Modèles Mm

Tout se passe comme pour M1. Pour tous $x \in \mathcal{A}$ et $x_{1:n} \in \mathcal{A}^n$, on demande

$$\mathbb{P}(X_n = x | X_{1:n-1} = x_{1:n-1}) = \mathbb{P}(X_n = x | X_{n-m:n-1} = x_{n-m:n-1}).$$

Paramètres :

- Loi initiale ν sur $\mathcal{A}^m$:

$$\nu(x_{1:m}) = \mathbb{P}(X_{1:m} = x_{1:m}).$$

- Transitions q de $\mathcal{A}^m$ vers $\mathcal{A}$:

$$q(x_{1:m}, x) = q(x|x_{1:m}) = \mathbb{P}(X_{n+m} = x | X_{n:n+m-1} = x_{1:m}).$$

Loi d'une séquence ($n \geqslant m$)

$$\mathbb{P}_\nu(X_{1:n} = x_{1:n}) = \nu(x_{1:m})\, q(x_{1:m}, x_{m+1}) \dots q(x_{n-m:n-1}, x_n).$$

Log-vraisemblance ($n \gg m$)

$$\log \mathbb{P}_\nu(X_{1:n} = x_{1:n}) \approx \sum_{x,\mathtt{w}} N_n(\mathtt{w}x) \log q(x|\mathtt{w}),$$

somme sur les lettres x et les mots $\mathtt{w}$ de longueur m, et $N_n(\mathtt{w}x)$ le nombre d'occurrences du mot $\mathtt{w}x$ dans la séquence $x_{1:n}$.

Remarque fondamentale $(X_n)_n$ suit un modèle Mm si et seulement si $(Y_n)_n$ suit un modèle M1, avec

$$Y_n = X_{n:n+m-1} = (X_n, X_{n+1}, \dots, X_{n-m+1}).$$

Conséquence : tous les résultats démontrés pour les processus M1 fonctionnent aussi pour $(X_n)_n$ mais il faut passer par $(Y_n)_n$.

• Convergence de $(Y_n)_n$ vers un équilibre stochastique π_m unique et indépendant de la distribution de départ. Et π_m est l'unique distribution sur $\mathcal{A}^m$ solution du système suivant : pour toute lettre x de $\mathcal{A}$ et tout mot $\mathtt{w}$ de longueur $m-1$,

$$\pi_m(\mathtt{w}x) = \sum_{y\in\mathcal{A}} \pi_m(y\mathtt{w})\, q(y\mathtt{w}, x).$$

• Conséquence pour $(X_n)_n$: pour toute lettre x,

$$\lim_{n\to\infty} \mathbb{P}_\nu(X_n = x) = \rho(x).$$

Pour toute lettre x dans $\mathcal{A}$ et tout $0 \leqslant i \leqslant m-1$,

$$\rho(x) = \sum_{\mathtt{w}} \pi_m(x\mathtt{w}) = \sum_{\mathtt{w}} \pi_m(\mathtt{w}x) = \sum_{\mathtt{w}',\mathtt{w}''} \pi_m(\mathtt{w}'x\mathtt{w}''),$$

où les deux premières sommes portent sur les mots $\mathtt{w}$ de longueur $m-1$ et la dernière somme porte sur les mots $\mathtt{w}'$ et $\mathtt{w}''$ de longueurs respectives i et $m-1-i$.

• Convergence des fréquences empiriques de $(X_n)_n$ (qui sont aussi des fréquences empiriques de $(Y_n)_n$) : pour tout mot $\mathtt{w}$,

$$\lim_{n\to\infty} R_n(\mathtt{w}) = \Pi(\mathtt{w}).$$

Si $|\mathtt{w}| = \ell < m$,

$$\Pi(\mathtt{w}) = \pi_m(\mathtt{w} \times \mathcal{A}^{m-\ell}) = \sum_y \pi_m(\mathtt{w}y),$$

où la somme porte sur tous les mots y de longueur $m-\ell$.

Si $|\mathtt{w}| = \ell \geqslant m$,

$$\Pi(\mathtt{w}) = \pi_m(w_{1:m})\, q(w_{1:m}, w_{m+1}) \dots q(w_{\ell-m:\ell-1}, w_\ell).$$

Estimation statistique du modèle Mm

Pour toute lettre x et tout mot $\mathtt{w}$ de longueur m,

$$\widehat{q}(\mathtt{w}, x) = \frac{N_n(\mathtt{w}x)}{N_n(\mathtt{w})},$$

et

$$\widehat{\pi}_m(\mathtt{w}) = \frac{N_n(\mathtt{w})}{n}, \qquad \widehat{\rho}(x) = \frac{N_n(x)}{n}.$$

• **Remarque** Les fréquences des mots de longueur $\geqslant m+2$ sont toutes prédites par le modèle : par exemple, pour toutes lettres x et y et tout mot $\mathtt{w}$ de longueur m, $x\mathtt{w}y$ est un mot de longueur $m+2$ et il faut avoir

$$N_n(x\mathtt{w}y) \approx \frac{N_n(x\mathtt{w})\, N_n(\mathtt{w}y)}{N_n(\mathtt{w})}.$$

• **Remarque** Équilibre à trouver entre ordre du modèle et longueur de la séquence observée. Le modèle Mm comporte $|\mathcal{A}|^{m+1}$ paramètres (la matrice q) avec $|\mathcal{A}|^m$ contraintes puisque la somme de chaque ligne vaut 1, donc

$$|\mathcal{A}|^m\,(|\mathcal{A}|-1) \quad \text{paramètres.}$$

• **Un additif : l'état « Fin »**

Pas fait : si on veut modéliser des séquences de longueurs finies, on ajoute un état « Fin » tel que $q(\text{Fin}|\text{Fin}) = 1$ (un cimetière, pour les mathématiciens).

En résumé

• **Apprentissage dans un modèle Mm**

Comptage des mots jusqu'à la longueur $m+1$ incluse.

• **Vraisemblance dans un modèle Mm**

Les comptages des mots de longueur $m+1$ (et la loi initiale) suffisent à calculer $\mathbb{P}(\mathbf{x})$.

• **Discrimination entre modèles Mm**

On utilise la vraisemblance pour déterminer si une nouvelle séquence $\mathbf{x}$ est plutôt décrite par un modèle $+$ ou $-$, donc on calcule

$$\ell(\mathbf{x}) = \log\left(\frac{\mathbb{P}_+(\mathbf{x})}{\mathbb{P}_-(\mathbf{x})}\right) = \sum_{x,\mathtt{w}} N(\mathtt{w}x)\log\left(\frac{\mathbb{P}_+(x|\mathtt{w})}{\mathbb{P}_-(x|\mathtt{w})}\right).$$

Première partie des données : estimation de q_+ et q_-. Deuxième partie des données : loi empirique de $\ell(\mathbf{x})$ quand $\mathbf{x}$ suit le modèle $+$ puis quand $\mathbf{x}$ suit le modèle $-$.

Si les deux lois empiriques diffèrent nettement, on peut tester de nouvelles séquences, sinon, c'est raté.

Problème et suite du cours

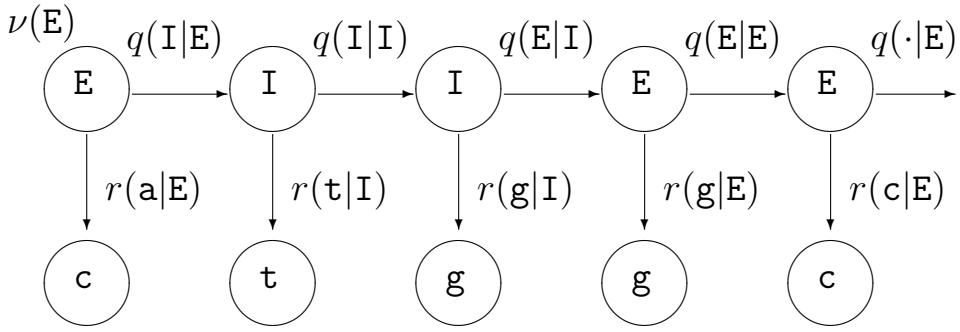
Dans tous les modèles Mm, la séquence est statistiquement homogène.

Pour l'ADN : gènes/régions intergéniques, introns/exons, etc.

Idée : décrire chaque type de région par un modèle Mm spécifique, puis recoller ces différents modèles.

Les chaînes de Markov cachées (HMM)

Exemple : recherche de structures dans un gène.
E = exon, I = intron.



Deux composantes :

• États $(S_n)_{n\geqslant 1}$: chaîne de Markov avec $S_n \in \mathcal{S}$, $\mathcal{S}$ fini. La loi initiale est ν avec $\nu(s) = \mathbb{P}(S_1 = s)$ et les transitions sont

$$\mathbb{P}(S_{n+1} = s'|S_n = s) = q(s, s') = q(s'|s).$$

• Observations $(X_n)_{n\geqslant 1}$: $X_n \in \mathcal{A}$, $\mathcal{A}$ fini et chaque état S_n émet l'observation X_n selon une loi qui dépend de S_n, donc

$$\mathbb{P}(X_n = x|S_n = s) = r(x|s) = r_s(x).$$

Dans l'exemple : $\mathcal{S} = \{\mathtt{E}, \mathtt{I}\}$ et $\mathcal{A} = \{\mathtt{a}, \mathtt{c}, \mathtt{g}, \mathtt{t}\}$.

Dans les vrais modèles, $\mathcal{S}$ est beaucoup plus gros : phases des codons, exons initiaux, intermédiaires, finaux, etc.

On a décrit le modèle « M1M0 » (on utilise en fait des modèles MmMk, voir plus tard).

Historique Reconnaissance de la parole (89). Génomique dès 1989. Mais aussi : modélisation de la croissance des plantes, courbes de consommation électrique, fiabilité de logiciels, etc.

Modèle complet M

$$\begin{cases} \text{Distribution de l'état initial :} \\ \qquad \nu(s) \text{ pour } s \in \mathcal{S} \\ \text{Probabilités de transition entre états :} \\ \qquad q(s'|s) \text{ pour } (s, s') \text{ dans } \mathcal{S}^2 \\ \text{Probabilités d'émission des observations :} \\ \qquad r(x|s) \text{ pour } (x, s) \text{ dans } \mathcal{A} \times \mathcal{S} \end{cases}$$

Trois calculs

On fixe un modèle $\mathbf{M}$ (c'est-à-dire ν, q et r) et une longueur T de séquences.
On note $\mathbf{x} = x_{1:T}$ une séquence d'observations dans $\mathcal{A}^T$.
On note $\mathbf{s} = s_{1:T}$ une séquence d'états dans $\mathcal{S}^T$.

- Loi des états seuls : $\mathbb{P}(S_{1:T} = \mathbf{s}) = \nu(s_1)\, q(s_2|s_1) \dots q(s_T|s_{T-1})$.
- Loi globale : $\mathbb{P}(X_{1:T} = \mathbf{x},\, S_{1:T} = \mathbf{s})$ vaut

$$\nu(s_1)\, r(x_1|s_1)\, q(s_2|s_1)\, r(x_2|s_2) \dots q(s_T|s_{T-1})\, r(x_T|s_T).$$

- Loi des observations seules : $\mathbb{P}(X_{1:T} = \mathbf{x})$ vaut

$$\sum_{\mathbf{s}} \nu(s_1)\, r(x_1|s_1)\, q(s_2|s_1)\, r(x_2|s_2) \dots q(s_T|s_{T-1})\, r(x_T|s_T),$$

où la somme porte sur toutes les séquences $\mathbf{s}$ dans $\mathcal{S}^T$.

Trois objectifs

• Le modèle $\mathbf{M}$ et la séquence $\mathbf{x}$ sont donnés. Calculer la probabilité $\mathbb{P}_{\mathbf{M}}(\mathbf{x})$ d'observer $\mathbf{x}$ sous le modèle $\mathbf{M}$.

[Évaluation : algorithmes avant/arrière]

• Le modèle $\mathbf{M}$ et la séquence $\mathbf{x}$ sont donnés. Déterminer la séquence d'états $\mathbf{s}$ qui donne la plus grande chance $\mathbb{P}_{\mathbf{M}}(\mathbf{x}|\mathbf{s})$ d'émettre $\mathbf{x}$ sous le modèle $\mathbf{M}$.

[Estimation : algorithme de Viterbi]

• La séquence $\mathbf{x}$ est donnée. Déterminer le modèle $\mathbf{M}$ qui donne la plus grande chance $\mathbb{P}_{\mathbf{M}}(\mathbf{x})$ d'émettre $\mathbf{x}$.

[Identification/apprentissage : algorithme de Baum-Welch]

Premier objectif : émission

On fixe $\mathbf{M}$. On veut calculer $\mathbb{P}(\mathbf{x})$. Approche directe :

$$\mathbb{P}(\mathbf{x}) = \sum_{\mathbf{s}\in\mathcal{S}^T} \mathbb{P}(\mathbf{x}|\mathbf{s})\,\mathbb{P}(\mathbf{s}).$$

Pour chaque suite d'états $\mathbf{s}$,

$$\begin{aligned}\mathbb{P}(\mathbf{x}|\mathbf{s}) &= r(x_1|s_1)\cdots r(x_T|s_T),\\ \mathbb{P}(\mathbf{s}) &= \nu(s_1)\,q(s_2|s_1)\cdots q(s_T|s_{T-1}).\end{aligned}$$

Pour chaque suite d'états $\mathbf{s}$, $2T$ multiplications, **mais $|\mathcal{S}|^T$ valeurs de s possibles**, donc $T\,|\mathcal{S}|^T$ opérations : boum !

La procédure « forward »

Pour chaque état s et chaque temps $1 \leqslant t \leqslant T$, on calcule la probabilité « forward »

$$\mathbf{f}_t(s) = \mathbb{P}(S_t = s,\, X_{1:t} = x_{1:t}).$$

• Début : au temps $t = 1$, pour chaque état s,

$$\mathbf{f}_1(s) = \nu(s)\,r(x_1|s).$$

• Récurrence $t \to t+1$ avec $1 \leqslant t \leqslant T-1$: pour chaque état s,

$$\mathbf{f}_{t+1}(s) = r(x_{t+1}|s)\sum_{s'\in\mathcal{S}} \mathbf{f}_t(s')\,q(s|s').$$

• Fin : $\mathbb{P}(\mathbf{x}) = \sum_{s\in\mathcal{S}} \mathbf{f}_T(s)$.

Nombre d'opérations : $T\,|\mathcal{S}|\,(|\mathcal{S}|+1)$. Taille mémoire : $T\,|\mathcal{S}|$ (car on veut souvent garder tous les $\mathbf{f}_t(s)$, voir plus bas).

Exemple : $|\mathcal{S}| = 5$, $T = 100$, on est passé de 10^{72} à 3000 opérations.

La procédure « backward »

Pour chaque état s et chaque temps $1 \leqslant t \leqslant T$, on calcule la probabilité « backward »

$$\mathbf{b}_t(s) = \mathbb{P}(X_{t:T} = x_{t:T} | S_t = s).$$

- Début : au temps $t = T$, $\mathbf{b}_T(s) = r(x_T|s)$.
- Récurrence $t \to t-1$ avec $2 \leqslant t \leqslant T$: pour chaque état s,

$$\mathbf{b}_{t-1}(s) = \sum_{s' \in \mathcal{S}} q(s'|s)\, r(x_{t-1}|s')\, \mathbf{b}_t(s').$$

- Fin : $\mathbb{P}(\mathbf{x}) = \sum_{s \in \mathcal{S}} \mathbf{b}_1(s)\, \nu(s)$.

Nombre d'opérations : $T\,|\mathcal{S}|\,(2|\mathcal{S}|)$. Taille mémoire : $|\mathcal{S}|$.

Conséquences des procédures avant/arrière

1) Loi a posteriori des états

La loi de l'état S_t sachant $\mathbf{x}$ se déduit de

$$\mathbb{P}(S_t = s,\, \mathbf{x}) = \mathbf{f}_t(s)\, \mathbf{b}_t(s).$$

Donc la loi conditionnelle de S_t vaut

$$\mathbb{P}(S_t = s|\, \mathbf{x}) = \frac{\mathbf{f}_t(s)\, \mathbf{b}_t(s)}{\mathbb{P}(\mathbf{x})}.$$

Pas besoin de calculer $\mathbb{P}(\mathbf{x})$ puisque, par exemple,

$$\mathbb{P}(\mathbf{x}) = \sum_{s' \in \mathcal{S}} \mathbf{b}_1(s')\, \nu(s') = \sum_{s' \in \mathcal{S}} \mathbf{f}_t(s')\, \mathbf{b}_t(s').$$

Rappel : les sommes sur $\mathcal{S}$ sont accessibles, pas celles sur $\mathcal{S}^T$.

2) Chemin uniformément optimal

Chemin $\mathbf{s}^{**} = s^{**}_{1:T}$ avec, pour chaque $1 \leqslant t \leqslant T$,

$$s^{**}_t = \text{argmax}\ \mathbb{P}(S_t = s|\mathbf{x}),\ \ s \in \mathcal{S}.$$

Attention : le chemin $\mathbf{s}^{**}$ dans l'espace d'états peut être illégal.

3) Évaluation d'une fonctionnelle des chemins

Soit g une fonction sur l'espace d'états $\mathcal{S}$. On peut calculer

$$G(t|\mathbf{x}) = \mathbb{E}(g(S_t)|\mathbf{x}) = \sum_s g(s)\, \mathbb{P}(S_t = s|\mathbf{x}).$$

Deuxième objectif : décodage

Le modèle $\mathbf{M}$ est fixé. La séquence $\mathbf{x} = x_{1:T}$ est fixée.

Objectif : « décoder » la séquence d'observations $\mathbf{x}$, c'est-à-dire trouver le chemin $\mathbf{s} = s_{1:T}$ dans l'espace $\mathcal{S}$ des états qui a engendré ce chemin $\mathbf{x} = x_{1:T}$ dans l'espace $\mathcal{A}$ des observations.

Chemin le plus probable :

$$\mathbf{s}^* = \operatorname{argmax} \mathbb{P}_{\mathbf{M}}(\mathbf{s}|\mathbf{x}).$$

Exemple typique : détection de gènes eucaryotes. Version (très) simplifiée : trois états cachés introns/exons/intergénique.

Concrètement, étant donnée la séquence

`ccgtactagctgtagctgtgac...atcgggggctctggatctgcagactgg`

où sont les exons ?

Tester tous les chemins est impossible. Donc algorithme de programmation dynamique.

L'algorithme de Viterbi

Pour chaque état s et chaque temps $1 \leqslant t \leqslant T$, on calcule la vraisemblance « partielle »

$$\mathbf{v}_t(s) = \max_{\mathbf{u}} \mathbb{P}_{\mathbf{M}}(S_{1:t-1} = \mathbf{u},\, S_t = s, X_{1:t} = x_{1:t}).$$

• Début : si $t = 1$, pour chaque état s,

$$\mathbf{v}_1(s) = \nu(s)\, r(x_1|s).$$

• Récurrence $t \to t+1$ avec $1 \leqslant t \leqslant T-1$: pour chaque état s,

$$\mathbf{v}_{t+1}(s) \;\; = \;\; r(x_{t+1}|s) \, \max_{s' \in \mathcal{S}} \Big(\mathbf{v}_t(s')\, q(s|s') \Big).$$

On garde en mémoire les états

$$\mathbf{m}_t(s) \;\; = \;\; \mathrm{argmax}_{s' \in \mathcal{S}} \Big(\mathbf{v}_t(s')\, q(s|s') \Big).$$

• Fin et rétro-propagation : $s_T^* = \mathrm{argmax}_{s' \in \mathcal{S}}\, \mathbf{v}_T(s')$.

Pour chaque temps $1 \leqslant t \leqslant T-1$,

$$s_t^* = \mathbf{m}_t(s_{t+1}^*).$$

Nombre d'opérations : $T\,|\mathcal{S}|\,(|\mathcal{S}|+1)$. Taille mémoire : $T\,|\mathcal{S}|$.

Problèmes d'underflow

On multiplie des petites probabilités. Par exemple, pour des séquences génomiques de $100'000$ bases, probabilités de l'ordre de $10^{-100'000}$.

Solution : le logarithme de $\prod_i$ vaut $\sum_i \log$ donc on manipule

$$\mathbf{w}_t(s) = \log \mathbf{v}_t(s).$$

L'étape de récurrence $t \to t+1$ devient : pour chaque s dans $\mathcal{S}$,

$$\mathbf{w}_{t+1}(s) \;=\; \log r(x_{t+1}|s) + \max_{s' \in \mathcal{S}} \Big(\mathbf{w}_t(s') + \log q(s|s') \Big).$$

On a toujours

$$\mathbf{m}_t(s) \;=\; \mathrm{argmax}_{s' \in \mathcal{S}} \Big(\mathbf{w}_t(s') + \log q(s|s') \Big).$$

La rétropropagation est similaire :

$$s_T^* = \mathrm{argmax}_{s' \in \mathcal{S}}\, \mathbf{w}_T(s'), \quad s_t^* = \mathbf{m}_t(s_{t+1}^*).$$

Stabilité numérique de l'algorithme « avant »

Même problème mais on ne peut pas passer au logarithme.

Une solution : renormaliser par a_t au temps t et calculer

$$\widetilde{\mathbf{f}}_t(s) = \mathbf{f}_t(s) \prod_{i \leqslant t} a_i.$$

Nouvelle récurrence ? (Exercice.)

Un exemple « historique » : les îlots CpG

Attention : CpG désigne `c` puis `g` sur un même brin, et non pas une paire complémentaire `c`-`g` en un locus donné des deux brins.

Principe biologique : la cytosine `c` des CpG a tendance à être méthylée, souvent en thymine `t`. Donc les dinucléotides `cg` sont plus rares que le produit des fréquences de `c` et de `g`...

...Sauf autour des promoteurs de certains gènes, où la méthylation est réprimée !

Fait d'expérience : plus de `cg` et de `c` et de `g` autour des régions promotrices qu'ailleurs ; on parle d'îlots CpG.

Objectif : trouver les îlots CpG.

Remarque : problème de dinucléotides donc M1 naturel.

Référence : Durbin, Eddy, Krogh, Mitchison (1998).

Ensemble d'entraînement de 60 kb, 48 îlots CpG.

Deux modèles M1 par EMV (comptages), notés + pour les îlots CpG et − pour le reste.

$$q_+ = \begin{pmatrix} .180 & .274 & .426 & .120 \\ .171 & .368 & .274 & .188 \\ .161 & .339 & .375 & .125 \\ .079 & .355 & .384 & .182 \end{pmatrix}.$$

$$q_- = \begin{pmatrix} .300 & .205 & .285 & .210 \\ .322 & .298 & .078 & .302 \\ .248 & .246 & .298 & .208 \\ .177 & .239 & .292 & .293 \end{pmatrix}.$$

Premier problème Identifier une séquence $\mathbf{x}$ comme étant un îlot CpG ou non.

Calculs de vraisemblance : le (log)score de $\mathbf{x}$ est

$$\log\left(\frac{\mathbb{P}_+(\mathbf{x})}{\mathbb{P}_-(\mathbf{x})}\right) = \sum_{x,x'\in\mathcal{A}} N_{\mathbf{x}}(x,x') \log\left(\frac{q_+(x,x')}{q_-(x,x')}\right).$$

Deuxième problème Trouver la place des îlots CpG dans une séquence donnée.

Approche naïve : utiliser des fenêtres glissantes et calculer le (log)score de chaque fenêtre. Inconvénient : quelle(s) longueur(s) de fenêtre choisir ?

En fait : HMM.

Option de Durbin et al. un peu dégénérée : en passant de + à − ou vice versa, on saute vers une des 4 lettres choisies avec la même probabilité.

Chemin +/− le plus probable estimé par Viterbi.

Donc prédiction des îlots CpG d'une nouvelle séquence.

Exemple :

```
a c g a t c g c g c c a c g g t t t a t a t a a g c a a
−−−−−++++++++++−−−−−−−−−−−−−−−−
```

La suite de + est une île prédite.

Troisième objectif : estimation

La séquence $\mathbf{x} = x_{1:T}$ est fixée.

Objectif : trouver le modèle $\mathbf{M}$ qui rende le mieux compte de la séquence d'observations $\mathbf{x}$. Modèle le plus probable :

$$\mathbf{M}^* = \operatorname{argmax} \mathbb{P}_{\mathbf{M}}(\mathbf{x}).$$

On utilise un cas particulier de l'algorithme EM (pour expectation/maximisation) : ré-estimation itérative et convergence vers un optimum local.

L'algorithme de Baum-Welch

0) Principe

On part d'un modèle $\mathbf{M}$; on ré-estime les valeurs des paramètres du modèle, ce qui donne $\widehat{\mathbf{M}}$ avec

$$\mathbb{P}(\mathbf{x}|\mathbf{M}) \leqslant \mathbb{P}(\mathbf{x}|\widehat{\mathbf{M}}).$$

Puis on recommence avec $\widehat{\mathbf{M}}$ en lieu et place de $\mathbf{M}$.

1) Notations

On fixe un modèle $\mathbf{M}$. Pour des états s et s',

$$\mathbf{c}_t(s, s') = \mathbb{P}_{\mathbf{M}}(S_t = s,\, S_{t+1} = s'|\mathbf{x}),$$

et

$$\mathbf{c}_t(s) = \mathbb{P}_{\mathbf{M}}(S_t = s|\mathbf{x}) = \sum_{s' \in \mathcal{S}} \mathbf{c}_t(s, s').$$

Si on somme $\mathbf{c}_t(s,s')$ et $\mathbf{c}_t(s)$ le long de la séquence, on obtient les quantités

$$\begin{aligned}\mathbf{C}(s,s') &= \sum_{t=1}^{T} \mathbf{c}_t(s,s') = \mathbb{E}_{\mathbf{M}}(N_T(s,s')|\mathbf{x}),\\ \mathbf{C}(s) &= \sum_{t=1}^{T} \mathbf{c}_t(s) = \mathbb{E}_{\mathbf{M}}(N_T(s)|\mathbf{x}).\end{aligned}$$

Enfin, on peut sommer $\mathbf{c}_t(s)$ le long de la séquence en ne gardant que les sites t où l'observation x_t vaut x, soit

$$\mathbf{C}_x(s) = \sum_{t=1}^{T} \mathbf{c}_t(s)\, \mathbf{1}(x_t = x).$$

2) Rappel sur avant/arrière

Pour un modèle $\mathbf{M}$ donné,

$$\mathbf{c}_t(s,s') = \frac{\mathbb{P}_{\mathbf{M}}(S_t = s,\, S_{t+1} = s',\, \mathbf{x})}{\mathbb{P}_{\mathbf{M}}(\mathbf{x})},$$

soit

$$\mathbf{c}_t(s,s') = \frac{\mathbf{f}_t(s)\, q(s'|s)\, r(x_{t+1}|s')\, \mathbf{b}_{t+1}(s')}{\mathbb{P}_{\mathbf{M}}(\mathbf{x})}.$$

Donc on peut calculer $\mathbf{c}_t(s)$, $\mathbf{C}(s,s')$, $\mathbf{C}(s)$ et $\mathbf{C}_x(s)$ comme des sommes (au moins à un facteur près).

3) L'étape $\mathbf{M} \to \widehat{\mathbf{M}}$

On est prêt à estimer les transitions des états par une nouvelle matrice $\widehat{q}$ et les émissions par une nouvelle matrice $\widehat{r}$. On utilise les estimateurs du maximum de vraisemblance dans le modèle $\mathbf{M}$, donc

$$\widehat{q}(s'|s) = \frac{\mathbf{C}(s,s')}{\mathbf{C}(s)}, \quad \widehat{r}(x|s) = \frac{\mathbf{C}_x(s)}{\mathbf{C}(s)}.$$

Le nouveau modèle $\widehat{\mathbf{M}}$ utilise les paramètres $\widehat{q}$ et $\widehat{r}$.

4) L'algorithme de Baum-Welch

• Initialiser la valeur de $\mathbf{M}$.

• Appliquer l'étape $\mathbf{M} \to \widehat{\mathbf{M}}$.

• Comparer les vraisemblances $\mathbb{P}(\mathbf{x}|\mathbf{M})$ et $\mathbb{P}(\mathbf{x}|\widehat{\mathbf{M}})$.

• Retourner à l'étape $\mathbf{M} \to \widehat{\mathbf{M}}$ jusqu'à stabilisation de la vraisemblance.

Remarque À chaque étape, $\mathbb{P}(\mathbf{x}|\mathbf{M}) \leqslant \mathbb{P}(\mathbf{x}|\widehat{\mathbf{M}})$.

Remarque Nécessité de définir un « critère d'arrêt » : différence des vraisemblances inférieure à un seuil absolu ; gain proportionnel inférieur à un seuil absolu ; idem sur un nombre d'itérations fixé à l'avance ; etc.

5) Extension au cas de plusieurs séquences observées

On suppose (souvent abusivement) que les I séquences observées $\mathbf{x}^i$ sont indépendantes les unes des autres et issues d'un même modèle $\mathbf{M}$, donc leur vraisemblance jointe sous $\mathbf{M}$ vaut

$$\mathbb{P}(\mathbf{x}^1|\mathbf{M})\,\mathbb{P}(\mathbf{x}^2|\mathbf{M})\cdots\mathbb{P}(\mathbf{x}^I|\mathbf{M}).$$

Pour chaque séquence $\mathbf{x}^i$ observée, on peut calculer $\mathbf{C}(s,s'|\mathbf{x}^i)$, $\mathbf{C}(s|\mathbf{x}^i)$ et $\mathbf{C}_x(s|\mathbf{x}^i)$ comme expliqué ci-dessus dans le cas d'une séquence.

Ensuite, on utilise $\widehat{q}(s'|s) = \dfrac{\mathbf{C}(s,s')}{\mathbf{C}(s)}$ et $\widehat{r}(x|s) = \dfrac{\mathbf{C}_x(s)}{\mathbf{C}(s)}$, en ayant posé $\mathbf{C}(s,s') = \displaystyle\sum_{i=1}^{I} \mathbf{C}(s,s'|\mathbf{x}^i)$, $\mathbf{C}_x(s) = \displaystyle\sum_{i=1}^{I} \mathbf{C}_x(s|\mathbf{x}^i)$, etc.

Tout se passe comme si on avait concaténé toutes les séquences en une seule.

6) Stabilité

En itérant Baum-Welch, on augmente la vraisemblance de la collection de séquences observées. Donc la vraisemblance converge.

Mais il n'y a pas (forcément) convergence dans l'espace des modèles. En pratique, la suite $\mathbf{M}_n$ des modèles obtenus après n itérations peut osciller violemment, même si le score de $\mathbf{M}_n$ converge.

Par ailleurs : problème des maxima locaux. Solution : utiliser plusieurs valeurs initiales différentes, faire tourner l'algorithme pour chacune de ces valeurs initiales, et espérer.

Ou partir de valeurs de $\mathbf{M}$ significatives biologiquement.

En conclusion

Procédure : 1. Choisir un ensemble d'états : codant/non codant, introns/exons/intergénique, prendre en compte les phases, lessignaux peptidiques, etc. 2. Choisir les transitions licites.
En pratique, permettre toutes les transitions donne de mauvais modèles (problèmes d'estimation). Donc utiliser les connaissances biologiques.
Une fois que la classe du modèle est définie, utiliser un ensemble d'entraînement pour estimer q et r. Ensuite, on peut analyser de nouvelles séquences par Viterbi, avant/arrière, etc.

Critiques Attention aux nouvelles séquences trop éloignées (au sens biologique) des séquences d'entraînement.

De réels problèmes Comment choisir l'ordre du modèle : critères BIC, AIC, etc. L'ordre k du modèle markovien engendrant les observations sous l'état s peut dépendre de s : VLHMM (Variable Length HMM). Pour éviter de faire exploser le nombre total de paramètres, on se permet des longueurs plus grandes dans certains états seulement. Modèles voisins des HMM : semi-chaînes de Markov cachées, etc.

Quelques applications des HMM à la génomique Hétérogénéité des séquences sans renseignements a priori. Transferts horizontaux de gènes. Recherche de motifs. Prédiction et annotation de gènes. Alignements de séquences. Reconstruction d'arbres phylogénétiques. Prédiction de structures secondaires. Etc.

Moralité On cherche à détecter une structure composée de modules élémentaires, chacun des modules est puisé dans une collection finie, on veut connecter les modules entre eux mais la mosaïque est inconnue.

– Fin –

VLMC (variable length Markov chains)

Synonyme : arbres de contexte.

On dispose d'un contexte c définie sur $\mathcal{A}^\infty$ qui associe à chaque passé $x_{-\infty:-1}$ son contexte de taille finie

$$c(x_{-\infty:-1}) = x_{-\ell:-1}.$$

Donc $\ell = \ell(x_{-\infty:-1})$ et on suppose que ℓ est toujours finie.

On impose souvent une contrainte logique sur les contextes : $c(x_{-\infty:-1}x) \leqslant c(x_{-\infty:-1}) + 1$.

Le contexte c (ou ℓ) étant donnée, on impose pour tout état x, tout temps t et tout passé $x_{-\infty:t}$ que

$$\mathbb{P}(X_{t+1} = x | X_{-\infty:t} = x_{-\infty:t}) = \mathbb{P}(X_{t+1} = x | X_{t-\ell+1:t} = x_{t-\ell+1:t}),$$

avec $\ell = \ell(x_{-\infty:t})$.

• Si ℓ est constante et vaut k, modèle Mk.

• Exemple :

$$c(x_{-\infty:-1}) = \begin{cases} \texttt{aa} & \text{si } x_{-2} = \texttt{a} \text{ et } x_{-1} = \texttt{a}, \\ \texttt{ca} & \text{si } x_{-2} = \texttt{c} \text{ et } x_{-1} = \texttt{a}, \\ \texttt{ga} & \text{si } x_{-2} = \texttt{g} \text{ et } x_{-1} = \texttt{a}, \\ \texttt{ta} & \text{si } x_{-2} = \texttt{t} \text{ et } x_{-1} = \texttt{a}, \\ \texttt{c} & \text{si } x_{-1} = \texttt{c}, \\ \texttt{g} & \text{si } x_{-1} = \texttt{g}, \\ \texttt{t} & \text{si } x_{-1} = \texttt{t}. \end{cases}$$

On peut construire des HMM à ordre variable, voir `Glimmer`, `Eugene`, etc.