

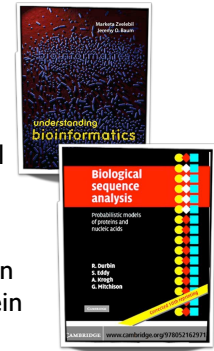
Introduction à la Bioinformatique

Des motifs et des modèles Markoviens cachés

1

Bibliographie

- Zvelebil et Baum, Understanding bioinformatics
- Durbin, Eddy, Krogh et Mitchison, Biological sequence analysis: probabilistic models of proteins and nucleic acids
- Krogh et al (1994) Hidden Markov models in computational biology: applications to protein modeling. J. Mol. Biol. 235:1501-1531
- Rabiner (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77(2): 257-286



2

Objectifs

- Être capable d'expliquer la différence entre les modèles Markoviens normaux et cachés
- Décrire les éléments qui définissent les deux modèles
- Comprendre les différences entre le problème d'évaluation, d'encodage et de l'entraînement
- Être capable d'expliquer comment les algorithmes "forward" et "backward" fonctionnent
- Être capable d'expliquer l'algorithme de Viterbi et sa relation avec l'algorithme "forward" et "backward"
- Comprendre pourquoi on a parfois besoin de "pseudocounts" pour la construction des MMC
- Comprendre pourquoi il est parfois mieux d'utiliser des logarithmes de probabilités plutôt que les probabilités
- Être capable d'expliquer la différence entre l'algorithme de Baum-Welsh et l'entraînement de Viterbi
- Comprendre comment ces algorithmes sont utilisés pour la construction des profils MMC
- Être capable d'expliquer la structure des profils MMC

3

Détecter les motifs

```

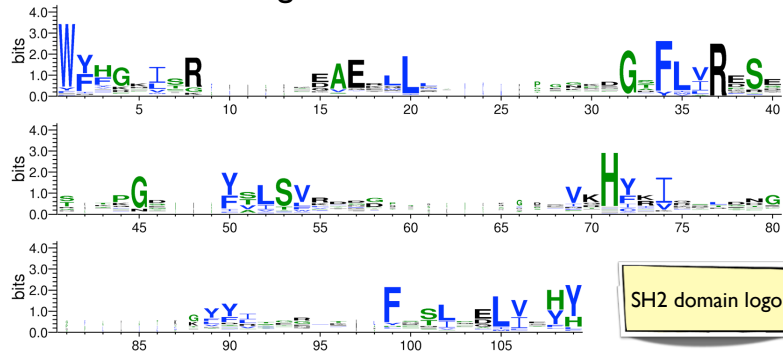
P06241 | 149-246      WYFGKLGKDAERQLLSFGN--PRGTFLIRESSETT-KGAYSLSIRDWDDMKGDHVKHXYKI
Q06124 | 6-102        WFHPNITGVAEENLLLTR-G--VDGSFLARPSKSN-PGDFTLVSVRRNG-----AVTHIKI
P62993 | 60-152       WFFGKIPRAKAEMLSKQ-R--HDGAFILRESSEA-PGDFSLSVKFGN-----DVQHFKV
P12931 | 151-248      WYFGKITRRESERLLNNAEN--PRGTFLVRESSETT-KGAYCLSVSDFDNARGLNVKHYKI
P41240 | 82-171       WFHGKITREQAERLLYPET---GLFLVRESSTNY-PGDYTLVCSVDG-----KVEHYRI
P00519 | 127-217      WYHGPVSRNAEYLLSSGIN---GSFLVRESSES-PGQRSISLRVYG-----RVYHYRI
P20936 | 181-272      WYHGKLDRTIAERLRQAGK---SGSYLIRESDRR-PGSFVLSFLSQMN-----VVNHFR
P42224 | 573-670      WNDGCMGFISKERERALLKDDQPGTFLRLRFSESSESGAITFTWVERS-----QNGGEPD
O60674 | 401-482      --HGPI$MDFAI$K$K$KAGN--QTGLVYLRC$PKD-FNKYFLTF$VEREN-VIEY$K$C$LI
      :      :      * : : *      .      :

P06241 | 149-246      RKLDNGGYITTRAQ-FETLQQLVQHY$ERAAGLC-CRLVVP-----
Q06124 | 6-102        QNTGDYDLYGGE-K-FATLAEVLQYMEHHGQLK-EKNGDVIELKYPL
P62993 | 60-152       LRDGAGKYFLWV-V-K-FNSLNLVDYHRS--V-SRNQOIFLRDIE-
P12931 | 151-248      RKLD$G$FYITSRTQ-FNSLQQLVAY$K$HADGLC-HRLT$TV$C-----
P41240 | 82-171       MYHAS-KL$IDE$VY-FENLMQLVEHYTS$ADGLC-TRLIKPK-----
P00519 | 127-217      NTASDGKLYV$SESR-FNTLAELVHH$STVADGLI-TTLHYPA-----
P20936 | 181-272      IAMCGDYIYIGGR--R-FSSLSDLIGY$HVSCLLK$EKL$LYP-----
P42224 | 573-670      F$HAVEPYTK$EL$AVTFPD$IRNYK$VMAAENIPENPLKYLYPN-----
O60674 | 401-482      TKNENE$YNL$G$TKNF$SLK$DLLNCYQ-----
      *      :
    
```

4

Les Motifs

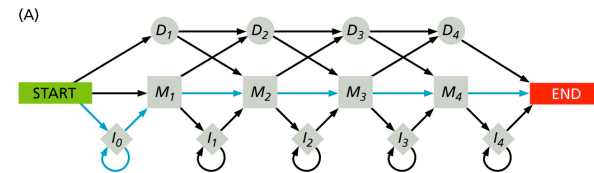
Les profils et les logo's donnent de l'information sur des motifs globaux au sein d'un MSA



5

Modèles Markoviens cachés

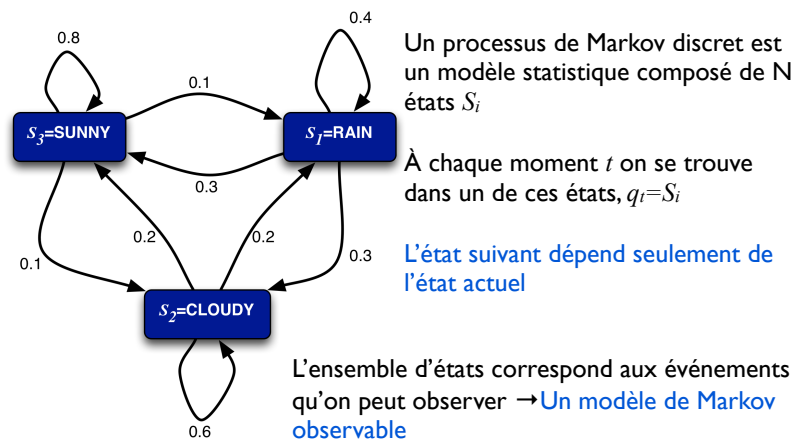
Les modèles Markoviens cachés (MMC) sont des modèles comme des PSSM qui enregistrent les motifs de conservation dans des alignements.



Ce sont des modèles probabilistes qui contiennent toutes les informations: les associations, les substitutions, les insertions et les délétions

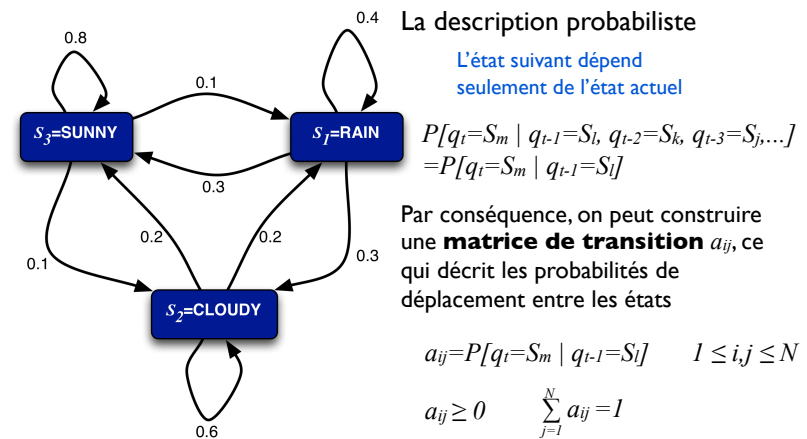
6

Les processus Markoviens discrets



7

Les processus Markoviens discrets



8

Les processus Markoviens discrets

La description probabiliste

L'état suivant dépend
seulement de l'état actuel

a_{ij}	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

S=sunny

R=rain

C=cloudy

$$P[q_t=S_m \mid q_{t-1}=S_l, q_{t-2}=S_k, q_{t-3}=S_j, \dots] \\ = P[q_t=S_m \mid q_{t-1}=S_l]$$

Par conséquent, on peut construire une matrice de transition a_{ij} , ce qui décrit les probabilités de déplacement entre les états

$$a_{ij} = P[q_t=S_m \mid q_{t-1}=S_l] \quad 1 \leq i, j \leq N$$

$$a_{ij} \geq 0 \quad \sum_{j=1}^N a_{ij} = 1$$

9

Les processus Markoviens discrets

Exemple 1

Quelle est la probabilité que le temps pour les sept jours suivants sera : sun-sun-rain-rain-sun-cloudy-sun, quand vous savez que l'état actuel est sunny?

a_{ij}	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq N$$

$$O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$$

$$\begin{aligned} P[O \mid \text{Model}] &= P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3] \\ &= P[S_3] \cdot P[S_3 \mid S_3] \cdot P[S_3 \mid S_3] \cdot P[S_1 \mid S_3] \\ &\quad \cdot P[S_1 \mid S_1] \cdot P[S_3 \mid S_1] \cdot P[S_2 \mid S_3] \cdot P[S_3 \mid S_2] \\ &= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\ &= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 0.0001536 \end{aligned}$$

10

Les processus Markoviens discrets

Exemple 2

Combien de jours on reste dans l'état rainy?

a_{ij}	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

En utilisant le modèle, la réponse est $(1/0.6) = 1.67$ jours

$$O = \{S_1, S_2, S_3, \dots, S_d, S_{d+1} \neq S_i\}$$

$$P[O \mid \text{Model}, q_1 = S_i] = (a_{ii})^{d-1} (1 - a_{ii}) = p_i(d)$$

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d)$$

$$= \sum_{d=1}^{\infty} d (a_{ii})^{d-1} (1 - a_{ii}) = \frac{1}{1 - a_{ii}}$$

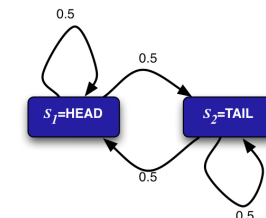
11

Les modèles Markoviens cachés

Voici un modèle Markovien construit pour une expérience de lancement d'une pièce de monnaie dans laquelle on peut observer la pièce:



a_{ij}	H	T
H	0,5	0,5
T	0,5	0,5



Supposez maintenant qu'on ne peut **pas observer la pièce** de monnaie (le lancement est fait derrière un rideau) et que le lanceur peut se servir d'une **pièce juste ou influencée**

12

Les modèles Markoviens cachés

Le lanceur sait :

que la pièce de monnaie juste a les probabilités suivantes pour *head* et *tail* : $P[H]=P[T]=0.5$ et la pièce influencée a $P[H]=0.75$ et $P[T]=0.25$

Pour des raison de sécurité le lanceur n'aime pas changer la pièce trop souvent. Donc $P[F|B]=P[B|F]=0.1$

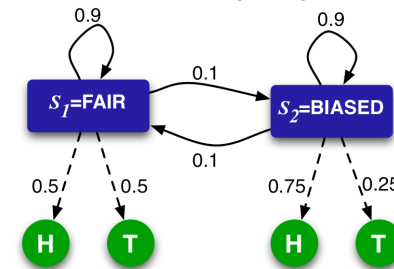
Les spectateurs verront seulement les symboles H ou T, et pas la pièce de monnaie qui a produit le symbole

Nous voulons déterminer quelle séquence de pièces de monnaie a été utilisée pour produire la séquence de symboles H et T

13

Les modèles Markoviens cachés

On peut construire un MMC qui représente ce problème

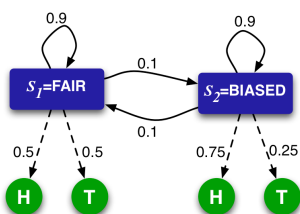


La grande différence avec le modèle Markovien discret est que les états du modèle et les événements (les symboles qui peuvent être observés) sont séparés

14

Les modèles Markoviens cachés

Un MMC est caractérisé par les éléments suivants:



N états : Généralement le modèle est ergodique, signifiant que chaque état peut être atteint à partir de chaque autre état

La matrice de transitions

a_{ij}	F	B
F	0,9	0,1
B	0,1	0,9

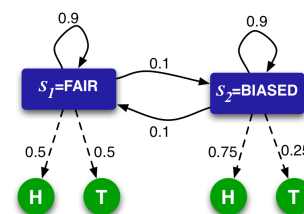
M observations distinctes par état

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

15

Les modèles Markoviens cachés

Un MMC est caractérisé par les éléments suivants:



La distribution des probabilités pour chaque symbole par état

$$B = \{b_j(k)\}$$

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j] \quad 1 \leq j \leq N$$

La distribution des états initiaux

$$\pi = \{\pi_i\}$$

$$\pi_i = P[q_t = S_i] \quad 1 \leq i \leq N$$

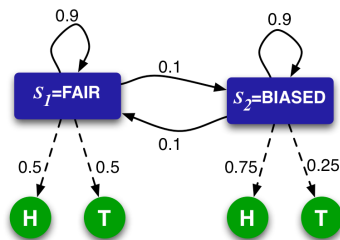
$$1 \leq k \leq M$$

$j=F$	$b_F(H)$	0,5
	$b_F(T)$	0,5
$j=B$	$b_B(H)$	0,75
	$b_B(T)$	0,25

16

Les modèles Markoviens cachés

Ainsi, un MMC est complètement défini par:



1. Trois paramètres pour la structure du modèle: N, M et V

2. un ensemble de 3 paramètres probabilistes: A, B and π

$$\lambda = (A, B, \pi)$$

17

Les modèles Markoviens cachés

On pourrait maintenant utiliser le modèle pour produire des séquences de symboles:

Algorithme:

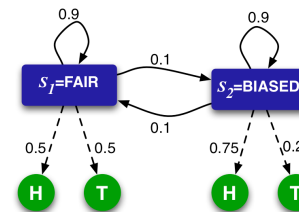
1. Choisissez un état S_j en utilisant les probabilités π

2. Mettez le temps t à 1, l'état au moment t est q_t ($=S_j$)

3. Choisissez un symbole v_k en utilisant la distribution $b_j(k)$ de l'état S_j

4. Déplacez-vous vers l'état suivant q_{t+1} en utilisant les probabilités de transition a_{jk} pour des liens qui commencent à l'état q_t

5. Augmentez t à $t+1$, retournez à l'étape 3 ou arrêtez-vous



THHHTHTTTTHT
HTHTHTTTTHT
...

18

Les modèles Markoviens cachés

Les MMC peuvent être utilisés pour résoudre 3 problèmes:

Le problème d'évaluation: Quelle est la probabilité qu'une séquence de symboles particulière (THHTTTTH) est produite par un certain modèle λ ?

19-1

Les modèles Markoviens cachés

Les MMC peuvent être utilisés pour résoudre 3 problèmes:

Le problème d'évaluation: Quelle est la probabilité qu'une séquence de symboles particulière (THHTTTTH) est produite par un certain modèle λ ?

Le problème de decodage: Etant donnée une séquence de symboles et un modèle λ , quelle est la séquence d'états $Q=q_1q_2...q_T$ la plus probable qui produirait cette séquence de symboles?

19-2

Les modèles Markoviens cachés

Les MMC peuvent être utilisés pour résoudre 3 problèmes:

Le problème d'évaluation: Quelle est la probabilité qu'une séquence de symboles particulière ($THHTTTTHH$) est produite par un certain modèle λ ?

Le problème de decodage: Etant donnée une séquence de symboles et un modèle λ , quelle est la séquence d'états $Q=q_1q_2...q_T$ la plus probable qui produirait cette séquence de symboles?

Le problème d'entraînement: Etant donnée la structure d'un modèle (N, M et V) et un ensemble de séquences, cherchez les paramètres optimaux pour le modèle λ qui produisent la meilleure adaptation aux données

19-3

Problème d'évaluation

$$P[O=\{THHTTTTHH\} | \lambda] ?$$

L'approche simple est de prendre chaque séquence d'états possibles avec la même taille que la séquence $THHTTTTHH$ et de déterminer la probabilité que cette séquence d'états produise la séquence de symboles

La somme de toutes ses probabilités est égale à $P[O | \lambda]$

Prenez par exemple la séquence d'états $Q=q_1q_2...q_L$

La probabilité que cette séquence produise la séquence $O=O_1O_2...O_L$ est

$$P[O|Q, \lambda] = \prod_{t=1}^T P[O_t|q_t, \lambda] = b_{q_1}(O_1) b_{q_2}(O_2) b_{q_3}(O_3) \dots b_{q_T}(O_T)$$

20

Problème d'évaluation 2

$$P[O|Q, \lambda] = \prod_{t=1}^T P[O_t|q_t, \lambda] = b_{q_1}(O_1) b_{q_2}(O_2) b_{q_3}(O_3) \dots b_{q_T}(O_T)$$

La probabilité de produire la séquence d'états $Q=q_1q_2...q_T$ est

$$P[Q|\lambda] = \pi_{q_1} a_{q_1q_2} a_{q_2q_3} \dots a_{q_{T-1}q_T}$$

$P[O | Q, \lambda] P[Q | \lambda]$ donne pour une séquence d'états la probabilité que cette séquence ait produit ces symboles. Pour toutes les séquences d'états possibles on obtient :

$$P[O | \lambda] = \sum_Q P[O | Q, \lambda] P[Q | \lambda]$$

La complexité est $O(2^{TN^T}) = \text{inefficace} !!!$

21

Problème d'évaluation 3

On peut simplifier ce processus en utilisant l'**algorithme forward** (vers l'avant) ou **backward** (vers l'arrière)

Pour l'implémentation de l'algorithme **vers l'avant** on définit $\alpha_t(i)$, qui est la **probabilité d'observer la séquence de symboles partielle** O_1, O_2, \dots, O_t et **d'arriver** dans l'état S_i à temps t , étant donné le modèle λ

$$\alpha_t(i) = P[O_1O_2...O_t, q_t=S_i | \lambda]$$

Cet $\alpha_t(i)$ peut être déterminé de manière inductive, en produisant la probabilité $P[O|\lambda]$ à la fin

La complexité est seulement $O(N^2T)$!!!

22

Problème d'évaluation 4

Le processus inductif:

Initialisation $\alpha_1(i) = \pi_i b_i(O_1)$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F	0,25							
B	0,125							

$$\alpha_1(F) = 0.5 \times 0.5 = 0.25$$

$$\alpha_1(B) = 0.25 \times 0.5 = 0.125$$

23

Problème d'évaluation 5

Le processus inductif :

induction $\alpha_{t+1}(j) = \left(\sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1})$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F	0,25	0,1188						
B	0,125	0,1031						

$S_1 = \text{FAIR}$ $\alpha_2(F) = [(0.25 \times 0.9) + (0.125 \times 0.1)] \times 0.5 = 0.1188$

$S_2 = \text{BIASED}$ $\alpha_2(B) = [(0.125 \times 0.9) + (0.25 \times 0.1)] \times 0.75 = 0.1031$

24

Problème d'évaluation 6

Le processus inductif :

induction $\alpha_{t+1}(j) = \left(\sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1})$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F	0,25	0,1188	0,0586					
B	0,125	0,1031	0,0785					

$S_1 = \text{FAIR}$ $\alpha_3(F) = [(0.1188 \times 0.9) + (0.1031 \times 0.1)] \times 0.5 = 0.0586$

$S_2 = \text{BIASED}$ $\alpha_3(B) = [(0.1031 \times 0.9) + (0.1188 \times 0.1)] \times 0.75 = 0.0785$

25

Problème d'évaluation 7

Le processus inductif :

induction $\alpha_{t+1}(j) = \left(\sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1})$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F	0,25	0,1188	0,0586	0,0303				
B	0,125	0,1031	0,0785	0,0191				

$S_1 = \text{FAIR}$ $\alpha_4(F) = [(0.0586 \times 0.9) + (0.0785 \times 0.1)] \times 0.5 = 0.0303$

$S_2 = \text{BIASED}$ $\alpha_4(B) = [(0.0785 \times 0.9) + (0.0586 \times 0.1)] \times 0.25 = 0.0191$

26

Problème d'évaluation 8

Le processus inductif :

terminaison $P[O|\lambda] = \sum_{i=1}^N \alpha_T(j)$ $P[THHTTTTHH|\lambda] = 0.0028$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F	0,25	0,1188	0,0586	0,0303	0,0146	0,0068	0,0031	0,0015
B	0,125	0,1031	0,0785	0,0191	0,0051	0,0015	0,0015	0,0013



27

Problème d'évaluation 9

L'algorithme vers l'avant

Initialisation

$$\alpha_1(i) = \pi_i b_i(O_1)$$

induction

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1})$$

terminaison

$$P[O|\lambda] = \sum_{i=1}^N \alpha_T(j)$$

28

Problème d'évaluation 10

Les mêmes résultats peuvent être obtenus en utilisant un algorithme qui fonctionne dans l'autre sens : **vers l'arrière**

Ici nous définissons $\beta_t(i)$ qui est la **probabilité d'observer une séquence de symboles partielle** $O_{t+1}, O_{t+2}, \dots, O_T$ qui **commence** à l'état S_i à temps t , étant donné le modèle λ

$$\beta_t(i) = P[O_{t+1}O_{t+2}\dots O_T, q_t = S_i | \lambda]$$

Ce $\beta_t(i)$ peut être obtenu aussi de manière inductive, produisant la probabilité $P[O|\lambda]$ quand on arrive au début ($t=0$)

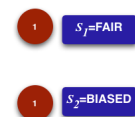
29

Problème d'évaluation 11

Le processus inductif :

Initialisation $\beta_T(i) = 1$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	T	T	T	H	H
F								1
B								1



30

Problème d'évaluation 12

Le processus inductif :

induction $\beta_t(i) = \sum_{j=1}^N a_{ij} \beta_{t+1}(j) b_j(O_{t+1})$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
F							0,525	1
B							0,725	1

$$\beta_7(F) = (1 \times 0.9 \times 0.5) + (1 \times 0.1 \times 0.75) = 0.5250$$

$$\beta_7(B) = (1 \times 0.9 \times 0.75) + (1 \times 0.1 \times 0.5) = 0.7250$$

31

Problème d'évaluation 13

Le processus inductif :

induction $\beta_t(i) = \sum_{j=1}^N a_{ij} \beta_{t+1}(j) b_j(O_{t+1})$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
F						0,2906	0,525	1
B						0,5156	0,725	1

$$\beta_6(F) = (0.5250 \times 0.9 \times 0.5) + (0.7250 \times 0.1 \times 0.75) = 0.2906$$

$$\beta_6(B) = (0.7250 \times 0.9 \times 0.75) + (0.5250 \times 0.1 \times 0.5) = 0.5156$$

32

Problème d'évaluation 14

Le processus inductif :

induction $\beta_t(i) = \sum_{j=1}^N a_{ij} \beta_{t+1}(j) b_j(O_{t+1})$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
F					0,1437	0,2906	0,525	1
B					0,1305	0,5156	0,725	1

$$\beta_5(F) = (0.2906 \times 0.9 \times 0.5) + (0.5156 \times 0.1 \times 0.25) = 0.1437$$

$$\beta_5(B) = (0.5156 \times 0.9 \times 0.25) + (0.2906 \times 0.1 \times 0.5) = 0.1305$$

33

Problème d'évaluation 15

Le processus inductif :

induction $P[O|\lambda] = \sum_{i=1}^N \beta_i(i) \pi_i b_i(O_1) \quad P[THHTTTTHH|\lambda] = 0.0028$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
F	0,0075	0,015	0,0315	0,0679	0,1437	0,2906	0,525	1
B	0,0071	0,0094	0,0116	0,0366	0,1305	0,5156	0,725	1

34

Problème d'évaluation 16

L'algorithme vers l'arrière

Initialisation

$$\beta_T(i) = 1$$

induction

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(O_{t+1})$$

terminaison

$$P[O|\lambda] = \sum_{i=1}^N \beta_1(i) \pi_i b_i(O_1)$$

35

Problème de décodage

Etant donnée une séquence de symboles $THHTTTTHH$ et un modèle λ , quelle est la séquence d'états $Q=q_1q_2...q_T$ la plus probable qui a produit cette séquence de symboles

$$\Rightarrow \text{maximisation de } P[Q=q_1q_2...q_T | THHTTTTHH, \lambda] ?$$

L' **Algorithme de Viterbi** est utilisée pour résoudre ce problème.

Pour l'implémentation nous définissons $\delta_t(i)$ qui est le **score** maximal dans **un chemin d'états** qui représente les t premières observations et qui **arrive** à l'état S_i , étant donné le modèle λ

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = S_i ; O_1, O_2, \dots, O_t | \lambda]$$

36

Problème de décodage 2

Un exemple de l'algorithme de Viterbi

Initialisation $\delta_1(i) = \pi_i b_i(O_1)$ and $\psi_1(i) = 0$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
$\delta(F)$	0,25							
$\delta(B)$	0,125							
$\psi(F)$	0							
$\psi(B)$	0							

$$\delta_1(F) = 0.5 \times 0.5 = 0.25$$

$$\delta_1(B) = 0.5 \times 0.25 = 0.125$$

37

Problème de décodage 3

Récursion $\delta_t(j) = \max_{i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$ and $\psi_t(j) = \operatorname{argmax}_{i \leq N} [\delta_{t-1}(i) a_{ij}]$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
$\delta(F)$	0,25	0,1125						
$\delta(B)$	0,125	0,0844						
$\psi(F)$	0	F						
$\psi(B)$	0	B						

$$\delta_2(F) = \max[(0.25 \times 0.9), (0.125 \times 0.1)] \times 0.5 = 0.1125$$

$$\delta_2(B) = \max[(0.125 \times 0.9), (0.25 \times 0.1)] \times 0.75 = 0.0844$$

$$\psi_2(F) = \operatorname{argmax}_{i \in \{F, B\}} [(0.125 \times 0.9), (0.25 \times 0.1)] = F$$

$$\psi_2(B) = \operatorname{argmax}_{i \in \{F, B\}} [(0.125 \times 0.9), (0.25 \times 0.1)] = B$$

38

Problème de décodage 4

Récursion $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$ and $\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
$\delta(F)$	0,25	0,1125	0,0506					
$\delta(B)$	0,125	0,0844	0,057					
$\psi(F)$	0	F	F					
$\psi(B)$	0	B	B					

$$\delta_3(F) = \max[(0.1125 \times 0.9), (0.0844 \times 0.1)] \times 0.5 = 0.0506$$

$$\delta_3(B) = \max[(0.0844 \times 0.9), (0.1125 \times 0.1)] \times 0.75 = 0.0570$$

$$\psi_3(F) = \operatorname{argmax}_{i \in \{F,B\}} [(0.1125 \times 0.9), (0.0844 \times 0.1)] = F$$

$$\psi_3(B) = \operatorname{argmax}_{i \in \{F,B\}} [(0.0844 \times 0.9), (0.1125 \times 0.1)] = B$$

39

Problème de décodage 5

Récursion $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$ and $\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
	T	H	H	T	T	T	H	H
$\delta(F)$	0,25	0,1125	0,0506	0,0228				
$\delta(B)$	0,125	0,0844	0,057	0,0128				
$\psi(F)$	0	F	F	F				
$\psi(B)$	0	B	B	B				

$$\delta_4(F) = \max[(0.0506 \times 0.9), (0.0570 \times 0.1)] \times 0.5 = 0.0228$$

$$\delta_4(B) = \max[(0.0570 \times 0.9), (0.0506 \times 0.1)] \times 0.25 = 0.0128$$

$$\psi_4(F) = \operatorname{argmax}_{i \in \{F,B\}} [(0.0506 \times 0.9), (0.0570 \times 0.1)] = F$$

$$\psi_4(B) = \operatorname{argmax}_{i \in \{F,B\}} [(0.0570 \times 0.9), (0.0506 \times 0.1)] = B$$

40

Problème de décodage 6

Terminaison $p^* = \max_{1 \leq i \leq N} [\delta_T(i)]$ and $q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	
	T	H	H	T	T	T	H	H	
$\delta(F)$	0,25	0,1125	0,0506	0,0228	0,0103	0,0046	0,0021	0,0009	p^*
$\delta(B)$	0,125	0,0844	0,057	0,0128	0,0029	0,0007	0,0005	0,0003	
$\psi(F)$	0	F	F	F	F	F	F	F	q_T^*
$\psi(B)$	0	B	B	B	B	B	B	B	

$$p^* = \max_{1 \leq i \leq N} [0.0009, 0.0003] = 0.0009 \quad q^* = \operatorname{argmax}_{1 \leq i \leq N} [0.0009, 0.0003] = F$$

En commençant, à $t=T$ on peut produire la séquence d'états optimale = FFFFFFFF

$$q_{t-1}^* = \psi_{t-1}(q_t^*)$$

41

Problème de décodage 7

L'algorithme de Viterbi

Initialisation

$$\delta_1(i) = \pi_i b_i(O_1) \text{ and } \psi_1(i) = 0$$

Récursion

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

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

Terminaison

$$p^* = \max_{1 \leq i \leq N} [\delta_T(i)] \text{ and } q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$$

Marche arrière

$$q_{t-1}^* = \psi_{t-1}(q_t^*)$$

42

Problème de décodage 8

Remarquez que les probabilités peuvent devenir très petites pour des longues séquences

Cela peut mener à des problèmes pour la représentation des probabilités au sein de l'ordinateur

Pour résoudre ce problème on peut changer les probabilités en **logarithmes de probabilités**

Toutes les multiplications deviennent des sommes !

Utiliser cette approche dans les algorithmes vers l'avant/vers l'arrière peut causer des problèmes

43

Problème de décodage 9

L'algorithme de Viterbi utilisant les transformation en logarithmes (e.g. $B_j(O_t) = \log(b_j(O_t))$)

Initialisation

$$\Delta_1(i) = \Pi_i + B_i(O_1) \text{ and } \psi_1(i) = 0$$

Récursion

$$\Delta_t(j) = \max_{1 \leq i \leq N} [\Delta_{t-1}(i) + A_{ij}] + B_j(O_t)$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\Delta_{t-1}(i) + A_{ij}]$$

Terminaison

$$p^* = \max_{1 \leq i \leq N} [\Delta_T(i)] \text{ and } q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\Delta_T(i)]$$

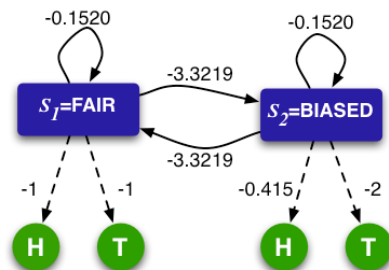
Marche arrière

$$q_{t-1}^* = \psi_{t-1}(q_t^*)$$

44

Problème de décodage 10

Pour réduire le nombre de calculs de logarithmes, les paramètres du modèle sont d'abord transformés



La fonction \log_2 est utilisé pour les transformations

45

Problème de décodage 11

Les résultats pour le même exemple et l'algorithme de Viterbi transformé

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	q_T^*
	T	H	H	T	T	T	H	H	
$\Delta(F)$	-2	-3,152	-4,304	-5,456	-6,608	-7,76	-8,912	-10,064	F
$\Delta(B)$	-3	-3,567	-4,1341	-6,2861	-8,4381	-10,5901	-11,1571	-11,7242	
$\Psi(F)$	0	F	F	F	F	F	F	F	
$\Psi(B)$	0	B	B	B	B	B	B	B	

La séquence d'états optimale est FFFFFFFF

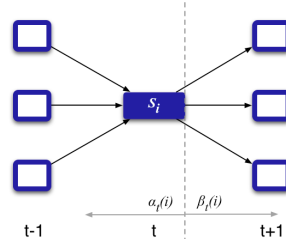
46

Problème de décodage 12

Le problème de décodage peut aussi être résolu en utilisant les algorithmes vers l'avant (*forward*) et vers l'arrière (*backward*)

Cette relation est définie par la **probabilité de traverser l'état S_i à temps t , étant donnée une séquence de symboles O et un modèle λ**

$$\gamma_t(i) = P[q_t = S_i | O, \lambda]$$

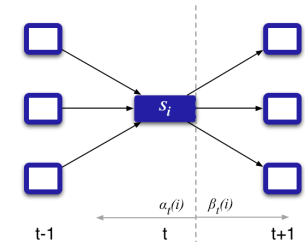
$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P[O|\lambda]} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \quad \sum_{i=1}^N \gamma_t(i) = 1$$


47

Problème de décodage 13

L'état q_t le plus probable au moment t est le maximum de ses probabilités $\gamma_t(i)$

$$q_t = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \quad 1 \leq t \leq T$$



48

Problème d'entraînement

Étant donnée la structure d'un modèle (N, M et V) et un ensemble de séquences de symboles, cherchez les paramètres optimaux pour le modèle $\lambda = (A, B, \pi)$ qui donnent la meilleure association avec les données

Il n'y a pas une solution analytique pour ce problème. Pour cette raison on a besoin des méthodes heuristiques qui rendent des solutions approximatives

Dans les approches on utilise des ensembles de séquences indépendantes

Deux approches

1. Chaque état associé avec les positions dans les séquences de symboles **est connu**
2. Chaque état associé avec les positions dans les séquences de symboles **n'est pas connu**

49

Problème d'entraînement 2

Supposons les deux ensembles (séquences de symboles et séquences d'états) suivants (normalement la taille de chaque ensemble sera plus grande que 20):

observations états

THHHTHTT
HHTTHHHH
HHHTTTTH
THTHTHHH
TTTTHHHH
HTHHHTHT
TTHHTTTH
THHTTHHH
...

BBBBFBFB
BBFFFBFB
BBBFFFFF
FFFFFBFB
FFFFFBFB
BBBFBFBF
FFFFFBFB
FFFBFBFB
...

Énumérez simplement le nombre d'émissions de chaque symbole, le nombre de transitions entre les états et le nombre de fois qu'on commence avec un certain état pour calculer les probabilités : $b_i(O_t)$, a_{ij} et π_i respectivement

50

Problème d'entraînement 3

Supposons les deux ensembles (séquences de symboles et séquences d'états) suivants (normalement la taille de chaque ensemble sera plus grande que 20):

observations	états	La probabilité départ π_i
THHHTHTT	BBBBFBBB	$\pi_F = \frac{\text{nombre de fois } F}{\text{nombre de fois } F \text{ et } B}$
HHTTHHHH	BBFFFBBB	
HHHTTTTH	BBBFFFFF	
THHTTHHH	FFFFFBBB	
TTTTHHHH	FFFFFBBB	$= 4/8 = 0.5$
HTHHHTHT	BBBFBBB	$\pi_B = \frac{\text{nombre de fois } B}{\text{nombre de fois } F \text{ et } B}$
TTHHTTTH	FFFFFFFF	
THHTTHHH	FFFB BBBB	
...	...	
		$= 4/8 = 0.5$

51

Problème d'entraînement 4

Supposons les deux ensembles (séquences de symboles et séquences d'états) suivants (normalement la taille de chaque ensemble sera plus grande que 20):

observations	états	La probabilité de transition a_{ij}
THHHTHTT	BBBBFBBB	$a_{FF} = \frac{\text{nombre de fois de } F \text{ à } F}{\text{nombre de fois de } F \text{ à } F \text{ ou } F \text{ à } B}$
HHTTHHHH	BBFFFBBB	
HHHTTTTH	BBBFFFFF	
THHTTHHH	FFFFFBBB	
TTTTHHHH	FFFFFBBB	$= 25/31 = 0.81$
HTHHHTHT	BBBFBBB	
TTHHTTTH	FFFFFFFF	
THHTTHHH	FFFB BBBB	
...	...	

a_{ij}	F	B
F	0,81	0,19
B	0,16	0,84

52

Problème d'entraînement 5

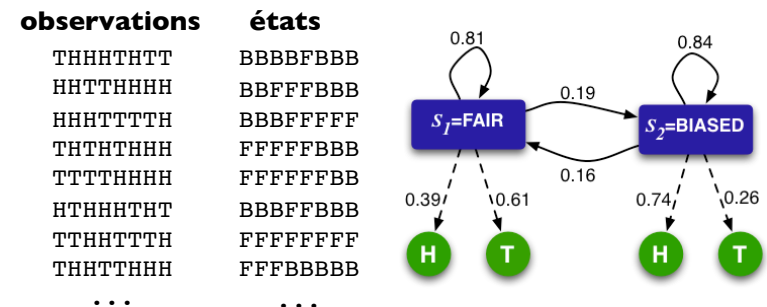
Supposons les deux ensembles (séquences de symboles et séquences d'états) suivants (normalement la taille de chaque ensemble sera plus grande que 20):

observations	états	La probabilité d'émission $b_i(O_i)$
THHHTHTT	BBBBFBBB	$b_F(T) = \frac{\text{nombre de fois } T \text{ est rendu de } F}{\text{nombre de fois que } F \text{ est rendu}}$
HHTTHHHH	BBFFFBBB	
HHHTTTTH	BBBFFFFF	
THHTTHHH	FFFFFBBB	
TTTTHHHH	FFFFFBBB	$= 20/33 = 0.61$
HTHHHTHT	BBBFBBB	
TTHHTTTH	FFFFFFFF	
THHTTHHH	FFFB BBBB	
...	...	

$j=F$	$b_F(H)$	0,39
	$b_F(T)$	0,61
$j=B$	$b_B(H)$	0,74
	$b_B(T)$	0,26

53

Problème d'entraînement 6



54

Problème d'entraînement 7

On appelle cette méthode le **maximum likelihood estimation (MLE)** ou **l'évaluation de probabilité maximale**

A_{kl} est défini comme le nombre de fois qu'il y a une transition entre les états k et l dans les séquences d'états

$B_k(s)$ est défini comme le nombre de fois que l'état k donne le symbole s dans les séquences d'états

$$a_{kl} = \frac{A_{kl}}{\sum_i A_{ki}} \quad b_k(s) = \frac{B_k(s)}{\sum_z B_k(z)}$$

Le plus grand problème de cette approche est sa tendance de surestimer les probabilités si la quantité de données est insuffisante (utilisez aussi les *pseudocounts* !!)

55

Problème d'entraînement 8

Quand on n'a pas l'ensemble d'états qui ont produit les séquences d'observations, le problème d'entraînement devient plus difficile

Utiliser des algorithmes qui essaient d'améliorer les estimations pour les paramètres A, B et π du modèle λ itérativement

1. Viterbi training

2. Baum-Welch algorithm

! A priori on prend un nombre fixe d'états N (e.g. 3 states) et symboles M (H or T)!

56

Problème d'entraînement 9

L'algorithme d'entraînement de Viterbi

Initialisez les paramètres A, B et π du modèle aléatoirement ou en utilisant les meilleurs résultats jusqu'à maintenant

Calculez pour chaque séquence de symboles le chemin le plus probable Q^* utilisant l'algorithme de Viterbi

Calculez A_{kl} et $B_k(s)$ comme en MLE en utilisant les chemins optimaux (utilisez aussi les *pseudocounts*)

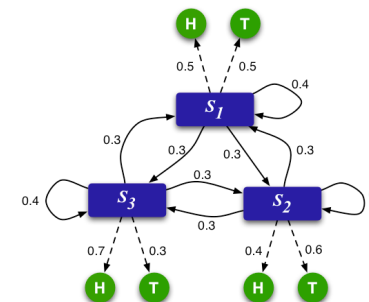
Calculez les nouveaux paramètres $a_{kl}, b_k(s)$ et π_k (comme en MLE)

Si le modèle a convergé, arrêtez le processus. Sinon, retournez au début de l'algorithme

57

Training problem 10

Par exemple, supposez qu'on a 3 états au lieu de 2.



observations

THHHTHTT
HHTTHHHH
HHHTTTTH
THTHTHHH
TTTTHHHH
HTHHHTHT
TTHHTTTH
THHTTHHH
...

58

Training problem 11

On calcule le chemin optimal pour chaque séquence en utilisant l'algorithme de Viterbi

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
	T	H	H	H	T	T	H	H
$\delta(S_1)$	0,1667	0,05	0,015	0,0047	0,002	0,0006	0,0002	0,00005
$\delta(S_2)$	0,2	0,048	0,0115	0,0038	0,0024	0,0006	0,0002	0,00007
$\delta(S_3)$	0,1	0,056	0,0235	0,0099	0,0018	0,0007	0,0001	0,00003
$\psi(S_1)$	0	1	1	3	3	1	1	1
$\psi(S_2)$	0	2	2	3	3	2	2	2
$\psi(S_3)$	0	2	3	3	3	3	3	2

q_T^*

The optimal sequence is 22333222

59

Problème d'entraînement 12

observations

états

THHHTHTT	22333222
HHTTTHHH	33322333
HHHTTTTH	33332223
THTHTHHH	22222333
TTTTTHHH	22222333
HTHHHTHT	33333222
TTHHTTTH	22233223
THHTTTHH	22332233
...	...

Maintenant on peut faire la même chose que pour le MLE pour calculer les nouveaux paramètres

60

Problème d'entraînement 13

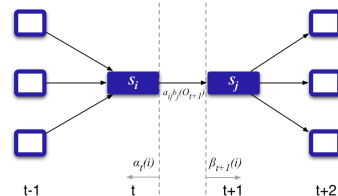
L'algorithme de Baum-Welch est une autre approche qui produit une meilleure estimation mais qui est plus complexe que l'approche précédente

On doit calculer la probabilité de traverser l'état S_i au moment t et l'état S_j au moment $t+1$, étant donné le modèle λ et une séquence d'observations

$$\xi_t(i,j) = P[q_t = S_i, q_{t+1} = S_j \mid O, \lambda]$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P[O \mid \lambda]}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}$$



61

Problème d'entraînement 14

En utilisant la probabilité $\xi_t(i,j)$ on peut calculer la probabilité d'être en état S_i au moment t pour une certaine séquence d'observations et un modèle λ

$$\gamma_t(i) = P[q_t = S_i \mid O, \lambda]$$

Cette probabilité peut être calculée en utilisant $\xi_t(i,j)$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i,j)$$

La probabilité $\gamma_t(i)$ est utilisée pour l'estimation du nombre de fois que l'état S_i est visité dans les séquences d'états =

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

62

Problème d'entraînement 15

De la même façon, on peut estimer le nombre de transitions entre les états S_i et S_j =

$$\sum_{t=1}^{T-1} \xi_t(i, j)$$

Ces deux estimations sont utilisées pour le calcul des nouveaux paramètres du modèle λ

On doit simplement calculer les valeurs $\alpha_t(i)$ (vers l'avant) et $\beta_t(i)$ (vers l'arrière) pour chaque séquence de symboles

63

Problème d'entraînement 16

Le calcul pour a_{kl} , $b_k(s)$ et π_k (pour une séquence de taille T):

La **probabilité de départ dans l'état i** au moment $t=1$ est

$$\pi'_i = \gamma_1(i)$$

La **probabilité de transition entre l'état i et l'état j**

$$A'_{ij} = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^T \gamma_t(i)}$$

La **probabilité d'observer le symbole k dans l'état S_j**

$$B'_{j(k)} = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \sum_{k \in \Omega} \gamma_t(j)}$$

Dans le numérateur on fait seulement la somme de tous les $\gamma_t(j)$ des états j qui produiraient le symbole v_k au moment t

64

Problème d'entraînement 17

L'algorithme de Baum-Welch

Prenez des paramètres arbitraires pour le modèle $\lambda = (A, B, \pi)$

Mettez les valeurs des variables temporaires A' et B' égales à leurs pseudocounts (ou zéro)

Pour chaque séquence $j = 1 \dots n$:

Calculez $\alpha_t(i)$ pour j en utilisant l'algorithme vers l'avant

Calculez $\beta_t(i)$ pour j en utilisant l'algorithme vers l'arrière

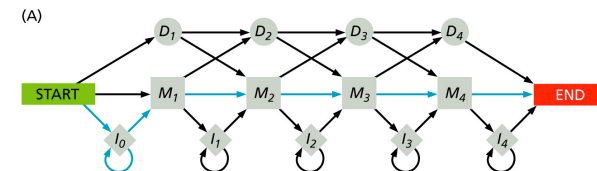
Ajoutez les contributions de la séquence j à A' et B'

Calculez les nouveaux paramètres a_{kl} , $b_k(s)$ et π_k (voyez MLE)

Arrêtez quand le *log-likelihood* du modèle est plus petit qu'un seuil ou qu'un nombre maximal d'itérations est passé

65

Les profils MMC



Comme décrit au début, les MMC donnent de l'information sur un alignement de plusieurs séquences

C'est une représentation alternative pour des PSSM ou profils

66

Les profils MMC 2

La structure pour un profil MMC est définie par les situations qu'on peut trouver entre des résidus dans des alignements

Deux résidus peuvent être associés à une certaine position = **l'état d'association** M_u



Les états d'association sont connectés par des transitions (la somme de toutes les transitions qui sortent de M_u est égal à 1)

Chaque état d'association M_u peut produire des symboles (p.e. un certain acide aminé)

67

Les profils MMC 3

Les deux autres possibilités sont les insertions et les délétions

Le résidu dans la séquence q peut être une insertion relative à une position de référence = **un état d'insertion** I_u



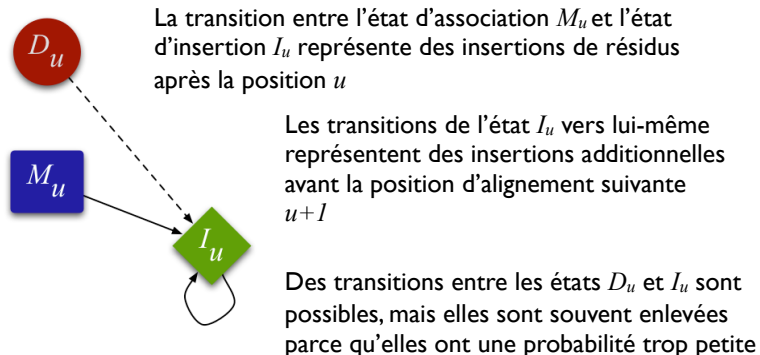
Le résidu dans la séquence q pourrait représenter une position qui apparaît plus tard dans la référence = **un état de délétion** D_u



68

Les profils MMC 4

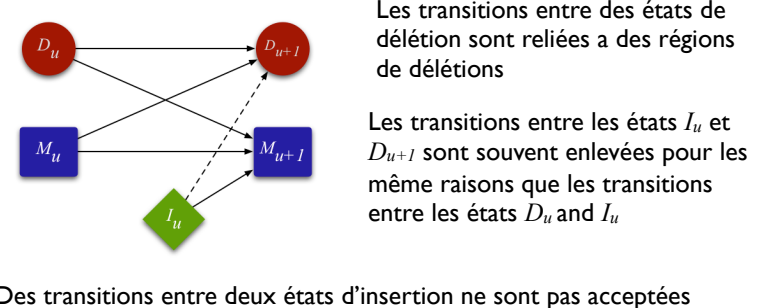
Quelques transitions entre les états dans l'étape u sont permises



69

Les profils MMC 5

Plusieurs transitions sont possibles entre les positions u et $u+1$ dans l'alignement



70

Les profils MMC 6

De la même façon que l'état d'association, l'état d'insertion peut émettre des symboles

Les symboles pour les états M_u

Les probabilités d'émission pour les états d'association sont dépendent des fréquences des acides aminés dans les colonnes de l'alignement de séquences

Les symboles pour les états I_u

Les probabilités d'émission pour les états d'insertion ne sont pas nécessairement liées à l'information dans l'alignement
On y assigne souvent les fréquences de base des résidus p_a (voyez les données swiss-prot)

71

Les profils MMC 7

Le nombre d'états d'association utilisé pour le MMC ne doit pas être égal à la taille des séquences

On prend parfois la moitié de la taille des séquences puisque ça réduit le nombre d'états d'insertion et d'enlèvement dans la structure

Quand on connaît le MSA, on prend pour les états d'association toutes les colonnes avec quelques espaces et pour les d'état d'insertion les colonnes avec beaucoup d'espaces

C'est souvent utile d'essayer des structures différentes = **model surgery**

72

Les profils MMC 8

Le nombre de probabilités de transition et d'émission peut devenir énorme

Pour faire une évaluation correcte pour toutes ces probabilités beaucoup de données (séquences) sont exigées

Le **chemin qu'on suit dans le profil MMC** montre quels résidus peuvent être alignés (état d'association) ou quels résidus ne peuvent pas (états d'insertion et d'enlèvement)

Quand le chemin contient beaucoup d'états d'insertion ou d'enlèvement la séquence n'est pas reliée au MMC

Tous les profils MMC ont **des états initial et final silencieux**

73

Les profils MMC 9

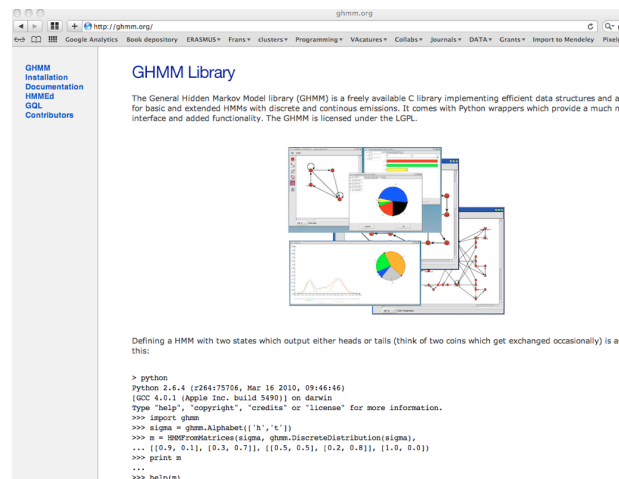
Le **problème d'évaluation** : La détermination de la probabilité qu'une séquence particulière est produite par une certaine séquence d'états dans un modèle donné

Le **problème de décodage** : La détermination de l'alignement d'une séquence à un profil MMC le plus probable

Le **problème d'entraînement** : Apprendre les paramètres pour un MMC qui représente un ensemble de séquences alignées (ou non alignées)

74

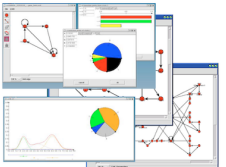
L'Outil



The screenshot shows the GHMM Library website. The browser address bar displays 'http://ghmm.org/'. The page title is 'GHMM Library'. A sidebar on the left contains links: 'GHMM', 'Installation', 'Documentation', 'HMMEd', 'GQL', and 'Contributors'. The main content area describes the GHMM library as a freely available C library for efficient data structures and algorithms for basic and extended HMMs. It includes a diagram of a Hidden Markov Model with states and transitions, and a code block showing a Python script to define and use an HMM.

GHMM Library

The General Hidden Markov Model library (GHMM) is a freely available C library implementing efficient data structures and algorithms for basic and extended HMMs with discrete and continuous emissions. It comes with Python wrappers which provide a much more interface and added functionality. The GHMM is licensed under the LGPL.



Defining a HMM with two states which output either heads or tails (think of two coins which get exchanged occasionally) is as this:

```
> python
Python 2.6.4 (r264:75766, Mar 16 2010, 09:46:46)
[GCC 4.0.1 (Apple Inc. build 5400)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> import ghmm
>>> sigma = ghmm.Alphabet(['h','t'])
>>> m = ghmm.FromMatrices(sigma, ghmm.DiscreteDistribution(sigma),
... [15.9, 0.1], [0.3, 0.7]], [[0.5, 0.5], [0.2, 0.8]], [1.0, 0.0])
>>> print m
...
>>> htm(m)
```