Assimilation des données

Yanis Tazi

Décembre 2018

Contents

1 Introduction

L'assimilation de données est l'ensemble des méthodes qui permettent de combiner de manière optimale les informations disponibles sur un système :
– équations décrivant un modèle
– observations ou mesures physiques de la réalité
– statistiques d'erreurs (erreurs d'observation, bruits,...)
Ces informations sont souvent hétérogènes en nature, en quantité et en qualité.

Il existe différentes approches mathématiques pour résoudre ces problèmes d'assimilation des données. Parmi les plus courantes, on retrouve le traitement du signal , la théorie du contrôle et la théorie de l'estimation.

Dans ce cours, nous allons nous intéresser plus particulièrement à la théorie de l'estimation en étudiant plus en détails les réseaux bayésiens , les modèles de markov cachés ainsi que le filtre de kalman et ses principales variantes. Pour finir , nous étudierons en détail un article scientifique appliquant les modèles de markov cachés pour des applications en génétique.

Comme nous le verrons , les champs d'application de ces différents modèles sont très vastes. Historiquement développés pour des applications en géophysique et plus particulièrement en météorologie, ces modèles se sont très vite étendus à partir de la fin des années 90 . Les applications se généralisent aujourd'hui dans tous les domaines en passant de l'astronomie à la médecine.

Pour terminer cet introduction, citons quelques applications de l'assimilation des données:
–en météorologie: utiliser les observations des jours et des heures précédentes pour prédire la météo.
–en océanographie: prédire la montée des eaux et les possibilités de crues
–épidémiologie: prédiction précoce d'épidémie

2 Réseaux Bayésiens

2.1 Rappel

Le théorème de Bayes est à la base de l'inférence statistique et est utilisé pour actualiser les estimations d'une probabilité ou d'un paramètre à partir des observations faites et de leurs lois de probabilités.

$$\boxed{\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A)\mathbb{P}(B \mid A)}{\mathbb{P}(B)}}$$

De ce théorème, on en déduit directement la formule des probabilités totales:

$$\boxed{\mathbb{P}(B) = \sum_{i=1}^{n} \mathbb{P}(B \mid A_i) * \mathbb{P}(A_i)}$$

2.2 Définition

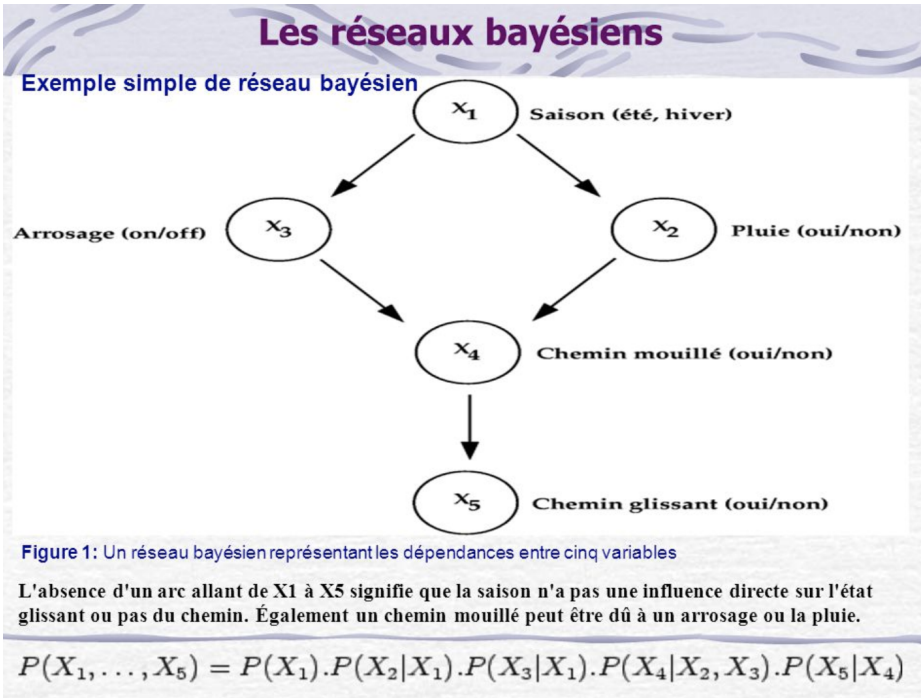

Figure 1: Réseau bayésien

Un modèle graphique probabiliste est une représentation graphique d'un ensemble de lois de probabilités multivariés. Il existe deux grandes familles de modèles graphique probabiliste: les graphes orientés et les graphes non orientés. Un réseau bayésien est un modèle graphique probabiliste qui représente des variables aléatoires sous forme de **graphe acyclique orienté**. Plus simplement, c'est un graphe qui ne contient pas de boucle dans lequel les noeuds représentent les variables aléatoires et les liens des influences entre ces variables. Les flèches permettent, quant à elles, de représenter les relations probabilistes ou déterministes entre ces variables.

2.3 Motivations

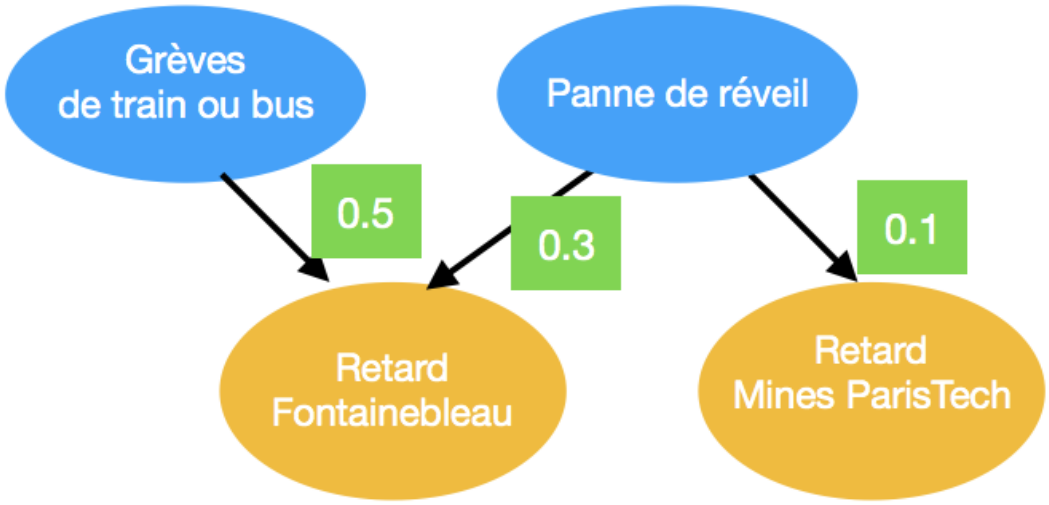

Figure 2: Réseau bayésien d'un élève optionnaire en géostatistique

Nous nous limitons dans ce cours aux noeuds représentant des variables aléatoires discrètes mais le lecteur doit savoir qu'il existe des applications aux variables aléatoires continues. Dans cet exemple, nos variables aléatoires sont catégoriques (vrai / faux ou 0 / 1) ce qui comme on le voit pose un problème dans le cas de variable continu et nécessite de définir un cadre strict et précis pour le cas continu.
Les arcs représentent dans un ordre descendant les relations de cause à effet. Pour simplifier, nous avons limiter la taille du réseau ici . Nous aurions pu rajouter un noeud "soirée arrosée" qui serait cause de la "panne de réveil" elle-même une cause des retards. La motivation des réseaux bayésiens est de pouvoir représenter de manière simple et efficace des relations non déterministes ainsi

que l'ensemble de l'information. Ainsi, afin de déterminer n'importe quelle probabilité relative à ce réseau, il suffit uniquement de connaître les probabilités des noeuds parents pour une des deux catégories (vrai ou faux).On déduit l'autre par complémentarité.
Supposons que $\mathbb{P}(\ Greves\ De\ Train\ Ou\ De\ Bus\ =\ Vrai) = 0.1$ et que $\mathbb{P}(\ Panne\ De\ Reveil\ =\ Vrai) = 0.2$, alors nous avons une vision exhaustive de notre réseau .
En effet, nous pouvons calculer la probabilité d'arriver en retard à Fontainebleau comme suit:

Notons: RF pour "RetardFontainebleau=Vrai" ; GTB pour "Greves De Train Ou De Bus=Vrai" et PR pour "Panne De Reveil=Vrai".

$$\boxed{\mathbb{P}(RF) = \mathbb{P}(GTB) * \mathbb{P}(RF \mid GTB) + \mathbb{P}(PR) * \mathbb{P}(RF \mid PR)}$$

Enfin, une des façons de représenter un réseau bayésien est d'exprimer la loi de probabilité jointe sur l'ensemble des variables aléatoires associées à ce réseau.
La formule générale est comme suit:

$\boxed{\mathbb{P}(E) = \prod_{x \in E} \mathbb{P}(x \mid pa(x))}$ avec $pa(x)$ les parents de x.

Exemple concrêt de réduction de variables:
-2 maladies A et B non mutuellement exclusives sous forme de **2** variables aléatoires **binaires**
-4 saisons corrélés aux maladies sous forme d'**1** variable aléatoire avec **4** catégories
-2 symptomes S_1 et S_2 sous forme de **2** variables aléatoires **binaires**
Pour ces 5 variables , l'espace de probabilité est constitué de $64 = (2^2) * (4) * (2^2)$ valeurs.
En choisissant un graphe adapté:

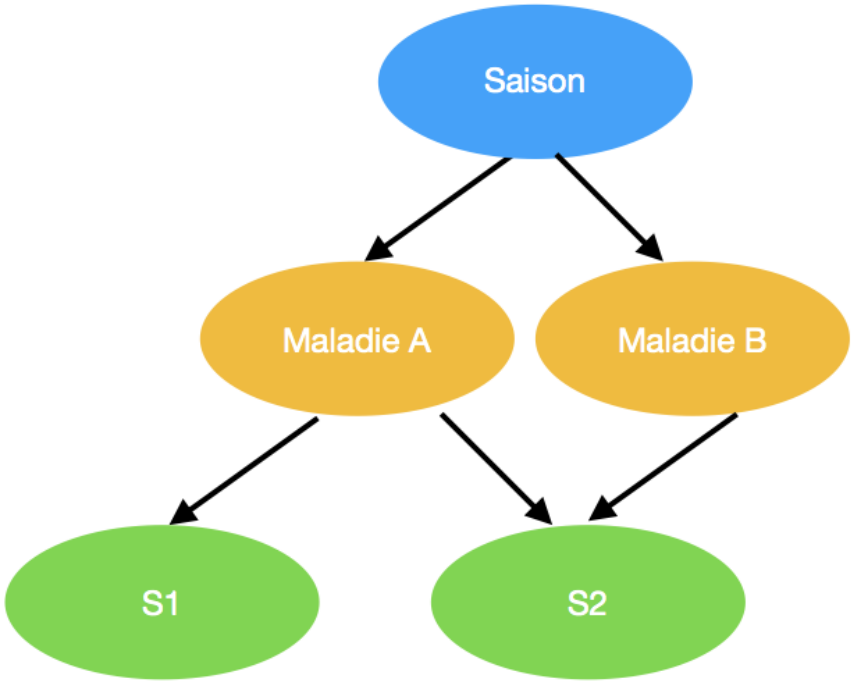

Figure 3: Réseau bayésien pour exemple maladie

Le réseau bayésien propose une factorisation de la loi de probabilité jointe en utilisant : $3+4+4+4+2 = 17$ paramètres au lieu de $64-1 = 63$ paramètres!

2.4 Algorithme simple de construction d'un réseau bayésien

1. Choisir un ensemble variables pertinentes ordonnées X1, X2, ..., Xm.
2. Pour i=1 à m
   1. Ajouter Xi au graphe
   2. Parents(Xi) = sous-ensemble minimal de X1, ..., Xi-1 tel que indépendance conditionnelle de Xi et des éléments de X1, ..., Xi-1 étant donné Parents(Xi)
   3. Définir la table de probabilités $\mathbb{P}(Xi = k \mid \{valeurs\ affectees\ aux\ Parents(Xi)\})$

2.5 Conclusion

La distribution de probabilité jointe dans une table nous permet de déduire n'importe quelle probabilité conditionnelle. Cependant, la taille de la table est exponentielle au nombre de variables. Il convient donc de trouver une représentation plus adaptée et d'assouplir certaines hypothèses lorsque cela est possible afin de remplacer cette table par un réseau bayésien dont la taille est

largement inférieure. La représentation explicite de la distribution de probabilité jointe pose problème à la fois en terme de puissance de calcul , de ressources humaines et numériques.

3 Modèles de Markov cachés

3.1 Modèle de Markov

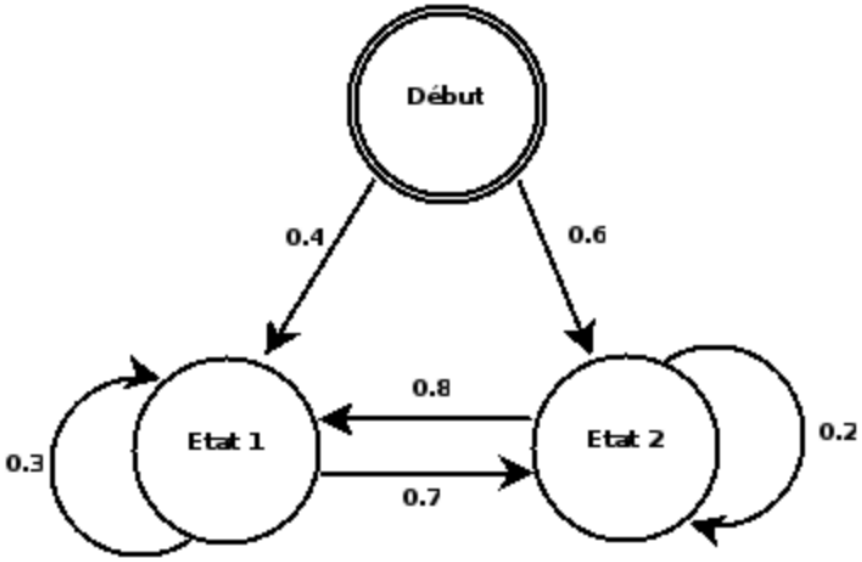

Figure 4: Modèle de Markov

Tout d'abord , rappelons ce qu'est un modèle de Markov ou chaîne de Markov d'ordre 1 . C'est un modèle statistique composé d'états et de transitions unidirectionnelles représentant la probabilité de passer d'un état à un autre ou de rester dans le même état. C'est un processus stochastique possédant la propriété de Markov i.e l'information pour prédire le future est entièrement connue à l'état présent et ne dépend pas du passé.

$$\boxed{\mathbb{P}(X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, ..., X_1 = i_1) = \mathbb{P}(X_{t+1} = j \mid X_t = i)}$$

3.2 Modèle de Markov caché

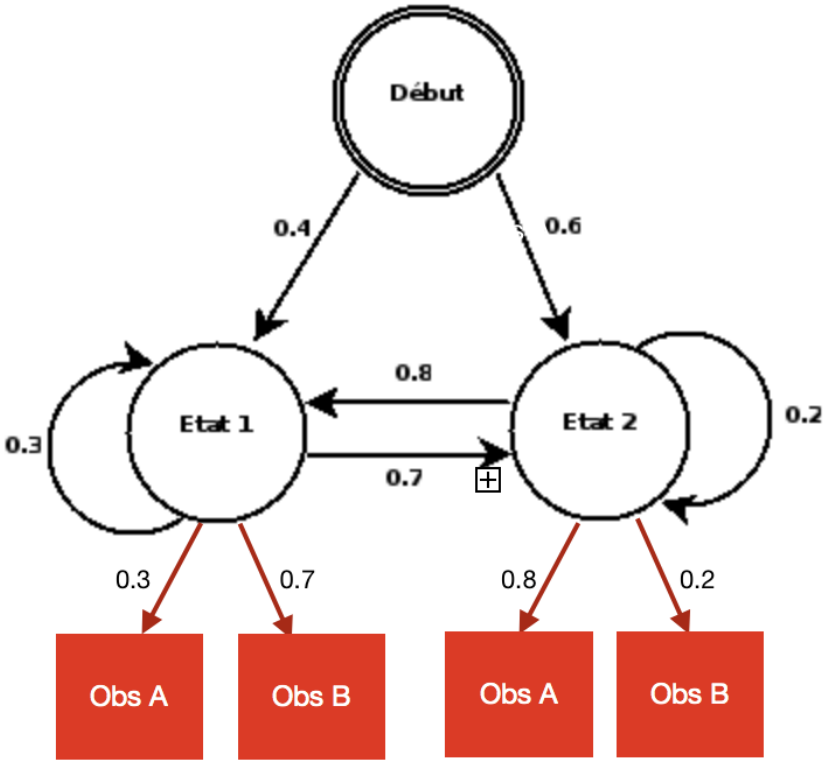

Figure 5: Modèle de Markov caché

Contrairement à un modèle de Markov simple, nous ne sommes pas en mesure d'observer directement les états : les états , comme le nom du modèle l'indique, sont cachés. En revanche , nous avons accés aux observations émises par ces états. Ainsi, notre modèle contient un paramètre probabilistique en plus du modèle de markov simple : les probabilités d'émissions représentant la probabilité d'émission des observations pour chaque état représentée par les flèches rouge dans le schéma ci-dessus.
La probabilité jointe du couple $(x, y) = (x_0, ..., x_n, y_0, ..., y_n)$ où les x_i sont les observations correspondant aux états y_i se calcule comme suit:

$$\mathbb{P}(x, y) = \mathbb{P}(y_0) \prod_{i=0}^{n-1} \mathbb{P}(y_{i+1} \mid y_i) \prod_{i=0}^{n} \mathbb{P}(z_i \mid y_i)$$

3.3 Modèle de mélange et modèle de mélange gaussien

Un modèle de mélange ou mixture model en anglais est un modèle probabiliste de représentation de sous-population au sein d'une population globale . L'objectif est de définir un mélange de distribution représentant la distribution de probabilité des observations au sein de la population.

Le modèle de mélange classique est le modèle de mélange gaussien qui est un modèle statistique utilisé pour exprimer la distribution de variables aléatoires comme une somme pondérée de plusieurs gaussiennes. Ce modèle est donc caractérisé par la moyenne , l'écart-type et la pondération de chacune de ces gaussiennes.

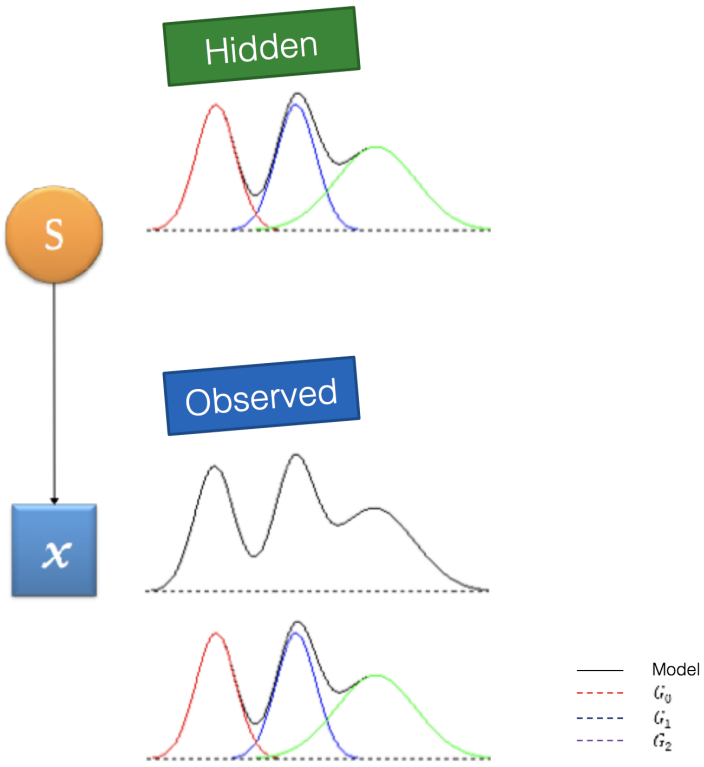

Figure 6: Modèle de mélange gaussien

Une des méthodes d'apprentissage des paramètres des modèles de mélange est d'utiliser les modèles de markov cachés comme suit:

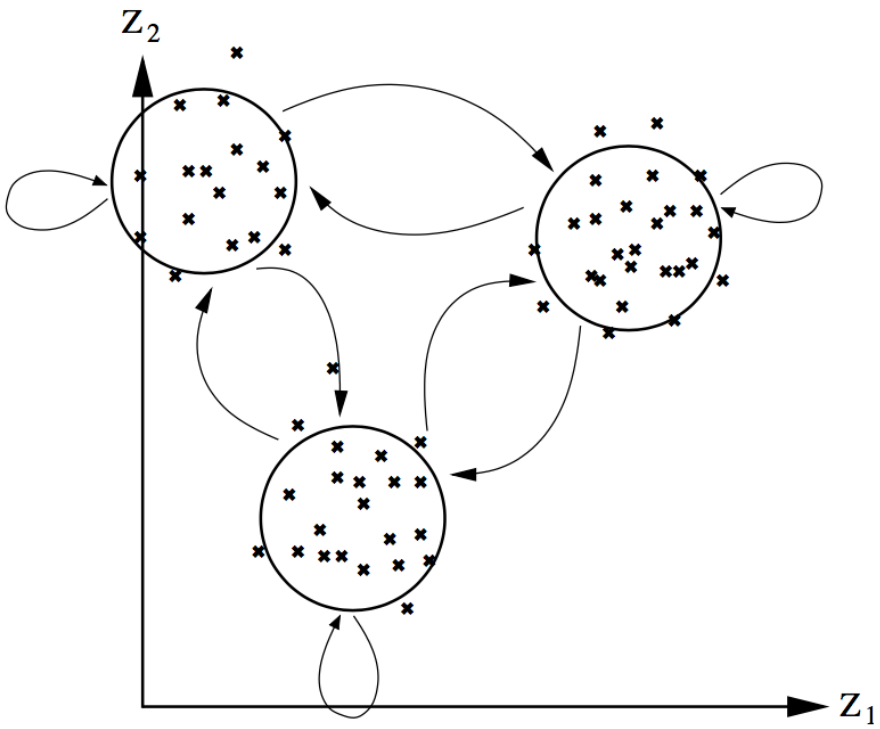

Figure 7: Généralisation des modèles de mélanges par modèle de markov caché

La généralisation se fait en introduisant une dépendance entre l'instant t et $t+1$ autorisant les composantes du modèle de mélange dépendre du choix de ces dernières à l'étape précédentes.

4 Filtre de Kalman

Le filtre de Kalman est un ensemble d'équations mathématiques permettant d'obtenir une meilleure estimation des états futur d'un système malgré des incertitudes et incomplétudes de mesure. C'est un ensemble d'équations efficace d'obtention de la solution optimale d'un problème dont la connaissance est partielle.

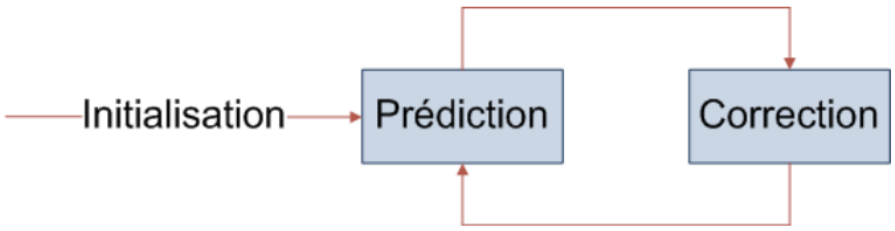

Figure 8: Etape du filtre de kalman

4.1 Filtre de Kalman linéaire et représentation d'état

La représentation d'état permet de modéliser un système dynamique en utilisant des variables d'états. Bien que le filtre de Kalman et le modèle de markov ont été développés initialement indépendemment l'un de l'autre, une représentation d'état peut-être vu comme un modèle de markov caché particulier utilisant un modèle gaussien de probabilité et où les noeuds sont des vecteurs réels.

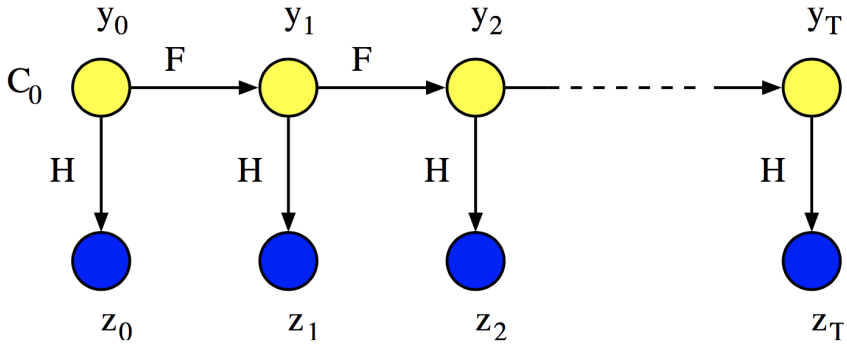

Figure 9: Représentation d'état graphique

$y_{0,\dots,T}$ resp $z_{0,\dots,T}$ sont les états successifs resp les observations successives et C_0 la covariance de l'état initial t_0.

Equation linéaire de la représentation d'état:

$y_{t+1} = Fy_t + Gq_t$, avec $q_t \sim \mathcal{N}(0_{\Re^n}, Q)$.

$\Rightarrow (y_{t+1} \mid y_t) \sim \mathcal{N}(Fy_t, GQG^T)$.

Equation linéaire de l'observation:

L'observation z_t est une transformation linéaire de l'état y_t à laquelle il faut ajouter un terme d'erreur nul en moyenne u_t.

$z_t = Hy_t + u_t$, avec $u_t \sim \mathcal{N}(0_{\Re^n}, R)$

$\Rightarrow (z_t \mid y_t) \sim \mathcal{N}(Hy_t, R)$

Distribution de l'état:

- $y_0 \sim \mathcal{N}(0_{\Re^n}, C_0)$ et $y_{t+1} = Fy_t + Gq_t$ avec $q_t \sim \mathcal{N}(0_{\Re^n}, Q)$

$\Rightarrow \forall t \in T$, y_t est d'espérance nulle (simple récurrence sous condition que y_0 soit d'espérance nulle)

$C_{t+1} = \mathbb{E}((y_{t+1} - 0_{\Re^n})(y_{t+1} - 0_{\Re^n})^T) = \mathbb{E}((Fy_t + Gq_t)(Fy_t + Gq_t)^T)$

Comme q_t est indépendant de y_t et de moyenne nulle:

$\Rightarrow \boxed{C_{t+1} = F\mathbb{E}(y_t y_t^T)F^T + GQG^T}$

Problème:

Le problème d'inférence de représentation d'états consiste à calculer la probabilité à posteriori des états connaissant la séquence des observations.
Il s'agit là d'un problème récursif qui se décompose comme suit:
-Filtrage : $\mathbb{P}(y_t \mid z_0, ..., z_t)\ \forall t \in 0, ..., T$

-Prediction: $\mathbb{P}(y_{t+n} \mid z_0, ..., z_t)\ \forall t \in 0, ..., T\ ,\ n = 1, 2, ...$

-Lissage: $\mathbb{P}(y_t \mid z_0, ..., z_T)\ \forall t \in 0, ..., T$

Nous voulons estimer la probabilité de l'état y_t connaissant les observations $z_0, ..., z_t$ ce qui revient à calculer : $\mathbb{P}(y_t \mid (z_0, ..., z_t))$ avec :

$\hat{y}_{t|t} = \mathbb{E}(y_t \mid (z_0, ..., z_t))$ et $C_{t|t} = \mathbb{E}[(y_t - \hat{y}_{t|t})(y_t - \hat{y}_{t|t})^T \mid (z_0, ..., z_t)]$.

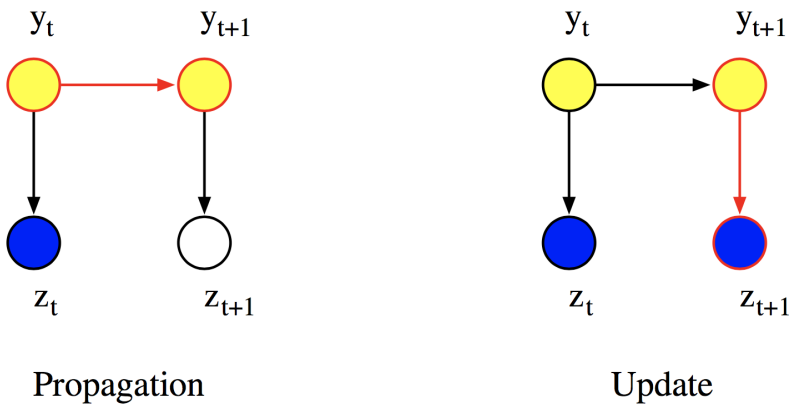

Figure 10: Transformation en deux étapes pour le filtre de kalman

A chaque incrément temporel, nous voulons passer de $(\hat{y}_{t|t}, C_{t|t})$ à $(\hat{y}_{t+1|t+1}, C_{t+1|t+1})$. Pour cela, il faut suivre le schéma et décomposer la transformation en **2** étapes:

1) Propagation: $\mathbb{P}(y_t \mid (z_0, ..., z_t)) \rightarrow \mathbb{P}(y_{t+1} \mid (z_0, ..., z_t))$

Il suffit alors de calculer la moyenne et la matrice de covariance en utilisant $\hat{y}_{t|t}$ et $C_{t|t}$.

$$\hat{y}_{t+1|t} = \mathbb{E}(y_{t+1} \mid (z_0, ..., z_t)) = \mathbb{E}[(Fy_t + Gq_t) \mid (z_0, ..., z_t)]$$

$$\hat{y}_{t+1|t} = F\hat{y}_{t|t} + G\mathbb{E}(q_t \mid (z_0, ..., z_t))$$

De plus, q_t étant indépendant par rapport à $(z_0, ..., z_t)$ et de moyenne nulle, on a:

$$\Rightarrow \boxed{\hat{y}_{t+1|t} = F\hat{y}_{t|t}}$$

$$C_{t+1|t} = \mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^T \mid (z_0, ..., z_t))$$

$$C_{t+1|t} = \mathbb{E}[(Fy_t + Gq_t - F\hat{y}_{t|t})(Fy_t + Gq_t - F\hat{y}_{t|t})^T \mid (z_0, ..., z_t)]$$

Comme q_t est indépendant de y_t et de $(z_0, ..., z_t)$ ainsi que de moyenne nulle, on peut simplifier cette expression:

$$C_{t+1|t} = F\mathbb{E}[(y_t - \hat{y}_{t|t})(y_t - \hat{y}_{t|t})^T \mid (z_0, ..., z_t))F^T + G\mathbb{E}(qtqt^T)G^T$$

$$\Rightarrow \boxed{C_{t+1|t} = FC_{t|t}F^T + GQG^T}$$

2) Mise à jour : $\mathbb{P}(y_{t+1} \mid (z_0, ..., z_t)) \rightarrow \mathbb{P}(y_{t+1} \mid (z_0, ..., z_t, z_{t+1}))$

Il s'agit de mettre à jour y_{t+1} en incorporant dans le système l'observation z_{t+1}.

Nous souhaitons maintenant connaître l'espérance et la matrice de covariance de la variable aléatoire conditionnelle $(y_{t+1} \mid z_0, ..., z_{t+1})$:

Rappel:

Pour cela, rappelons les propriétés des distributions conditionnelles de loi normale multidimensionnelle:

$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ et $\Sigma = \begin{pmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{pmatrix}$ et $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}(\mu, \Sigma)$ alors:

$(X_1 \mid X_2 = a) \sim \mathcal{N}(\mu_a, \Sigma')$ tels que:

$\boxed{\mu_a = \mu_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(a - \mu_2)}$ $(*)$ et $\boxed{\Sigma' = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}}$ $(**)$.

Distribution conditionnelle de $y_{t+1} \mid (z_0, ..., z_t, z_{t+1})$**:**

Calculons tout d'abord l'espérance conditionnelle de $z_{t+1} \mid (z_0, ..., z_t)$:

$\hat{z}_{t+1|t} = \mathbb{E}(z_{t+1} \mid (z_0, ...z_t)) = \mathbb{E}(Hy_{t+1} + u_{t+1} \mid (z_0, ...z_t))$

$\hat{z}_{t+1|t} = H\mathbb{E}(y_{t+1} \mid (z_0, ...z_t))$ car u_{t+1} est de moyenne nulle.

$\boxed{\hat{z}_{t+1|t} = H\hat{y}_{t+1|t}}$ $(***)$

Calculons maintenant sa matrice de covariance conditionnelle :

$\mathbb{E}((z_{t+1} - \hat{z}_{t+1|t})(z_{t+1} - \hat{z}_{t+1|t})^T \mid (z_0, ...z_t))$

$= \mathbb{E}((Hy_{t+1} + u_{t+1} - H\hat{y}_{t+1|t})(Hy_{t+1} + u_{t+1} - H\hat{y}_{t+1|t})^T \mid (z_0, ...z_t))$

$= H\mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^T \mid (z_0, ...z_t))H^T + \mathbb{E}(u_{t+1}u_{t+1}^T)$

$= \boxed{HC_{t+1|t}H^T + R}$ en utilisant les mêmes astuces de simplification avec u_t indépendant de y_t et de $(z_0, ..., z_t)$, de moyenne nulle et de matrice de covariance R.

Enfin, calculons la matrice conditionnelle de variance croisée afin de reproduire la situation de rappel ci-dessus et de pouvoir utiliser les formules de distributions conditionnelles:

$$\mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(z_{t+1} - \hat{z}_{t+1|t})^T \mid (z_0, ...z_t))$$

$$= \mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(Hy_{t+1} + u_{t+1} - \hat{z}_{t+1|t})^T \mid (z_0, ...z_t))$$

$$= \mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(Hy_{t+1} + u_{t+1} - H\hat{y}_{t+1|t})^T \mid (z_0, ...z_t)) \text{ en utilisant } (***)$$

$$= \mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})(H(y_{t+1} - \hat{y}_{t+1|t}) + u_{t+1})^T \mid (z_0, ...z_t))$$

$$= H\mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})((y_{t+1} - \hat{y}_{t+1|t})^T \mid (z_0, ...z_t)) + \mathbb{E}((y_{t+1} - \hat{y}_{t+1|t})u_{t+1}^T \mid (z_0, ...z_t))$$

$= \boxed{HC_{t+1|t}}$ en utilisant toujours u_t indépendant de y_t et de $(z_0, ..., z_t)$ ainsi que de moyenne nulle.

Similairement, $\mathbb{E}((z_{t+1} - \hat{z}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^T \mid (z_0, ...z_t)) = C_{t+1|t}H^T$.

Pour résumé, la distribution jointe de y_{t+1} et z_{t+1} conditionnellement à$(z_0, ..., z_t)$ a pour moyenne : $\mu = \begin{pmatrix} \hat{y}_{t+1|t} \\ H\hat{y}_{t+1|t} \end{pmatrix}$ et pour matrice de covariance: $\Sigma = \begin{pmatrix} C_{t+1|t} & C_{t+1|t}H^T \\ HC_{t+1|t} & HC_{t+1|t}H^T + R \end{pmatrix}$

Ceci nous permet de déduire la distribution conditionnelle de $(y_{t+1|t} \mid z_{t+1|t})$ où $y_{t+1|t}$ et $z_{t+1|t}$ sont eux-même conditionnés sur $(z_0, ..., z_t)$.
Ainsi, en remplaçant avec les rappels (*),(**) et les calculs précédemment, on est en mesure de retrouver simplement:

$$\boxed{\hat{y}_{t+1|t+1} = \hat{y}_{t+1|t} + C_{t+1|t}H^T(HC_{t+1|t}H^T + R)^{-1}(z_{t+1} - H\hat{y}_{t+1|t})}$$

$$\boxed{C_{t+1|t+1} = C_{t+1|t} - C_{t+1|t}H^T(HC_{t+1|t}H^T + R)^{-1}HC_{t+1|t}}$$

avec le gain de Kalman apparaissant dans les deux équations: $\boxed{K_{t+1} = C_{t+1|t}H^T(HC_{t+1|t}H^T + R)^{-1}}$

Algorithme de récursion du filtre de kalman:

Nous avons désormais toutes les équations nécessaires pour l'algorithme :

Initialisation : $\hat{y}_{0|-1} = 0 et C_{0|-1} = C_0$

Equations de récursions:

-$\hat{y}_{t+1|t} = F\hat{y}_{t|t}$

-$C_{t+1|t} = FC_{t|t}F^T + GQG^T$

-$\hat{y}_{t+1|t+1} = \hat{y}_{t+1|t} + C_{t+1|t}H^T(HC_{t+1|t}H^T + R)^{-1}(z_{t+1} - H\hat{y}_{t+1|t}$

-$C_{t+1|t+1} = C_{t+1|t} - C_{t+1|t}H^T(C_{t+1|t}H^T)^{-1}HC_{t+1|t}$

4.2 Filtres de kalman non linéaires

Dans cette partie, nous étudions 2 types de filtres.

4.2.1 FKE : Filtre de Kalman étendu

C'est la version non linéaire du filtre de kalman. Il suffit que les modèles soient des **fonctions différentiables** de l'état. On utilise la fonction F pour prédire un état à partir de l'état précédent.

Equations	Filtre de kalman linéaire	Filtre de kalman NON linéaire
Equation d'état	$y_{t+1} = Fy_t + q_t$	$y_{t+1} = F(y_t) + q_t$
Prédiction d'état par méthode numérique	$\hat{y}_{t+1} = Fy_t^*$	$\hat{y}_{t+1} = F(y_t^*)$
Prédiction de covariance d'erreur par méthode numérique	$\hat{C}_{t+1} = FC_t^*F^T + C_t$	$\hat{C}_{t+1} = F_t'C_t^*F_t'^T + C_t^m \ (***)$

Table 1: Comparaison des équations pour les filtres linéaires et non linéaires

avec:

-$y^* = \hat{y} + K(z - H\hat{y})$,

-$C* = (I - KH)\hat{C}$,

-$K = \hat{C}H^T(H\hat{C}H^T + C_0)^{-1}$,

-$F_t' = \frac{\partial F(y_t)}{\partial y_t}$

Calculons $(**)$ dans le cas scalaire pour ne pas allourdir les calculs et généralisons:

$\hat{C}_{t+1} = \mathbb{E}[(y_{t+1} - \hat{y}_{t+1})^2] = \mathbb{E}[(F(y_t) + q_t - F(y_t^*))]^2$.

Par Taylor: $F(y_t) = F(y_t^*) + F'(y_t^*)(y_t - y_t^*) + \frac{1}{2}F''(y_t^*)(y_t - y_t^*)^2 + ... + C_t^m$

$\Rightarrow F(y_t) - F(y_t^*) \simeq F'(y_t^*)(y_t - y_t^*) + \frac{1}{2}F''(y_t^*)(y_t - y_t^*)^2 + ... + C_t^m$

$\Rightarrow \hat{C}_{t+1} \simeq \mathbb{E}[(y_t - y_t^*)^2]F'(y_t^*)^2 + C_t^m$, on néglige les moments d'ordre ≥ 3

$\Rightarrow \hat{C}_{t+1} \simeq C_t^* F'(y_t^*)^2 + C_t^m$

Dans le cas plus générale: $\boxed{\hat{C}_{t+1} \simeq F_t' C_t^* F_t'^T + C_t^m}$

Ces équations calquées sur le filtre linéaire, ce modèle est simple à mettre en place et s'adapte bien aux systèmes non linéaires qui sont généralement les systèmes à traiter. Cependant, si les estimations initiales ne sont pas correctes, le modèle diverge rapidement et ce modèle requiert beaucoup de mémoire de stockage en grande dimension.

4.2.2 Filtre de kalman d'ensemble

Cette variante du filtre de kalman est adapté aux problèmes de grande dimension utilisés notamment en géophysique.Elle utilise les méthodes de Monte Carlo et le théorème de Bayes pour la mise à jour des paramètres(étape d'analyse). Le filtre de kalman d'ensemble ne maintient pas la matrice de covariance au cours du temps comme ce qui était fait dans le filtre de kalman classique. L'idée est d'utiliser un échantillon aléatoire appelé ensemble que l'on assume représentatif et de remplacer la matrice de covariance par la covariance de cet ensemble.

La matrice de covariance est ici remplacé par la covariance d'échantillon:

$\hat{C} = \mathbb{E}[(\hat{y} - \mathbb{E}(\hat{y}))(\hat{y} - \mathbb{E}(\hat{y}))^T]$

$C^* = \mathbb{E}[(y^* - \mathbb{E}(y^*))(y^* - \mathbb{E}(y^*))^T]$

$C_0 = \mathbb{E}(u_0 u_0^T)$ tels que $z_j = z + u_{0,j}$ où z_j est l'observation j parmi l'ensemble .

L'étape d'analyse ici consiste à mettre à jour les observations des membres de l'ensemble j, $\{j = 1, ...N\}$ où N est le nombre d'éléments de l'ensemble:

$y_j^* = \hat{y_j} + K(z_j - H\hat{y_j})$ avec $K = \hat{C}H^T(H\hat{C}H^T + C_0)^{-1}$

Le filtre de kalman d'ensemble est une généralisation aux filtres de kalman linéaire et étendu .

On peut retrouver l'expression de l'étape d'analyse du filtre de kalman linéaire: $\bar{y^*} = \bar{\hat{y}} + K(\bar{z} - H\bar{\hat{y}})$ (1)

L'analyse d'erreur devient comme suit: $y_j^* - \bar{y^*} = (I - KH)(\hat{y}_j - \bar{\hat{y}} + K(z_j - \bar{z})$ (2)

En utilisant (1) et (2), on retrouve la covariance d'erreur:

$\mathbb{E}[(y^* - \mathbb{E}(y^*))(y^* - \mathbb{E}(y^*))^T] = (I - KH)\hat{C}$

Cette solution est donc une bonne alternative aux méthodes de filtrage classique pour les problèmes en grande dimension.

5 Article scientifique: Modèle de Markov caché et génétique

5.1 Présentation et objectif

Cet article présente une approche d'encodage génétique basée sur les modèles de markov cachés. L'étude porte sur l'identification des gènes encodant la glycoprotéine à surface variable(VSG) qui est présente dans le génome des trypanosomas brucei, espèce de parasite présente en Afrique provoquant des maladies chez les humains et les animaux. Ces modèles ayant fait leur preuve en reconnaissance vocale , de part leur adaptabilité aux variabilités de longueur des séquences, il paraît donc normal d'utiliser ces modèles en génomique où l'un des principaux problèmes est la non adaptabilité des algorithmes aux propriétés de variabilité de longueurs requérant plutôt une taille fixée en input. Dans cet article, les auteurs compareront les performances selon plusieurs métriques pour des modèles de markov cachés à 2 et 3 états utilisant un dataset public: GenBank database.

Les modèles de markov cachés ont demontré leur efficacité dans les problèmes de classification de données de séries temporelles et de données spatiales séquentielles. Ils ont longtemps été utilisés en biologie pour les problèmes de prédiction de structure de gène et de création de modèles statistiques de famille de protéine appelé profilage.

L'objectif de cet article est de développer un outil de machine learning basé sur les modèles de markov cachés permettant d'identifier les zones homogènes de présence du VSG dans le génome des trypanosomas brucei. L'approche proposée est une approche hybride qui prend en compte à la fois la localisation du gêne et les informations cachés de la séquence d'ADN à travers un modèle de markov.

5.2 Application des modèles de markov dans le contexte biologique

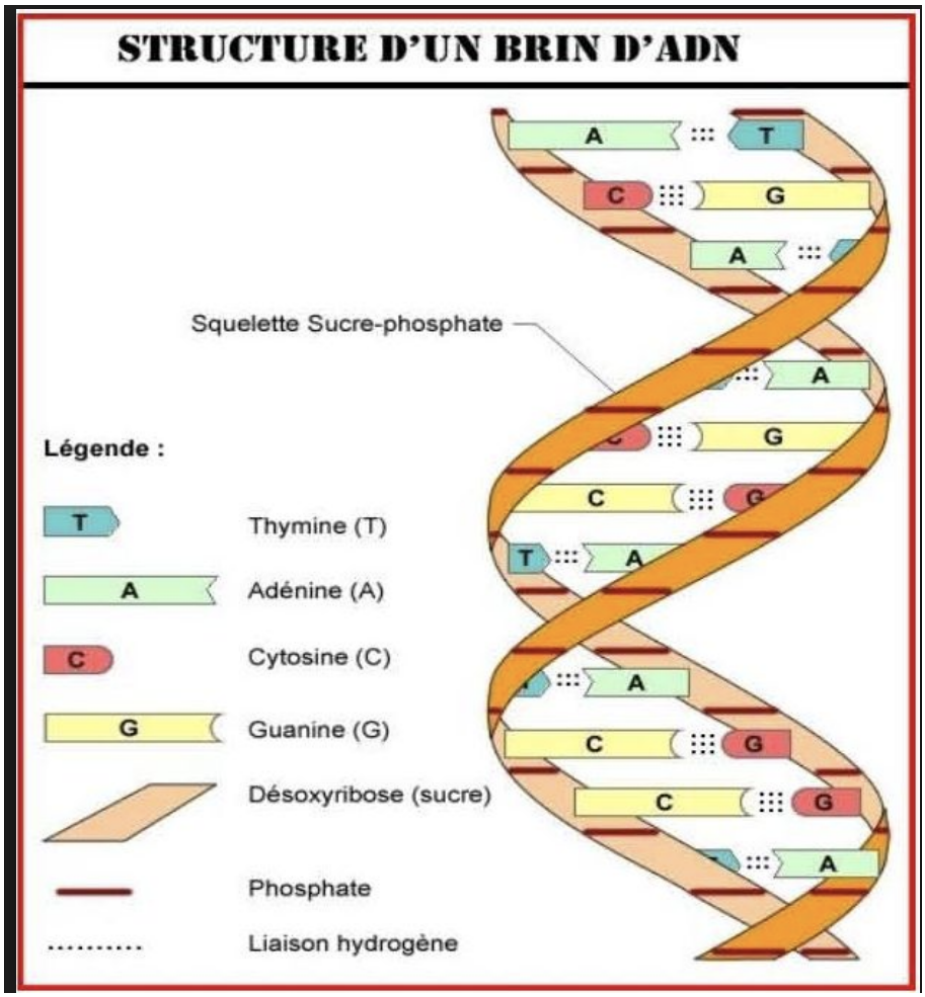

Figure 11: Structure simplifiée de l'ADN

L'ADN est une macromolécule biologique composée d'un assemblage de molécule organique simple: les nucléotides. Un nucléotide est un monomère formé d'une base nucléique A,C,G ou T lié au désoxyribose qui lui-même est lié à un groupe phosphate. Les molécules d'ADN sont des cellules vivantes formées de deux brins orientés parallèlement l'un à l'autre et enroulés pour former comme on le voit sur le schéma une double hélice.

Un modèle de markov caché est caractérisé par sa séquence d'observation

ici $(x_1, ..., x_n)$ avec $x_i \in \Lambda = \{A, C, G, T\} \forall i \in [1, n]$ et par ses états $\epsilon = \{VSG\ gene\ , no\ VSG\ gene\}$ ou $\epsilon = \{VSG\ gene\ , no\ VSG\ gene\ , other\ structures\}$. L'objectif est donc d'assigner à chaque observation $x_i \in \Lambda$ l'état $y_i \in \epsilon$ qui lui correspond.

5.3 Entraînement du modèle

L'entraînement du modèle est basé sur l'estimateur du maximum de vraisemblance dont l'objectif est de maximiser les paramètres pour une séquence d'observation générée par le modèle : $\hat{\theta} = \text{argmax}_\theta \, \mathbb{P}_\theta(x)$.

Pour cela, on utilise l'alogrithme de Baum-Welch décomposé en deux étapes récursives: forward puis backward pour obtenir les paramètres puis l'algorithme de Viterbi pour définir la séquence d'états optimal à une séquence d'observation donnée.

5.3.1 Algorithme forward-backward

Algorithm 1: Forward Algorithm.

// Initialization

$\alpha_k(1) = \mathbb{P}(x_1, y_1 = k) = \mathbb{P}(y_1 = k)\mathbb{P}(x_1 | y_1 = k)$, $\forall k = 1, \dots K$;

for *(i = 1 to n)* **do**

  $\alpha_k(i) = \mathbb{P}(x_i | y_i = k) \sum_j \alpha_j(i-1)\mathbb{P}(y_i = k | y_{i-1} = j)$;

$\mathbb{P}(x) = \sum_k \alpha_k(n)$;

Return $\mathbb{P}(x)$ and $\alpha_k(n)$ for all k;

La probabilité forward : $\alpha_k(n) = \mathbb{P}(x_1, ..., x_n, y_n = k)$ correspond à la probabilité d'être à l'état k à l'étape n avec une séquence d'observation jusqu'à l'étape n bien définie.

$\alpha_k(n) = \sum_j \mathbb{P}(x_1, ..., x_n, y_n = k, y_{n-1} = j)$ par la formule des probas totales.

$$\alpha_k(n) = \sum_j \mathbb{P}(x_n \mid x_1, ..., x_{n-1}, y_{n-1} = j, y_n = k) \\ * \mathbb{P}(y_n = k \mid y_{n-1} = j, x_1, ..., x_{n-1}) * \mathbb{P}(y_{n-1} = j, x_1, ..., x_{n-1})$$

Comme x_n ne dépend que de y_n et que y_n ne dépend que de y_{n-1}, on a :

$\mathbb{P}(x_n \mid x_1, ..., x_{n-1}, y_{n-1} = j, y_n = k) = \mathbb{P}(x_n \mid y_n = k)$ et

$\mathbb{P}(y_n = k \mid y_{n-1} = j, x_1, ..., x_{n-1}) = \mathbb{P}(y_n = k \mid y_{n-1} = j)$

$$\Rightarrow \boxed{\alpha_k(n) = \mathbb{P}(x_n \mid y_n = k) \sum_j \alpha_j(n-1)\mathbb{P}(y_n = k \mid y_{n-1} = j)}$$

Algorithm 2: Backward Algorithm.

// Initialization

$\beta_k(n) = 1 \ \forall k = 1, \dots K$;

for *(*$i = n - 1$ *to* 1*)* **do**

  $\beta_k(i) = \sum_l \mathbb{P}\big(x_{i+1} | y_{i+1} = l\big)\beta_l(i+1)\mathbb{P}\big(y_{i+1} = l | y_i = k\big)$;

$\mathbb{P}(x) = \sum_k \mathbb{P}(y_1 | y_0 = k)\mathbb{P}\big(x_1 | y_1 = k\big)\ \beta_k(1)$;

Return $\mathbb{P}(x)$ and $\beta_k(n)$ for all k;

Similairement, avec les mêmes propriétés d'indépendance ,la probabilité backward se calcule:

$$\boxed{\beta_k(n) = \mathbb{P}(x_{n+1}, ..., x_n \mid y_n = k) = \sum_l \mathbb{P}(x_{n+1} \mid y_n + 1 = l)\beta_l(n+1)\mathbb{P}(y_{n+1} = l \mid y_n = k)}$$

L'algorithme est comme suit:

1) on commence par calculer progressivement les probabilités d'obtenir les observations dans une séquence définie pour chaque état du modèle.

2) on calcule rétroprogressivement les probabilités d'obtention d'une séquence d'observation en amont d'une étape fixée avec un état fixé.

3) combiner ces deux étapes pour avoir la probabilité d'un état caché à un instant donné connaissant la séquence totale d'observation.

Connaissant $x^{(t)}$ la séquence d'observation, on obtient alors toutes les probabilités de transition de n'importe quel état k à n'importe quel état l à n'importe quel instant i:

$$\boxed{\mathbb{P}(y_{i+1} = l, y_i = k \mid x^{(t)}) = \frac{\alpha_k^{(t)}(i)\mathbb{P}(y_{i+1} = l \mid y_i = k)\mathbb{P}(x_{i+1}^{(t)} \mid y_{i+1} = l)\beta_l^{(t)}(i+1)}{\mathbb{P}(x^{(t)})}}$$

Nous pouvons maintenant définir de manière explicite les composantes de

$\theta(\hat{p}_{kl}, \hat{e}_k)_{k,l\in\epsilon}$ avec:

- $p_{kl} = \mathbb{P}(y_{i+1} = l \mid y_i = k)\ \forall k, l \in \epsilon$

- $P_{kl} = \sum_{t\in T}\sum_i \mathbb{P}(y_{i+1} = l, y_i = k \mid x^{(t)})$ où T correspond aux différentes séquences d'entraînement.

- $e_k(b) = \mathbb{P}(x_i = b \mid y_i = k)\ \forall b \in \Lambda$

-$E_k(b) = \sum_{t\in T} \frac{1}{\mathbb{P}(x^{(t)})} \sum_{\left\{i:x_i^{(t)}=b\right\}} \alpha_k^{(t)}(i)\beta_k^{(t)}(i)$

- $\hat{p}_{kl} = \frac{P_{kl}}{\sum_{j\in\epsilon} P_{kj}}$ avec P_{kl} le nombre total de transitions de l'état k à l'état l pour toutes les séquences d'entraînement et donc $\hat{p}_{kl}$ correspond au pourcentage total des séquences d'entraînement à l'état k qui vont passer à l'état l.

- $\hat{e}_k(b) = \frac{E_k(b)}{\sum_{j\in\Lambda} E_k(j)}$ avec $E_k(b)$ le nombre de fois total que l'état k a émis l'observation b durant toutes les séquences d'entraînement d'où $\hat{e}_k(b)$ représente le pourcentage total des séquences d'entraînement à l'état k émettant l'observation b .

Ayant démontré que la fonction de vraisemblance augmente à chaque itération de l'algorithme, il suffit de définir un seuil pour arrêter l'algorithme. Dans cet article le seuil a été choisi de sorte à ce que la distance euclidienne entre 2 itérations successives soit supérieure à $\lambda = 0.000001$ c'est à dire que : $\mathbb{P}_{\hat{\theta}^{(h+1)}}(x) - \mathbb{P}_{\hat{\theta}^{(h)}}(x) < \lambda$. L'algorithme s'arrête lorsque la performance entre deux itérations de l'algorithme pour mettre à jour les paramètres augmente de moins que λ.

5.3.2 Algorithme de Viterbi

Après avoir estilmé les paramètres du modèle à l'aide de l'algorithme forward-backward, il est utile d'utiliser l'algorithme de Viterbi afin de trouver la séquence d'état représentant au mieux notre séquence d'observation c'est à dire de trouver la séquence d'état ayant la plus grande probabilité de générer nos observations.Cette séquence d'état caché la plus probable est appelée le chemin de Viterbi. Il suffit donc de trouver:

$y* = \text{argmax}_y\, \mathbb{P}(y \mid x) = \text{argmax}_y\, \frac{\mathbb{P}(x,y)}{\mathbb{P}(x)}$.

Comme le dénominateur ne dépend pas de y, il suffit de trouver:

$$\boxed{y* = \underset{y}{\text{argmax}}\, \mathbb{P}(x, y)}.$$

Pour cela, l'algorithme qui a été développé est un algorithme qui calcule itérativement à chaque étape i, la séquence d'états cachés la plus probable $(y_1, ..., y_i)$ pour générer les observations $(x_1, ..., x_i)$ et ceux pour chaque état

$\{y_i = k\ ,\ k \in \epsilon\}$.

L'algorithme calcule à chaque étape i:

$V_{i+1}(l) = \max_k[V_i(k)\mathbb{P}(y_{i+1} = l \mid y_i = k)] * \mathbb{P}(x_{i+1} \mid y_{i+1} = l)$ où:

$V_i(l)$ représente la probabilité de la meilleure séquence d'états générant l'état l à l'étape i connaissant les observations $(x_1, ..., x_i)$

Il suffit alors à chaque étape i de stocker l'état qui maximise la probabilité V_i afin d'obtenir la séquence d'état la plus probable.

5.3.3 Méthodologie

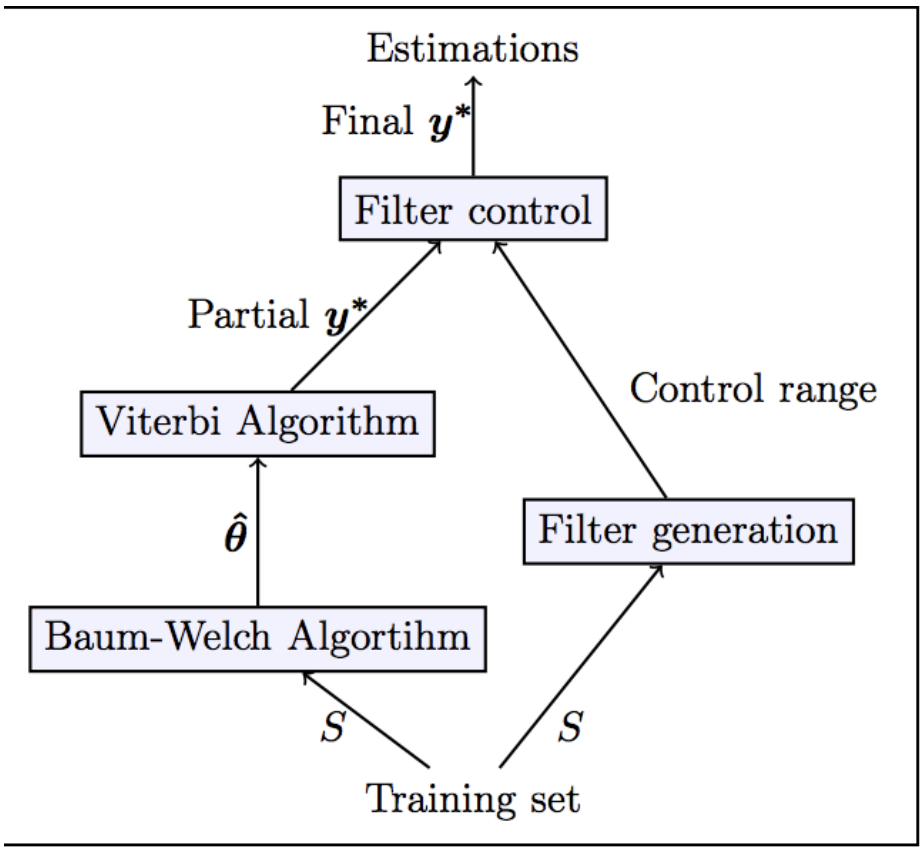

Figure 12: Procédure d'entraînement du modèle

La collection S de séquence du génôme est découpée en 9 séquences de tailles et de regroupement différents du gène VSG. Pour l'entraînement, la collection S est décomposée aléatoirement en une partie d'entraînement S^T et une partie de validation S^V afin de tester les performances du modèle sur un nouvel ensemble. L'algorithme forward-backward est appliqué à S^T afin d'avoir les paramètres $\hat{\theta}$ du modèle représentant les probabilités de transition et d'émission. Une fois que l'algorithme converge, on applique l'algorithme de Viterbi sur S^V avec les paramètres estimés précédemment afin d'obtenir la meilleure séquence d'états cachés. Avant d'évaluer le modèle, une petite étape manuel est ajouté pour améliorer les performances: les auteurs ont définis un intervalle de contrôle en utilisant un intervalle borné défini dans V^T dans lequel est présent le gène VSG. Il suffit donc de garder dans les résultats obtenus par l'algorithme, uniquement ceux qui sont à l'intérieur de l'intervalle. Enfin, on évalue le modèle en comparant les annotations avec les résultats obtenus par l'algorithme combiné au filtre de contrôle.

5.4 Résultats et conclusions

Les résultats du modèle à 2 états sont mitigés. En effet, la sensibilité est plutôt bonne (0.81 au pire des cas) tandis que la précision laisse à désirer d'où le taux élevé de faux positif ce qui peut-être quelque chose que l'on souhaite éviter notamment dans les applications médicales.

Pour remédier à ce problème, les auteurs ont introduit et évalué les performances d'un modèle à 3 états cachés au lieu de 2 ce qui a permis d'améliorer sensiblement les résultats.

Enfin , il est important de souligner le fait qu'une des hypothèses sur l'indépendance des différentes séquences doit être légitimement remise en question. En effet, en séquence génomique, nous ne pouvons affirmer l'indépendance entre les gènes et segment de gènes dans une séquence étant donné que les séquences génétiques ont la même origine. Ici, pour mesurer la corrélation entre ces différentes séquences, les auteurs ont utilisé le pourcentage de base identique pour assumer l'indépendance et donc établir un cadre mathématique propre. Cependant, les résultats numériques des corrélations n'étant pas donnés, je pense que ces hypothèses d'indépendance sont contestables...

Pour conclure, les auteurs de cet article ont utilisé un modèle de markov caché afin de résoudre un problème de classification génétique. Ils ont proposé et détaillé deux approches avec un nombre d'états cachés différents et ont introduit la notion de filtrage des résultats pour identifier de manière simple les gènes mal classifiés afin d'améliorer les performances du modèle.

References

-Hans Wackernagel. *Data Assimilation slides.* Mines ParisTech

-Bruno Bouzy. *Cours sur les réseaux bayésiens.*
http://www.math-info.univ-paris5.fr/ bouzy/Doc/AA1/ReseauxBayesiens.pdf
Paris-V

-F.Moutarde, S.Manitsaris. *Présentation sur les chaines de markov cachés.*
http://perso.mines-paristech.fr/fabien.moutarde/ES_MachineLearning/Slides/slides_Gestures-HMM_Sotiris.pdf
Mines ParisTech

-Jan Mandel. *Les filtres de kalman.*
https://arxiv.org/pdf/0901.3725.pdf
University of Colorado

-Gabriel A. Terejanu. *Les filtres de kalman étendus.*
https://www.cse.sc.edu/ terejanu/files/tutorialEKF.pdf
University at Buffalo

-A. Mesa, S. Basterrech, G. Guerberoff et F. Alvarez-Valin.
Hidden Markov Models for Gene Sequence Classification: Classifying the VSG genes in the Trypanosoma brucei Genome.
https://arxiv.org/pdf/1508.05367.pdf
Universidad de la República

-Gabriel A. Terejanu. *Algorithme forward backward.*
https://people.eecs.berkeley.edu/∼stephentu/writeups/hmm-baum-welch-derivation.pdf
University of California