

Planification markovienne du choix d'agents de classification pour la poursuite de classes évolutives

Markov planification of classification tools for dynamic classes tracking

Pierre Loonis¹

Hervé Locteau²

¹L3I

Laboratoire Informatique, Image, Interaction
Avenue Michel Crépeau - 17024 La Rochelle Cedex 1
pierre.loonis@univ-lr.fr

²PSI

Laboratoire Perception, Système, Information
FRE CNRS 2645
Place Emile Blondel - 76821 Mont-Saint-Aignan Cedex
Herve.Locteau@univ-rouen.fr

Résumé

Lorsque les observations ne sont pas statiques, la plupart des méthodes numériques montrent leurs limites. L'évolution des données, qu'elle soit lente, cyclique, brusque, ... peut alors être traitée en construisant des systèmes de reconnaissance hiérarchisés : plutôt que de modéliser numériquement le bas niveau, on conceptualise pour traiter à une granularité plus large, en symbolique. Notre approche consiste à déployer des outils de classification adaptés sur des paysages de données où leurs performances sont optimales. Le passage d'un outil à un autre est effectué de manière probabiliste, par l'élaboration d'un modèle de Markov caché. L'apprentissage des matrices de transitions et d'observations s'effectue par l'algorithme EM. Cette étude montre que la planification des outils de classification est robuste et permet d'ordonner des outils, sans que le temps ne soit explicité.

Mots Clef

classification, coopération, planification, graphe markovien

Abstract

Most of numerical methods are not allowed to class dynamic data. An other way to track the data is due to hierarchical system : the generalization permits to explain the evolution with symbolic meanings, more than to approximate it numerically. Our approach deals with the spread of classification tools onto partial data landscape in which the reliability is optimised. A Hidden Markov Model is used to establish the transition and observation matrices. This

study shows that this kind of planification in an upper level is robust et allows to model choices, without defining an explicit notion of time index.

Keywords

hybrid classification tools, planification, HMM.

1 Introduction

En toute généralité, l'apprentissage de modèles à partir des données tend, étape par étape, à conceptualiser un ensemble de mesures en entités représentatives de catégories homogènes et mutuellement exclusives, selon un critère de performance fixé *a priori*.

En particulier, dans le domaine de la Reconnaissance des Formes (RdF), cette chaîne de traitements peut être réalisée, soit en une seule étape (par exemple avec l'apprentissage d'un classifieur paramétrique) [20], soit en plusieurs niveaux hiérarchiques (par exemple par l'association de classifieurs de types différents) [21].

Lorsque les données évoluent en fonction du temps, il est rare que les modèles eux mêmes ne doivent pas évoluer aussi. Si les Formes soumises aux lois d'évolution restent, sur tout l'horizon, localisées dans les limites des frontières de décisions construites, alors le modèle est suffisamment générique (il prend en compte l'évolution et la répartition, soit parce que l'évolution est faible par rapport à la répartition, soit parce que le modèle est riche).

Les problèmes se posent lorsque les lois d'évolution modifient sensiblement le paysage des données, ce qui peut s'exprimer par des changements des lois de répartitions (apparition de multi modalité, variation de topologie, trajectoires

différentes, ...) : le modèle construit à $t = 0$ n'est donc plus valide à $t = \tau$, car il n'est plus représentatif des données. Lorsque l'on connaît la loi d'évolution, il est possible d'effectuer des remises à jour des paramètres du modèle des données (par exemple dans le cas des méthodes fondées sur le filtre de Kalman) [4]; *a contrario*, si aucune connaissance explicite n'est disponible, on utilisera les approches dites "de révision des connaissances", qu'elles s'expriment dans le cadre bayésien classique, en estimation stochastique ou encore fondées sur des approches plus proches de l'Intelligence Artificielle (théorie des croyances, graphe bayésien, ...). A noter que, dans le cas particulier où les changements sont brusques, comme lorsque l'on passe de manière discontinue d'un mode de fonctionnement nominal à un mode de panne, les méthodes fondées sur les tests d'hypothèses [3] permettent de détecter des ruptures de modèles selon des critères de modification de la moyenne et/ou de la variance des observations.

Afin de résoudre les problèmes complexes de RdF, la conception d'un système sous la forme d'une collection d'outils coopérants a initié très tôt des recherches actives, dans divers domaines.

Nous nous intéressons plus particulièrement à la combinaison d'outils de classification [21, 16]. Le problème de l'association de plusieurs classifieurs peut être abordé selon deux approches. La première approche essaie d'apporter une solution à la question suivante : **comment combiner les résultats de plusieurs classifieurs afin d'obtenir un meilleur taux de reconnaissance** ? La très grande majorité des travaux traitent exclusivement du problème de la **combinaison**, c'est à dire une association parallèle de classifieurs. La règle de combinaison la plus élémentaire, et néanmoins efficace dans bien des cas, réside en une procédure de vote majoritaire.

La seconde approche pose les **interrogations structurelles** suivantes :

1. *Combien de classifieurs impliquer ?*
2. *Quels types de classifieurs utiliser ?*
3. *Comment associer à chaque entrée de classifieur un ensemble de Représentations spécifique ?*

De là, dérivent les deux problématiques de la **sélection** et de la **coopération**. La coopération de classifieurs consiste à réaliser une association en parallèle et en cascade. Cette technique pose de nombreuses questions encore à ce jour, dont la principale réside en : **comment planifier** l'ensemble des relations entre les outils ?

C'est sur ce point que porte ce travail, et plus précisément sur la prise en compte de la planification des changements de classifieurs pour suivre l'évolution des données. On suppose que les classifieurs eux mêmes ont été construits hors-ligne, à l'aide de données d'apprentissage sur un horizon fini ou infini dénombrable. Il s'agit alors de disposer, en-ligne, d'une représentation du choix possible d'un classifieur particulier, en fonction de l'évolution des données à classer.

Si l'on ne souhaite pas prendre en compte le temps de manière explicite, une solution est donnée, par exemple, par l'analyse multi-modèles [20] : les données permettent de paramétrer plusieurs modèles. Le choix du modèle à prendre en compte peut s'effectuer alors par maximum de vraisemblance *a posteriori* des estimés issues de chaque modèle [9], ou bien en seillant un test de qualité sur l'innovation évaluée par la distance de Mahalanobis [6]. L'intérêt de ce type d'approches réside en la réactivité des étages décisionnels et la possibilité d'utiliser toutes les ressources en chaque instant. Mais la sélection elle-même de chaque modèle n'est pas interprétable : elle est conduite de façon numérique par la maximisation d'une fonction de coût.

Pour gagner en interprétabilité, les travaux de Gunes [13] proposent une approche avec la notion temporelle explicite, fondée sur une mise en oeuvre des modèles conditionnée par un index temporel flou, dans le formalisme des réseaux de Petri. Une étude similaire est réalisée par Rombault et al dans [19]. L'intérêt de la séquentialité explicite de ce type d'approches est contrebalancé par la complexité de la représentation de l'entité temporelle et du raisonnement associé.

De plus, pour une mise en oeuvre dans un cadre applicatif temps réel, il est nécessaire d'intégrer les contraintes de coûts en temps et en mémoire de l'approche envisagée. Dans les approches de type multi-modèles, on doit forcément conserver en mémoire tous les modèles à chaque instant. Or la loi d'évolution des données, sous réserve qu'elle ne soit pas aléatoire, induit naturellement une séquence de modèles possibles entre deux instants.

Il semble donc pertinent de planifier explicitement plusieurs modèles de classification ordonnés non pas par un **indice temporel** mais plutôt par une **mesure de sélection fondée sur la fréquence d'occurrence**, à partir d'un ensemble de données d'apprentissage.

Après avoir précisé ce que nous entendons par *données évolutives*, après avoir défini la notion d'*agent de classification*, comme une fonction d'association de classifieurs, nous présentons dans les sections suivantes un rappel sur les définitions élémentaires de la formalisation des processus markovien, puis une adaptation des Modèles de Markov Caché (MMC) à la planification d'agents de classification. Un exemple simple précise notre contexte d'étude et la manière de générer des problèmes.

La section suivante montre, sur un exemple synthétique plus réaliste, car plus complexe, plusieurs tests : la recherche du nombre d'états optimal, la robustesse de ce nombre par rapport à du bruit sur les conditions initiales, la robustesse des paramètres du MMC, par rapport à la dispersion des séquences générées, qui valident notre approche, dans le cadre particulier des approches probabilistes. Nous concluons alors en explicitant les limitations et en ouvrant sur nos perspectives d'applications, issues d'un problème de reconnaissance de la fraîcheur de poissons débarqués en criée à l'aide de vision artificielle et de nez élec-

tronique.

2 Coopération d'agents de classification

2.1 les données évolutives

Dans la terminologie de la RdF, lorsque les caractéristiques d'objets subissent des évolutions, leur classe d'interprétation est communément appelée *classe évolutive* ou *classe dynamique*.

Par exemple, les caractéristiques physiques d'une espèce de poissons pêchée sur nos côtes varient tout au long de l'année. Ainsi, ce sont les observations d'un poisson de la même espèce (un même individu au sens de la RdF) qui varient périodiquement dans l'espace de représentation.

A ces variations, plutôt de nature cyclique, peuvent s'ajouter des variations acycliques dues au vieillissement de ces individus. Ces dérives (par exemple : vieillissement des visages, l'usure d'une machine, ...) sont très difficilement détectables quand elles s'expriment par un glissement lent des caractéristiques du modèle [3, 2, 7]. Il est alors intéressant de travailler globalement sur une notion de trajectoire. La trajectoire d'une classe se décrit dans l'espace de représentation des individus. Parce qu'il existe du bruit, indifféremment d'observation ou d'état, la trajectoire est un nuage de points évoluant dans l'espace de représentation comme sur la figure 1.

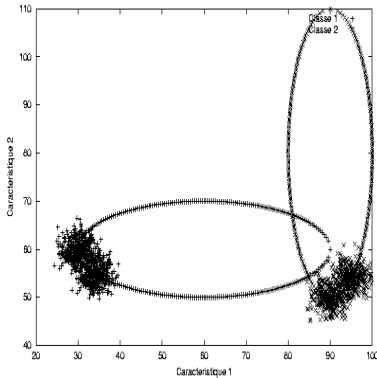


FIG. 1 – Exemple de problème à deux classes, dont les trajectoires sont elliptiques.

Le choix de l'intégration des données au cours du temps dans un espace de représentation particulier, ainsi que le niveau d'abstraction dans lequel on prend l'indice temporel en compte, est un *a priori* sur la définition de la granularité du temps.

La réactivité du système de reconnaissance, bien évidemment, en découle : plus on s'éloigne de l'espace des observations, plus on ajoute des connaissances symboliques (complication *a priori* du système, données contextuelles, ...), et moins le système est réactif.

C'est plutôt au niveau hiérarchique supérieur que nous cadrerons notre étude, portant sur la planification d'outils de

classification.

2.2 agent de classification

Soient :

- $X \subset \mathbb{R}^p$, un ensemble de vecteurs de Formes,
- Ω , le cadre de discernement,
- 2^Ω , l'ensemble des parties de Ω ,
- $x \in X$, un individu à classer,
- ϕ , le vecteur d'interprétations, ou vecteur de décisions.

Dans toute la suite de ce travail, on appelle **classifieur**, une fonction e :

Définition 1

$$e : X \longrightarrow 2^\Omega$$

$$x \longmapsto \phi = e(x)$$

La constatation du fait que des approches de classement différentes mises en compétition ne produisent pas la même erreur de classement [10] ont conduit différentes équipes, dès les années 1980, à concevoir des approches hybrides, que nous appelons *agent de classification (AC)*, afin de minimiser cette variation du comportement de l'erreur. Pour associer différentes approches, la seule hypothèse forte, mais toutefois applicable, est que les classifieurs élémentaires aient des erreurs indépendantes.

La gestion d'une structure hybride peut apporter des solutions sur les points suivants :

1. l'association d'outils fournit un taux de reconnaissance meilleur que le meilleur des outils [21, 11, 12].
2. la complexité de la forme des classes (chevauchement, multi modalités, ...) nécessite souvent la construction de frontières de décisions complexes, ce qu'apporte un réseau connexionniste, par exemple, mais de manière non lisible [14].
3. la dérive temporelle des classes nécessite de planifier des stratégies de reconnaissance en fonction du paysage évolutif à reconnaître (détection de changements de modèles, planification d'outils, ...) [13, 5].

Ces intérêts différents, dus à des spécificités de l'application à traiter, font apparaître des solutions différentes quant à la nature même de l'association à réaliser entre les outils. Soit $E = \{e_i\}, i \in [1, I]$, l'ensemble des classifieurs et $\Omega = \{\omega_j\}_{j \in [1, J]}$, les différentes classes d'interprétation. Un agent de classification doit permettre d'obtenir des performances supérieures à celles d'un seul classifieur. Ces performances peuvent être déterminées selon certains critères (le taux de reconnaissance, le critère AIC [1], le critère z_{01} [17], ...).

Par exemple, le problème de la discrimination des deux classes dont les trajectoires sont illustrée fig. 1 a été décomposé en 4 paysages de données (selon l'effondrement des taux de reconnaissance) sur lesquels sont affectés 4 agents de classification constitués de classifieurs bayésiens (fig.2). On pourra se reporter à [15] pour plus de détails sur les algorithmes.

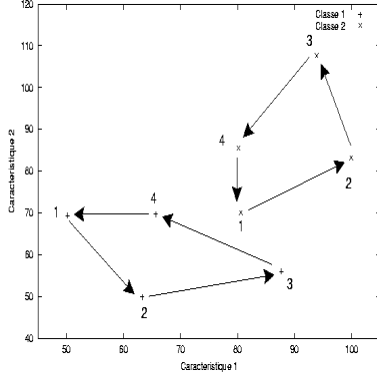


FIG. 2 – Résultat de l'apprentissage des 4 agents de classification.

Un AC est un classifieur, au sens de la définition 1. Ainsi, pour suivre la dérive des données dans l'espace des caractéristiques, il est alors possible de construire de manière hiérarchique des associations d'ACs. Dans la suite de ce travail, nous cherchons à ordonner un ensemble d'ACs à un niveau hiérarchique immédiatement supérieur : l'étage de planification.

3 Chaîne de Markov à temps discret

Soit S un ensemble dénombrable dont chaque élément s est considéré comme un état qu'une Forme (au sens de la RdF) peut occuper à un instant donné $n \in \mathbb{N}$ (le temps est discret). S peut être fini ou infini dénombrable et est nommé *espace des états*. Notons X_n la position de la Forme à l'instant n . Si la Forme évolue de manière aléatoire, X_n est une variable aléatoire à valeurs dans S , et on appelle la suite de variables aléatoires $(X_n, n \geq 0)$ un processus stochastique à valeurs dans S .

3.1 Modèle de Markov

Supposons que l'évolution future de l'Objet ne dépend que de son passé immédiat. Soit $\mathbb{X} = (X_n; n \geq 0)$, une suite de variables aléatoires à valeur dans S ; choisissons une suite d'états $i, j, i_0, \dots, i_{n-1} \in S$, telle que $P[(X_0, X_1, \dots, X_{n-1}, X_n) = (i_0, i_1, \dots, i_{n-1}, i)] > 0$ alors la propriété de Markov s'énonce, $\forall j \in S$, par :

Propriété 3.1

$$\mathbb{P}[X_{n+1} = j | (X_0, X_1, \dots, X_{n-1}, X_n) = (i_0, i_1, \dots, i_{n-1}, i)] \\ = \mathbb{P}(X_{n+1} = j | X_n = i)$$

De plus, si la loi régissant les étiquettes ne varie pas elle même avec le temps, on a la propriété d'homogénéité :

Propriété 3.2

$$\mathbb{P}(X_{n+1} = j | X_n = i) = P_{ij} \quad (i, j) \in E$$

Muni des propriétés 3.1 et 3.2, on dit que $\mathbb{X} = (X_n, n \geq 1)$ est une chaîne de Markov homogène à valeurs dans S .

Définition 2 La loi initiale de $\mathbb{X}$ est la distribution de probabilité $\nu = (\nu_i; i \in S)$ définie par :

$$\nu_i = \mathbb{P}[X_0 = i]$$

En supposant que l'ensemble $\{n \in \mathbb{N} : \mathbb{P}[X_n = i] > 0\}$ est non vide, $\forall i \in S$, on a :

Définition 3 Soit $\mathbb{X} = (X_n, n \geq 1)$, une chaîne de Markov sur S . La matrice stochastique $\mathcal{P} = (P_{i,j} \quad i \in S, j \in S)$ définie par l'équation :

$$P_{i,j} = \mathbb{P}[X_{n+1} = j | X_n = i]$$

s'appelle la matrice des probabilités de transitions de la chaîne de Markov $\mathbb{X}$.

3.2 Modèle de Markov caché : MMC

Lorsque que le processus ne peut-être que partiellement observé, le nombre d'états ne correspond plus comme précédemment au nombre d'états réels du processus physique, mais plutôt au nombre d'intervalles homogènes quant aux densités de probabilité des observables. La définition d'un modèle de Markov caché (ce sont les états eux mêmes qui sont cachés) demande donc la spécification d'un nombre distinct de symboles d'observations, qui correspondent pratiquement aux sorties effectives du processus à analyser :

soit l'ensemble des symboles $V = \{v_k\} \forall k \in [1, K]$

A cela on doit fixer une distribution de probabilités des symboles d'observation par état :

$\forall j \in S, B = \{b_j(v_k)\}$ où

$$b_j(v_k) = \mathbb{P}[v_k(t) | X_n = j] \quad \forall k \in V$$

4 Planification markovienne discrète en RdF

Notre objectif est de trouver une suite d'états dont l'exécution des agents associés parviendra à réaliser la "meilleure" performance de classification à chaque instant.

Nous avons cherché à conserver l'aspect séquentiel tout en rendant implicite la notion de transition temporelle. Ainsi, comme Gunes, nous construisons une structure de graphe qui représente les choix possibles d'un ou plusieurs modèles selon que l'on utilise tel modèle à l'instant considéré; et comme dans l'analyse multi-modèles, le choix d'un noeud s'effectue non pas selon un indice temporel, mais selon une mesure de probabilité.

4.1 Génération de séquences

Issue de l'apprentissage des agents de classification, l'information temporelle portée par le graphe de la figure 2 peut être également représentée par l'abaque de la figure 3 qui s'apparente à un ensemble de chronogrammes; chaque

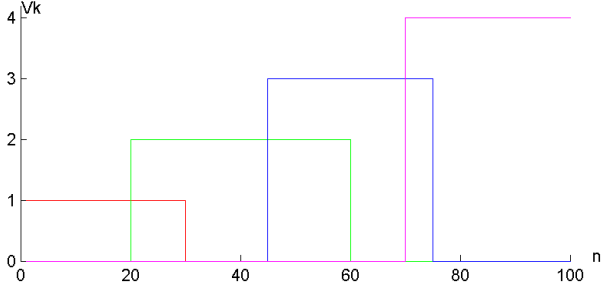


FIG. 3 – Abaque représentant les séquences de symboles des trajectoires 1

chronogramme représente la durée de validité d'un agent de classification comme un intervalle temporel. Ainsi, nous faisons apparaître des intervalles temporels I_l pour lesquels au moins un agent de classification intervient.

Supposons que les données obéissent à un cycle de période T .

Un cycle sera décomposé en une succession d'intervalles temporels contenant chaque instant :

$$\begin{aligned} \Gamma : [1, T] &\mapsto \{I_l\}_{l \in [1, L]} \\ n &\leftarrow \Gamma(n) = [n_l, n_{l+1} - 1] \end{aligned}$$

avec $n_1 = 1$ et $n_{L+1} = 1 + T$

Les agents de classification, pertinents sur un intervalle temporel I_j de ce cycle, seront représentés par l'occurrence d'un symbole v_k , conditionné à la distribution de probabilités $b_j(v_k)$.

Nous appellerons une suite de cycles, une séquence d'observations ; ces observations devront respecter l'abaque ; toutefois les listes de symboles entre deux cycles de la même séquence pourront différer grâce aux distributions de probabilités associées. Notons que les $b_j(v_k)$ sont constantes pour tous les cycles constituant la séquence.

Dans cet article, afin d'analyser précisément des cas typiques de comportement de la planification, nous supposons que l'extraction des suites d'agents de classification à partir des données a déjà été réalisée lors d'une phase de classification ; dans ce qui suit, nous avons écrit l'algorithme 1 afin de générer les séquences de symboles : un ensemble de N séquences $\mathbb{S} = \{s_i\}_{i \in [1, N]}$ de décisions (i.e. dont l'élément constituant s_i^n avec $n \in [1, T]$ est le choix d'utilisation d'un agent de classification adapté à cet instant n).

Algorithme 1 Création d'une séquence de T symboles

```

pour  $n = 1$  to  $T$  faire
   $l \leftarrow \text{IntervalleTemporel}(n, \text{chronogramme})$ 
   $v_{\text{Aleatoire}} \leftarrow \text{Aleatoire}()$ 
   $v_k \leftarrow \text{Observation}(v_{\text{Aleatoire}}, b_l)$ 
   $\text{sequence}(n) \leftarrow v_k$ 
fin pour

```

Par la suite, les séquences ont toutes la même durée T , mais notre approche s'étend aux séquences de durée variable. Par exemple, sur l'abaque de la figure 4 avec 3 intervalles temporels et 4 agents de classification), posons la matrice d'observations B :

$$B = \begin{bmatrix} 0 & 0.3 & 0.7 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0.4 \end{bmatrix}$$

La sémantique associée stipule que l'intervalle temporel I_1 est défini avec une occurrence de 30% d' AC_2 et de 70% d' AC_3 ; que durant I_2 , seul l' AC_1 survient ; et enfin que AC_2 et AC_4 interviennent respectivement avec 60 et 40%.

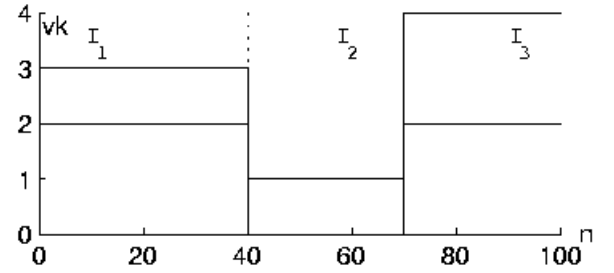


FIG. 4 – Exemple d'abaque ayant des intervalles temporels reliés à plusieurs symboles d'observation

Issue de l'abaque de la figure 4, la figure 5 illustre une séquence de deux cycles pour $T = 100$ de 4 agents de classification représentés respectivement par les symboles 1, 2, 3 et 4.

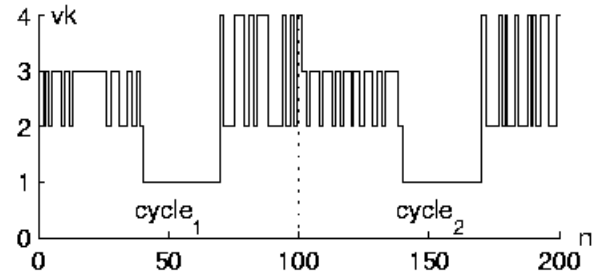


FIG. 5 – Exemple de séquences générées conformément à l'abaque

4.2 Planification par chaîne de Markov

A partir de l'abaque présenté figure 4, nous pouvons conceptualiser les séquences générées par une chaîne de Markov à 4 symboles $(v_1, v_2, v_3, v_4) = 1, 2, 3, 4$ sur la figure 6. Cette chaîne nous permet de générer la séquence de la figure 7. Il apparaît qu'il existe une probabilité non nulle de l'apparition de la suite des symboles v_3, v_2, v_4 au cours d'une séquence alors que l'abaque stipule que le symbole $v_1 \in V_2 = \{v_1\}$ doit forcément survenir entre les ensembles $\{v_3, v_2\}$ et $\{v_2, v_4\}$.

Algorithme 2 Détermination de Q^*

```

 $Q \leftarrow Q^0$ 
pour  $e = 1$  to  $N_e$  faire
   $[\pi \ A \ B] \leftarrow \text{Aleatoire}(Q, K)$ 
  répéter
    convergence  $\leftarrow$  faux
     $[LL \ \pi \ A \ B \ n] \leftarrow EM(\pi, A, B, n_{max}, \Delta_{LL})$ 
     $etatsCourts \leftarrow \text{EtatsCourts}(A, \theta_a)$ 
    si  $|etatsCourts| > 0$  alors
      si  $Q - |etatsCourts| < K$  alors
         $Q \leftarrow Q^0$ 
         $[\pi \ A \ B] \leftarrow \text{Aleatoire}(Q, K)$ 
      sinon
         $Q \leftarrow Q - |etatsCourts|$ 
         $[\pi \ A \ B] \leftarrow \text{Aleatoire}(Q, K)$ 
      fin si
    sinon
      si  $n_{max} > n$  alors
        convergence  $\leftarrow$  vrai
        enregistrer ce résultat
         $Q \leftarrow Q^0$ 
      fin si
    fin si
  jusqu'à convergence
fin pour

```

$$a_{k,l} = a_{q,i}^0 \times a_{i,j}^0 + a_{q,j}^0$$

avec $q = k$ si $k < i, k + 1$ sinon
avec $j = l$ si $l < i, l + 1$ sinon

Une distance entre MMC est une fonction qui assigne à tous couples de modèles $\lambda^\nu = (A^\nu, B^\nu)$ et $\lambda^\alpha = (A^\alpha, B^\alpha)$, un réel positif $d(\lambda^\nu, \lambda^\alpha)$. La distance de Bhattacharyya apparaît dans la littérature comme étant un choix possible. Une métrique simple pour des MMC discrets ayant tous deux N états, est la distance euclidienne [8] entre les vecteurs d'observation des matrices B^ν et B^α :

$$d_{ec}(\lambda^\nu, \lambda^\alpha) = \sqrt{\frac{1}{N} \sum_{i=1}^N \|b_i^\nu - b_i^\alpha\|^2}$$

Plus généralement, la distance euclidienne minimale [8] est définie par :

$$d_{mec}(\lambda^\nu, \lambda^\alpha) = \sqrt{\frac{1}{N} \sum_{i=1}^N \min_j \sum_{k=1}^M |b_{ik}^\nu - b_{jk}^\alpha|^2}$$

Toutefois, ces métriques ne prennent pas en compte la notion temporelle des chaîne de Markov en s'appuyant uniquement sur la matrice d'observation. Les différences de matrices d'observation sont généralement retranscrites plus amplement au niveau des métriques que celle de la matrice de transitions. Kwong et al dans [18] utilise la métrique de

Juang et Rabiner :

$$d(\lambda^\nu, \lambda^\alpha) = \lim_{T_\nu \rightarrow \infty} \frac{1}{T_\nu} (\log P(O^\nu | \lambda^\nu) - \log P(O^\nu | \lambda^\alpha))$$

où O^ν désigne une séquence générée par le modèle λ^ν . En pratique, on utilise :

$$d(\lambda^\nu, \lambda^\alpha) = \frac{1}{T_\nu} (\log P(O^\nu | \lambda^\nu) - \log P(O^\nu | \lambda^\alpha))$$

Etude de la robustesse des autres paramètres. La génération de modèles à partir de séquences d'apprentissage peut se heurter à l'absence de générativité. Une fois que λ^* est déterminé, nous vérifions la constance de la log vraisemblance associée sur de nouvelles séquences, afin de valider la robustesse quant à l'étalement des données.

Nous étudions aussi la convergence des paramètres lorsque l'initialisation est perturbée par un bruit uniforme d'amplitude variable ($\lambda_0 \approx \lambda^*$) puis lorsque l'initialisation est totalement aléatoire.

Influence du nombre de séquences d'apprentissage. Enfin, on montre l'évolution de la log vraisemblance du modèle λ en fonction du nombre de séquences d'apprentissage.

5 Application et discussion

5.1 un exemple d'étude

Pour montrer la pertinence des MMC estimés, on effectuera des comparaisons avec celui issu de l'abaque. Par exemple, soit l'abaque défini par la figure 9 dont la matrice d'observations est :

$$B^T = \begin{pmatrix} 0 & 0.5 & 0.3 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0.3 & 0.5 & 0.3 & 0 & 0 \\ 1 & 0.5 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.2 & 0.5 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 1 & 0.5 & 0.3 & 0 & 0.5 & 0.5 & 0.3 & 0.5 & 0 \end{pmatrix}$$

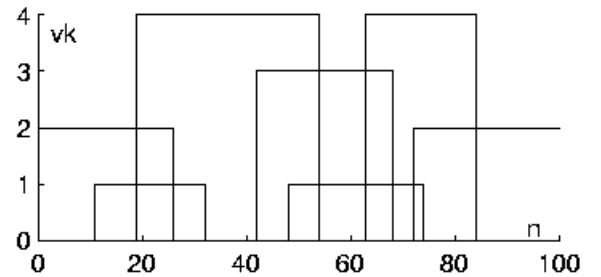


FIG. 9 – Abaque d'un cas de synthèse

Le nombre d'intervalles temporels est de $Q_{theo} = 13$. Toutefois, tous les I_k ayant un même vecteur d'observations V_k sont susceptibles d'être retranscrits par un même état. De plus, les états q_k associés aux intervalles temporels ayant une courte durée ($a_{kk} < \theta_a$) sont appelés à fusionner avec les états voisins (c.-à-d. avec les q_j tels que

$a_{k,j} > 0$ ou $a_{j,k} > 0$). Sur notre cas, puisque $V_1 = V_{13}$ et que $V_4 = V_{11}$, Q_{theo} se ramène à 11. Enfin, selon le seuil θ_a que l'on se fixe pour la durée d'un état, cette valeur diminue à nouveau.

5.2 détermination de Q^*

Nous disposons d'un ensemble de 300 séquences pour cette phase apprentissage. Comme l'indique le graphe de la figure 10, il semble que $Q^* = 9$. Les résultats de cette première série de tests (66 au total) ont été obtenus avec $Q_{max} = 50$, $\theta_{LL} = 10^{-6}$, un nombre maximum d'itérations pour une passe de l'algorithme EM de 80 et $\theta_a = 1/3$, ce qui correspond à une durée minimale de 1.5. Pour l'ensemble des MMC ayant $Q = 9$ durant cette phase, seules 3 structures distinctes apparaissent (précision de 10^{-2} près pour chaque probabilité); la figure 12 nous donne les caractéristiques de celle qui maximise la log vraisemblance.

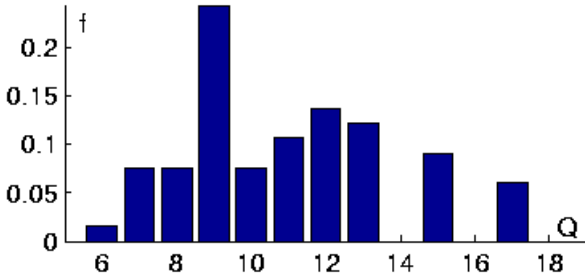


FIG. 10 – Détermination du nombre d'états

$Q^* \neq Q_{theo} = 11$ car certains états se sont regroupés. En analysant les sous-graphes obtenus, il est possible d'établir des associations d'états entre MMC_{theo} et MMC_{EM} comme le montre le tableau 1.

$\mathbb{X}_{EM}$	1	2,3	4	3,4	9	5	6	7,8
$\mathbb{X}_{theo}$	1	2	3,4	5	6	7	8	9,10,11
V_k	0	0.49	0.41	0.3	0.02	0	0.01	0.45
	1	0.48	0.09	0.35	0.48	0	0	0
	0	0	0	0	0	0	0.45	0.22
	0	0.03	0.5	0.35	0.50	1	0.54	0.33
	0	0.5	0.4	0.3	0	0	0	0.41
V_k	1	0.5	0.1	0.4	0.5	0	0	0
	0	0	0	0	0	0	0.5	0.29
	0	0	0.5	0.3	0.5	1	0.5	0.3
$a_{k,k}$	0.96	0.85	0.92	0.7	0.91	0.9	0.83	0.99
$a_{k,k}$	0.95	0.87	0.97	0.5	0.9	0.9	0.83	0.99

TAB. 1 – Association des états entre ceux du MMC_{theo} et ceux de MMC_{EM}

En analysant plus en détails les structures obtenues, on s'aperçoit que les états du MMC_{theo} formant localement une chaîne Gauche - Droite ont une tendance forte à se regrouper lorsque leurs vecteurs d'observations sont proches. De ces regroupements, on obtient généralement un état q_k

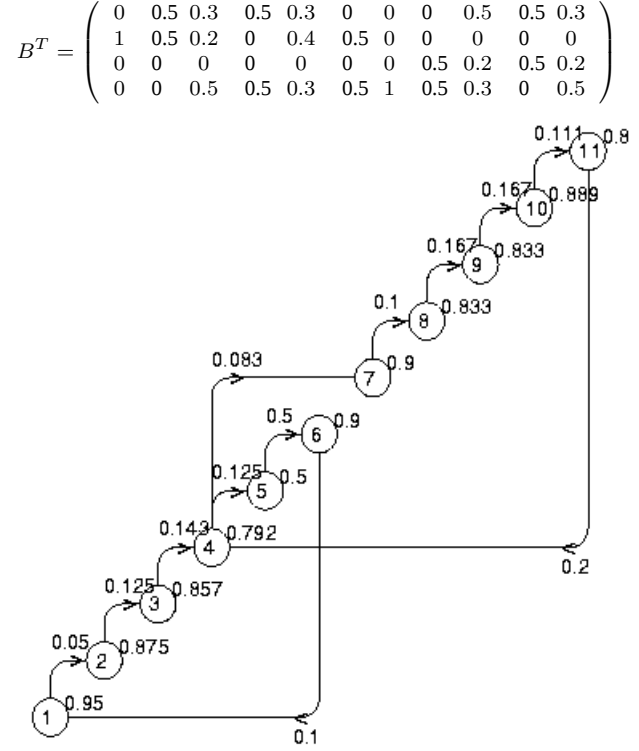


FIG. 11 – MMC_{theo} résultant des critères de fusions

pour lequel $a_{k,k} > 0.9$. Il conviendrait donc, en plus de θ_a qui impose une durée minimale, d'introduire θ_A concernant une durée maximale afin que les fusions ne soient pas effectuées de manière exagérée d'une part (modification des vecteurs d'observation), et, d'autre part, que la génération de séquences issues du MMC n'implique qu'un nombre limité d'états ($a_{k,j} \approx 0 \forall j \in [1, Q]$). On observe que la plus grande différence au niveau des d_k se situe pour l'état 4 du MMC_{EM} . Toutefois, cette variation n'implique qu'un prolongement de 2 unités de temps ($\frac{1}{1-0.97} = 10.31$ tandis que $\frac{1}{1-0.92} = 12.5$) ce qui permet de valider un formalisme où le temps est implicite.

5.3 Robustesse par rapport aux données

La figure 13, obtenue à partir de 36 expérimentations, montre qu'avec une variance de la log vraisemblance de 0.3576 qui ne représente que 0.52% de la valeur moyenne, le MMC^* généralise la diversité des séquences générées. Par ailleurs, la moyenne et l'écart type de d_{LL} pour différents essais avec des paramètres initiaux de λ est resp. de 2.7 et 0.7; cela montre la robustesse de la planification des agents de classification selon les conditions initiales de λ .

5.4 Influence du nombre de séquences d'apprentissage

Alors que la généralité du λ^* obtenu a été validé, la figure 14 indiquent, de manière prévisible, qu'avec les mêmes paramètres pour l'algorithme de Baum-Welch, plus l'en-

$$B^T = \begin{pmatrix} 0 & 0.94 & 0.08 & 0.41 & 0 & 0.01 & 0.47 & 0.43 & 0.02 \\ 1 & 0.06 & 0.86 & 0.09 & 0 & 0 & 0 & 0 & 0.48 \\ 0 & 0 & 0 & 0 & 0 & 0.45 & 0.39 & 0.03 & 0 \\ 0 & 0 & 0.06 & 0.5 & 1 & 0.54 & 0.14 & 0.55 & 0.50 \end{pmatrix}$$

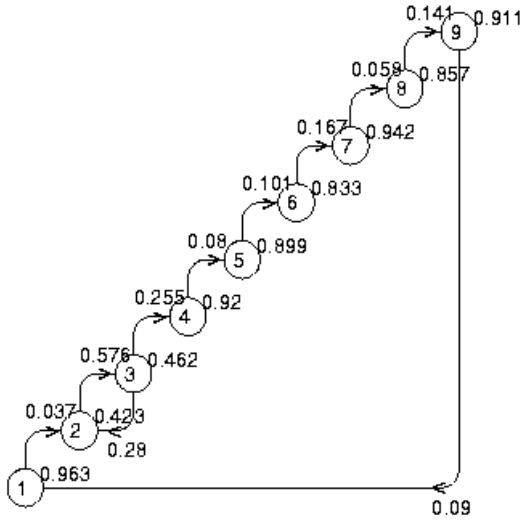


FIG. 12 – MMC_{EM} retenu lors de la détermination de Q^*

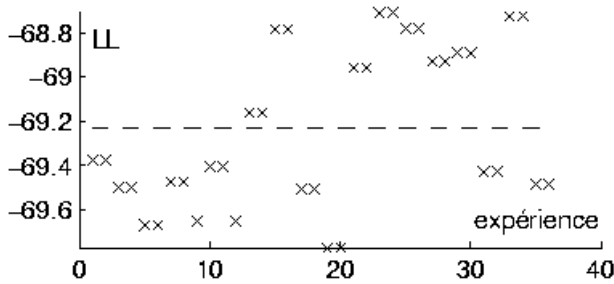


FIG. 13 – Stabilité de la log vraisemblance pour différents ensembles d'apprentissage

semble d'apprentissage est important, meilleures sont les estimations.

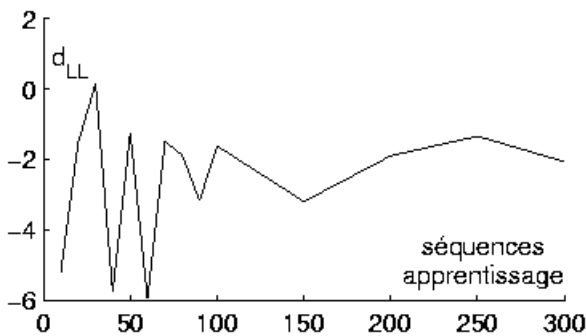


FIG. 14 – Distance entre les modèles et λ^*

6 Conclusion

Dans un problème de planification classique, on considère déterministe la dynamique du système commandé, pour un état initial entièrement défini. Mais ces hypothèses ne sont souvent pas vérifiées, et il est nécessaire de prendre en compte l'incertain dans la représentation du domaine. Classiquement, les approches probabilistes, qui nécessitent des informations de nature très quantitatives pour l'élaboration de fréquences, sont pertinentes pour représenter l'incertitude.

Ce travail concerne la modélisation du déploiement d'outils de classification à l'aide du formalisme probabiliste des processus de Markov. Il fait apparaître classiquement des notions d'états, d'actions, de transitions dans un espace probabilisé. La planification optimale résultante repose sur la maximalisation de l'espérance d'un gain, effectué par l'algorithme EM.

Notre problème de planification des différents agents de classification selon l'évolution du paysage des données à classer est alors représenté sous forme d'un processus markovien en horizon infini dénombrable ; un couple (état ; observation) décrit les probabilités de sélection des agents de classification, à un instant donné. A partir des séquences observables, nous générons des "plans" décrivant des situations de référence qui permettent de proposer dynamiquement des outils de classement.

De nombreux tests synthétiques, dont un exemple significatif est donné dans cet article, montrent plusieurs qualités de notre approche, notamment sa robustesse à la variation des conditions initiales, lorsque la dimension des séquences est suffisante.

Cette approche, fondée sur une modélisation hors ligne, si elle ne présente pas d'obstacles théoriques, induit les inconvénients pratiques classiques :

- hypothèse de monde clos : nécessite de représenter et de traiter tous les états possibles,
- coût algorithmique important.

Ces modèles doivent donc être employés lorsque le nombre d'états possibles et l'espace mémoire nécessaire sont relativement faibles, ce qui est le cas sur nos modèles de type Gauche-Droite, induisant des matrices plutôt diagonales.

Cela semble être le cas, par exemple, dans le cadre du projet eFISH (évaluation de la Fraîcheur par l'Instrumentation des Sens Humains) où l'on cherche à suivre les caractéristiques visuelles et olfactives des poissons *post-mortem* pour déterminer un indice de fraîcheur. Les ruptures de modèles associés à l'évolution naturelle des objets vivants dans leur environnement naturel font apparaître des modifications tout autant des moyennes que des variances des distributions. Ces évolutions naturelles sont de deux sortes :

- une dérive lente, associée à des phénomènes à longue période (réchauffement climatique, ...), conduisant à une dynamique faible des données ;
- une rupture nette, associée à des variations saisonnières, ou bien à des événements rares (comme des modifications climatiques exceptionnelles, des pollutions, ...),

conduisant à un changement de modèle
Ceci induit un temps présentant des embranchements. Par le passé, nous avons modélisé explicitement cette possibilité de différents futurs, à l'aide de réseau de Petri flou. Nos travaux actuels visent à comparer les performances de ces modélisations implicite et explicite du temps, pour nos applications.

Références

- [1] H. Akaike. Factor analysis and aic. *Psychometrika*, 52 :317–332, 1987.
- [2] Michèle Basseville. Identification aveugle de structure propre sous excitation non-stationnaire. In *GRETSI*, 2001.
- [3] Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes. Theory and application*. Prentice Hall Information and System science series, 1993.
- [4] S. Blackman and R.F. Popoli. *Design and analysis of modern tracking systems*. artech house, 1999.
- [5] E. Boutleux. *Diagnostic et suivi d'évolution de l'état d'un système par reconnaissance des formes floues. Application au modèle du réseau téléphonique français*. PhD thesis, Université de Compiègne, nov 1996.
- [6] . L. Dorst and F. van Lambalgen, M.and Voorbraak. *Reasoning with uncertainty in robotics*. springer, 1996.
- [7] P. Fabiani. *Représentation dynamique de l'incertain et stratégie de perception pour un système autonome en environnement évolutif*. PhD thesis, ENSAE, 1996.
- [8] M. Falkhausen, H. Reininger, and D. Wolf. Calculation of distance measures between hidden markov models. In *Eurospeech Madrid*, pages 1487–1490, 1995.
- [9] C. Fargeon. *Réalisation par microprocesseur d'un filtre de Kalman*. PhD thesis, Ecole nationale supérieur de l'aéronautique et de l'espace, 1979.
- [10] J. Francke and E. Mandler. A comparison of two approaches for combining the votes of cooperating classifiers. In *11th Int. Conf. on Pattern Recognition*, pages 611–614, the Hague, Netherlands, 1992.
- [11] G. Giacinto and F. Roli. Adaptive selection of image classifiers. In *ICIAP'97-9th Int. Conf. on Image Analysis and Processing*, pages 38–45, Florence, Italy, 1997.
- [12] B. Gosselin. Cooperation on multilayer perceptron classifiers. In *8th Work. on Circuits, Systems and Signal Processing*, pages 187–190, Mierlo, Netherlands, 1997.
- [13] V. Gunes, P. Loonis, and M. Ménard. A fuzzy petri net for pattern recognition : Application to dynamic classes. *Knowledge And Information Systems (KAIS)*, 4(1) :112–128, 2002.
- [14] V. Gunes, M. Ménard, and P. Loonis. A multiple classifier system using ambiguity rejection for Clustering-Classification Cooperation. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, 8(6) :747–762, World Scientific, Déc. 2000.
- [15] Veyis Gunes. *Système adaptatif dans le cadre de la reconnaissance des objets complexes évolutifs*. PhD thesis, Université de La Rochelle, Laboratoire d'Informatique et d'Imagerie Industrielle, October 2001.
- [16] T. K. Ho, J. J. Hull, and S. N. Srihari. On multiple classifier for pattern recognition. In *11th ICPR IAPR*, pages 84–87, Lahaye, august 1992. Chinese University of Hong Kong.
- [17] P. Loonis and M. Ménard. z_{01} , a reliability limit in pattern recognition application in exterior scenes area and automatic fish sorting. In *Information Processing and Management of Uncertainty in Knowledge Based Systems*, Paris, july 1998.
- [18] K. F. Man Q. H. He, S. Kwong and K. S. Tang. An improved maximum model distance approach for hmm-based speech recognition systems. *Pattern Recognition*, 33(10) :1749–1758, October 2000.
- [19] M. Rombaut, S. Loriette-Rougegrez, and J.M. Nigro. Temporal sequence recognition using uncertain sensor data. In *FUSION 2000*, 2000.
- [20] E. Walter and L. Pronzato. *Identification de modèles paramétriques à partir de données expérimentales*. Masson, 1994.
- [21] L. Xu, A. Krzyzak, and Suen C.Y. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3) :418–435, 1992.