Représentation et comparaison de séquences par visualisation

Christine Largeron (*), Cedric Dreissia (**)

Université Jean Monnet de Saint-Etienne
(*) EURISE
23, rue du docteur Paul Michelon
(**) CREUSET
6, rue Basse des Rives
42023 Saint-Etienne Cedex 2
Christine.Largeron@univ-st-etienne.fr

Résumé. Dans cet article, nous présentons un outil de visualisation de séquences modélisées par des arbres de suffixes probabilistes (Prediction suffix trees - PST). Ce type d'arbre permet de représenter une chaîne de Markov d'ordre variable. Dans différentes applications, il s'est avéré plus efficace qu'une chaîne de Markov d'ordre fixe, avec un coût calculatoire moindre. Pour ces raisons, il nous a paru intéressant d'exploiter le caractère arborescent de ce mode de représentation des séquences, non seulement d'un point de vue algorithmique, mais aussi d'un point de vue visuel. Le logiciel que nous avons développé dans ce but fournit une représentation graphique d'un PST appris à partir de séquences et, il permet de le comparer à un autre. Dans un contexte de classement supervisé d'une nouvelle séquence, il apporte une information complémentaire par rapport au PST en mettant en évidence les sous-séquences qui n'ont pas été observées dans la nouvelle séquence bien qu'elles soient caractéristiques du modèle sous-jacent à sa classe d'affectation. Ainsi, il permet de mieux appréhender la structure des séquences et d'améliorer le processus de fouille de données par leur visualisation.

1 Introduction

Les travaux précurseurs en fouille visuelle de données (Visual Data Mining) remontent à Bertin ou encore à Tufte [Bertin, 1977, Tufte, 1983]. Ils portaient sur la représentation graphique de données. Jusqu'à un passé proche, les techniques de visualisation étaient principalement employées dans deux étapes lors du processus de traitement de données :

- au début de la chaîne du traitement, dans une phase exploratoire des données brutes,
- à la fin du traitement, dans une phase de présentation des résultats sous une forme souvent plus synthétique.

Avec l'émergence de la fouille visuelle de données [Card *et al.*, 1999, Spence, 2001, Keim, 2002, Davidson et Soukup, 2002, Poulet, 2004], elles interviennent dans la phase principale du processus de fouille, afin d'impliquer plus directement l'utilisateur

dans l'extraction de connaissances à partir des données et dans l'élaboration des modèles.

C'est dans cette perspective, que nous avons développé un logiciel de représentation et de comparaison de séquences par visualisation. Par séquence, nous entendons une suite de valeurs observées dans le temps. Il peut s'agir par exemple en climatologie du temps observé quotidiennement dans une région pendant une période donnée, en bioinformatique de séquences d'ADN se présentant sous la forme d'une suite de nucléotides (a, c, g ou t), ou encore en robotique, d'une suite de déplacements dans l'espace. Pour modéliser de telles séquences, il existe de nombreux modèles. On peut distinguer d'une part, les modèles non probabilistes tels que les automates finis déterministes ou non déterministes ou encore différentes familles de grammaires et d'autre part, des modèles probabilistes tels que les automates stochastiques, les chaînes de Markov, les modèles de Markov cachés. Le choix d'une classe de modèles dépend notamment des caractéristiques des séquences. Si on suppose que le phénomène étudié présente une dépendance temporelle ; ce qui signifie que x_t, la valeur observée en t dépend des valeurs observées antérieurement ou du moins de certaines d'entre elles, on peut avoir recours à un modèle de Markov d'ordre variable. Par rapport à un modèle de Markov d'ordre fixe L, le modèle d'ordre variable exploite l'idée que dans certaines séquences naturelles la longueur de la mémoire dépend du contexte et n'est pas fixe. Cette hypothèse signifie qu'il suffit dans certains cas d'observer un seul état précédent pour établir une prédiction (par exemple l'état b sera très vraisemblable après a) alors que dans d'autres cas, il faut observer une séquence plus longue d'états (par exemple l'état d sera très probable après *cbbc* tandis que ce sera e après *cbbb*). Ce type de situation peut apparaître par exemple, dans des séquences biologiques où des parties codantes alternent avec des parties non codantes et où toutes les parties codantes n'ont pas nécessairement la même longueur. Il peut aussi se rencontrer lors de la détection de panne en robotique où un accident grave donnera lieu de façon quasi systématique à une intervention tandis qu'il faudra une succession d'incidents mineurs pour déclencher telle ou telle réparation.

Plusieurs algorithmes d'apprentissage de modèles de Markov d'ordre variable ont été proposés notamment par Willems [Willems *et al.*, 1995, Willems, 1998] et Seldin [Seldin *et al.*, 2001]. Celui développé par Ron [Ron *et al.*, 1996] utilise un arbre de suffixes probabilistes, appelé Prediction Suffix Tree (PST). Cet arbre est construit lors de la phase d'apprentissage supervisé et il permet de représenter le modèle sous-jacent aux séquences utilisées dans cette étape. Dans une seconde étape, il peut être employé pour classer de nouvelles séquences. Outre son pouvoir prédictif, ce modèle présente de nombreux avantages, notamment du point de vue de l'apprenabilité. Cependant, à notre connaissance, les possibilités offertes par le mode de représentation arborescent des séquences n'ont pas été exploitées dans une perspective de fouille visuelle de données. C'est la raison pour laquelle, nous avons conçu un outil de visualisation et de comparaison de séquences reposant sur ce modèle. Cet outil présente un double intérêt : il permet à l'utilisateur d'une part de mieux appréhender les données et d'autre part, d'interagir avec les modèles. Il sera décrit dans la troisième section ; la suivante étant

consacrée au modèle de Markov d'ordre variable et à sa représentation sous forme de PST.

2 Chaînes de Markov d'ordre variable et Arbres de suffixes probabilistes

2.1 Chaîne de Markov d'ordre variable

Les chaînes de Markov d'ordre variable (ou de longueur de mémoire variable - Variable Memory Markov Model) ont été introduites initialement par Rissanen [Rissanen, 1983] puis développées par Bühlmann et Wyner [Buhlmann et Wyner, 1999]. Par rapport à une chaîne de Markov d'ordre L à valeur dans un espace d'états fini Σ^L qui vise à exploiter les Σ^L historiques possibles, ce modèle considère uniquement ceux qui sont les plus vraisemblables. Ceci conduit en fait à conserver un historique de longueur maximale L dans certains contextes mais à le limiter lorsque la prise en compte d'un événement supplémentaire ne modifie pas significativement la distribution des probabilités conditionnelles.

Définition 1 *Une chaîne de Markov d'ordre variable L est un processus stationnaire et ergodique* $(X_t, t \in N)$ *à valeur dans un espace d'états* Σ^L *tel que :*

$$P(X_t = x_t / X_0 = x_0,...,X_{t-2} = x_{t-2},...,X_{t-1} = x_{t-1}) =$$

$$P(X_t = x_t / X_{t-c} = x_{t-c},...,X_{t-2} = x_{t-2},...,X_{t-1} = x_{t-1})$$

où

$$c = min\{1 \leq k \leq L | P(X_t = x_t / X_0 = x_0,...,X_{t-1} = x_{t-1}) = P(X_t = x_t / X_{t-k} = x_{t-k},...,X_{t-1} = x_{t-1})\}$$

Dans la suite, Σ sera appelé l'alphabet. De plus, on notera :

- Σ^*, l'ensemble de toutes les séquences que l'on peut définir sur Σ
- Σ^l, l'ensemble de toutes les séquences de Σ^* de longueur l, $\forall l \in N^*$
- $\Sigma^{\leq l}$, l'ensemble de toutes les séquences de Σ^* de longueur inférieure ou égale à l
- $s_i = (s_{it}, t = 1..l_i)$, une séquence de $\Sigma^{\leq l}$ générée par X (où l_i est la longueur de la séquence s_i : $l_i \leq l$, et $s_{it} \in \Sigma, \forall t = 1..l_i$
- le plus long suffixe de s_i différent de s_i est $(s_{i2},...,s_{il_i})$ et l'ensemble de tous les suffixes de s_i est $\{s_{it},...s_{l_i} | 1 \leq t \leq l_i\} \cup \{\varepsilon\}$ où ε désigne le mot vide.

2.2 Arbres de suffixes probabilistes

Un modèle de Markov d'ordre variable peut être représenté par un arbre de suffixes probabilistes (Prediction Suffix Tree - PST) défini dans Ron [Ron *et al.*, 1996] de la

façon suivante :

Définition 2 *Un PST S défini sur un alphabet fini* Σ *est un arbre* $|\Sigma|$*-aire de racine* ε *qui vérifie les propriétés suivantes :*

- *la racine ainsi que chaque noeud interne est l'extrémité initiale de* $|\Sigma|$ *arcs correspondant chacun à un symbole distinct et unique de* Σ*.*
- *chaque noeud de l'arbre, à l'exclusion de la racine* ε*, est étiqueté par un couple* (k,φ^k) *où*
  - *k est le mot correspondant au chemin parcouru depuis ce noeud jusqu'à la racine* ε *de l'arbre. Pour alléger les notations, dans la suite k désignera aussi ce noeud de l'arbre.*
  - φ^k *est un vecteur de dimension* $|\Sigma|$ *dont chaque composante* φ^{kj} *est égale à la probabilité conditionnelle d'observer le caractère j de* Σ *après le mot k*
- *la racine de l'arbre est étiquetée par* $(\varepsilon,\varphi^\varepsilon)$ *où* φ^ε *est un vecteur de dimension* Σ *dont chaque composante* $\varphi^{\varepsilon j}$ *est égale à la probabilité d'observer le caractère j de* Σ
- *r est le nombre maximum de noeuds internes de l'arbre.*

Par exemple, si on considère l'alphabet $\Sigma = \{a,b\}$ et la séquence $s = (aabaabaabaab)$ alors, le PST S associé à s pour une longueur de mémoire L égale à 2, est décrit dans la figure 1.

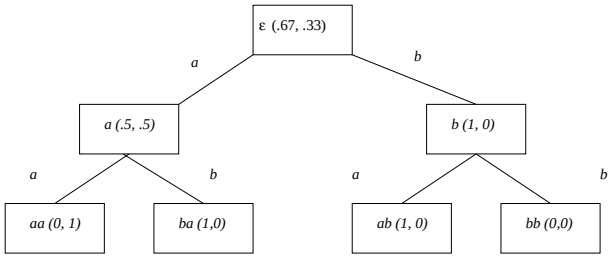

FIG. 1 – *PST S associé à la séquence* $s = (aabaabaabaab)$

2.2.1 Elagage et lissage des probabilités

Lorsque l'arbre est complet (ie tous ses noeuds internes ont exactement $|\Sigma|$ fils) et que le nombre de caractères du mot associé à une feuille de l'arbre est L, comme c'est le cas dans l'exemple précédent, le PST représente une chaîne de Markov d'ordre fixe L. Une des limites de ce modèle d'ordre fixe est la croissance exponentielle du nombre de paramètres à estimer en fonction de l'ordre L qui induit un coût de stockage élevé même pour une longueur de mémoire L limitée. En élaguant l'arbre ou, ce qui est équivalent, en n'insérant pas tous les noeuds, on obtient un arbre dont certaines branches sont de profondeur L et d'autres de profondeur inférieure ou égale à L et qui correspond à une chaîne de Markov d'ordre variable. Ainsi, l'élagage des branches

de l'arbre permet de réduire les coûts de stockage et, si besoin est, d'augmenter localement l'ordre du modèle de façon à améliorer du même coup son pouvoir prédictif dans un contexte de classement supervisé. Cette diminution du coût de stockage dépend bien évidemment des valeurs des paramètres du modèle. Elle s'est avérée significative, pour une capacité prédictive équivallente, dans les expériences que nous avons réalisées dans plusieurs applications [Largeron, 2001, Largeron-Leténo, 2003].

Cependant, il existe une autre difficulté liée à l'estimation des paramètres à partir de séquences d'apprentissage pour les chaînes de Markov d'ordre fixe comme d'ordre variable, et ce même si on dispose d'un échantillon d'apprentissage de taille élevée et pour un ordre L limité. Il s'agit du cas où certaines probabilités conditionnelles empiriques φ_i^{kj} prennent une valeur nulle car les sous-séquences [1] k correspondantes n'ont pas été observées dans l'échantillon d'apprentissage bien qu'elles puissent apparaître dans une nouvelle séquence générée par le modèle. Ces probabilités nulles, vont bien évidemment altérer les performances du modèle en phase de prédiction car elles conduiront à attribuer une probabilité nulle à la nouvelle séquence. Pour éviter cela, il est possible d'effectuer un lissage des probabilités. Plusieurs méthodes de lissage ont été proposées comme le décomptage [Henikoff et Henikoff, 1996], le lissage par deleted interpolation de Jelinek et Mercer [Jelinek et Mercer, 1980], la technique de back-off [Katz, 1987], le back-off amélioré par Kneser-Ney [Ney *et al.*, 1994]. Le principe général du lissage consiste à attribuer une probabilité conditionnelle empirique faible mais différente de zéro aux sous-séquences qui n'ont pas été observées lors de l'apprentissage tout en diminuant d'autant les probabilités associées aux sous-séquences qui ont été observées sur l'échantillon d'apprentissage de façon à conserver une somme des probabilités égale à 1.

2.2.2 Apprentissage de chaînes de Markov d'ordre variable

L'algorithme d'apprentissage de modèles de Markov d'ordre variable proposé par Ron, Singer et Tishby [Ron *et al.*, 1996] combine l'élagage de l'arbre et le lissage des probabilités. La construction d'un PST S à partir d'une séquence $s_i = (s_{it}, t = 1,..l_i)$ peut être réalisée de façon descendante (top-down) en n'insérant pas tous les noeuds dans l'arbre ou de façon ascendante (bottom-up) en supprimant certains noeuds dans l'arbre complet. Les deux schémas sont équivalents en ce sens qu'ils conduisent au même PST mais le premier est moins coûteux d'un point de vue calculatoire et est généralement employé. [2] Dans l'algorithme de Bejerano et Yona [Bejerano et Yona, 2001], directement inspiré de celui de Ron, le lissage fait intervenir un paramètre positif $ymin$ de sorte que chaque probabilité conditionnelle φ_i^{kj} est remplacée par f_i^{kj} :

$$f_i^{kj} = (1 - |\Sigma| \times y\min)\varphi_i^{kj} + ymin \quad (1)$$

1. On rappelle qu'une sous-séquence (ou sous-chaîne) est connexe alors qu'un sous-mot ne l'est pas nécessairement. Ainsi, 1123 est une sous-séquence de la séquence 211231 alors que 2231 est seulement un sous-mot

2. Bien que le terme d'élagage appliqué au premier schéma puisse paraître abusif, il est généralement employé même dans ce cas dans la littérature.

où :

- φ_i^{kj} est égale à la probabilité conditionnelle empirique d'observer le caractère j de Σ après la sous-séquence k de s_i :

$$\varphi_i^{kj} = \frac{n_i^{kj}}{n_i^{k*}} \tag{2}$$

avec :

- n_i^{kj} le nombre de fois où on a observé le caractère j après k dans s_i
- n_i^{k*} le nombre de fois où on a observé un caractère quelconque de Σ après k dans la séquence s_i

- $\varphi_i^{\varepsilon j}$ la probabilité empirique d'observer le caractère j de Σ dans la séquence s_i de longueur l_i :

$$\varphi_i^{\varepsilon j} = \frac{n_i^{\varepsilon j}}{l_i}$$

Quant à l'élagage, il consiste à insérer un noeud k dans l'arbre que si ce noeud correspond à une sous-séquence $(k_1 k_2 \ldots k_l)$ suffisamment fréquente dans la séquence s_i et si les probabilités conditionnelles d'apparition des caractères de l'alphabet (φ_i^{kj},$\forall j \in \Sigma$) après cette sous-séquence sont significativement différentes des probabilités conditionnelles observées pour le père du noeud k, correspondant à $(k_2 \ldots . k_l)$, le suffixe de k. Ce qui formellement, revient à vérifier les conditions suivantes :

(1) $\varphi_i^k \geq pmin$ où φ_i^k est la probabilité empirique d'observer la sous-séquence $k = (k_1 k_2 \ldots k_l)$ dans s_i:

$$\varphi_i^k = \frac{n_i^k}{l_i - l} \tag{3}$$

où n_i^k désigne le nombre de fois où on a observé la sous-séquence k de longueur l dans la séquence s_i de longueur l_i
et pour au moins un caractère j de Σ :

(2) $\varphi_i^{kj} \geq (1 + a)ymin$
(3) $\varphi_i^{kj} \geq r\, \varphi_i^{suf(k)j}$ ou $\varphi_i^{kj} \leq 1/r\, \varphi^{suf(k)j}$

Ainsi, en reprenant l'exemple précédent et en utilisant comme paramètre de lissage $ymin$ = 0,001 et comme paramètres d'élagage $pmin$ = 0,001, r = 1,05 et a = 0, on obtient avec Σ = {a,b} et L = 2 pour la séquence $s = (aabaabaabaab)$, le PST S' de la figure 2.

Le modèle de Markov d'ordre variable offre donc de nombreux avantages :

- il permet de capturer des dépendances à long terme présentes dans la séquence en adaptant la taille de la mémoire en fonction du contexte,
- sa topologie et sa complexité sont déterminées par les données,
- il est PAC apprenable [Kearns *et al.*, 1994, Vapnik, 1995, Vapnik, 1998] en un temps polynomial en L à l'aide de l'algorithme précédent [Ron *et al.*, 1996] alors qu'une chaîne de Markov d'ordre fixe L est apprenable en un temps exponentiel en L,

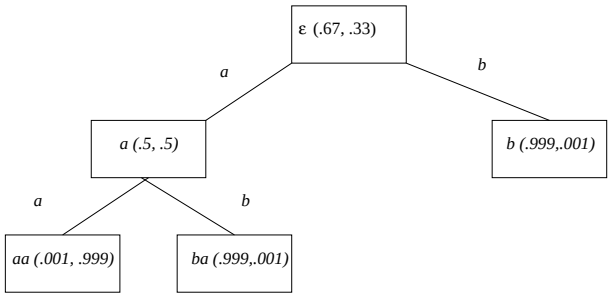

FIG. 2 – *PST S' associé à la séquence $s = (aabaabaabaab)$*

– enfin, dans un contexte de classement supervisé, il permet très facilement de classer une nouvelle séquence.

2.2.3 Prédiction à partir de PST

En effet, dans la phase d'apprentissage, un PST représentatif de chaque classe est construit à partir de séquences d'un échantillon d'apprentissage pour lesquelles on connaît la classe d'appartenance. Puis, dans la phase de classement, chacun de ces PST est employé pour calculer la probabilité que la nouvelle séquence ait été générée par ce modèle.

Ainsi, étant donnée une séquence $s_i = (s_{i1},...s_{il_i})$ de Σ^{l_i} et S un PST défini sur Σ, la probabilité que s_i ait été générée par S est définie par :

$$P^S(s_i) = \prod_{t=1}^{l_i} \varphi_S^{sufmax(s_{i1},...s_{it-1})s_{it}} \tag{4}$$

où

– $sufmax(s_{i1},...s_{it-1})$ désigne le plus long suffixe de $(s_{i1},...s_{it-1})$ figurant dans l'arbre S,

– $sufmax(s_{i1},...s_{it-1})s_{it}$ est la concaténation de ce suffixe avec le caractère s_{it} de s_i,

– $s_{i0} = \varepsilon$.

Ainsi, la probabilité attribuée à la séquence $s' = (abaab)$ avec le PST S décrit dans la figure 1 est :

$$\begin{aligned} P^S(s') &= P^S(a)P^S(b/a)P^S(a/ab)P^S(a/aba)P^S(b/abaa) \\ &= \varphi^{\varepsilon a}\varphi^{ab}\varphi^{aba}\varphi^{baa}\varphi^{aab} \\ &= 0.67 \times 0.5 \times 1 \times 1 \times 1 \\ &= 0.33 \end{aligned}$$

Notons que $P(b/abaa)$ est estimée par φ^{aab} car aa est le plus long suffixe de $abaa$ dans ce PST.

Enfin, la séquence est affectée à la classe du PST correspondant à la plus forte probabilité.

La combinaison de ces améliorations (lissage de probabilité et élagage de l'arbre) a permis d'obtenir des résultats intéressants dans plusieurs domaine d'application, comme en bioinformatique pour la catégorisation de familles de protéines [Apostolico et Bejerano, 2000, Bejerano et Yona, 2001] ou en chronobiologie [Largeron-Leténo, 2003].

3 Représentation et comparaison visuelle de séquences

Compte tenu des nombreux avantages et des performances de ce modèle, il nous a paru intéressant d'exploiter dans une perspective de fouille visuelle de données son mode de représentation arborescent. L'intérêt d'une telle approche est d'une part d'améliorer la perception des données pour l'utilisateur et d'autre part de lui permettre d'interagir avec le modèle. Ceci nous a conduit à développer un logiciel graphique de visualisation et de prévision de séquences. Cet outil présente les fonctionnalités suivantes :

- représentation d'une séquence sous forme de PST,
- représentation de plusieurs séquences sous forme de PST avec mise en évidence des noeuds communs,
- représentation des noeuds d'un PST (ie des sous-séquences de la chaîne de Markov) intervenant dans le classement d'une nouvelle séquence et affichage de la probabilité attribuée à la séquence par le PST.

Ces différentes fonctionnalités sont décrites plus précisément dans ce paragraphe et illustrées à partir d'un exemple simple de prévision climatique.

Dans cet exemple, on se propose à partir du temps observé quotidiennement dans une région de caractériser son climat. On distingue quatre types de climat possibles : océanique, continental, méditerranéen et montagnard. Pour identifier le climat d'une zone géographique, on relève chaque jour le temps qu'il fait dans cette zone pendant une période de temps suffisamment longue. Chaque jour, le temps peut être décrit suivant quatre modalités : beau temps, pluie, vent, nuageux notées respectivement b, p, v et n. On suppose que non seulement la fréquence d'apparition de chacune de ces modalités mais aussi leur succession permettent de différencier les différents climats. On peut donc employer un modèle de Markov d'ordre variable pour modéliser chaque climat. Pour construire ce PST pour chaque type de climat, on utilise en phase d'apprentissage supervisé, une séquence d'observations effectuées dans une zone déjà identifiée par un expert comme relevant de ce climat. A partir des modèles représentatifs des différents climats, on peut ensuite, en phase de classement, prédire le climat d'une nouvelle zone en disposant uniquement de la séquence des relevés de temps effectués quotidiennement.

3.1 Représentation d'une séquence par PST

La première fonctionnalité offerte par le logiciel est la représentation visuelle sous forme d'arbre du modèle de markov d'ordre variable sous-jacent à une séquence. Dans un contexte de fouille de données, cette première fonctionnalité paraît d'autant plus importante que le modèle appris en phase d'apprentissage se présente sous un formalisme logique qui ne fait pas clairement apparaître la dimension structurelle des données. Si ce formalisme répond bien aux contraintes d'implémentation et permet ensuite lors de la phase de classement d'effectuer une prévision en limitant les coûts de traitement, en revanche, il s'avère difficilement compréhensible. Ainsi, dans l'exemple climatique cité précédemment, on obtient un PST S_m à partir d'une séquence s_m de temps relevés pendant 86 jours dans une région caractérisée comme ayant un climat montagneux par un expert et définie par :
$s_m = (bnpvbvbvbpvnbvbvbvbvbnvbvbvnbvbvbpbvbvbvbvbvbvbvnbvbvbvbvbvbvbvbvbvnbvbvbvbvbnvbvbbv)$.
Le fichier décrivant le PST S_m appris[3] à partir de la séquence s_m est le suivant[4] :

!0.001000 4 bnpv 0.452674418605 0.082069767442 0.035744186047 0.429511627907

b [0.026538 0.077615 0.052077 0.843769] (
bb [0.001000 0.001000 0.001000 0.997000] ( )
nb [0.001000 0.001000 0.001000 0.997000] ( )
pb [0.001000 0.001000 0.001000 0.997000] ( )
vb [0.032125 0.063250 0.063250 0.841375] (
bvb [0.035345 0.069690 0.069690 0.825276] (
nbvb [0.001000 0.001000 0.001000 0.997000] ( )
pbvb [0.001000 0.001000 0.001000 0.997000] ( )
vbvb [0.042500 0.084000 0.084000 0.789500] (
bvbvb [0.001000 0.095857 0.095857 0.807286] ( )
nvbvb [0.499000 0.001000 0.001000 0.499000] ( )
pvbvb [0.001000 0.001000 0.001000 0.997000] ( )))
nvb [0.001000 0.001000 0.001000 0.997000] ( )
pvb [0.001000 0.001000 0.001000 0.997000] ( )))
n [0.570143 0.001000 0.143286 0.285571] (
bn [0.001000 0.001000 0.333000 0.665000] (
vbn [0.001000 0.001000 0.001000 0.997000] ( ))
vn [0.997000 0.001000 0.001000 0.001000] ( ))
p [0.333000 0.001000 0.001000 0.665000] (
bp [0.499000 0.001000 0.001000 0.499000] ( )
np [0.001000 0.001000 0.001000 0.997000] ( ))
v [0.886333 0.111667 0.001000 0.001000] (
bv [0.903625 0.094375 0.001000 0.001000] (
nbv [0.997000 0.001000 0.001000 0.001000] ( )
pbv [0.997000 0.001000 0.001000 0.001000] ( )

3. Le PST S_m a été construit à l'aide de l'algorithme de Bejerano et Yona [Bejerano et Yona, 2001]
4. Fichier : montagnePST.TXT

vbv [0.886333 0.111667 0.001000 0.001000] (
bvbv [0.872500 0.125500 0.001000 0.001000] (
nbvbv [0.997000 0.001000 0.001000 0.001000] ( )
pbvbv [0.997000 0.001000 0.001000 0.001000] ( )
vbvbv [0.839737 0.158263 0.001000 0.001000] ( ))
nvbv [0.997000 0.001000 0.001000 0.001000] ( )
pvbv [0.997000 0.001000 0.001000 0.001000] ( )))
nv [0.997000 0.001000 0.001000 0.001000] ( )
pv [0.499000 0.499000 0.001000 0.001000] (
bpv [0.001000 0.997000 0.001000 0.001000] ( )
npv [0.997000 0.001000 0.001000 0.001000] ( )))

On peut noter la réduction du coût de stockage puisque le PST comporte 36 noeuds alors que sans élagage il en aurait 340. On peut aussi remarquer que si ce fichier permet d'établir efficacement des prédictions dans un contexte de classement supervisé, il ne permet pas, même à un utilisateur averti, d'appréhender la séquence sous-jacente et a fortiori de la comparer à une autre décrite suivant le même formalisme.

Le but de l'application que nous avons développée est précisément de construire, à partir du fichier, la représentation graphique arborescente correspondante. Ainsi, dans l'exemple, on obtient le graphique de la figure 3 plus facilement compréhensible par l'utilisateur et exploitable visuellement grâce à différentes capacités offertes par le logiciel telles que :

- le zoom vers des points d'intérêt,
- l'élagage des éléments les moins intéressants,
- la personnalisation de la représentation graphique selon les préférences de l'utilisateur : affichage horizontal ou vertical, représentation arborescente sous une forme classique ou sous la forme d'un explorateur plus adapté aux PST de grandes tailles, repositionnement manuel des noeuds dans la fenêtre de visualisation.

De plus, l'utilisateur a accès au vecteur de probabilités associé aux feuilles de l'arbre en cliquant dessus (cf. Figure 4). Ces graphiques révèlent la prédominance des modalités beau temps (b) et vent (v) pour le climat montagneux.

Lorsque le PST comporte un grand nombre de branches et de noeuds, il n'est pas très facile, même en jouant sur les différents modes d'affichage, d'appréhender globalement la structure de la séquence sous-jacente. Dans ce cas, l'outil doit plutôt être employé pour analyser certaines sous-séquences correspondant à des parties de l'arbre accessibles grâce au zoom.

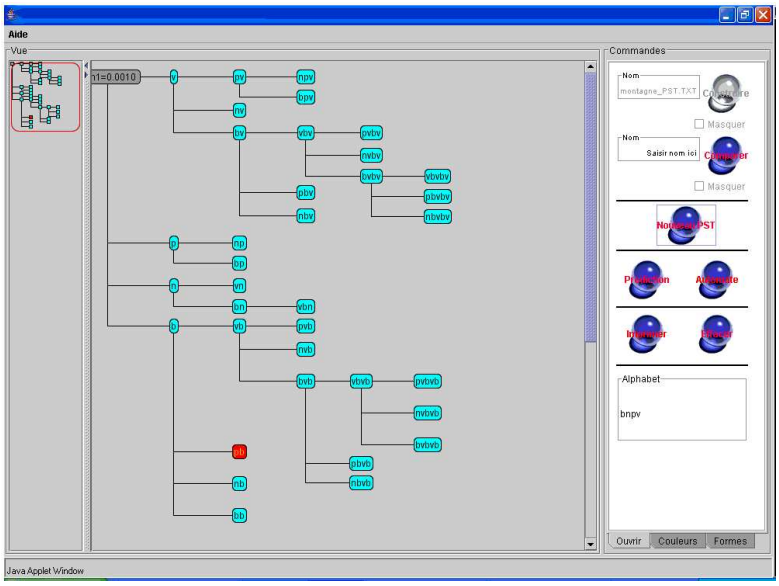

FIG. 3 – *Visualisation graphique arborescente du PST S_m*

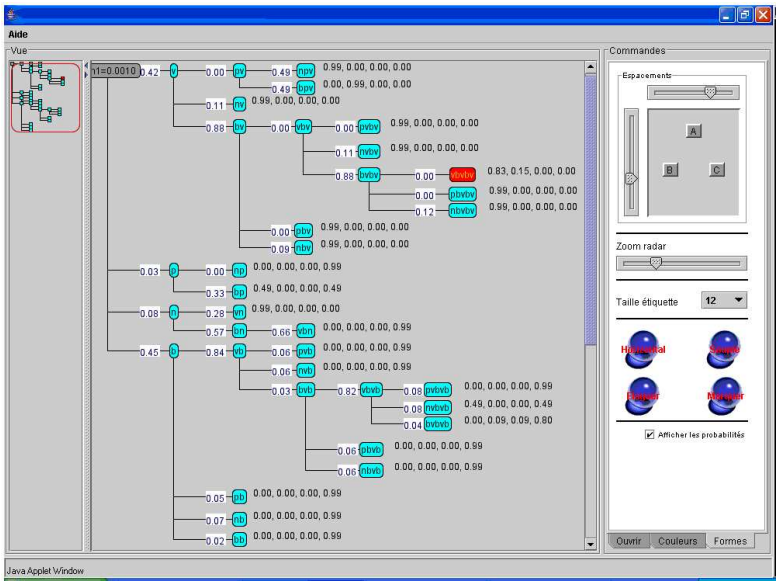

FIG. 4 – *Visualisation graphique arboresente du PST S_m avec probabilités*

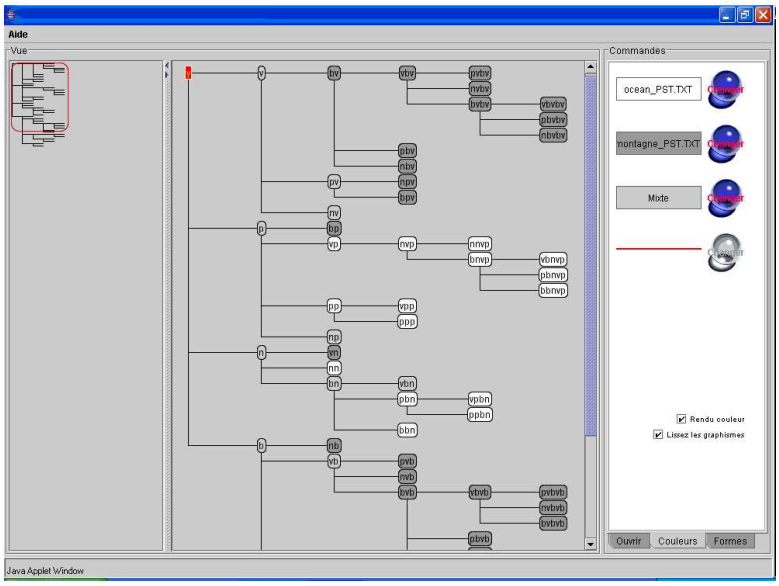

FIG. 5 – *Visualisation graphique simultanée de S_m et de S_o*

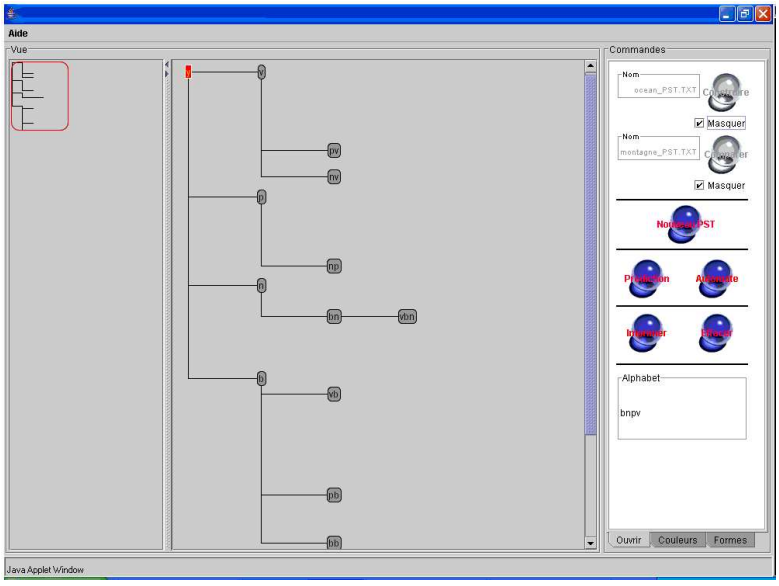

FIG. 6 – *Visualisation des noeuds communs à S_m et S_o*

3.2 Représentation simultanée de plusieurs séquences

Pour permettre une analyse comparative de deux modèles, l'application fournit simultanément leur représentation graphique arborescente. Des couleurs différentes indiquent les noeuds qui appartiennent au premier modèle, au second et ceux qui sont communs aux deux. L'utilisateur peut ensuite personnaliser les couleurs de représentation ou masquer un de ces trois types de noeuds.

Ainsi, dans l'exemple climatique, la figure 5 fait apparaître en plus du modèle S_m précédent, le modèle S_o [5] caractérisant le climat océanique et défini [6] à partir de la séquence s_o suivante : s_o = (*bnpvbnvpbnnvpbbbnvpbnvpbnvpppbbbnvpbnvpbnvpbnvpbn vpbbbnvpbnvpbnvpbnvpbbbnvpbnvppbnvpppbnvpbnvp*). Les modalités nuageux (n) et pluie (p) apparaissent davantage dans ce graphique que dans le précédent car elles sont plus fréquentes dans le climat océanique que dans le climat montagneux.

Dans la figure 6 l'utilisateur n'a conservé que les noeuds communs dans les deux modèles, mettant ainsi en évidence le nombre peu élevé de sous séquences communes aux deux climats et par conséquent leur différence.

3.3 Classement d'une nouvelle séquence

La dernière fonctionnalité offerte par l'interface graphique est le classement d'une nouvelle séquence. En effet, comme indiqué précédemment, un PST S est un modèle de classement supervisé. Il permet de prédire le modèle qui a le plus probablement généré une nouvelle séquence s. Après avoir visualisé le modèle, le logiciel permet la saisie de cette nouvelle séquence puis l'affichage sur le modèle des noeuds intervenant dans le calcul de la probabilité et enfin du résultat final.
Ainsi, en reprenant l'exemple climatique, la figure 7 illustre le déroulement de la phase de classement où les noeuds intervenant dans le calcul de la probabilité sont successivement mis en évidence. Dans cet exemple, on suppose que l'on a relevé une séquence s' = (*bnvbvbvbpvnbvnbvvbvbnvnvbvnvbv*) dans une région dont on souhaite identifier le climat. Alors, la valeur attribuée à s' par S_m et S_o est respectivement égale à -22,6 et -61,36 [7]. On en déduit que la séquence correspond plutôt à un climat montagnard qu'océanique.

5. Fichier oceanPST.TXT
6. Le PST S_o a été construit à l'aide du même algorithme (Bejerano et al., 2001)
7. Pour des raisons algorithmiques, il est d'usage d'effectuer une transformation logarithmique

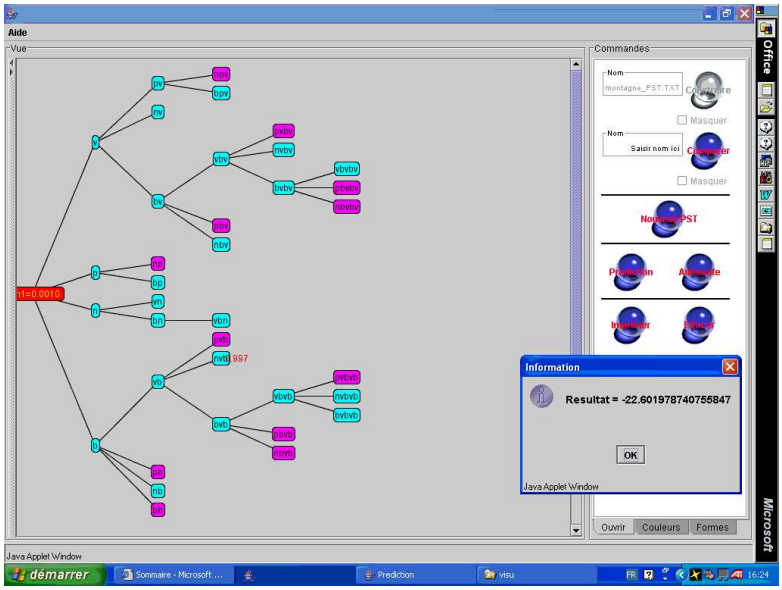

FIG. 7 – *Prévision d'une séquence à partir de* S_m

4 Conclusion

Les arbres de suffixes probabilistes (Probabilistic Sufix Trees - PST) permettent de représenter des modèles de Markov d'ordre variable. En ce sens, ils fournissent une généralisation des modèles de Markov d'ordre fixe, capables grâce à leur taille de mémoire variable, de capturer des dépendances à long terme présentes dans des séquences. Outre cet intérêt théorique et algorithmique, ils se sont avérés aussi particulièrement efficaces dans différentes applications. Compte tenu de leurs nombreux avantages, il nous a paru intéressant d'exploiter, autrement que d'un point de vue purement algorithmique, la représentation arborescente des séquences qu'ils offrent. Ceci nous a amené à développer un outil graphique de visualisation et de comparaison de séquences. Cet outil permet de représenter une séquence sous forme de PST, puis de la comparer à une autre en distinguant les noeuds communs aux modèles appris à partir des deux séquences et ceux qui appartiennent à chacun d'eux. Cet outil de visualisation peut aussi être employé pour classer une nouvelle séquence. Dans ce contexte de classement supervisé, il apporte une information complémentaire par rapport au modèle de Markov d'ordre variable en mettant en évidence les sous-séquences qui n'ont pas été observées dans la nouvelle séquence bien qu'elles soient caractéristiques du modèle. Ainsi, cet outil permet de mieux appréhender la structure des séquences et d'améliorer le processus de fouille de données par leur visualisation. Il est disponible en ligne via internet à l'adresse.

`http://eurise.univ-st-etienne.fr/~largeron/RNTI_visualisation/index.htm`

Les fichiers correspondants aux figures sont aussi disponibles dans ce répertoire où figurei.GIF et figurei.ps correspondent à la ième figure. Ce répertoire contient également les fichiers montagnePST.TXT et oceanPST.TXT permettant de tester le logiciel.

Références

[Apostolico et Bejerano, 2000] A. Apostolico et G. Bejerano. Optimal amnesic probabilistic automata or how to learn and classify proteins in linear time and space. *Journal Comput. Biol.*, 7(3):381–393, 2000.

[Bejerano et Yona, 2001] G. Bejerano et G. Yona. Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, 17(1):23–41, 2001.

[Bertin, 1977] J. Bertin. *La graphique et le traitement de l'information.* Flammarion, Paris, 1977.

[Buhlmann et Wyner, 1999] P. Buhlmann et A. Wyner. Variable length markov chains. *The annals of Statistics*, 27(2):480–513, 1999.

[Card *et al.*, 1999] K. Card, J.D. Mackinlay, et B. Schneiderman. *Readings in information visualization: using vision to think.* Morgan Kaufmann, 1999.

[Davidson et Soukup, 2002] I. Davidson et T. Soukup. *Visual data mining.* Wiley, 2002.

[Henikoff et Henikoff, 1996] J.G. Henikoff et S. Henikoff. Using substitution probabilities to improve position-specific scoring matrices. *Com. App. Biosci*, 12(2):135–143, 1996.

[Jelinek et Mercer, 1980] F. Jelinek et R.L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *E. Gelsema E. Kanal L.N Ed. Pattern Recognition in practice*, pages 381–397, Amsterdam Hollande, 1980.

[Katz, 1987] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[Kearns *et al.*, 1994] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, et L. Sellie. On the learnability of discrete distributions. *26th Annual ACM Symposium on Theory of Computing*, pages 273–282, 1994.

[Keim, 2002] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. on visualizations and computer graphics*, 7(1):100–107, 2002.

[Largeron-Leténo, 2003] C. Largeron-Leténo. Prediction suffix tree for supervised classification of sequences. *Pattern recognition letters*, 24:3153–3164, 2003.

[Largeron, 2001] C. Largeron. Algorithme de comparaison d'arbres : application au classement de séquences. In *7eme conférence de la Société Francophone de Classification*, Pointe-à-Pitre, France, 2001.

[Ney *et al.*, 1994] H. Ney, U. Essen, et R. Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer speech and language*, 8:1–38, 1994.

[Poulet, 2004] F. Poulet. Towards visual data mining. In *Proceedings of the 6th International Conference on Enterprise Information Systems ICEIS*, pages 349–356, Portugal Porto, 2004.

[Rissanen, 1983] J. Rissanen. A universal data compression system. *IEEE Trans Infor Theory*, 29(5):656–664, 1983.

[Ron *et al.*, 1996] D. Ron, Y. Singer, et N. Tishby. The power of amnesia : learning probabilistic automata with variable memory length. *Machine learning*, 25:117–149, 1996.

[Seldin *et al.*, 2001] Y. Seldin, G. Bejerano, et N Tishby. Unsupervised sequence segmentation by mixture of switching variable memory sources. In *Proceedings of the Eighteenth International Conference of Machine Learning*, pages 513–520. ICML, 2001.

[Spence, 2001] B. Spence. *Information visualization.* Addison-Wesley, ACM Press, 2001.

[Tufte, 1983] E.R. Tufte. *The visual display of quantitative information.* Graphics Press, reprint edition 1992 1983.

[Vapnik, 1995] V.N. Vapnik. *The nature of statistical learning theory.* Springer, 1995.

[Vapnik, 1998] V.N. Vapnik. *Statistical learning theory.* John Wiley, 1998.

[Willems *et al.*, 1995] F.M.J. Willems, Y.M. Shtarkov, et T.J. Tjalkens. The context tree weighting method: basic properties. *IEEE Trans. Infor. Theory*, 41(3):653–664, 1995.

[Willems, 1998] F.M.J. Willems. The context tree weighting method: extensions. *IEEE Trans. Infor. Theory*, pages 792–798, 1998.

Summary

This paper presents a visual tool for sequences analysis. Sequences are represented by prediction suffix trees (PST). PST can be used to efficiently describe variable order chain. It performs better than the Markov chain of order L and at a lower cost. For this reason, it is interesting to exploit the tree representation not only from a computational point of view but also from a visual one. The tool, developped in this aim, provides a graphic representation of a PST constructed from sequences. It can also be used to compare two models. In supervised classification context, it brings more information than the model by underlying sub-sequences observed in the new sequence and which are not in the model of its predicted class. By the way of visualisation, this tool upgrades the data mining process and the comprehension of the information encoded in sequences.