications des HMM en deux catégories :
  - La première traite les problèmes de **reconnaissance** ou de **classification** (reconnaissance d'un mot parmi un ensemble de mots, reconnaissance d'une écriture manuscrite,….)
  - Le second type des applications des HMM concerne les problèmes de **segmentation** des séquences i.e le découpage d'une séquence en sous séquences de différents types, On utilise pour ces applications un HMM dont les états sont typés.

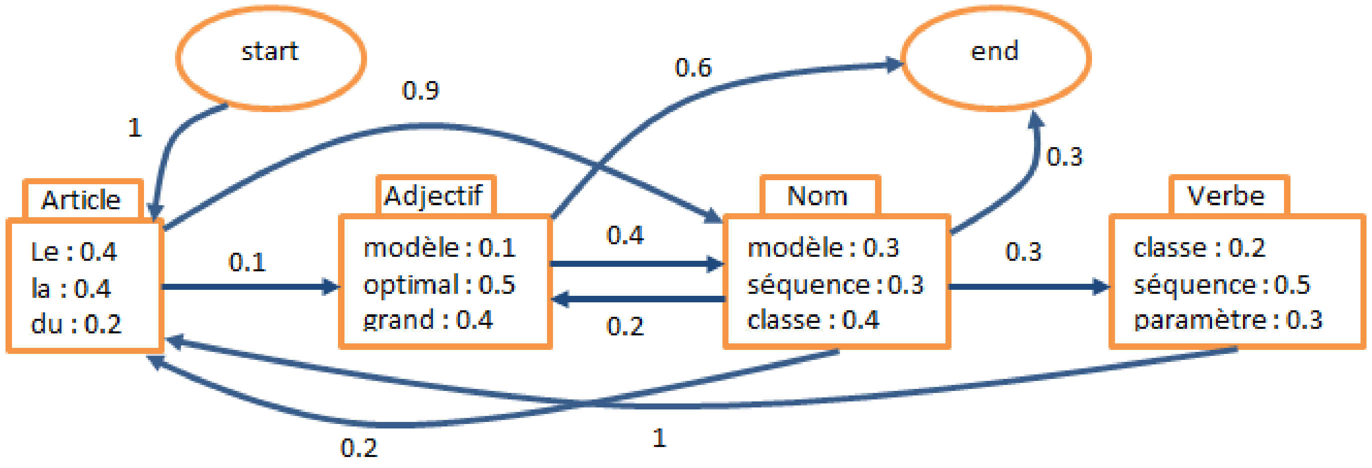
Application d'un HMM à la segmentation

La procédure de segmentation d'une séquence de symboles consiste alors à calculer à l'intérieur d'un HMM H, le chemin qui a la probabilité maximale de générer cette séquence, on associe ensuite à chaque symbole de la séquence son type en fonction du type de l'état correspondant dans le chemin calculé.

Exemple : On considère l'HMM suivant



Application d'un HMM à la segmentation



En remarquant que deux mots apparaissent dans plusieurs états :

- Le mot 'Modèle' qui peut être utilisé comme 'Adjectif' ou 'Nom'.

- Le mot 'Classe' qui peut être utilisé comme 'Nom' ou 'Verbe'.

Considérant la phrase suivante : « **Le modèle classe la séquence** ».

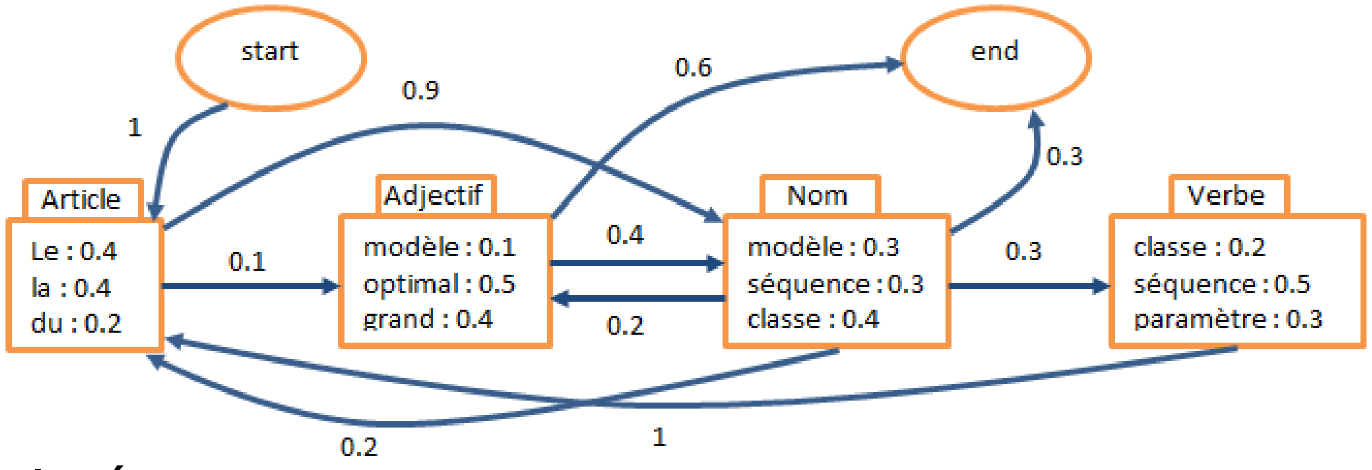
Notre objectif est de segmenter cette phrase i.e affecter à chaque mot son type.

Deux (02) chemins peuvent générer cette phrase :

Chemin 1 : start-article-adjectif-nom-article-nom-end. Avec P=0.018 x 10^{-3}

Chemin 2 : start-article-nom-verbe-article-nom-end. Avec P=0.21 x 10^{-3}

Application d'un HMM à la segmentation



Considérant la phrase suivante : « **Le modèle classe la séquence** ».

Chemin 1 : start-article-adjectif-nom-article-nom-end. Avec P=0.018 x 10^{-3}

Chemin 2 : start-article-nom-verbe-article-nom-end. Avec P=0.21 x 10^{-3}

Conclusion : la segmentation la plus probable est donc celle introduite par le second chemin. Ce qui nous permet de conclure que 'Classe' est utilisé comme 'Verbe' dans cette phrase et non pas comme 'Nom'.

- Si on étudie de plus près les paramètres de ce HMM, on se rend compte que le mot Modèle en tant que nom a plus de chance (0.9) de suivre immédiatement un article.
- Ces paramètres dépondent bien évidemment du corpus de phrases étudiés pour construire le HMM (Apprentissage).

Problèmes classiques des HMM

a. Soit un HMM H et une séquence de symbole $\mathbf{O = O_1O_2 \ldots O_t}$ donnée. Quelle est la probabilité de générer **O** avec **H**.

Solution : Algorithme de Forward-backward

b. Quelle est la séquence d'état $\mathbf{S=S_1S_2 \ldots S_t}$ de H qui a la probabilité maximale de générer **O**.

Solution : Algorithme de Viterbi

c. Comment ajuster les paramètres de H (probabilités de transition et de génération) de manière à ce qu'il représente au mieux les séquences manipulées.

Solution : Algorithme de Baum-Welch

Algorithme Forward-Backward

- Si on veut générer une séquence de symboles **O** par une approche directe, on doit calculer la probabilité de génération pour chaque chemin possible et faire la somme de ces probabilités, ce qui est très complexe dans les situations réelles.
  - N : nombre d'états
  - T : longueur de la séquence
  - Nombre de chemins possibles = N^T
  - Nombre d'opérations /chemin = T opérations
  - Nombre d'opérations pour tous les chemins possibles = $T \times N^T$

Algorithme Forward-Backward

- N : nombre d'états
- T : longueur de la séquence
- Nombre de chemins possibles = N^T
- Nombre d'opérations /chemin = T opérations
- Nombre d'opérations pour tous les chemins possibles = $T \times N^T$

Exemple : N = 5 et T = 100

Nombre de chemins possibles = $\mathbf{5^{100}}$

Nombre d'opérations /chemin = **100** opérations

Nombre d'opérations pour tous les chemins possibles = $\mathbf{100} \times \mathbf{5^{100}} \approx \mathbf{10^{72}}$

➢ L'algorithme de Forward/Backward présente une procédure bien plus efficace.

Algorithme Forward

- On définit la variable Forward $\boldsymbol{\alpha_t}$**(S)=(O$_1$,O$_2$,..,O$_t$, S$_t$=S/H)** qui exprime la probabilité d'avoir la séquence **O$_1$,O$_2$,..,O$_t$** en partant de l'état **start** et en arrivant à l'état **S$_t$** (S à l'instant t).
- Cette variable peut être calculée d'une manière inductive (incrémentale) avec l'algorithme suivant :

Algorithme Forward :

-Initialisation à t=1

$\alpha_1(S) = P(\text{start} \rightarrow S)\ .\ P(O_1/S)$

-Induction : pour t=2 jusqu'à T

$\alpha_t(S) = (\sum_{S' \in S} \alpha_{t-1}(S').P(S' \rightarrow S))\ .\ P(O_t/S)$

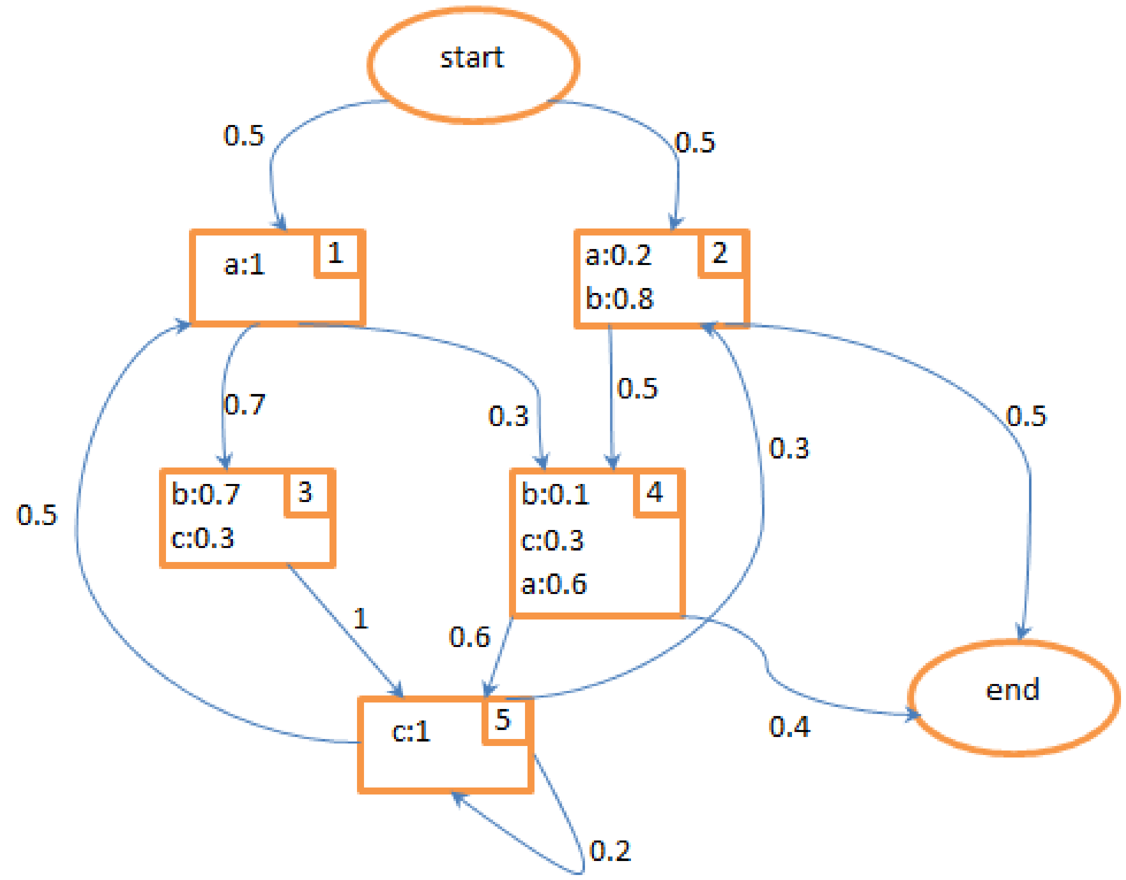
-En fin

$P(O/H) = \sum_{S' \in S} \alpha_T(S').P(S' \rightarrow end)$

Algorithme Forward

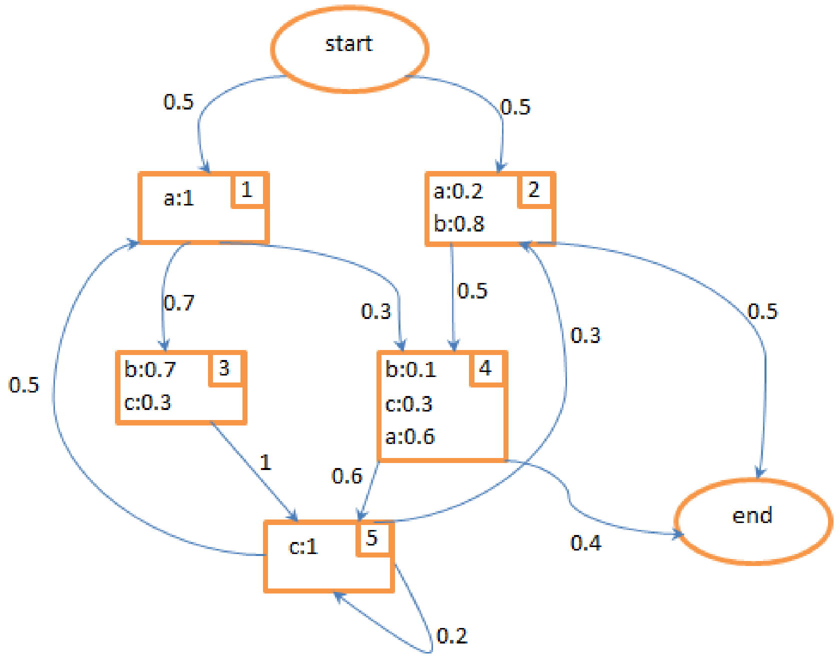
- Exemple :

Calculer **P("abcb" / H)** en utilisant l'approche directe ensuite l'algorithme forward **?**



- Approche directe :
  - Trouver tous les chemins générant la séquence abcb
  - Calculer la probabilité de génération pour chaque chemin
  - La probabilité P("abcb" / H) est la somme des probabilités des différents chemins

Algorithme Forward



- Algorithme Forward:

-Initialisation à t=1

$\alpha_1(S) = P(start \rightarrow S) \,.\, P(O_1/S)$:

$\alpha_1(1) = P(start \rightarrow 1) \,.\, P(a/1) = 0.5 \times 1 = 0.5$

$\alpha_1(2) = P(start \rightarrow 2) \,.\, P(a/2) = 0.5 \times 0.2 = 0.1$

$\alpha_1(3) = P(start \rightarrow 3) \,.\, P(a/3) = 0 \times 0 = 0$

$\alpha_1(4) = P(start \rightarrow 4) \,.\, P(a/4) = 0 \times 0.6 = 0$

$\alpha_1(5) = P(start \rightarrow 5) \,.\, P(a/5) = 0 \times 0 = 0$

-Induction : pour t=2 jusqu'à T (t=2)

$\alpha_t(S) = (\sum_{S' \in S} \alpha_{t-1}(S').P(S' \rightarrow S)) \,.\, P(O_t/S)$

$\alpha_2(1) = [\alpha_1(1)\times P(1 \rightarrow 1)+\alpha_1(2)\times P(2 \rightarrow 1)+\alpha_1(3)\times P(3 \rightarrow 1)+\alpha_1(4)\times P(4 \rightarrow 1)+\alpha_1(5)\times P(5 \rightarrow 1)] \times P(b/1) = [0.5\times 0 + 0.1\times 0 + 0\times 0 + 0\times 0 + 0\times 0.5] \times 0 = 0$

$\alpha_2(2) = [\alpha_1(1)\times P(1 \rightarrow 2)+\alpha_1(2)\times P(2 \rightarrow 2)+\alpha_1(3)\times P(3 \rightarrow 2)+\alpha_1(4)\times P(4 \rightarrow 2)+\alpha_1(5)\times P(5 \rightarrow 2)] \times P(b/2)$

$= [0.5\times 0 + 0.1\times 0 + 0\times 0 + 0\times 0 + 0\times 0.3] \times 0.8 = 0$

$\alpha_2(3) = [\alpha_1(1)\times P(1 \rightarrow 3)+\alpha_1(2)\times P(2 \rightarrow 3)+\alpha_1(3)\times P(3 \rightarrow 3)+\alpha_1(4)\times P(4 \rightarrow 3)+\alpha_1(5)\times P(5 \rightarrow 3)] \times P(b/3)$

$= [0.5\times 0.7 + 0.1\times 0 + 0\times 0 + 0\times 0 + 0\times 0] \times 0.7 = 0.245$

$\alpha_2(4) = 0.06$

$\alpha_2(5) = 0$

Algorithme Forward

-Induction : pour t=2 jusqu'à T (t=3)

$\boldsymbol{\alpha_t(S) = (\sum_{S' \in S} \alpha_{t-1}(S').P(S' \to S)) \,.\, P(O_t/S)}$

$\alpha_3(1) = [\alpha_2(1)\times P(1 \to 1)+\alpha_2(2)\times P(2 \to 1)+\alpha_2(3)\times P(3 \to 1)+\alpha_2(4)\times P(4 \to 1)+\alpha_2(5)\times P(5 \to 1)] \times P(c/1)$

$= [0\times 0 + 0\times 0 + 0.245\times 0 + 0.06\times 0 + 0\times 0.5] \times 0 = 0$

$\alpha_3(2) = [\alpha_2(1)\times P(1 \to 2)+\alpha_2(2)\times P(2 \to 2)+\alpha_2(3)\times P(3 \to 2)+\alpha_2(4)\times P(4 \to 2)+\alpha_2(5)\times P(5 \to 2)] \times P(c/2)$

$= [0\times 0 + 0\times 0 + 0.245\times 0 + 0.06\times 0 + 0\times 0.3] \times 0 = 0$

$\alpha_3(3) = 0$

$\alpha_3(4) = 0$

$\alpha_3(5) = 0.281$

(t=4)

$\alpha_4(1) = [\alpha_3(1)\times P(1 \to 1)+\alpha_3(2)\times P(2 \to 1)+\alpha_3(3)\times P(3 \to 1)+\alpha_3(4)\times P(4 \to 1)+\alpha_3(5)\times P(5 \to 1)] \times P(b/1)$

$= [0\times 0 + 0\times 0 + 0\times 0 + 0\times 0 + 0.281\times 0.5] \times 0 = 0$

$\alpha_4(2) = 0.0674$

$\alpha_4(3) = 0$

$\alpha_4(4) = 0$

$\alpha_4(5) = 0$

-En fin

$\boldsymbol{P(O/H) = \sum_{S' \in S} \alpha_T(S').P(S' \to end)}$

$P(\text{'abcb'} / H) = \alpha_4(1)\times P(1 \to end)+ \alpha_4(2)\times P(2 \to end)+ \alpha_4(3)\times P(3 \to end)+ \alpha_4(4)\times P(4 \to end)+ \alpha_4(5)\times P(5 \to end)$

$= 0\times 0+0.0674\times 0.5+0\times 0+0\times 0.4+0\times 0=0.337$

Algorithme Forward

Cet algorithme est appelé **Forward** car l'induction est réalisée en allant en avant. On calcule tout d'abord la probabilité de générer le premier symbole de la séquence, puis à chaque étape de l'induction on rajoute un symbole et on ré-itère la procédure jusqu'à calculer la probabilité de génération de la séquence entière.

N : nombre d'états, T : longueur de la séquence	
Approche directe	Algorithme Forward
Nombre d'opérations : $T \times N^T$	Nombre d'opérations : $T \times N^N$
Exemple : $N = 5$, $T = 100$	
$100 \times 5^{100} \approx 10^{72}$	$100 \times 5^5 = 312500$

Il existe un algorithme similaire qui consiste à faire le chemin inverse, c'est l'algorithme de **Backward**.

Algorithme Forward

Cet algorithme est appelé **Forward** car l'induction est réalisée en allant en avant. On calcule tout d'abord la probabilité de générer le premier symbole de la séquence, puis à chaque étape de l'induction on rajoute un symbole et on ré-itère la procédure jusqu'à calculer la probabilité de génération de la séquence entière.

N : nombre d'états, T : longueur de la séquence	
Approche directe	Algorithme Forward
Nombre d'opérations : $T \times N^T$	Nombre d'opérations : $T \times N^N$
Exemple : N = 5, T = 100	
$100 \times 5^{100} \approx 10^{72}$	$100 \times 5^{5} = 312500$

Il existe un algorithme similaire qui consiste à faire le chemin inverse, c'est l'algorithme de **Backward**.