Apprentissage statistique de modèles de comportement multimodal pour les agents conversationnels interactifs

Alaeddine Mihoub

UNIVERSITÉ GRENOBLE ALPES

THÈSE

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Signal, Image, Parole, Telecom (SIPT)**

Arrêté ministériel : 7 août 2006

Présentée par :

Alaeddine MIHOUB

Thèse dirigée par **Gérard BAILLY**
codirigée par **Christian WOLF**

préparée au sein du **GIPSA-Lab et le LIRIS**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique & Traitement du Signal (EEATS)**

financée par **une allocation doctorale de recherche de la Région Rhône Alpes**

Apprentissage statistique de modèles de comportement multimodal pour les agents conversationnels interactifs

Thèse soutenue publiquement le **08 octobre 2015**
devant le jury composé de :

M Mohamed CHETOUANI
Professeur, UPMC, ISIR/Paris (Rapporteur)
M Olivier PIETQUIN
Professeur, Univ. Lille, LIFL/Lille (Rapporteur)
M Abdel-Illah MOUADDIB
Professeur, Univ. Caen, Greyc/Caen (Examinateur)
M Frederic BEVILACQUA
Directeur de recherche, IRCAM/Paris (Examinateur)
M Gérard BAILLY
Directeur de recherche CNRS, GIPSA-lab/Grenoble, (Directeur de thèse)
M Christian WOLF
Maitre de conférences (HDR), INSA, LIRIS/Lyon (Co-directeur de thèse)

Université Joseph Fourier / Université Pierre-Mendès-France /
Université Stendhal / Université Savoie Mont Blanc / Grenoble INP

Résumé

L’interaction face-à-face représente une des formes les plus fondamentales de la communication humaine. C'est un système dynamique multimodal et couplé – impliquant non seulement la parole mais de nombreux segments du corps dont le regard, l'orientation de la tête, du buste et du corps, les gestes faciaux et brachio-manuels, etc – d'une grande complexité. La compréhension et la modélisation de ce type de communication est une étape cruciale dans le processus de la conception des agents interactifs capables d’engager des conversations crédibles avec des partenaires humains. Concrètement, un modèle de comportement multimodal destiné aux agents sociaux interactifs fait face à la tâche complexe de générer un comportement multimodal étant donné une analyse de la scène et une estimation incrémentale des objectifs conjoints visés au cours de la conversation. L’objectif de cette thèse est de développer des modèles de comportement multimodal pour permettre aux agents artificiels de mener une communication co-verbale pertinente avec un partenaire humain. Alors que l’immense majorité des travaux dans le domaine de l’interaction humain-agent repose essentiellement sur des modèles à base de règles, notre approche se base sur la modélisation statistique des interactions sociales à partir de traces collectées lors d'interactions exemplaires, démontrées par des tuteurs humains.

Dans ce cadre, nous introduisons des modèles de comportement dits "sensori-moteurs", qui permettent à la fois la reconnaissance des états cognitifs conjoints et la génération des signaux sociaux d’une manière incrémentale. En particulier, les modèles de comportement proposés ont pour objectif d’estimer l’unité d’interaction (IU) dans laquelle sont engagés de manière conjointe les interlocuteurs et de générer le comportement co-verbal du tuteur humain étant donné le comportement observé de son/ses interlocuteur(s). Les modèles proposés sont principalement des modèles probabilistes graphiques qui se basent sur les chaînes de markov cachés (HMM) et les réseaux bayésiens dynamiques (DBN). Les modèles ont été appris et évalués – notamment comparés à des classifieurs classiques – sur des jeux de données collectés lors de deux différentes interactions face-à-face. Les deux interactions ont été soigneusement conçues de manière à collecter, en un minimum de temps, un nombre suffisant d'exemplaires de gestion de l’attention mutuelle et de deixis multimodale d’objets et de lieux. Nos contributions sont complétées par des méthodes originales d’interprétation et d’évaluation des propriétés des modèles proposés. En comparant tous les modèles avec les vraies traces d’interactions, les résultats montrent que le modèle HMM, grâce à ses propriétés de modélisation séquentielle, dépasse les simples classifieurs en termes de performances. Les modèles semi-markoviens (HSMM) ont été également testés et ont abouti à un meilleur bouclage sensori-moteur grâce à leurs propriétés de modélisation des durées des états. Enfin, grâce à une structure de dépendances riche apprise à partir des données, le modèle DBN a les performances les plus probantes et démontre en outre la coordination multimodale la plus fidèle aux évènements multimodaux originaux.

Mots Clés

Interaction face-à-face, traitement des signaux sociaux, apprentissage statistique, modèles sensori-moteurs de comportement multimodal, modèles séquentiels incrémentaux, classifieurs, SVM, arbres de décision, modèles probabilistes graphiques, HMM, HSMM, DBN, reconnaissance de l'unité interactionnelle, génération de regard, génération de gestes, histogramme de coordination.

Abstract

Face to face interaction is one of the most fundamental forms of human communication. It is a complex multimodal and coupled dynamic system involving not only speech but also numerous segments of the body among which gaze, the orientation of the head, the chest and the body, the facial and brachiomanual movements, etc. The understanding and the modeling of this type of communication is a crucial stage for designing interactive agents capable of committing (hiring) credible conversations with human partners. Concretely, a model of multimodal behavior for interactive social agents faces with the complex task of generating gestural scores given an analysis of the scene and an incremental estimation of the joint objectives aimed during the conversation. The objective of this thesis is to develop models of multimodal behavior that allow artificial agents to engage into a relevant co-verbal communication with a human partner. While the immense majority of the works in the field of human-agent interaction (HAI) is scripted using ruled-based models, our approach relies on the training of statistical models from tracks collected during exemplary interactions, demonstrated by human trainers.

In this context, we introduce "sensorimotor" models of behavior, which perform at the same time the recognition of joint cognitive states and the generation of the social signals in an incremental way. In particular, the proposed models of behavior have to estimate the current unit of interaction (IU) in which the interlocutors are jointly committed and to predict the co-verbal behavior of its human trainer given the behavior of the interlocutor(s). The proposed models are all graphical models, i.e. Hidden Markov Models (HMM) and Dynamic Bayesian Networks (DBN). The models were trained and evaluated - in particular compared with classic classifiers - using datasets collected during two different interactions. Both interactions were carefully designed so as to collect, in a minimum amount of time, a sufficient number of exemplars of mutual attention and multimodal deixis of objects and places. Our contributions are completed by original methods for the interpretation and comparative evaluation of the properties of the proposed models. By comparing the output of the models with the original scores, we show that the HMM, thanks to its properties of sequential modeling, outperforms the simple classifiers in terms of performances. The semi-Markovian models (HSMM) further improves the estimation of sensorimotor states thanks to duration modeling. Finally, thanks to a rich structure of dependency between variables learnt from the data, the DBN has the most convincing performances and demonstrates both the best performance and the most faithful multimodal coordination compared to the original multimodal events.

Keywords

Face-to-face interaction, social signal processing, machine learning, sensorimotor models of multimodal behavior, incremental sequential models, classifiers, SVM, decision trees, probabilistic graphical models, HMM, HSMM, DBN, recognition of the interaction unit, gaze generation, gesture generation, coordination histogram.

Remerciements

D'abord, je remercie la région Rhône-Alpes qui a financé cette thèse dans le cadre d'une bourse régionale de type ARC6.

J'adresse mes remerciements ensuite à mon directeur de thèse Gérard Bailly et mon co-directeur Christian Wolf pour m'avoir donné l'opportunité de travailler sur ce sujet captivant et novateur. Je remercie sincèrement Gérard pour son encadrement de grande qualité, son investissement et son écoute tout au long de la thèse. Je remercie chaleureusement Christian pour ses suggestions pertinentes, ses conseils précieux et sa disponibilité malgré la distance géographique. Je tiens à remercier également Frédéric Elisei pour ses remarques constructives et surtout pour tout le support qu'il m'a apporté lors de mes manipulations dans la plateforme Mical.

Mes remerciements s'adressent ensuite aux membres du jury pour avoir accepté l'évaluation de mes travaux de thèse. Je souhaite exprimer ma gratitude aux rapporteurs pour leur investissement, et aux examinateurs d'avoir pris le soin de lire et d'examiner ce manuscrit.

Je tiens à remercier également les membres du Gipsa-lab et du laboratoire LIRIS pour leur sympathie et leur gentillesse. Je tiens à remercier en particulier les membres du département parole et cognition du Gipsa-lab, et surtout ceux de mon équipe CRISSP (Thomas, Pierre…) pour toutes les discussions intéressantes que j'ai pu entretenir, et qui m'ont permis d'améliorer significativement mon travail.

Enfin, mes remerciements les plus chaleureux vont à mes chers parents pour tous leurs sacrifices afin que je puisse étudier et travailler dans les meilleures conditions. Je remercie également mes deux petites sœurs, tous les autres membres de la famille, et tous mes amis pour les encouragements qui m'ont été apportés le long de cette thèse. Un remerciement particulier va à ma chère épouse pour son amour et son soutien qui m'ont permis de bien mener et achever cette thèse.

Table des matières

Table des Figures

Table des Tableaux

Glossaire

BN: Bayesian Network
BSTV: Bounded Short Time Viterbi
CH: Coordination Histogram
CHMM: Coupled Hidden Markov Model
CRF: Conditional Random Fields
DBN: Dynamic Bayesian Network
DHMM: Discrete Hidden Markov Model
EM: Expectation-Maximization
FX: Fixations du regard sur un objet d'intérêt donné
GT: Geste déictique
HMM: Hidden Markov Model
HSMM: Hidden Semi-Markov Model
HTK: Hidden Markov Model Toolkit
ID: Interlocutor Dependent (qualifie les modèles entrainés sur une unique dyade)
IDHMM: Incremental Discrete Hidden Markov Model
IDHSMM: Incremental Discrete Hidden Semi-Markov Model
II: Interlocutor Independent (qualifie les modèles entrainés sur l'ensemble des dyades)
IU: Interaction Unit (intervalle de temps consacré par les interlocuteurs à une tâche cognitive conjointe)
kNN: k-Nearest Neighbors
LN: Hélène (sujet principal dans le jeu "répétition de phrases")
MoCap: Motion Capture
MP: Manipulation
NN: Neural Networks
SMO: Sequential Minimal Optimization
SMS: Sensori-Motor State (Etat non observable qui structure le déroulement d'une IU et résulte de la factorisation d'actions conjointes entre interlocuteurs)
SP: Speech Processing
SSP: Social Signal Processing
STV: Short-Term Viterbi
SVM: Support Vector Machine

Introduction

L'interaction face-à-face représente une des formes les plus élémentaires de la communication humaine dans la vie quotidienne [1]. Néanmoins, elle reste un phénomène multimodal bidirectionnel et complexe dans lequel les interlocuteurs en permanence communiquent, perçoivent, interprètent et réagissent aux signaux verbaux et non verbaux de l'autre [2]. Bien que la parole constitue un canal de communication assez puissant dans les conversations face-à-face [3], une bonne partie de l'information est transmise de façon non verbale parallèlement au signal acoustique [4] [5]. Cette communication non verbale mobilise plusieurs canaux de transmission et peut être à la fois consciente (pour répondre à des motivations, des objectifs ou des intentions spécifiques) et inconsciente (l'effet de processus internes et automatiques) [6]. Les signaux non verbaux sont perçus phonétiquement - à travers la prosodie - et visuellement - par des gestes corporels tels que la posture du corps, les gestes brachio-manuels, les mouvements de la tête, les expressions faciales, et le regard. Ils participent vivement à l'encodage et au décodage de l'information linguistique, paralinguistique et non linguistique et sont largement impliqués dans le maintien de l'attention mutuelle et de l'attachement social [7].

La compréhension et la modélisation de ce type de communication face-à-face est une étape cruciale dans le processus de la conception des systèmes interactifs, notamment les robots et les agents conversationnels sociaux. Aujourd'hui, les systèmes interactifs représentent un axe de recherche très dynamique qui ne cesse de s'élargir et de se développer tant au niveau académique qu'industriel. La capacité à mener une communication face-à-face pertinente et fluide avec un ou plusieurs partenaires humains représentera, dans les prochaines années, une compétence et une technologie indispensables pour un système destiné à la commercialisation à grande échelle. Par conséquent, le champ d'application de cette thèse regroupe tout type de robot ou agent artificiel social destiné au domaine du loisir, assistance aux personnes, éducation, animation, réception, orientation, etc.

Doter ces systèmes d'interaction d'intelligence sociale située passe alors nécessairement par une approche bio-inspirée qui consiste à l'observation et l'étude approfondie des subtilités du comportement humain multimodal durant les interactions face-à-face sociales. Plusieurs de ces études ont confirmé que les interactions humaines sont gérées par des boucles de perception et d'action, multi-niveaux et multimodales [8] [9]. Ces boucles de perception-action planifient les différentes actions transmises en prenant en compte plusieurs paramètres extraits de l'analyse de l'environnement. Elles peuvent se manifester dans des simples boucles réflexes jusqu'à des boucles délibératives plus sophistiquées (cf. Figure 1) qui permettent la gestion de rapports sociaux et culturels [8]. Les systèmes interactifs sociaux visant à engager des conversations efficaces et crédibles avec des partenaires humains devraient ainsi être capables d'imiter et intégrer ce système de boucle d'interaction dans un processus plus large de dialogue social.

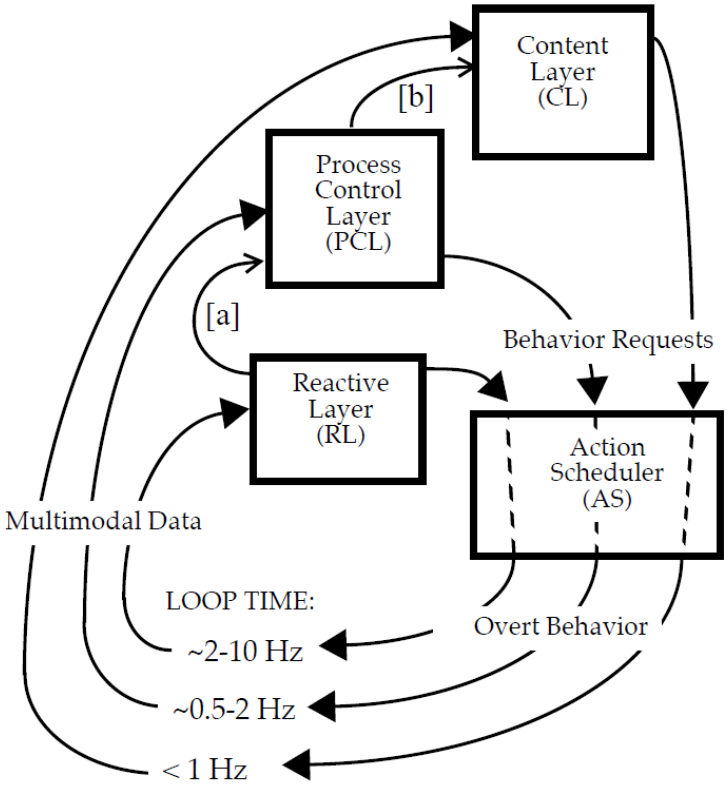


Figure 1: Boucles de perception-action imbriquées et multi-niveaux. La figure montre trois niveaux d'action correspondant à trois niveaux de priorité: RL (priorité élevée c.-à.-d boucle rapide), PCL (priorité moyenne) et CL (priorité basse). Le module RL assure les boucles réactives rapides. Le module PCL permet de modéliser les processus du dialogue en cours (qui correspondent à la notion d'unité d'interaction que nous présentons dans le Chapitre 2). Le module CL permet une interprétation plus poussée des données et donc des comportements plus adéquats. [a] et [b] illustrent qu'une partie des données multimodales prétraitées peuvent passer d'un module à un autre afin d'avoir une analyse plus approfondie. Les trois modules envoient les actions au module de la planification AS qui permet l'affichage du comportement ("overt behavior"). La Figure est reproduite de [9].

Concrètement, le système doit mener en parallèle deux tâches principales: (1) l'analyse de la scène et (2) la génération des actions multimodales adéquates. Un modèle de comportement multimodal destiné aux agents artificiels sociaux est donc face à la tâche complexe de générer un comportement multimodal pertinent étant donné (a) une analyse de la scène et (b) une estimation des intentions et des objectifs visés au cours de la conversation, le tout en mode incrémental. Notre objectif dans cette thèse est de concevoir des modèles statistiques de comportement multimodal qui apprennent à partir des traces d'interactions de partenaires humains et qui permettent de faire la correspondance entre le flux perceptuel et le flux moteur, cette correspondance étant conditionnée à l'état d'avancement de la tâche conjointe. Dans ce cadre, nous avons exploité la notion de modèles statistiques sensori-moteurs – organisant les séquences d'interaction en états et unités – qui permettent à la fois la reconnaissance et la génération des signaux sociaux d'une manière incrémentale. En particulier, les modèles de comportement proposés avaient pour double objectif d'estimer l'unité d'interaction (IU) dans laquelle est engagé un sujet principal et de générer son comportement co-verbal de manière à être cohérent avec l'IU estimée.

Les problématiques de modélisation, analyse et synthèse des signaux sociaux auxquelles un modèle de comportement doit faire face suscitent ces dernières années de plus en plus d'intérêt dans plusieurs communautés scientifiques, vu le potentiel scientifique et technologique qu'elles offrent. Le SSP ("Social Signal Processing") [10][11] est un axe de recherche émergent dont le but est de traiter les différentes problématiques liées au traitement des signaux sociaux. Une revue des travaux relatifs au domaine du SSP ainsi que les défis encore à surmonter sont abordés avec plus de détails dans le Chapitre 1 de l'état de l'art.

Le manuscrit comporte 5 chapitres: le chapitre 1 est une revue de l'état de l'art, dans laquelle nous revisitons la littérature de l'analyse et la synthèse des signaux sociaux dans les interactions face-à-face. Le chapitre 2 décrit les deux scénarios que nous avons exploités pour collecter des jeux de données de type homme-homme, permettant l'étude et l'apprentissage du comportement humain multimodal. Les trois chapitres suivants décrivent les différents modèles de comportements que nous avons proposés en partant des modèles les plus simples (classifieurs) jusqu'aux modèles plus élaborés en terme de représentation graphique et complexité des relations temporelles entre les variables (Réseaux Bayésiens dynamiques) en passant par les modèles séquentiels (chaînes de Markov cachées). Ainsi, le chapitre 3 expose notre première approche de modélisation basée sur des classifieurs classiques et largement utilisés dans la littérature (SVM et arbres de décisions). Le chapitre 4 présente des modèles basés sur le paradigme des chaînes de Markov cachées (HMM) qui permettent une modélisation séquentielle de l'interaction sociale à l'encontre des classifieurs qui effectuent une mise en correspondance (un "mapping") directe entre perception et action sans aucune modélisation séquentielle explicite. Dans le même chapitre, nous testons les modèles semi-markoviens (HSMM) et nous comparons leurs apports par rapport aux simples HMM. Le chapitre 5 propose un modèle probabiliste graphique se basant sur les réseaux Bayésiens dynamiques (DBN) ainsi qu'une nouvelle méthode pour évaluer la proximité des interactions générées par rapport aux interactions réelles. Nous conclurons enfin sur nos différentes contributions et résultats et développerons les perspectives qui découlent de ces travaux.

Chapitre 1 Etat de l'art

1.1 Introduction

Ibn Khaldoun[1] - historien et philosophe arabe - a introduit dans son Muqaddimah (Introduction à l'histoire universelle et à la sociologie moderne, écrite au 14e siècle) [12] la théorie de la sociabilité naturelle dans laquelle il expliquait que l'interaction sociale des hommes et des organisations est une chose naturelle et nécessaire afin de subsister à leurs besoins et de s'entraider mutuellement. Avec le progrès technologique, les neuroscientifiques ont identifié des neurones appelés "miroirs" [13] dont l'un des rôles présumés est d'assurer et faciliter l'apprentissage et la gestion de l'interaction sociale entre individus. Ces neurones en effet semblent être impliqués dans un circuit de traitement de l'information multimodale permettant de coopérer, d'apprendre des autres par observation et imitation [14], de partager des émotions et généralement d'améliorer la conscience d'autrui ainsi que d'estimer leurs intentions. Aujourd'hui, avec l'intérêt grandissant pour l'interaction sociale et la multiplication des interfaces numériques, il est devenu nécessaire de réduire l'écart social entre les hommes et les machines. Une nouvelle vision de l'informatique centrée sur l'humain s'impose [15] et l'adoption d'une dimension sociale et interactive pour les machines est devenue un besoin crucial pour le futur [16]. Le challenge sera de rendre ces machines capables d'incorporer les modes naturels de l'interaction sociale et les notions élémentaires de la communication humaine [11]. Outre la robotique cognitive et l'informatique affective – i.e. "affective computing" – [17], le traitement des signaux sociaux – i.e. "Social Signal Processing" (SSP) – [10], [11], [18], [19] est un domaine émergent non seulement dans le domaine de la perception, le traitement du signal et de l'image, mais aussi en sciences humaines et sociales, en impliquant notamment la sociologie, la psychologie et l'anthropologie. Au cours des dernières années, il est devenu un domaine de recherche attrayant et il y a une prise de conscience croissante sur ses défis technologiques et scientifiques. Le SSP vise à doter les machines et particulièrement les systèmes interactifs de l'intelligence sociale (SI) [20], définie par Vinciarelli et al. [11], comme "[...] la facette de nos capacités cognitives qui nous guide à travers nos interactions sociales quotidiennes, celles qui nous obligent à être un collègue respecté, un parent attentif, ou tout simplement une personne agréable". L'intelligence sociale, comme définie par Albrecht [20] est un ensemble de compétences sociales pratiques pour interagir harmonieusement avec les gens dans tous les contextes. Albrecht a ainsi intégré 5 points-clés (conscience de la Situation, Présence, Authenticité, Clarté, et Empathie) dans son modèle "SPACE" pour décrire cette intelligence et fournir un ensemble de recommandations pour la développer et l'entretenir. Pour un agent artificiel, l'intelligence sociale vise à la perception correcte, l'interprétation précise, et l'affichage approprié de signaux sociaux [21]. En d'autres termes, elle vise à assurer une analyse pertinente de l'interaction et une génération appropriée du comportement multimodal à afficher. Ce sont les deux tâches principales d'un modèle de génération de comportement.

[1] http://fr.wikipedia.org/wiki/Ibn_Khaldoun

Dans ce chapitre, nous commençons par définir l'ensemble des signaux sociaux, ensuite nous proposons une revue des travaux existants sur l'analyse et la synthèse de ces signaux ainsi que les problématiques afférentes.

1.2 Les signaux sociaux

Un signal social (voir exemple Figure 2) est défini par Poggi et al. [22] [11] comme "[...] *a communicative or informative signal or a cue which, either directly or indirectly, provides information about social facts, that is, about social interactions, social emotions, social attitudes, or social relations*". Dans cette définition, Poggi et al. distinguent les signaux de communication des signaux d'information : un signal est dit "communicatif" s'il est généré dans le but de transmettre un message. Il est "informatif" si l'interlocuteur donne à ce signal une signification ou un sens même si le locuteur n'avait pas l'intention de transmettre un message particulier. De manière similaire, Campbell [23] distingue "fillers" et "wrappers" dans sa conception de la structure du langage parlé, où les secondes unités de langage, à l'image des marqueurs phatiques – "back-channels" – ou régulateurs du discours – comme "tu vois ce que je veux dire" –, facilitent la transmission du contenu informationnel essentiellement assuré par les premières unités.

Pour les signaux sociaux, on distingue 4 types de fonctions : les actions/interactions sociales, les émotions, les attitudes et les relations [10] [11].

- Actions sociales : une action est dite sociale si un agent X la réalise en liaison avec un agent Y c.-a.-d. que l'agent Y est présent dans le processus cognitif de l'action effectuée par X [24]. Dans les interactions face-à-face, on cite deux exemples d'action: la prise de tour – "turntaking" – et les marqueurs phatiques – "backchannel". La prise de tour est le processus par lequel les interlocuteurs négocient et prennent la parole durant une conversation quelconque. La prise de tour de parole est déclenchée par des signaux non verbaux comme l'ouverture de la bouche ou le contact oculaire [25]. Quant au marqueur phatique, il est souvent transmis par des hochements de tête ou aussi des signaux para-verbaux (par ex. "uhuh" et "mm-hmmh"). Les marqueurs phatiques informent le locuteur si son interlocuteur est effectivement en train de suivre, de comprendre, et éventuellement d'être d'accord avec ce qu'il dit [26]. Ceci va lui permettre le cas échéant d'améliorer ou ajuster en partie son discours. Ces marqueurs phatiques sont alors des éléments typiques de la conversation humaine et leur absence peut perturber de manière significative la capacité des interlocuteurs à communiquer naturellement.
- Emotions sociales : ici il faut savoir distinguer les émotions liées à un événement personnel des émotions sociales. Ainsi l'admiration, l'empathie, la haine, la jalousie, la honte sont des exemples typiques d'émotions sociales. D'autre part, des émotions individuelles comme la joie, l'anxiété etc. peuvent être considérées comme sociales si elles ne sont pas liées simplement à l'état interne de la personne mais plutôt liées à l'interaction et à la présence des partenaires [27].

- Attitudes sociales : Une attitude est le résultat d'un ensemble de croyances, d'opinions, de préférences et d'émotions qui se forment pendant une interaction [19]. On cite par exemple le mépris, la supériorité, ou aussi la dominance. Selon Gilbert [28], une attitude peut être définie comme "une évaluation positive ou négative d'une personne ou d'un ensemble de personnes". L'accord, de son point de vue, est considéré comme une attitude positive alors que le désaccord comme une attitude négative. Parmi les signaux affichés en situation d'accord ou de désaccord, on trouve les bras croisés, les mouvements de tête, les sourires, etc. [29].
- Relations sociales : une relation est dite sociale si elle concerne deux ou plusieurs interlocuteurs dont les objectifs sont liés [30]. On trouve plusieurs types de relations, par exemple : coopération, compétition, dépendance, professionnelle, amitié, amour, etc. Plusieurs signaux peuvent nous renseigner sur les éventuelles relations : la façon de parler, la prosodie, la proximité spatiale, la fréquence et les régions fixées par le regard etc. D'autres signaux peuvent s'avérer également pertinents dans la détection des relations dans les groupes, par exemple la manière de s'habiller ou aussi la disposition spatiale qui indiquent d'éventuelles relations de pouvoir ou d'autorité.

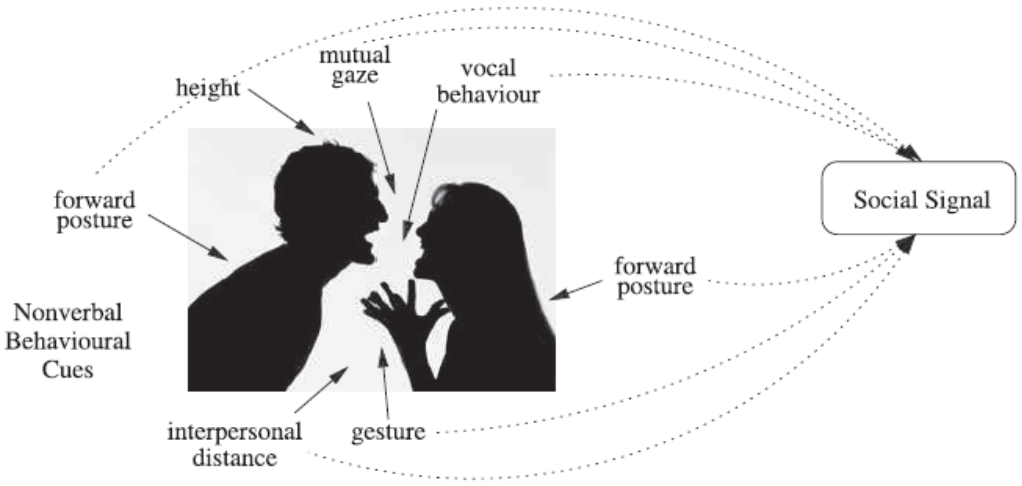


Figure 2: Les comportements non verbaux. Le comportement non verbal permet de comprendre facilement que les signaux sociaux échangés sont des signaux de désaccord, d'hostilité et d'agressivité, et que les deux personnes ont une relation tendue (figure reproduite de [11]).

1.3 Analyse des interactions sociales

L'analyse automatique des conversations [31], [32] a suscité beaucoup d'attention ces dernières années dans plusieurs communautés scientifiques. L'analyse conversationnelle essaye d'inférer à partir des données audio-visuelles brutes [33] [34] des connaissances sur les actions, les émotions, les attitudes et les relations existantes dans les interactions sociales. Plusieurs modèles computationnels ont été proposés pour traiter ces problématiques. Dans cette section, nous donnons un aperçu sur les sujets qui ont été le plus investis par les différents intervenants du domaine, ainsi que les modèles d'analyse et de description concrète de ces interactions les plus utilisés.

1.3.1 Détection de la structure et des actions sociales

Pentland et ses collègues [35] [36] [37] ont modélisé les conversations face-à-face en utilisant des capteurs portables appelés des sociomètres (cf. Figure 3). Ces capteurs permettent de caractériser les gestes/mouvements corporels, et d'estimer l'emplacement et la proximité des interlocuteurs en utilisant des balises infrarouges (IR pour "Infra-Red") et des accéléromètres. Ils ont construit un modèle basé sur les chaînes de Markov cachés et couplés (CHMM, pour "Coupled Hidden Markov Model") pour décrire les interactions entre deux personnes, quantifier leur degré de couplage et prédire leur dynamique. Ce modèle est appelé modèle d'influence. Ce modèle a été utilisé par exemple pour prédire l'issue d'une négociation étant donné les premières minutes d'interaction. Pentland et al. ont ainsi montré que plus de 70% des issues de rencontres rapides "speed dating" peuvent être prédites par les 3 premières minutes d'interaction. D'une manière plus générale, les études de Pentland visent à mieux comprendre la dynamique des interactions dans les organisations complexes afin de gérer efficacement et améliorer leur fonctionnement.

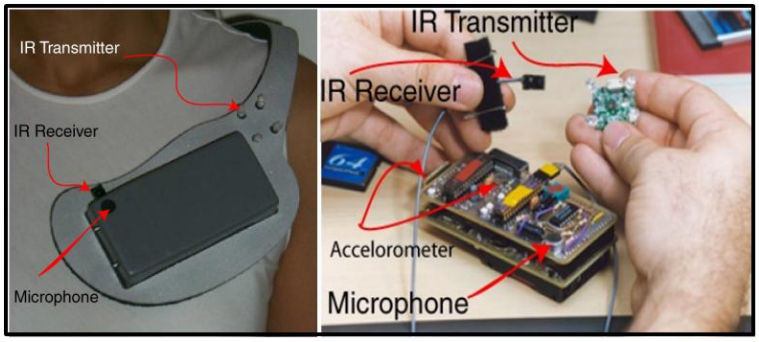


Figure 3: Le Sociomètre (figure reproduite de [38])

Otsuka et ses collègues [39] [40] ont utilisé les réseaux Bayésiens dynamiques – i.e. "Dynamic Bayesian Network" (DBN) – (pour une description détaillée, voir Chapitre 5) pour estimer la structure d'une interaction face-à-face multi-parties (4 personnes) et pour estimer l'adressage et la prise de tour depuis la seule observation de l'activité vocale et l'orientation de la tête des interlocuteurs. Le modèle DBN proposé (cf. Figure 5) est composé de trois couches: la première perçoit la parole, les directions et les mouvements de tête; la deuxième couche estime les motifs des regards (cf. Figure 4) tandis que la troisième couche estime les régimes conversationnels. La première couche est visible, alors que les autres sont latentes et doivent être estimées. Dans cette modélisation, c'est principalement les motifs des regards des participants qui vont indiquer le type du régime. Les régimes conversationnels estimés sont au nombre de trois : un régime de convergence où l'attention collective est portée sur un seul locuteur qui s'adresse donc à l'audience, un régime dyadique dans lequel seules deux personnes sont engagées et conversent en aparté, et un troisième régime de divergence qui représente des situations autres que les deux premiers régimes. Il s'agit alors d'une situation de non organisation ou simplement de silence collectif. D'autres modèles de prédiction de structure et des prises de tours ont été développés récemment par Otsuka et ses collègues [41] [42]. Ainsi, dans [41], les auteurs ont démontré d'une manière concrète une relation de

corrélation forte entre les modèles de transition de regard et le temps de démarrage du prochain locuteur. Dans [42], ils ont eu recours à une approche différente: ils ont réalisé une étude fondamentale sur la respiration dont les résultats révèlent notamment qu'un locuteur qui veut garder la parole inhale plus rapidement après la fin d'une unité d'énonciation. De manière opposée, un locuteur qui veut prendre le tour de parole prend un souffle beaucoup plus grand, lui permettant ainsi d'être prêt à prendre la parole sur une plus grande période.

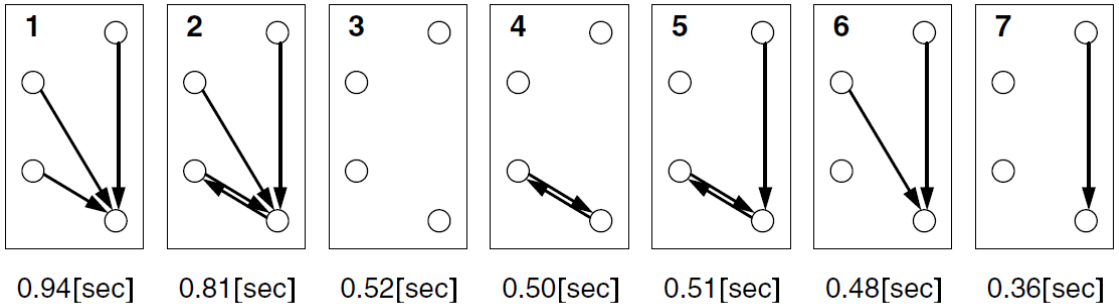


Figure 4: Les 7 motifs de regard les plus fréquents dans l'interaction face-à-face enregistrée par Otsuka et al. [39]. Les cercles représentent les 4 participants de l'interaction et les flèches représentent les directions de regard. La durée moyenne en dessous des motifs est donnée en secondes (figure reproduite de [39]).

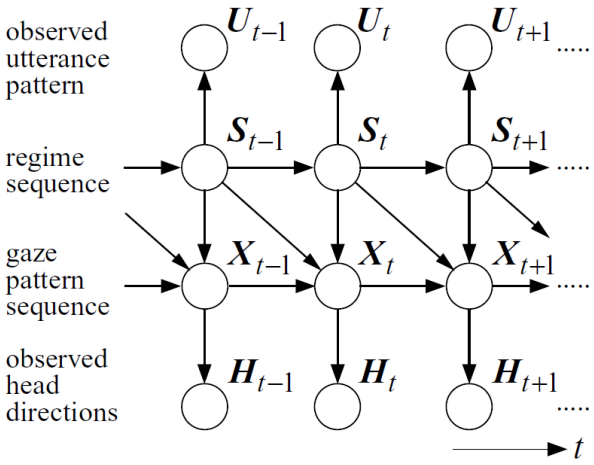


Figure 5: Le modèle DBN proposé par Otsuka et al. (figure reproduite de [39]). *U et H* (parole et direction de tête) sont observées alors que *X et S* (motif du regard et le régime conversationnel) sont à estimer.

D'autres auteurs se sont intéressés en particulier aux problèmes d'adressage. Jovanovic et al. [43]–[46] ont ainsi présenté un ensemble de caractéristiques verbales, non verbales, et contextuelles afin de déterminer les personnes adressées dans une interaction. Dans [44], ils ont annoté l'adressage dans les enregistrements de la base de données "AMI", qui est une collection publique de réunions de quatre personnes équipées de plusieurs microphones et de caméras vidéo [47]. L'annotation a été faite pour indiquer si le locuteur s'adresse à une seule personne, à un sous-groupe, à l'ensemble du public, ou si l'interlocuteur est inconnu. L'étude du processus manuel d'annotation [44] indique que la fiabilité était plus élevée sur les segments où le locuteur aborde une seule personne, et qu'il y avait des difficultés à distinguer entre les adressages aux sous-groupes et au groupe entier. Dans [46], la tâche de

reconnaissance a été confiée à des réseaux Bayésiens et une combinaison de caractéristiques extraites d'une manière manuelle et automatique. Ces caractéristiques contiennent des informations contextuelles (informations sur les interlocuteurs et les actes de dialogue), des informations lexicales (noms propres, pronoms personnels comme "you/we", pronoms possessifs comme "our/your", pronoms indéfinis comme "somebody/anyone", etc.) et les directions de regard. Le meilleur résultat était d'environ 75% de bonne détection.

On trouve aussi plusieurs travaux permettant de reconnaître les actions multimodales individuelles et collectives dans un groupe de personnes. Par exemple, Zhang et al. [48] ont proposé un HMM (HMM pour "Hidden Markov Model", pour une description détaillée voir Chapitre 4) à deux couches (cf. Figure 6). La première couche estime les actions individuelles à partir des données audiovisuelles (corpus AMI). Les caractéristiques acoustiques sont l'énergie, le pitch et le descripteur SRP-PHAT ("Steered Response Power-Phase Transform"). Les caractéristiques visuelles sont les positions et les mouvements de la main droite et la tête de chaque participant (cf. Figure 7). Les actions individuelles à estimer par la première couche sont "parler", "écrire" et "autre". Ensuite, la seconde couche déduit les actions de groupe en tenant compte des estimations de la première couche. A savoir que les actions de groupe sont essentiellement: "discussion", "monologue", "présentation", "prendre des notes", "écrire sur le tableau blanc" ou aussi des combinaisons comme "monologue+prendre des notes". Ce modèle a montré son efficacité par rapport à un modèle de base avec une seule couche de HMM. De plus, le modèle multimodal (utilisant l'ensemble des caractéristiques audiovisuelles) avait un taux d'erreur moins élevé qu'un modèle unimodal (utilisant uniquement les données acoustiques ou visuelles).

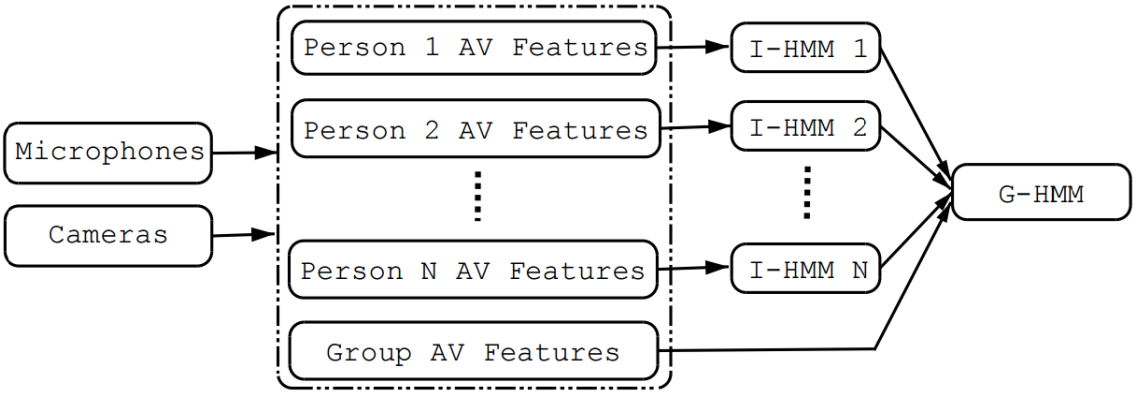


Figure 6: Le modèle proposé par Zhang et al. (figure reproduite de [48]) impliquant deux couches de HMM. I-HMM désigne "Individual action HMM" et G-HMM désigne "Group action HMM". AV désigne "Audiovisual".

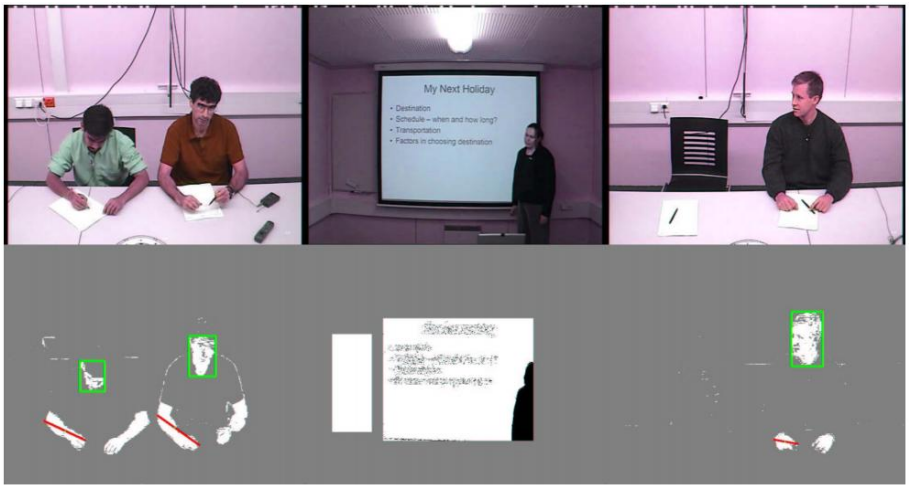


Figure 7: Extraction des caractéristiques visuelles utilisées dans le modèle de Zhang et al. (figure reproduite de [48])

1.3.2 Détection des émotions, des attitudes et des relations sociales

La recherche autour de la détection des émotions [49] [50] basiques et individuelles (par ex. la joie, la peur etc.), à partir de données audiovisuelles d'un environnement fermé, est relativement avancée. Quant à la détection des émotions sociales c.-à.-d. les flux affectifs collectifs, les études restent encore limitées et peu matures [10]. La détection de ces émotions est encore plus problématique si les scenarios étudiés sont ouverts et plus naturels (par ex. les consultations médicales, les entretiens d'emplois, les débats télévisés etc.). Pour un seul locuteur, Petridis et Pantic [51] ont présenté une approche audiovisuelle pour distinguer le rire de la parole en montrant sa pertinence par rapport à une approche uni-modale. Les caractéristiques utilisées étaient essentiellement les distances entre 20 points faciaux détectés dans les images (4 points pour les extrémités des sourcils, 8 points pour les deux yeux, 3 points pour le nez, 4 point pour la bouche et 1 point pour le menton) et 26 descripteurs acoustiques de type PLP ("Perceptual Linear Prediction coding features"). Le modèle utilise une combinaison de réseaux de neurones et d'Adaboost (la méthode Adaboost est utilisée ici pour la sélection des caractéristiques). Le modèle a atteint un taux de 86,9% pour le rappel et de 76,7% pour la précision. Kumano et al. [52] ont présenté une méthodologie pour modéliser les émotions sociales pendant les conversations face-à face multi-parties. En mettant l'accent sur l'empathie et l'antipathie partagée entre un couple de personnes, leur approche permet de caractériser les émotions par le biais d'observateurs externes (cf. Figure 8). En traitant les différences perceptuelles de l'état émotionnel comme une distribution de probabilité, le modèle computationnel proposé (basé sur un DBN, cf. Figure 9) a permis de décrire efficacement la relation entre l'émotion perçue et les principaux comportements non verbaux comme le regard (3 motifs: mutuel, à sens unique, et mutuellement évité) et les expressions faciales (plusieurs catégories: neutre, sourire, rire, penser, etc.). L'avantage de la méthode proposée est qu'elle facilite l'évaluation quantitative de plusieurs phénomènes, dont la vérité terrain est difficile à annoter. Le modèle DBN proposé a permis d'avoir un taux de reconnaissance de 64% pour l'empathie et 81% pour l'antipathie [53].

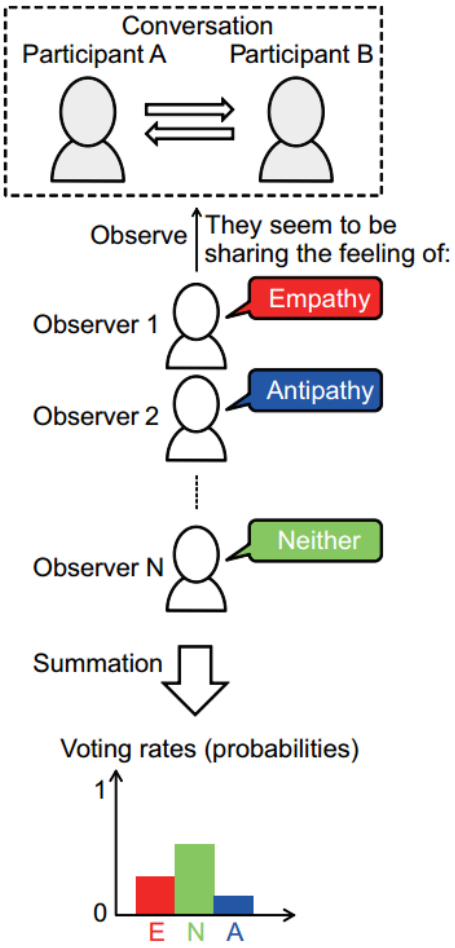


Figure 8: Etude de l'empathie et l'antipathie par des observateurs externes (figure reproduite de [52]) .

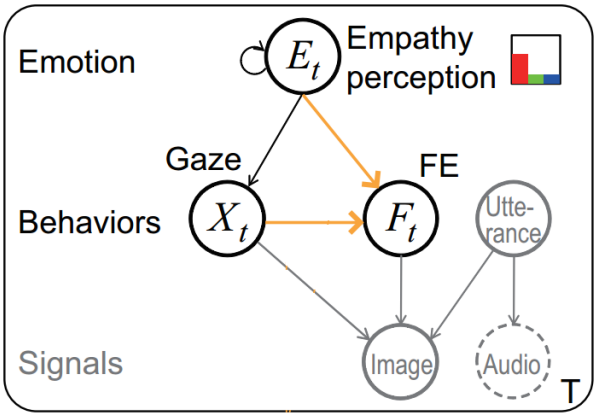


Figure 9: Le modèle DBN proposé par Kumano et al. (figure reproduite de [52]) décrit la relation entre l'émotion perçue et les principaux comportements non verbaux comme le regard et les expressions faciales (FE pour "Facial Expression").

Pour les attitudes sociales étudiées, l'attitude de dominance a été en particulier largement abordée dans la littérature [54]–[58]. Ce concept est bien étudié en psychologie sociale car il a des implications importantes sur la communication au sein des groupes. Il pourrait apporter des avantages dans certains contextes alors que dans d'autres, il pourrait léser la dynamique sociale d'un groupe en impactant négativement sa cohésion, son efficacité et surtout les relations sociales. Afin d'évaluer la domination de chaque personne impliquée dans une interaction multi-parties (dominant/non-dominant), Jayagopi et al. [56] ont développé un classifieur de type SVM (SVM pour "Support Vector Machines", pour une description détaillée voir Chapitre 3) en ayant recours à plusieurs caractéristiques auditives et visuelles. Les principaux résultats ont montré que les caractéristiques extraites de la modalité audio (par ex. l'énergie accumulée le long de la conversation, histogrammes de durée de parole, nombre de prise de tours, interruptions réussis, etc.) sont les plus pertinentes et que les signaux visuels (par ex. positions et mouvements de tête et des mains) contribuent relativement à l'amélioration de la puissance discriminative du classificateur. Le meilleur taux de reconnaissance pour les personnes dominantes était de 91%. Une fois cette identification effectuée, des actions peuvent être engagées hors-ligne par un système de recommandations,

soit de manière plus ambitieuse, en ligne par un système de régulation qui signale en temps-réel les recommandations - cf. le "negociation advisor" proposé par Pentland et al. [59]- voire modifie les signaux sociaux échangés.

De manière complémentaire aux attitudes, les relations sociales, et particulièrement les rôles, représentent un élément-clé pour comprendre les interactions humaines au sein des groupes étudiés. Formellement, Hare [60] définit le rôle comme "[...] un statut qui donne à une personne des droits et des devoirs envers une ou plusieurs personnes du groupe", comme par exemple un modérateur dans une réunion ou un animateur dans une émission diffusée par radio/télé, etc. [61]. D'autres rôles informels peuvent émerger au cours des interactions sociales [31]. Ces rôles non définis a priori caractérisent des situations particulières faisant évoluer les relations sociales au cours du temps, par exemple une situation de débat où on aura des supporteurs d'un coté et des opposants d'un autre coté. Plusieurs modèles ont été développées pour traiter cette problématique. Par exemple, un arbre de décision (pour une description détaillée voir Chapitre 3) est utilisé dans [62] pour la détection automatique des rôles. Basé principalement sur les caractéristiques acoustiques, le classificateur assigne des rôles à chaque participant: présentateur, participant de la discussion, fournisseur actuel d'information, récepteur d'information et autre. Les auteurs ont testé plusieurs fenêtres temporelles d'observation glissante et la meilleure avait comme longueur 20 secondes. Dans cette fenêtre temporelle, plusieurs caractéristiques ont été extraites parmi lesquelles on trouve: le nombre de fois où il y a changement de locuteur (différent du nombre de prises de tour car un marqueur phatique est considéré comme un changement), le nombre de personnes qui ont soulevé le même point, le nombre de chevauchements de parole, la durée totale de parole d'un participant et la durée totale de parole d'un participant en chevauchement avec la parole d'un autre. En combinant toutes ces caractéristiques, le modèle a permis d'avoir un taux de reconnaissance relativement modeste de 53%.

Une approche plus récente de Salamin et al. [61] est utilisée également pour la détection des rôles. Comme illustrée dans la Figure 10, elle se déroule en trois étapes: diarisation, extraction des caractéristiques et reconnaissance des rôles. La diarisation consiste à segmenter la conversation en plusieurs prises de parole et à assigner chaque segment de sortie à un interlocuteur particulier. Dans la deuxième étape, deux ensembles de caractéristiques ont été extraits. Le premier ensemble est basé sur l'organisation des prises de parole estimée dans la première étape (par ex. nombre de prises de tour pour le locuteur actuel, durée moyenne d'une prise de tour, durée moyenne de temps entre deux prises de tour etc.). Dans le deuxième ensemble, on trouve des informations sur la prosodie (par ex. la mélodie et l'énergie) et également la longueur de chaque segment parlé et non parlé. Ces différentes caractéristiques sont utilisées dans l'étape de la reconnaissance par un modèle se basant sur les champs aléatoires conditionnels – "Conditionnal Random Fields" (CRF) – pour estimer la séquence des rôles des différents interlocuteurs. Le domaine d'application de cette approche était la détection des rôles dans des émissions diffusées par radio. Le taux de reconnaissance du modèle CRF développé était de 85%.

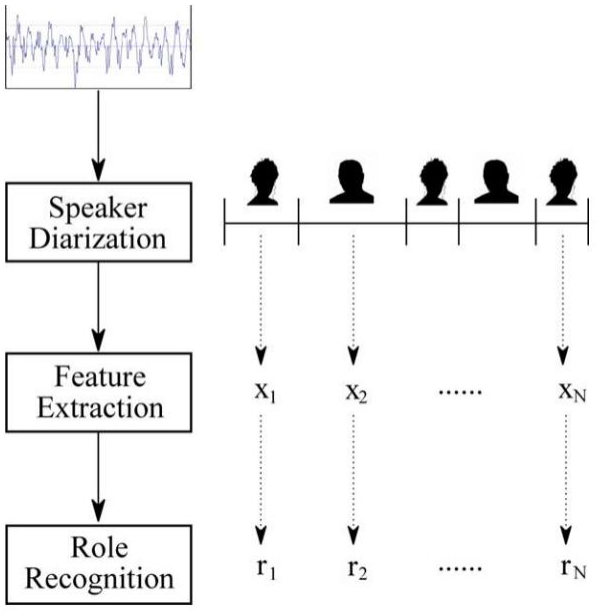


Figure 10: L'approche de Salamin et al. (figure reproduite de [61]) se déroule sur trois étapes: la diarisation, l'extraction des caractéristiques et enfin la reconnaissance des rôles avec un modèle de type CRF.

Plus de détails sur les problématiques ainsi que les modèles proposés pour traiter l'analyse automatique de l'interaction sociale non verbale peuvent être trouvés dans [32] [31] [11]. La plupart des modèles présentés ci-dessus ne traitent que des problématiques de perception et d'analyse de scène. Les modèles de comportement que nous proposons dans les prochains chapitres vont au-delà de cette dimension perceptuelle. Ils visent non seulement la compréhension de la scène perçue mais aussi la génération des actions pertinentes en retour. Dans un modèle de comportement, l'analyse est effectuée au service de la génération: nous insistons sur le fait que la perception est active et que les actions générées modifient constamment la perception des agents qui interagissent. Dans la section précédente, nous avons présenté brièvement l'état de l'art de l'analyse des interactions sociales. Dans la prochaine section, nous proposons une brève revue des travaux existants sur la génération des comportements.

1.4 Génération du comportement non-verbal

La génération de comportements sociaux non-verbaux pertinents et adaptés à la tâche, à l'interlocuteur et à la situation de communication est la deuxième portée du traitement des signaux sociaux (SSP). Elle nécessite des modèles computationnels robustes capables de prédire et synthétiser les bons signaux à transmettre. L'objectif est d'intégrer ces modèles dans des agents conversationnels virtuels, des robots humanoïdes, des interfaces intelligentes, ou tout autre dispositif capable d'interagir de manière naturelle et intuitive avec les utilisateurs, à l'image des gens interagissent les uns avec les autres [63]. Ces modèles de comportements visent à rendre ces systèmes interactifs capables d'afficher des actions, des émotions, des attitudes et de développer des relations sociales à travers leurs organes ou composants artificiels. Deux types de méthodes ont été proposées pour modéliser et synthétiser le

comportement humain: d'une part on trouve les modèles scriptés ou à base de règles, et d'autre part les modèles statistiques basés-données – i.e. "data-driven". Dans notre travail, nous nous intéressons en particulier aux approches basées données qui permettent de déduire automatiquement les modèles de comportement à partir de traces d'interactions – préalablement collectées par observation ou démonstration de comportements adéquats par des tuteurs humains – en utilisant des techniques d'apprentissage statistique.

1.4.1 Les modèles à base de règles

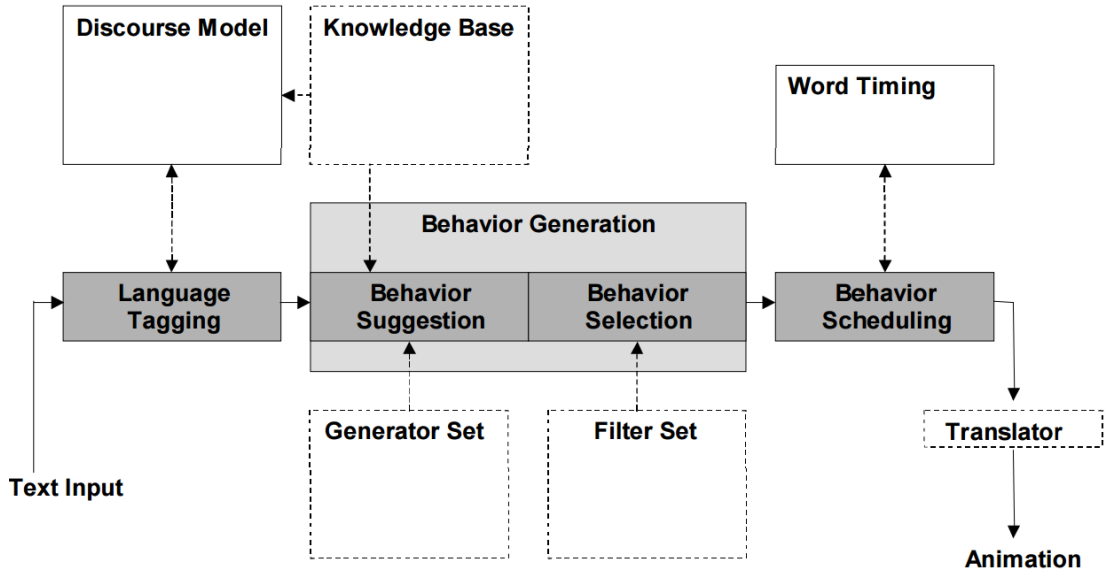


Figure 11: L'architecture du système BEAT (figure reproduite de [64])

Plusieurs méthodes à base de règles et de connaissances explicites ont été proposées pour les agents virtuels et les robots humanoïdes. Cassel et al. [64] ont développé le système BEAT ("Behavior Expression Animation Toolkit", cf. Figure 11) qui prend une entrée textuelle et génère des comportements synchronisés avec la parole comme l'intonation, le regard et les gestes iconiques. Le comportement non verbal synthétisé est attribué sur la base d'un ensemble de règles inspirées de la recherche sur le comportement conversationnel humain. Le premier module "Language Tagging" permet de faire une annotation contextuelle et linguistique de l'entrée textuelle (par ex. distinguer les verbes et les noms). En se basant sur cette première analyse, le deuxième module intitulé "Behavior Suggestion" va suggérer un ensemble de comportements pour la génération. Un exemple d'une règle de suggestion est la suivante: "If at end of utterance OR 73% of the time Suggest Gazing TOWARDS the user" ou aussi "If at beginning of utterance OR 70% of the time Suggest Gazing AWAY from user ". Ensuite le module "Behavior Selection" analyse toutes les suggestions, détermine les gestes qui sont source de conflits et ne sélectionne que les gestes qui sont compatibles entre eux et dont l'exécution physique est possible. Le module "Behavior Scheduling" transforme ensuite les comportements sélectionnés en une liste d'instructions temporelles et synchronisées prêtes pour l'animation finale.

Le système Ymir développé par Thörisson (cf. Fig.1) propose de structurer les règles de suggestion en plusieurs niveaux de traitement, du plus réactif au plus cognitif. Chaque niveau – réaction, gestion de processus et de contenu – s'appuie sur une base de règles œuvrant sur graphe d'états, chacun étant déclenché par des pré-conditions sur les observations multimodales, générant des actions spécifiques et actualisant par des post-conditions l'état du système.

MAX ("Multimodal Assembly eXpert") développé par Kopp et ses collègues [65], permet de réaliser des tâches collaboratives avec des humains dans un environnement de réalité virtuelle (cf. Figure 12). Max est en mesure de générer des actions réactives (des réflexes) et délibératives (qui nécessitent un raisonnement) via la synthèse vocale, les expressions faciales, le regard, et les gestes (cf. Figure 13). Pour guider son partenaire humain dans le tâche de construction et pour contrôler le bon déroulement de l'interaction, MAX comporte un planificateur de tâche, un planificateur de parole, un planificateur de prise de tour ainsi qu'un générateur de gestes synchronisés. MAX perçoit son environnement via un système de reconnaissance vocale et un ensemble de dispositifs visuels et tactiles pour suivre les manipulations et les gestes de son partenaire. Il dispose également d'un ensemble de mémoires (spatiale, temporelle et sémantique) permettant d'inférer les intentions de son interlocuteur. Les actions de l'utilisateur sont modélisées comme des actions intentionnelles permettant ainsi de mettre à jour les croyances de MAX sur les objectifs de son partenaire et sur l'évolution de la tâche en cours.

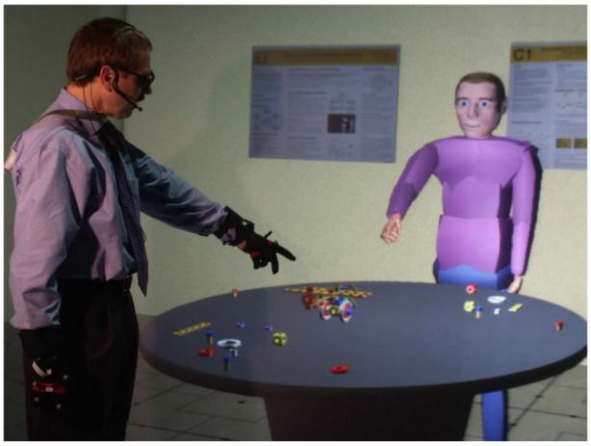


Figure 12: Une tâche de construction collaborative entre MAX et un humain dans un environnement de réalité virtuelle (figure reproduite de [65]).

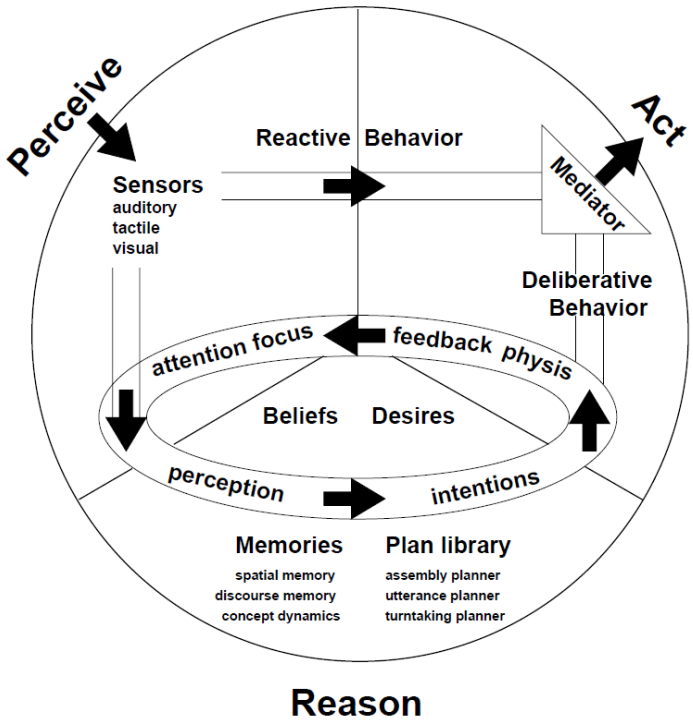


Figure 13: L'architecture de MAX (figure reproduite de [65])

Krenn [66] a introduit également le projet NECA ("Net Environment for Embodied Emotional Conversational Agents", cf. Figure 14) qui vise à développer une plateforme pour la mise en œuvre et l'animation d'agents conversationnels émotionnels pour des applications web. NECA comporte un générateur de scène permettant de définir le champ de l'application selon les préférences de l'utilisateur ("User Input"), un générateur de langage naturel permettant de transformer les actes communicatifs en un texte, un synthétiseur permettant de communiquer oralement le message avec une voix claire et une prosodie adéquate, et un module de génération permettant la planification de l'ensemble des comportements à produire. Le système produit une séquence d'évènements verbaux et non verbaux synchronisés pour tous les agents virtuels présents dans la scène. La sortie de chaque module est exprimée en RRL ("Rich Representation Language") qui a été conçu pour décrire la scène et les comportements des agents à l'issue de chaque étape. On cite deux exemples des règles utilisées dans NECA:

- Les syllabes accentuées sont le point d'ancrage pour les gestes déictiques, le lever des sourcils et les hochements de tête.
- Les positions de pause de parole sont utilisés pour la synchronisation des changements de posture, des mouvements respiratoires et des mouvement de tête.

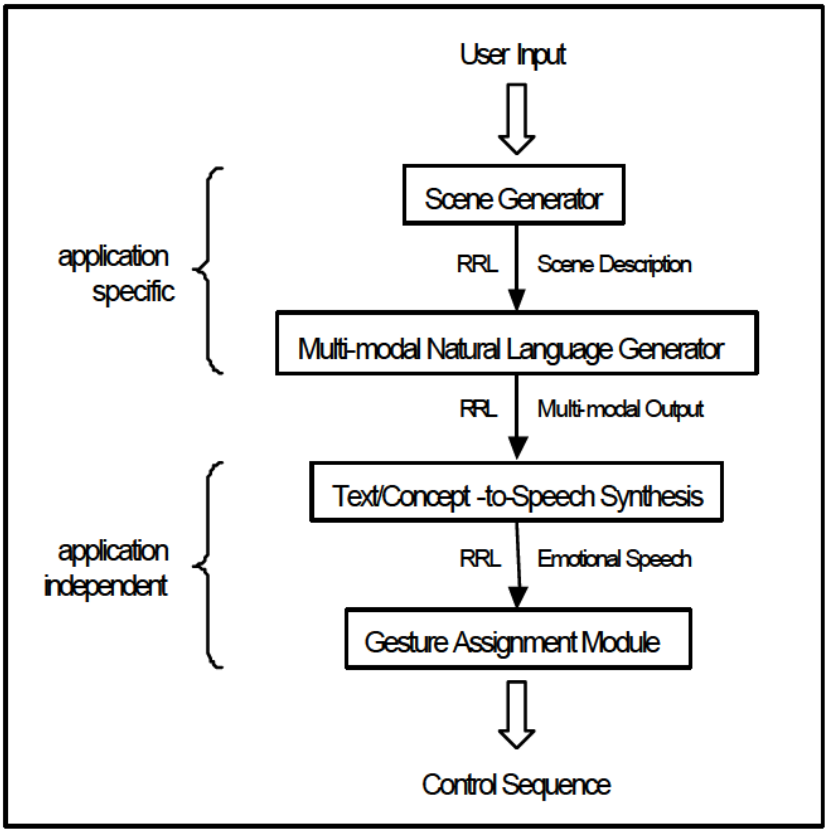


Figure 14: L'architecture de Neca (figure reproduite de [66])

Une autre contribution importante de la plateforme NECA est la création d'un répertoire d'animations et des gestes non verbaux prédéfinis – i.e. "Gesticon" [67] – permettant à la fois l'animation des supports virtuels et physiques. La structure d'une entrée "Gesticon" (au format XML) est montrée dans la Figure 15. Dans l'élément "verbatim", on trouve une description du geste. On trouve le type (iconique, déictique, etc) dans l'élément "function". L'élément "form" renseigne les propriétés physiques basiques du geste (par ex. la position). Les contraintes d'applicabilité sont mentionnées dans "restrictions" et les informations finales nécessaires pour l'animation sont mentionnées dans "playercode". Un exemple d'une entrée "Gesticon" est montrée dans l'Annexe A. Le couplage entre le "Gesticon" et le "RRL" a permis au système NECA d'avoir une représentation générique des gestes et une stratégie de génération pertinente.

```
<gesticonEntry>
      <verbatim/>
      <function/>
      <form/>
      <restrictions/>
      <playercode/>
</gesticonEntry>
```

Figure 15: La structure d'une entrée "Gesticon" (figure reproduite de [67])

Les plateformes mentionnées ci-dessus ainsi que d'autres systèmes [68] [69] présentent de nombreuses similitudes: les actions multimodales sont sélectionnées, programmées et intégrées selon des configurations enrichies par des règles à diverses étapes de traitement. La plateforme SAIBA ("Situation Agent Intention Behavior Animation") [70] a été développée pour établir une plateforme unique, unifier les normes et accélérer le progrès dans le domaine. Elle est organisée en trois composantes principales (cf. Figure 16): la planification des intentions, la planification des comportements, et la réalisation finale. SAIBA reprend le concept de Gesticon de la plateforme NECA et introduit deux nouveaux langages basés sur le XML: le BML ("Behavior Markup Language") [71] et le FML ("Functional Markup Language") [72]. Grâce à FML et BML, SAIBA vise à normaliser les flux d'information et à faire communiquer d'une manière standard les différents modules: FML représente la sortie du module "planification des intentions" et BML la sortie du module "planification des comportements". Un exemple de code BML décrivant une synchronisation entre un geste de battement, un hochement de tête et une fixation de regard envers un objet est montré dans la Figure 17. Il faut noter ici que SAIBA n'offre qu'un cadre général pour la construction de modèles de comportement, car chaque module est traité comme une "boîte noire" et c'est aux chercheurs de spécifier leurs propres modèles internes. De nombreux systèmes ont adopté le concept de SAIBA, en particulier, CADIA [73], GRETA [74] et le système SmartBody [75].

Un aspect majeur manquant à SAIBA était le traitement de la perception. Le PML ("Perception Markup Language") [2] a été récemment introduit pour combler cette lacune. Il représente une première étape vers une représentation standardisée des comportements verbaux et non verbaux perçus. PML a été inspiré par les efforts précédents dans le domaine de la génération non verbale de comportement (FML et BML) et a été conçu en synergie avec ces normes. Un exemple d'un système [2] adoptant le concept de SAIBA et ayant recours au PML, FML et BML est montré dans la Figure 18. Il s'agit d'un système de génération de comportement non verbal pour un agent conversationnel virtuel pour une application de consultation médicale. On voit que les données audiovisuelles recueillies sont exprimées en PML. Le module de gestion de dialogue exprime les intentions en FML et les comportements à générer au final sont exprimés en BML. Notons cependant un accès direct par PML à la génération de comportements non verbaux sans passer par FML: ceci permet d'implémenter des composants d'interaction réactifs qui ne passent pas nécessairement – sinon dans un second temps – par une interprétation fonctionnelle des actions d'autrui.

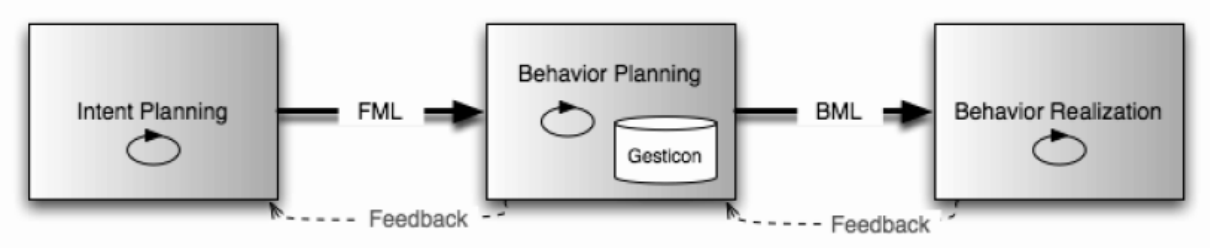


Figure 16: la plateforme SAIBA (figure reproduite de[70])

```
<bml>
  <gesture id="g1" type="beat"/>
  <head type="nod" stroke="g1:stroke"/>
  <gaze target="object1" start="g1:ready" end="g1:relax"/>
</bml>
```

Figure 17: Un exemple de code BML décrivant une synchronisation entre un geste de battement (en anglais "beat"), un hochement de tête et une fixation de regard envers un objet (figure reproduite de[70]).

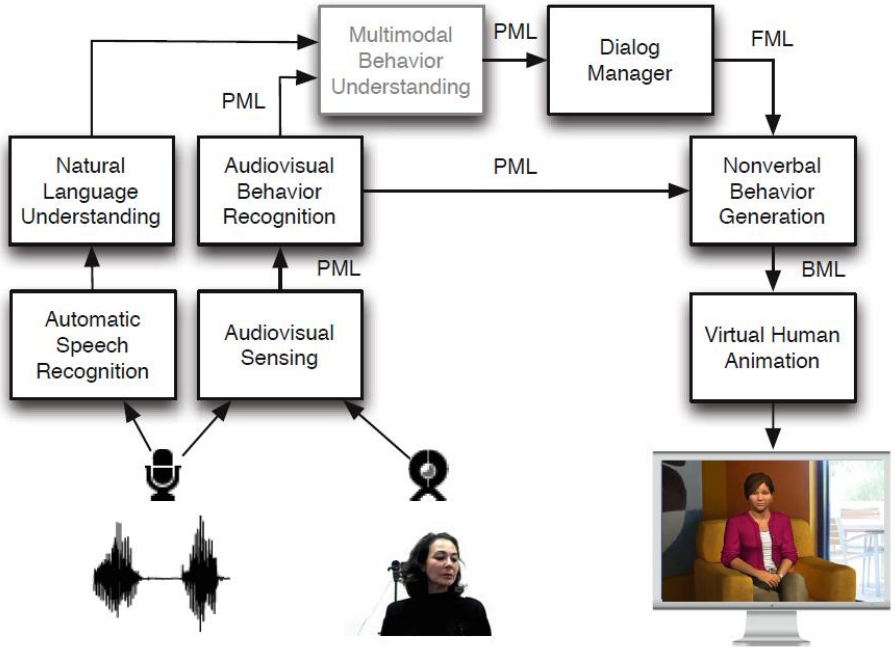


Figure 18: Un exemple de système (figure reproduite de [2]) adoptant le concept de SAIBA et ayant recours au PML, FML et BML. L'objectif était de générer le comportement d'un agent conversationnel virtuel destiné à une application médicale.

Les scénarios scriptés qui ont été disséqués ci-dessus sont tout à fait appropriés pour créer rapidement des prototypes interactifs. Néanmoins, la spécification des règles avec un langage déclaratif peut être contraignant et assez restrictif, surtout lorsque on est face à un grand nombre de comportements multimodaux ou lorsque de larges bases de données doivent être traitées [76]. Avec tels modèles, les briques interactifs résultantes sont également fortement sensibles aux décisions prises par le chercheur dans la phase de modélisation. Les méthodes basées-données ont été développées pour répondre à ces limitations. Utilisant les outils de l'apprentissage statistique, elles apprennent automatiquement les paramètres de modèles interactifs à partir des traces d'interactions préalablement démontrées par des humains.

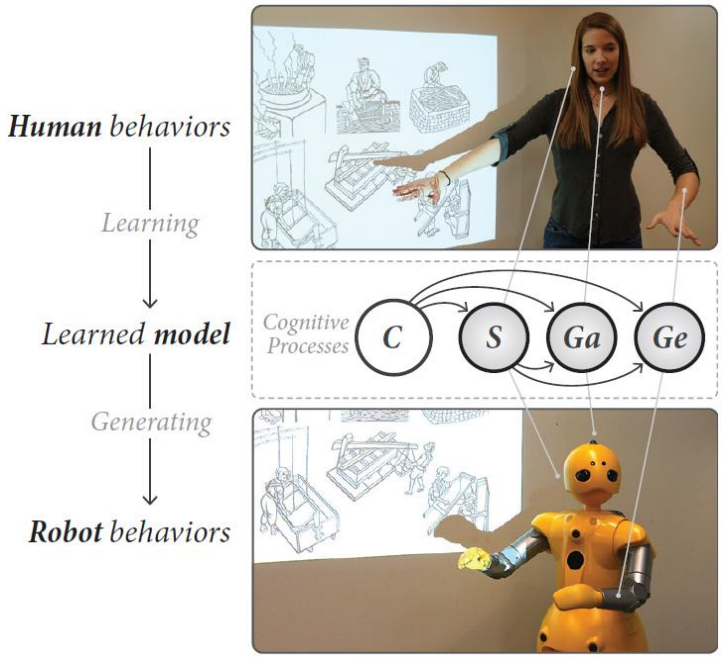


Figure 19: Une approche basée-donnée pour construire un modèle de comportement pour un robot impliqué dans une tâche de narration (figure reproduite de [76])

1.4.2 Les modèles basés-données

Ces dernières années, on remarque une tendance croissante à l'utilisation des modèles statistiques pour la génération de comportements pour les systèmes interactifs [77]. Huang et al. [76] ont ainsi étudié comment une approche fondée sur l'apprentissage statistique et destinée à la génération des comportements multimodaux pourrait répondre efficacement aux limites des modèles heuristiques. Ils ont utilisé les réseaux Bayésiens dynamiques (DBN) pour modéliser la coordination entre la parole, le regard et les gestes dans une tâche de narration (cf. Figure 19). Le modèle DBN proposé (cf. Figure 20) est constitué de 4 variables discrètes:

- C représente le processus cognitif qui contrôle la coordination entre le comportement verbal (la parole), et le comportement co-verbal (les gestes et le regard). C'est une variable latente i.e. non observable. En se basant sur des tests exploratoires, le nombre optimal d'états que peut prendre cette variable était de 3 états. La longueur moyenne de ces états – résultant de l'apprentissage automatique – est environ de 500ms, soit de l'ordre de la durée d'un mot.
- Ge représente les gestes qui sont codés en 5 événements possibles: iconique, déictique, métaphorique, battement et pas de geste.
- Ga représente le regard qui pouvait fixer 4 régions d'intérêt: la référence, le destinataire, le geste propre du narrateur ou autre.

- S: représente la parole, décrite par 12 valeurs booléennes. Chaque valeur correspond à la présence d'une caractéristique particulière, par exemple référence abstraite ("the first step"), objet concret ("two boards"), etc.

L'objectif était d'estimer C et le comportement co-verbal (Ge, Ga) à partir de l'activité verbale (S). Toutes les estimations étaient faites en mode hors ligne. Une première évaluation objective a donné un taux de génération correcte de 54% pour le regard et de 62% pour les gestes. L'implémentation de ce modèle dans le contrôle d'un robot et l'évaluation subjective par des observateurs externes a montré que cette approche basée-données bénéficiait d'avis similaires à une approche classique basée sur de règles. Cependant, cette approche permet de réduire considérablement l'effort impliqué dans la spécification du modèle et dans l'identification des motifs cachés du comportement humain.

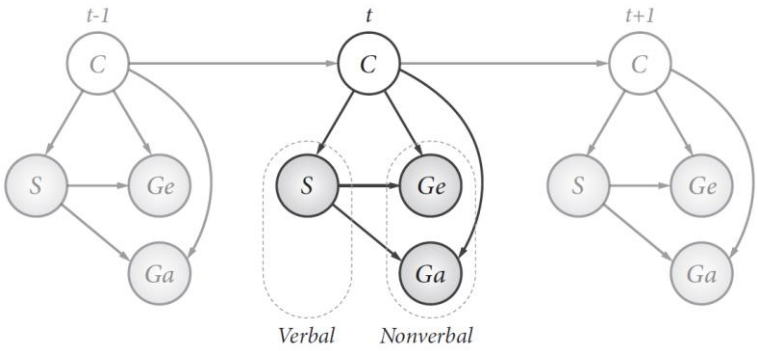


Figure 20: Le modèle DBN proposé par Huang et al. (figure reproduite de [76])

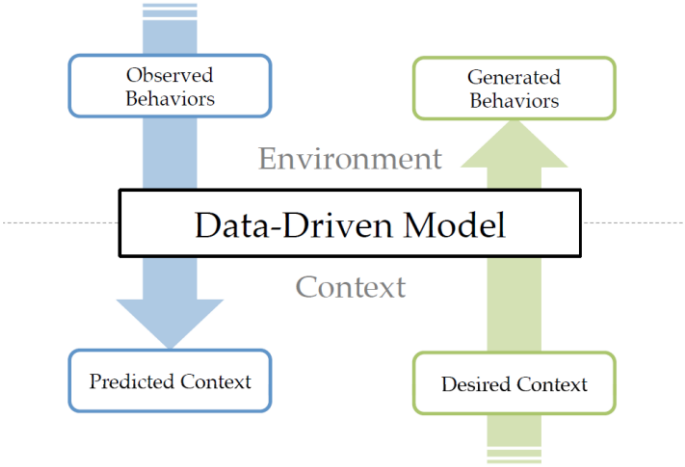


Figure 21: Le modèle proposé par Admoni et Scassellati (figure reproduite de [78]) permet de prédire le contexte d'un ensemble de comportements observés et aussi de générer des comportements selon un contexte désiré.

Admoni et Scassellati [78] ont également introduit un modèle probabiliste basé sur l'algorithme kNN ("k-Nearest Neighbors") pour la génération du comportement non verbal. Le modèle est destiné à être implémenté sur le long terme sur un robot d'assistance sociale pour une application de tutorat. En utilisant des données empiriques discrètes d'interactions homme-homme entre enseignants et élèves, le modèle peut être à la fois prédictif (c.-à.-d

reconnaître le contexte de nouveaux comportements non verbaux) et génératif (créer de nouveaux comportements non verbaux en se basant sur un contexte désiré) (cf. Figure 21). Parmi les contextes étudiés, les auteurs ont sélectionné les contextes suivants: l'énonciation d'un fait général, d'une explication, d'une question et d'une réponse. Une évaluation objective a été effectuée en ayant recours à une validation croisée sur les 10 interactions étudiées: on fait l'apprentissage sur 9 interactions et on teste sur la 10ème. Le taux moyen de la détection du contexte était de 45.9%. Pour la génération, le modèle réalise un taux moyen de 72% pour la production des gestes (7 types de gestes) et de 78% pour la génération du regard (4 régions d'intérêt).

Dans le même contexte de génération d'actions sociales, Morency et al. [79] ont opté pour des modèles probabilistes séquentiels, i.e. les HMM et les CRF ("Conditional Random Fields"), pour prédire les marqueurs phatiques d'un interlocuteur. Les modèles étaient appris à partir de traces d'interactions homme-homme en utilisant des caractéristiques multimodales extraites de l'activité du locuteur comme les mots énoncés, la prosodie et le regard. De kok et al. [80] ont présenté un modèle adaptatif pour prédire les réactions des auditeurs. Les données utilisées dans l'apprentissage des modèles sont issues de plusieurs interactions dyadiques impliquant un locuteur et un auditeur. Le locuteur avait comme tâche de résumer une petite vidéo ou aussi de décrire une recette de cuisine. Pour sa part, l'auditeur était appelé à mémoriser le maximum de détails sur les informations données. Les débuts de 886 réponses d'auditeurs ont été annotés manuellement, 90% des réponses annotées étaient des mouvements de tête et 10 % étaient des courtes vocalisations comme "okay" ou "uh-huh". L'approche proposée, qui se base sur une collection de modèles individuels (appris sur un seul locuteur), s'est avérée plus performante que le modèle de référence appris sur plusieurs locuteurs (voir notre propre analyse de ce problème de spécificité des modèles à la section 4.6.2). Tous les modèles proposés ici se basent sur les CRF. Un CRF apprend une correspondance entre une séquence d'observations – dans ce cas, les descripteurs décrivant le comportement du locuteur – et une séquence d'étiquettes – dans ce cas, l'existence ou pas de réponse de la part de l'auditeur. Les auteurs définissent un ensemble de paramètres qui caractérisent les locuteurs (par ex. la fréquence fondamentale moyenne, l'énergie moyenne, le pourcentage de temps de parole du locuteur, le pourcentage de temps où le locuteur regarde son partenaire, le nombre de saccades par minutes etc.). Dans la phase de test, lorsque on est face à un nouveau locuteur, on utilise ces paramètres pour identifier le style du locuteur le plus proche du locuteur actuel. Notons ici que la comparaison est faite grâce à la distance euclidienne de manière similaire à la méthode de l'i-vector [81] dans la reconnaissance des locuteurs. Une fois le style de communication le plus adéquat sélectionné, on utilise le modèle CRF individuel correspondant pour estimer les temps de réaction de l'interlocuteur. Dans une approche un peu similaire, en fonction de profils de gestes déduits d'annotations de comportements multimodaux, Neff et al. [82] ont proposé de générer des styles de gestes spécifiques capturant ainsi la variabilité comportementale entre plusieurs locuteurs humains. Le système prend en entrée des textes arbitraires et produit des gestes de conversation synchronisés avec le style d'un locuteur particulier.

Lee et al. [83] ont utilisé également une approche probabiliste pour générer les hochements de tête et les mouvements de sourcils pour un agent virtuel. Afin d'apprendre la dynamique de ces comportements, les auteurs ont exploré différents ensembles de caractéristiques et plusieurs algorithmes d'apprentissage, à savoir HMM, CRF et LDCRF ("Latent Dynamic CRF"). Le modèle LDCRF [84] se distingue d'un modèle CRF par une couche de variables latentes entre la couche des observations et celles des états à prédire. Ceci permet au modèle LDCRF d'avoir une structure interne suffisamment riche pour capter la variabilité des comportements à générer. Les données utilisées sont celles du corpus AMI. Les caractéristiques utilisées dans l'apprentissage sont de plusieurs natures: des informations de nature syntaxique (par ex. début d'une phrase, modalité finale de l'énoncé, etc.), un étiquetage des interventions en actes de dialogue, des caractéristiques paralinguistiques (par ex. des rires ou toute autre type de vocalisation non verbale) et d'autres descripteurs sémantiques. L'évaluation quantitative a montré que le modèle LDCRF réalise les meilleures performances, soulignant ainsi l'importance d'apprendre la dynamique et l'organisation interne sous-jacente des gestes étudiés dans cette application (les hochements de tête et les mouvements de sourcils).

Pour la génération du regard, Lee et Badler [85] ont mis en œuvre un modèle de mouvement des yeux basé sur des modèles empiriques de saccades et de modèles statistiques issus de données de suivi de regard. Ils ont observé notamment que les comportements du regard diffèrent selon que le sujet parle ou écoute et ils ont utilisé cette distinction dans leur modélisation. Notons que Bailly et al. [86] ont généralisé ces signatures de trajectoires oculaires à un ensemble plus important d'états cognitifs lors de vraies interactions dyadiques. Lee et Badler ont animé un visage virtuel en utilisant 3 différents types de mouvements oculaires: fixes, aléatoires, et basés sur le modèle proposé. L'évaluation subjective a montré que les mouvements oculaires générés par le modèle développé donnent un aspect plus naturel, amical et sociable. Plusieurs autres modèles utilisant l'apprentissage statistique existent dans la littérature [87]–[90].

Plus généralement, ces approches d'apprentissage utilisent fréquemment des modèles probabilistes graphiques en raison de leur capacité à fournir une représentation probabiliste de la dynamique du comportement humain (cf. Chapitre 4 et Chapitre 5), de modéliser les relations complexes entre évènements multimodaux et de prendre des décisions maximisant un critère objectif. Il est important de noter aussi que plusieurs approches hybrides ont été proposées [91] [92]. La plupart de ces approches utilisent des modèles statistiques pour la génération et des heuristiques pour affiner et mieux caractériser les comportements synthétisés.

1.5 Les challenges du SSP

La modélisation, l'analyse et la synthèse des signaux sociaux sont des problèmes loin d'être résolus et présentent plusieurs défis à surmonter dans les années à venir [93].

Le premier défi concerne la collecte, l'annotation et le partage des données [93]. En effet, la collecte des données est une étape importante sans laquelle il est impossible de

modéliser efficacement les interactions sociales. La collecte de nouveaux corpus de données s'est diversifiée et accélérée ces dernières années couvrant un large spectre de scénarios et de signaux. Néanmoins, la plupart de ces corpus ne traitent pas la totalité des phénomènes liés à l'interaction sociale multimodale et se limitent à des contextes et configurations particulières. La question d'avoir à la fois un scénario générique et un corpus riche de données reste encore ouverte. Vaut-il mieux opter pour une interaction ouverte et naturelle (par ex. un débat télévisé) ou plutôt un scénario contraint enregistré dans un laboratoire avec un dispositif technique et matériel lourd? Le premier type d'interactions permet de conserver la véracité des comportements en dépit de la qualité et quantité exploitable des données, alors que le deuxième type compromet la spontanéité des comportements tout en fournissant des données riches et massives sur une tâche restreinte en contrôlant a priori les variables indépendantes. D'autres problèmes existent, notamment l'annotation des données multimodales qui demeure une tâche laborieuse et sujette à débat, notamment auto-annotation par les interlocuteurs vs. annotations consensuelles par experts. On est également confronté au problème de partage des données (à cause des contraintes de propriétés intellectuelles, des restrictions éthiques, ou autre) ce qui altère le développement du domaine.

Le deuxième défi concerne l'analyse des comportements [11]. Comme déjà évoqué, plusieurs travaux ont tenté de relever ce challenge avec des méthodes et des objectifs divers. Par contre, certains points restent encore problématiques, notamment le choix des modalités et de la meilleure manière de les fusionner. En effet, il a été démontré que l'intégration de plusieurs modalités (visuelles et acoustiques) produit des résultats nettement meilleurs que les approches unimodales. Cependant, il n'existe pas encore une étude précise sur les meilleures modalités à utiliser selon le signal social à détecter (actions/émotions/attitudes/relations) ainsi que la manière optimale de les fusionner. Un autre problème dans l'analyse des comportement non verbaux est leur ambiguïté, car nous ne pouvons pas toujours les associer d'une manière automatique à un sens particulier. Par exemple, les distances physiques reflètent généralement les distances sociales mais parfois, ce n'est que l'effet de certaines contraintes spatiales. De plus, l'interprétation de ces comportements est fortement sensible au contexte et à la culture [94]. Certains suggèrent [15] que la meilleure solution est d'appréhender le problème comme un seul problème complexe et d'éviter la division en sous tâches (détection des comportements, détection du contexte, etc.).

Le troisième défi concerne la synthèse des comportements. Un être humain qui interagit avec un agent artificiel produit des signaux sociaux et attend de l'agent des réponses à ses signaux. L'agent doit alors non seulement percevoir et comprendre ces signaux mais aussi émettre des signaux qui affirment sa présence et contribuent à atteindre ses objectifs. On retrouve certains points problématiques déjà évoqués pour l'analyse des comportements notamment la relation complexe entre un comportement et le sens qu'il véhicule. Afin de clarifier la signification des comportements synthétisés, il est important pour un agent artificiel de prendre en considération plusieurs éléments contextuels et culturels même si cette tâche est loin d'être facile pour un ordinateur. La perception d'un geste démontré par un tuteur humain puis exécuté par un avatar est souvent bien éloignée de l'intention originale: la mise à l'échelle des gestes aux degrés de liberté physiques du robot et les a priori que les partenaires

humains projettent sur les technologies numériques biaisent fortement leur évaluation perceptive [95]. Certains suggèrent [93] que le robot doit être conscient de ses limites technologiques. Afin de remédier à des éventuelles ambiguïtés, il doit privilégier des comportements simples et clairs sous condition qu'ils restent suffisamment riches pour atteindre les objectifs communicatifs. Au contraire, d'autres études venant de l'industrie du dessin animé [96] montrent que, pour être crédible, les personnages doivent montrer un comportement fortement exagéré. Ceci suggère de trouver le bon compromis afin d'avoir un comportement à la fois clair, naturel et crédible. Dans le même contexte, Mori [97] a introduit la théorie de la vallée de l'étrange ("uncanny valley"). Cette théorie, bien que contestée récemment par plusieurs études [98], suggère que, plus un robot ressemble morphologiquement à un être humain, plus nous sommes exigeants vis à vis de son comportement. Etant donné les imperfections actuelles dans la génération des comportements sociaux, l'homme a tendance à mieux accepter un robot humanoïde qu'un robot androïde [99]. Avec une faible ressemblance à l'humain, un robot humanoïde reste toujours une machine et suscite même une certaine empathie, alors qu'un androïde est jugé comme un humain et suscite une sensation étrange face à son comportement artificiel et anormal.

Similairement à l'homme, un agent artificiel doit être capable d'adapter son comportement vis à vis de son interlocuteur en se basant sur leur historique conjoint d'interaction. Cette adaptation va permettre à l'agent de développer une relation durable avec son interlocuteur. Cela implique une capacité à mémoriser et à apprendre d'une manière incrémentale au vu de l'expérience interactive passée. L'apprentissage incrémental, l'importance donnée à l'expérience récente sont des verrous scientifiques et technologiques qui sont encore loin d'être matures. Des recherches futures doivent exploiter le potentiel de l'apprentissage incrémental afin d'arriver à des modèles de comportement continuellement adaptatifs. L'inférence – et donc la génération de comportements – doit elle-même être aussi incrémentale et continue pour ouvrir la porte aux applications de temps réel et surtout pour ne pas tomber dans une attitude négative d'ignorance sociale [19]. En effet, une large part des modèles de génération actuels ne sont évalués qu'en mode hors ligne. D'autre questions de génération restent encore en cours d'étude telle que : quand produire tel signal, comment le produire, quelles seront les modalités sollicitées, comment s'effectue la coordination, etc. Il est important de noter aussi que dans l'état de l'art actuel, les mouvements générés en rejouant des enregistrements effectués par des systèmes de capture de mouvements sont de loin plus naturels et fluides que les comportements des systèmes artificiels [100].

De nouveau challenges s'ouvrent et une nouvelle dimension de complexité s'ajoute pour la génération du comportement si on passe de l'interaction dyadique à l'interaction multi-parties [101]. Un agent artificiel engagé dans une interaction multi-partie doit avoir des compétences additionnelles en ce qui concerne l'analyse de la scène notamment la diarisation, la gestion de dialogue, la gestion de l'engagement et du désengagement, la gestion des prises de tour de parole et de l'adressage. L'agent doit être également capable d'identifier les sources et les destinations des signaux sociaux et surtout de gérer l'attention mutuelle avec plusieurs partenaires dont les objectifs et les états mentaux sont différents.

Dans de nombreux modèles d'interaction, les échanges sont encore modélisés d'une façon rudimentaire. Le comportement social généré n'est pas issu d'une représentation riche et profonde de l'agent et des interlocuteurs mais plutôt d'un ensemble de règles scriptés ou appris automatiquement. Selon Tomasello [102], l'interaction homme-homme peut être représentée par plusieurs niveaux allant de l'attention jointe jusqu'aux objectifs joints ou ce qu'on appelle aussi l'intentionnalité jointe. Dans la modélisation proposée par Tomasello et al, l'intentionnalité jointe représente le niveau le plus élevé et joue un rôle primordial dans la réussite d'une interaction sociale. Elle exige de nous la capacité à former un modèle efficace des états cognitifs des personnes qui nous entourent et donc la capacité de lire les esprits des autres [103] en décodant les signaux "overt" et "covert" produits pendant la conversation. Ceci suggère de recentrer de plus en plus les approches sur l'étude de la cognition humaine et de privilégier le développement des modèles computationnels qui s'en inspirent [93].

1.6 Conclusions

Dans ce chapitre nous avons présenté un aperçu sur les travaux et les approches qui traitent la modélisation, la synthèse et la génération des comportements non verbaux (par ex. postures, gestes, regard, prosodie, etc.). Ces thématiques relèvent du traitement des signaux sociaux (SSP) que l'être humain échange dans ses interactions sociales quotidiennes. Les signaux véhiculés peuvent être des actions sociales (par ex. la prise de tour), des émotions (par ex. l'empathie), des attitudes (par ex. la dominance) et des relations sociales (par ex. les rôles), voire des traits de personnalité [104]. Le SSP vise à doter les systèmes interactifs de l'intelligence sociale qui permettra la perception correcte, l'interprétation précise, et l'affichage approprié des différents types de signaux sociaux. Plusieurs modèles de comportement social et multimodal ont été présentés, notablement les modèles scriptés et les modèles basés-données. Le SSP n'est pas encore mature et plusieurs problématiques liées aux données, à l'analyse contextualisée des scènes, aux modèles de génération adaptatifs et incrémentaux, à la coordination multimodale et aux modèles cognitifs d'interaction, doivent être résolues par la communauté dans les travaux futurs. Dans les prochains chapitres, nous présentons deux bases de données ainsi que plusieurs modèles de comportements basés-données avec lesquels nous essayons de répondre à certaines problématiques d'analyse, de caractérisation et de génération de comportement co-verbal et donc de contribuer à donner plus de maturité au domaine particulier de la génération des signaux sociaux en situation d'interaction finalisée. Les modèles de comportements que nous proposons présentent plusieurs avantages: d'abord, ils sont basés sur l'apprentissage automatique et la modélisation statistique afin de coupler intrinsèquement la perception à la compréhension et à l'action. Ils organisent les séquences de percepts et d'actions en ce que nous appelons des comportements sensori-moteurs multimodaux. Deuxièmement, l'analyse et la génération sont effectuées d'une manière incrémentale, ce qui rend nos modèles utilisables pour des applications en temps réel. Troisièmement, nous montrons que nos modèles capturent des régularités et des propriétés caractérisant le comportement humain qui échappent souvent à l'expertise humaine et qui ne sont pas généralement évaluées dans les modèles existants dans l'état de l'art.

Chapitre 2 Interactions sociales

2.1 Introduction

L'approche suivie pour développer les modèles de comportement est une approche basée sur les données – data-driven. A partir de vraies traces d'interaction, nous allons utiliser l'apprentissage statistique pour construire des modèles de comportement multimodal. Grâce aux techniques d'apprentissage, ces modèles seront capables de capturer les co-variations subtiles entre les modalités du comportement et donc de capturer la coordination entre les événements multimodaux. Cette micro-coordination échappe souvent à l'expertise humaine, ce qui justifie en partie l'intérêt de cette méthodologie statistique par rapport à d'autres modèles à base de règles.

Dans ce chapitre, nous présentons deux interactions face-à-face qui ont été conçues afin d'étudier le comportement humain dans une interaction située, ainsi que modéliser ses boucles de perception et d'action. Ces deux interactions sont basées sur deux scénarios de jeux : un jeu de répétition de phrases et un jeu de cubes. Simples à première vue, ces deux interactions sont suffisamment riches et complexes pour répondre à notre objectif de modéliser et détecter les régularités dans les comportements joints. En outre, ils permettent de collecter en un minimum de temps de nombreuses répétitions d'unités de perception et d'action . Ceci permet d'être en capacité de caractériser la variabilité effective du couplage intra- et inter-modalités de perception et d'action des interlocuteurs. De plus, notre évaluation ne se limitera pas aux seuls taux de classification mais sera aussi une évaluation approfondie des interactions générées ainsi que leurs propriétés. Une fois que la modélisation et la fouille de ce type de données atteignent des performances acceptables sur plusieurs niveaux d'évaluation, il sera pertinent à ce moment de passer à des ensembles d'observations sensori-moteurs plus grands et, alors, plus complexes.

2.2 La première interaction: Jeu de répétition de phrases

2.2.1 Scénario

Ce scénario a été conçu et mis en œuvre dans le cadre de la thèse de S. Raidt [105]. Elle est inspirée par les travaux de Kendon [106], Argyle & Cook [107] et d'autres, dédiés à l'analyse du comportement du regard dans le dialogue et l'interaction sociale. Elle consiste en un jeu de répétition de phrases impliquant deux personnes : un sujet principal et plusieurs partenaires (10 en total). Les phrases choisies (20 au total) sont des phrases sémantiquement incompréhensibles (SUS) [108]. Elles sont grammaticalement correctes mais sans aucun sens valide (par ex: "la toile souffle le rateau qui pleure"). Cela oblige les sujets à être très attentifs aux signaux audio-visuels lors de l'écoute du locuteur, puisque les différents constituants (sujet, verbe…) n'aident pas à anticiper les éléments suivants (verbe, complément, qualificatif…). Les phrases étaient fournies sur un support papier: le locuteur lit, mémorise et énonce à chaque fois une seule phrase à son partenaire. L'interlocuteur est invité alors à

mémoriser et répéter exactement la phrase énoncée en une seule tentative si c'est possible. Après 10 phrases, il y a une inversion des rôles. Chaque sujet va être donc locuteur pour 10 phrases et interlocuteur pour 10 autres phrases. Pour motiver les sujets, ils étaient informés que le score des phrases répétées avec succès sera calculé.

Le but de maintenir un seul sujet principal en variant ses partenaires est d'acquérir un nombre important de données sur un seul sujet, permettant ainsi de développer un modèle imitant le comportement adaptatif de cette personne. En outre, un tel scénario va générer un parcours assez répétitif de l'interaction, ce qui va permettre de produire une quantité significative d'informations sur les segments de cette interaction et les comportements à modéliser, notamment celui du regard. L'interaction décrite était médiatisée par ordinateur afin de pouvoir capturer les comportements multimodaux des deux partenaires. Le dispositif expérimental est décrit avec plus de détails dans le paragraphe suivant.

2.2.2 Dispositif expérimental

Afin d'enregistrer l'interaction dyadique décrite précédemment, une plate-forme expérimentale a été mise en place (Figure 22). Grâce à cette plateforme, les deux sujets peuvent interagir à travers deux écrans pour leur donner l'impression que l'un est face à l'autre. Pour ceci, une petite caméra à sténopé (i.e. "pinhole camera" fonctionnant sur le principe de la camera obscura) est placée au centre d'un écran pour filmer le sujet qui est en face. Cette vidéo est alors affichée sur l'écran tourné vers l'interlocuteur, qui est équipé de manière symétrique. Les signaux audio sont échangés via des microphones et des écouteurs. Les signaux vidéo et audio ainsi que les directions du regard sont enregistrés au cours de l'interaction. Pour cela, deux écrans Tobii® intégrant des oculomètres, ont été utilisés.

Figure 22: Dispositif expérimental

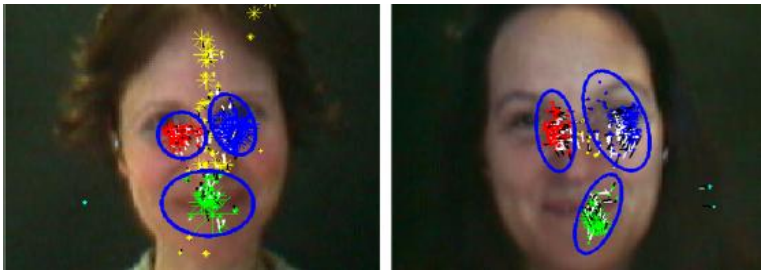


Figure 23: Détermination des régions d'intérêt pour le regard

2.2.3 Unité d'interaction

Comme proposé par Ford [109], chaque interaction de type homme-homme (ou aussi homme-machine) peut être segmentée en unités d'interaction (IU pour "Interaction Unit"), par exemple penser, informer, écouter, prendre le tour, etc. Dans notre modélisation, les IU sont annotées et labélisées en fonction des diverses tâches mentales – supposées conjointes – des deux sujets. Le séquencement de ces IU fournit une sorte de syntaxe ou de grammaire de comportements élémentaires. Le nombre, la longueur et l'ordre de ces IU dépendent alors de la tâche abordée dans l'interaction étudiée. Pour mieux comprendre le concept, une IU donnée peut être considérée comme une instance des états cognitifs conjoints des dyades de l'interaction. Par exemple si l'IU est "écouter ", cela suppose que l'interlocuteur est en train de parler alors que le sujet principal est en train de l'écouter. Un concept similaire a été utilisé par [110] dans lequel les auteurs proposent un modèle de regard (basé sur le modèle bien connu de Rickel [111] [112]) guidé par les opérations cognitives d'un agent virtuel. L'approche dans la modélisation du regard dans le modèle de Rickel suppose que le regard est étroitement lié aux opérations cognitives de l'agent et qu'il faut utiliser les événements critiques du regard pour segmenter l'interaction en un ensemble d'IU pertinentes. Ces opérations cognitives peuvent inclure la perception des événements, la mise à jour des croyances, la compréhension de la parole, la planification, etc. Dans notre modélisation, une unité d'interaction (IU) conditionne le couplage sensori-moteur entre les partenaires de la conversation étudiée : elle contextualise la façon dont les partenaires devraient à la fois signaler leur volonté d'initier, de réglementer, accepter, détecter ou d'abandonner des échanges d'informations. La réalisation d'une unité élémentaire d'interaction peut exiger le séquençage de plusieurs états sensori-moteurs (comme regard mutuel, hochement de tête, lecture labiale, gestes de battement, etc.). Les IU propres à l'interaction présentée ci-dessus ainsi que les signaux enregistrés sont présentés dans le prochain paragraphe.

2.2.4 Données

Dans chaque interaction, les séquences des SUS ont été subdivisées et annotées d'une manière semi-automatique en 7 unités d'interaction (à partir des fixations oculaires et des activités vocales des deux interlocuteurs) : lire, se préparer, parler, attendre, écouter, réfléchir et autre. Par rapport au comportement du regard enregistré, Raidt [105] a démontré que ces IU sont significativement différentes les unes des autres. Les fixations sont identifiées à partir des données brutes du regard par le biais d'un algorithme spécifique. Après avoir compensé les mouvements de la tête, toutes les fixations sont projetées sur l'écran de l'interlocuteur (Figure 23). Des régions elliptiques ont été définies par l'expérimentateur pour affecter les fixations à différentes régions d'intérêt (ROI), ce qui a donné 5 régions : œil gauche, œil droite, la bouche, le visage (autres parties que les trois précédentes, par exemple le nez) et autre (quand une fixation frappe d'autres parties de l'écran). En ce qui concerne la parole, elle a été alignée automatiquement avec les transcriptions phonétiques des phrases SUS. En résumé, 5 variables discrètes ont été extraites :

- IU : lire, se préparer, parler, attendre, écouter, réfléchir et autre

- Activité vocale du sujet principal: v_1 (pas de parole/parole)
- Activité vocale de l'interlocuteur: v_2 (pas de parole/parole)
- Regard du sujet principal: g_1 (œil gauche/œil droite/bouche/visage/autre)
- Regard de l'interlocuteur: g_2 (œil gauche/œil droite/bouche/visage/autre)

2.2.5 Modélisation

Dans notre modélisation nous allons nous intéresser à la reconnaissance de l'IU à partir de (v_1, v_2, g_2), et ensuite la génération du comportement co-verbal (le regard g_1). En effet, les modèles que nous allons proposer devraient être en mesure pour un sujet principal d'estimer d'abord l'unité d'interaction (IU) à partir d'observations perceptuelles (v_1, v_2, g_2) et deuxièmement de générer des actions appropriées (le regard g_1) qui reflètent l'unité interactionnelle courante et sa prise de conscience de l'évolution actuelle du plan ou de la tâche partagée. Nous rappelons que nous nous intéressons à la génération de la coverbalité, raison pour laquelle la variable v_1 (parole du sujet principal) est considérée comme une observation perceptuelle. La deuxième interaction (Jeu de cubes) que nous présentons dans la section suivante a été conçue de manière à être plus riche en termes de comportement co-verbal à synthétiser. Comme nous allons le voir, l'objectif sera de générer, non seulement les fixations oculaires du sujet principal, mais aussi ses gestes brachio-manuels déictiques.

2.3 La deuxième interaction : Jeu de cubes

2.3.1 Scénario

L'objectif de cette deuxième interaction face-à-face, que nous avons conçue et mise en œuvre dans le cadre de cette thèse, est de modéliser le comportement humain observé dans une tâche de type "mets-ça là" [113] . Cette tâche - simple à première vue - est une référence très intéressante pour l'étude des stratégies humaines multimodales utilisées pour maintenir l'attention mutuelle et coordonner la deixis d'objets et de lieux. Dans cette interaction, nous nous intéressons aux gestes co-verbaux, notamment aux gestes brachio-manuels et au regard.

Concrètement, le jeu implique un instructeur humain - dont on veut apprendre le comportement - et un manipulateur humain (Figure 24). L'instructeur doit faire reproduire une configuration cible d'un ensemble de cubes par le manipulateur, qui seul a la possibilité de s'en saisir et déplacer. La configuration cible n'est connue et accessible qu'à l'instructeur. Par conséquent, instructeur et manipulateur doivent interagir pour réaliser cette tâche de manière collaborative.

Chaque cube est marqué par un symbole coloré sur sa face supérieure. On dispose de quatre symboles (carré, croix, cercle et point) et de quatre couleurs (rouge, vert, bleu et noir), ce qui fait un total de 16 cubes. La table de jeu comprend trois zones, comme représenté sur la Figure 25:

1. Un damier cible où est affichée la configuration cible qui est composée de 10 cubes (Figure 26). Le plan de montage de ce damier est délivré au fur et à mesure par une tablette située devant l'instructeur. A chaque étape, une seule consigne de positionnement de cube est révélée sur la tablette et une fois que ce cube est correctement déplacé sur la table d'interaction, l'instructeur revient à la tablette pour obtenir l'instruction suivante. Ce plan de montage est généré par un algorithme spécifique afin de maitriser la variabilité des orientations, des symboles et des couleurs et aussi de contraindre l'instructeur et le guider. Seul l'instructeur est capable de voir ce plan.
2. Un damier tâche (d'une forme 5x5) où doit être reproduite la configuration cible.
3. L'espace manipulateur, où les cubes sont exposés face au manipulateur.

Au départ de chaque jeu, le damier tâche est vide et une nouvelle configuration cible est donnée à l'instructeur d'une manière progressive. L'instructeur doit faire garnir par le manipulateur le damier tâche avec les cubes, reproduisant ainsi la configuration cible. Lorsque un nouveau cube est positionné sur le damier cible pour l'instructeur, ce dernier doit donc le chercher dans l'espace manipulateur, puis demander au manipulateur de saisir ce cube en le désignant par son symbole coloré et de le placer à une certaine position du damier tâche. Les conventions d'espace sont "à gauche", "à droite", "devant" ou "au-dessous" et "derrière" ou "au-dessus" d'un cube déjà placé sur le damier tâche. Le damier tâche étant vidé au préalable de chaque jeu, le premier placement s'effectue au centre du damier. La seule désignation verbale des cubes et des placements permet au manipulateur d'effectuer la tâche mais le coût cognitif est élevé. On s'attend donc à ce que ces instructions verbales soient accompagnées de gestes co-verbaux (tête, regard et main) iconiques ou déictiques (par ex. pouce vers la bonne direction) qui facilitent le paramétrage du "mets-ça là".

Figure 24: Exemple d'une interaction, filmée par la caméra de scène montée sur la tête de l'instructeur

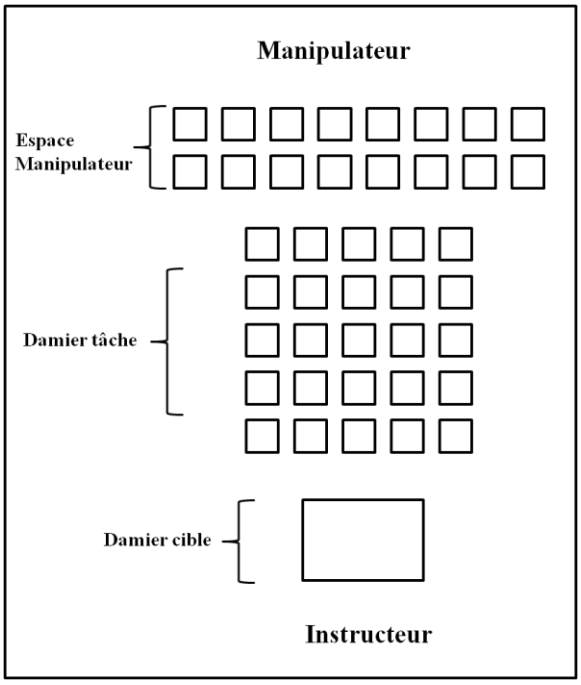
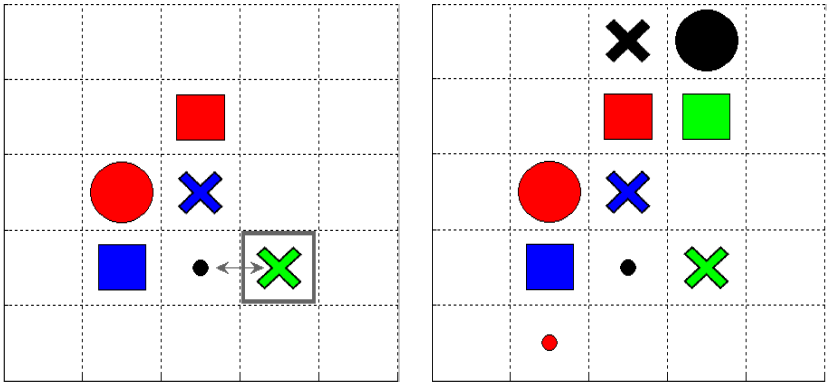


Figure 25: Table de jeu



(a) (b)

Figure 26: Exemple d'un jeu en cours (a) l'indication affichée à l'instructeur pour la transmettre au manipulateur est: "mets la croix verte à droite du point noir " (b) La configuration cible finale après plusieurs autres manipulations

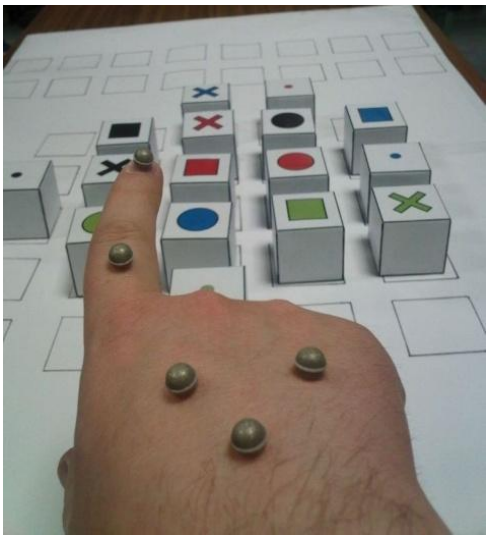
2.3.2 Dispositif expérimental

L'objectif de cette expérimentation est d'inférer un modèle de comportement pour l'instructeur en utilisant uniquement ses observations égocentriques. Ce choix va limiter la quantité d'information recueillie mais reste indispensable si on veut transférer le modèle appris à un robot humanoïde semblable à l'être humain en terme de perception et d'action. D'autant plus, par ce choix, nous essayons de développer un modèle qui ne permet pas l'analyse exhaustive et automatique d'une scène mais plutôt un modèle qui permet d'agir intelligemment pour mieux percevoir et comprendre, c'est ce que l'on appelle la perception active. Par conséquence, le manipulateur n'est pas équipé de capteurs et l'analyse de la scène est faite seulement du point de vue de l'instructeur. Les mouvements de l'instructeur sont capturés d'une manière synchrone par :

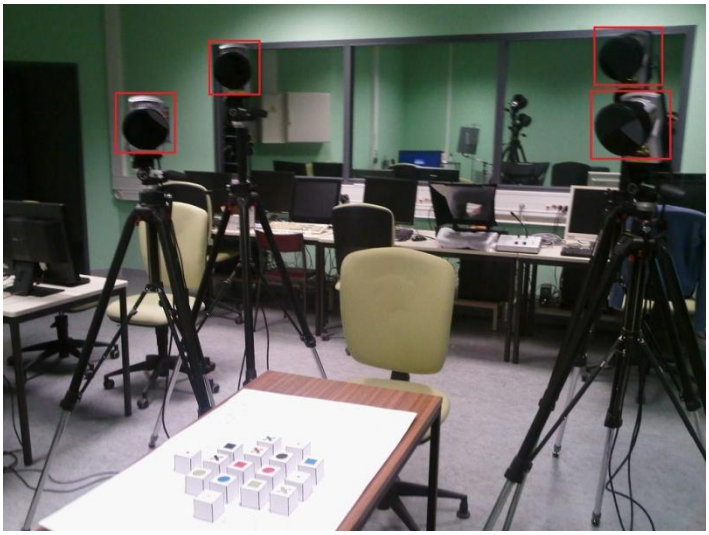
- Un système de capture de mouvements Qualysis® (MoCap). Cinq marqueurs ont été placés sur le casque de l'instructeur pour capturer les mouvements de tête. Cinq autres marqueurs ont été placés sur la main pour capturer ses gestes manuels. Dans cette expérience, le système MoCap utilisait quatre caméras optiques infrarouges, toutes étant en face de l'instructeur.
- Un oculomètre monoculaire Pertech® monté sur la tête et composé de deux caméras. Une caméra dirigée vers l'œil pour détecter ses mouvements et une caméra de scène pour filmer ce que l'instructeur perçoit.
- Un microphone monté sur la tête pour enregistrer la parole de l'instructeur.

Par souci de vérification et d'annotation de notre vérité terrain, nous avons également équipé l'environnement avec une caméra montée sur le plafond afin d'avoir accès à la scène complète. Un chronomètre est également placé dans le champ visuel des caméras de plafond et de scène afin de marquer et synchroniser avec précision les différents flux vidéos.

(a)



(b)



(c)

Figure 27: (a) Casque avec 5 marqueurs Qualisys + Oculomètre Pertech + Microphone (b) les 5 marqueurs de la main (c) les 4 cameras de Qualisys orientées vers l'instructeur

2.3.3 Données

Nous avons enregistré 30 jeux dans lesquels l'instructeur interagissait avec trois partenaires différents (10 jeux avec chacun). Chaque jeu consiste à placer 10 cubes dans le damier tâche initialement vide, en partant d'un espace manipulateur rempli de cubes (16 cubes en total). La durée moyenne d'un jeu est d'environ 1 minute et 20 secondes (~ 2 000 trames, 40 ms par trame). La durée totale des 30 jeux disponibles est de 45 minutes. Pour modéliser notre interaction, cinq variables discrètes ont été annotées semi-automatiquement:

- IU : comme dans la première interaction, nous supposons que les tâches cognitives sous-jacentes suivent une syntaxe organisée en unités interactionnelles (IU). Nous annotons et distinguons 7 IU différentes:
    1. s'informer: obtenir l'instruction de la tablette
    2. chercher: chercher le cube qui doit être placé
    3. pointer: pointer le cube à déplacer dans l'espace manipulateur
    4. indiquer: indiquer la position dans le damier tâche
    5. vérifier: vérifier la manipulation
    6. valider: valider la manipulation
    7. autre : autre activité
- MP : gestes de manipulation, nous distinguons cinq événements :
    1. position initiale (ini)
    2. saisir le cube (grasp)
    3. placer le cube (manip)
    4. fin de la manipulation (end)
    5. autre (else)
- SP : discours de l'instructeur avec 5 événements :
    1. cube à placer (cube)
    2. position comme devant, derrière, etc. (loc)
    3. cube de référence (ref)
    4. pas de parole (none)
    5. autre (else)
- GT : région d'intérêt pointé par l'index de l'instructeur avec 5 événements
    1. position initiale (ini)
    2. cube à déplacer (cube)
    3. position (loc)
    4. cube de référence (ref)
    5. autre (else)
- FX : fixations de l'instructeur avec huit régions d'intérêt :
    1. visage du manipulateur (face)
    2. espace manipulateur (ms)
    3. damier tâche (tks)

4. cube à placer (cube)
5. position (loc)
6. cube de référence (ref)
7. tablette (tablet)
8. autre (else)

MP et FX sont annotées manuellement en utilisant la vidéo de la camera de scène de Pertech. SP est détectée automatiquement par un système de reconnaissance vocale basé sur les HMM. GT est annotée semi-automatiquement en utilisant les signaux du MoCap Qualysis et la vidéo de la camera de scène de Pertech. Finalement, IU est annotée manuellement en se basant sur toutes les autres variables déjà annotées. Le logiciel d'annotation utilisé est ELAN [114] (cf. Figure 28). Dans les deux figures qui suivent (Figure 29 et Figure 30), nous montrons à titre d'exemple les distributions des différentes observations (MP,SP,GT et FX) pour deux IU différentes ("s'informer" et "pointer"). On voit que les observations sont bien cohérentes avec les IU correspondantes et que chaque IU dispose d'une signature propre. Par exemple, dans l'IU "s'informer" les observations dominantes sont : pour le manipulateur position initiale et fin de manipulation (ini et end), pour l'instructeur pas de parole (none), geste en position de repos (ini) et regard fixé sur la tablette (tablet). Pour l'IU "pointer" les observations dominantes sont : pour le manipulateur position initiale (ini), pour l'instructeur prononciation du cube à déplacer ou aussi pas de parole (cube et none), geste déictique vers le cube (cube) et regard fixé sur le cube (cube). Ces deux exemples montrent la pertinence de notre annotation semi-automatique ainsi que la variabilité contrainte des réalisations.

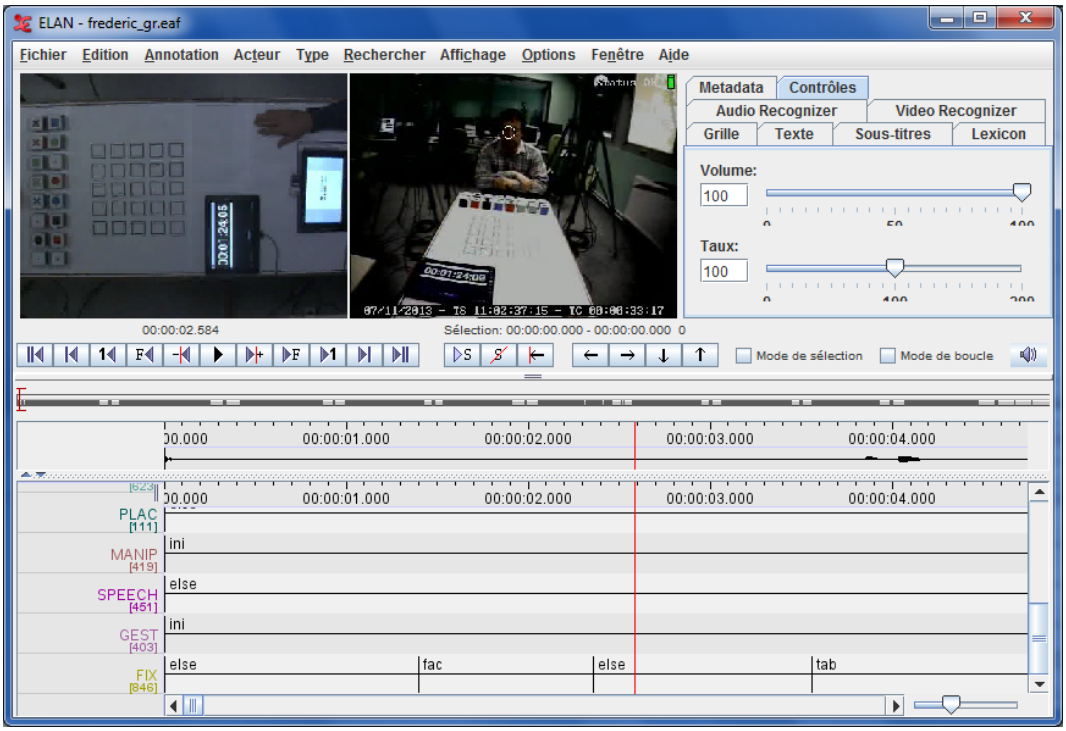


Figure 28: Le logiciel Elan utilisé dans l'annotation

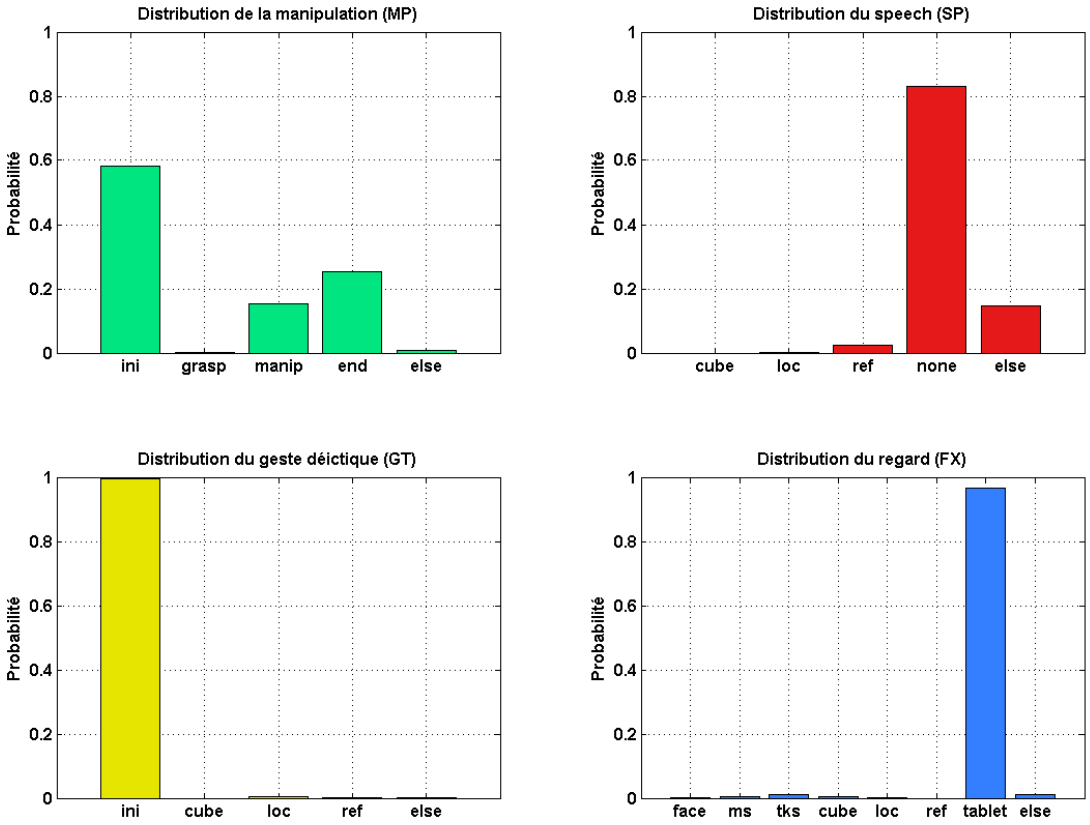


Figure 29: Distribution des observations pour l'IU "s'informer"

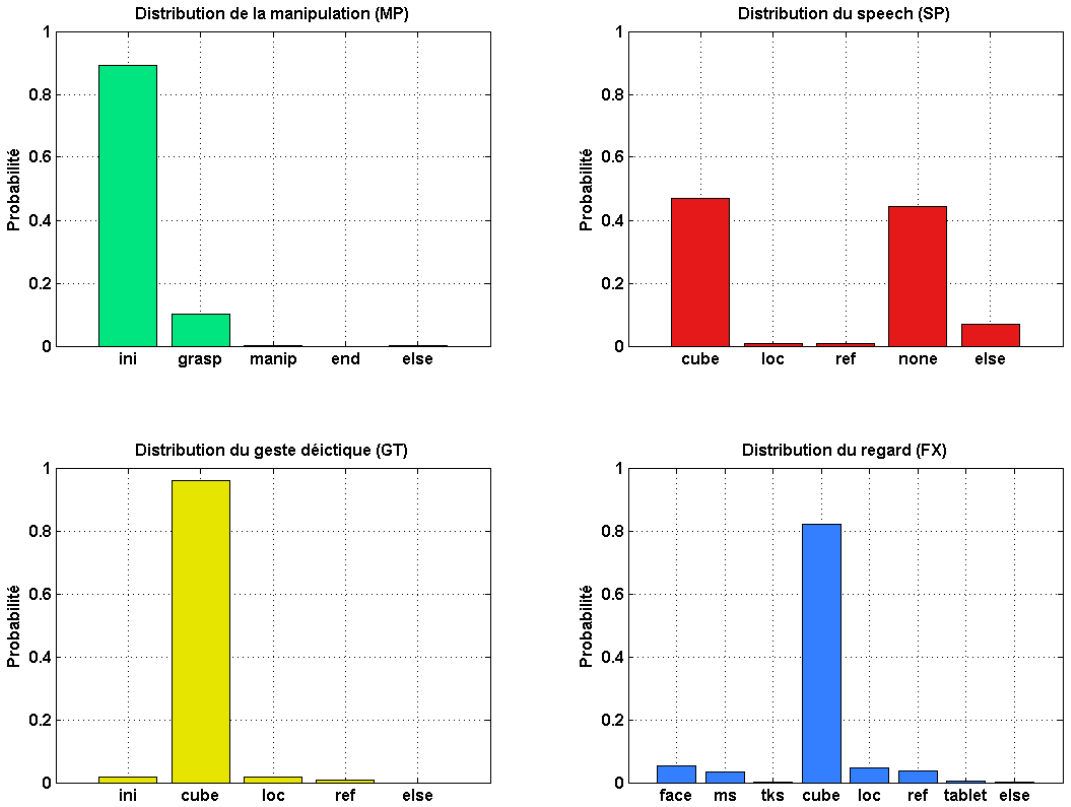


Figure 30: Distribution des observations pour l'IU "pointer"

2.3.4 Modélisation

Comme pour la première interaction (jeu de répétition de phrases), les modèles sensori-moteurs que nous proposons dans les chapitres suivants vont permettre à partir des données d'observations (ici MP et SP) d'estimer les segments d'une interaction (IU) et synthétiser par la suite le flux d'action le plus optimal (ici GT et FX). Nous rappelons que nous nous intéressons à la génération de la coverbalité, c'est pour cette raison que SP est considéré comme une observation perceptuelle d'entrée des modèles.

2.4 Conclusions

Nous avons présenté deux interactions face-à-face spécifiques: la première est un jeu de répétition de phrases et la deuxième est un jeu de référencement et de manipulation de cubes. Les deux interactions ont été soigneusement pensées et mises en place pour nous permettre de comprendre les mécanismes de perception et action que l'être humain utilise dans son interaction avec l'environnement. Avec le premier jeu, les expérimentateurs cherchaient à comprendre et modéliser le comportement du regard dans une interaction située. En particulier, ils essayaient de comprendre comment une unité d'interaction (qui correspond relativement bien à l'état mental d'un sujet) influence son comportement de regard, et aussi de clarifier à quel point le regard de l'un peut influencer le regard de l'autre. Avec le second jeu, nous avons élargi les capacités d'actions afin de pouvoir étudier de près les stratégies de l'attention mutuelle et de la deixis multimodale d'objets et de lieux. Par conséquent, ces deux interactions, malgré leur simplicité, fournissent des jeux de données suffisamment riches pour fouiller le comportement humain multimodal, extraire et évaluer des modèles sensori-moteurs pour l'interaction sociale. Grâce à l'apprentissage statistique, les modèles inférés d'une interaction particulière vont pouvoir 1/ situer l'activité du sujet principal dans l'IU la plus probable et 2/ générer les meilleures actions étant donnée cette unité d'interaction, les flux perceptuels et les objectifs joints des deux partenaires. Dans le prochain chapitre, nous appliquons notre démarche basée-données et nous explorons les premiers modèles appris sur nos jeux de données.

Chapitre 3 Modélisation par classification indépendante de chaque instant

3.1 Introduction

Afin de reconnaitre les unités constituant une interaction sociale entre plusieurs partenaires humains, ou aussi prédire le comportement d'un sujet cible à partir des traces interactionnelles observées, les modèles de classement, issus de l'apprentissage supervisé, constituent une solution intéressante à explorer. L'apprentissage supervisé permet à partir de données complètement labélisées (c.-à.-d. dont on connait la classe à laquelle elles appartiennent) de construire des modèles capables d'estimer la classe de nouvelles données non labélisées. Concrètement un classifieur prend un vecteur d'entrée x et l'affecte à une classe $c_k \; avec \; k = 1..K$, et K le nombre total des classes. Ce type de modèles a montré sa fiabilité et robustesse dans plusieurs domaines d'applications [115]. Dans l'état de l'art, on trouve plusieurs sortes de classifieurs, parmi lesquels on cite:

- Machines à vecteurs de support (SVM pour "Support Vector Machines") [116]
- Arbre des décisions [117]
- Réseaux de neurones (NN pour "Neural Networks") [118]
- Analyse discriminante [119]
- Méthode des k plus proches voisins ("kNN algorithm") [120]
- Classification naïve bayésienne [121]

Généralement, on ne peux pas dire en absolu qu'un classifieur est meilleur qu'un autre, tout dépendra du domaine d'application, de la nature des données et de leur distributions, de la quantité et la qualité des données disponibles pour l'apprentissage, etc.

Dans notre domaine d'application, l'estimation doit être faite d'une manière incrémentale et donc la rapidité du classifieur est un critère important. Les classifieurs connus pour être lents sont les méthodes du "lazy learning" notamment l'algorithme kNN. Son temps d'apprentissage est quasi nul et tout le calcul se fait lors du classement. En effet, le principe de cet algorithme est de retenir la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x. Par conséquent, sur le volet de la rapidité, le kNN n'est pas la meilleure méthode à retenir. En terme de précision, les SVM et les réseaux de neurones donnent généralement des taux plus élevés que d'autres méthodes [122]. Néanmoins, pour les réseaux de neurones, l'optimalité des performances n'est atteinte qu'avec de larges bases de données, ce dont nous ne disposons pas dans le cadre de notre application. Les arbres de décisions ont l'avantage de donner les meilleures résultats lorsqu'il s'agit de données catégorielles ou discrètes et c'est bien la nature de nos traces d'interactions. De plus, ils sont mieux interprétables que d'autres modèles (par exemple les SVM et NN) et

les stratégies d'élagage utilisées dans ces modèles permettent d'éviter le problème de sur-apprentissage et d'éliminer relativement aisément les données bruitées. Vu toutes ces propriétés, nous avons choisi dans notre application d'appliquer deux types de classifieurs, le premier classifieur sera basé sur les SVM et le deuxième sur les arbres de décisions.

D'autres méthodes appelées les méthodes d'ensemble [123][124] permettent de combiner plusieurs classifieurs afin de donner un résultat plus précis. On cite par exemple le Bagging [125], le Boosting [126] et le Stacking [127]. Leur principe est d'utiliser les points forts d'un classifieur pour remédier aux faiblesses de l'autre. On a recours à ce type d'approche pour avoir le meilleur taux de précision, car il est difficile, voire impossible, de trouver un seul modèle qui soit aussi performant qu'un ensemble de modèles [128]. Malgré cet avantage, les méthodes d'ensemble sont des modèles peu interprétables, assez compliqués vu le nombre de paramètres à gérer, et l'inconvénient majeur reste les temps d'apprentissage et de classification qui sont de loin beaucoup plus importants qu'un simple et unique classifieur.

Dans la prochaine section nous allons présenter brièvement les deux modèles que nous avons choisi d'appliquer, c'est à dire les SVM et les arbres de décisions.

3.2 Les SVM et les arbres de décision

3.2.1 Les SVM

Les machines à vecteurs de support (SVM) sont des techniques qui permettent de résoudre des problèmes non seulement de discrimination mais aussi de régression [129]. Les SVM (appelées aussi séparateurs à vastes marges) se basent sur deux principes clés: le principe d'hyperplan à marge maximale et le principe de fonction noyau. Ces deux principes étaient bien formalisées depuis plusieurs années avant qu'elles ne soient mises en commun pour construire les SVM en 1995 par Vapnik [130]. À partir de cette date, les SVM ont rapidement gagné de popularité vu les garanties théoriques apportées, et les bons résultats en pratique.

Les deux idées clés des SVM, permettent de résoudre des problèmes de classement non-linéaire. La première est celle de la marge maximale. La marge représente la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés les vecteurs supports. Le problème consiste à trouver la frontière séparatrice optimale (qui maximise la marge), à partir de données d'apprentissage. Si la frontière est linéaire, ceci revient à un problème d'optimisation quadratique, pour lequel il existe plusieurs algorithmes connus dans l'état de l'art. Lorsque les données sont linéairement inséparables, la fonction noyau (deuxième idée clé des SVM) permet de transformer l'espace de représentation initial des données en un espace de plus grande dimension, où il est probable de trouver une séparatrice linéaire. La fonction noyau doit respecter les conditions du théorème de Mercer [131] et a l'avantage de ne pas exiger la spécification explicite de la transformation à réaliser pour le changement d'espace.

3.2.1.1 Principe

Données linéairement séparables

Dans un problème de discrimination entre deux classes on essaye d'estimer une fonction $f: R^N \rightarrow \{\pm 1\}$ à partir d'une base d'apprentissage $(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_l, y_l) \in R^N \times \{\pm 1\}$ avec x_i le vecteur d'entrée et y_i la classe correspondante $(i = 1..l)$. Si les données sont linéairement séparables il existe un couple (w, b) tel que :

$$\begin{aligned} w^T.x_i + b \geq 0 \;\; si\; y_i = 1 \\ w^T.x_i + b \leq 0 \;\; si\; y_i = -1 \end{aligned} \qquad (1)$$

La fonction de décision f pour un vecteur de test x est définit par :

$$f(x) = sign(w^T.x + b) \qquad (2)$$

La marge est la plus petite distance entre les échantillons d'apprentissage et l'hyperplan séparateur. En résumé, on montre que maximiser la marge (qui est égale à $\frac{1}{\|w\|}$) revient à minimiser $\frac{1}{2}\|w\|^2$ sous les contraintes $y_i(w^T.x_i + b) \geq 1$. La solution de ce problème d'optimisation représente une combinaison linéaire d'un ensemble de points x_i. Ces points sont les points les plus proches de l'hyperplan optimale, on les appelle les points ou les vecteurs supports. La solution w s'écrit donc sous la forme : $w = \sum_i v_i x_i \;\; avec\; v_i \in R^*$ et la fonction décision finale s'écrit sous la forme :

$$f(x) = sign\left(\sum_{i=1}^{l} v_i(x.x_i) + b\right) \qquad (3)$$

On ne rentre pas dans les détails de l'algorithme. Il faut juste noter que la fonction de décision finale ne dépend que du produit scalaire $(x.x_i)$. Cette propriété est importante car elle va permettre de généraliser l'algorithme pour le cas non linéaire.

Données linéairement non- séparables

Beaucoup de données liées à des problèmes du monde réel sont linéairement inséparables et donc il n'existe pas un hyperplan qui permet de séparer les instances positives des instances négatives dans la base d'apprentissage. Une solution à ce problème d'inséparabilité est de revoir le problème dans un espace de dimension plus élevée (voire infinie) et de trouver ensuite un hyperplan optimal de séparation.

Concrètement, on applique aux vecteurs d'entrée x_i une transformation non-linéaire ϕ ($\phi : R^N \rightarrow H$), l'espace d'arrivée H est appelé l'espace de redescription. Une séparation linéaire dans le nouvel espace correspond bien à une séparation non linéaire dans l'espace d'origine. En analogie avec l'algorithme décrit dans le cas de séparabilité linéaire, l'algorithme d'apprentissage ne va dépendre que du produit scalaire $\phi(x).\phi(x_i)$. Maintenant s'il existe une fonction noyau K tel que $K(x, x_i) = \phi(x).\phi(x_i)$, on n'utilisera que la fonction K dans l'apprentissage et on n'aura jamais besoin de spécifier ϕ.

De toute évidence, si H est de grande dimension, le calcul sera assez coûteux. C'est pour ça que, dans certains cas, il existe des noyaux simples et qui peuvent être évalués de manière efficace. Par exemple, le noyau polynômial : $K(x, y) = (x, y)^d$. Une fois qu'on a choisi le bon noyau, on substitue chaque instance x_i par $\phi(x_i)$ et on applique l'algorithme de l'hyperplan optimal. La fonction de décision résultante est non linéaire et elle se présente sous la forme suivante :

$$f(x) = sign\left(\sum_{i=1}^{l} v_i K(x, x_i) + b\right) \quad (4)$$

Pour les deux cas présentés ci-dessus l'algorithme d'apprentissage atteint un minimum global ce qui représente un avantage significatif par rapport à d'autres modèles comme les réseaux de neurones.

3.2.1.2 Multi-classes SVM

Pour appliquer la méthodologie expliquée ci-dessus aux cas impliquant multiples classes, plusieurs méthodes ont été proposées [132]. Parmi elles, deux méthodes sont très connues et largement utilisées non seulement pour les SVM mais aussi pour tout classifieur binaire. La première est appelée "one versus all" et la deuxième "one versus one".

On considère K classes $(c_1, .., c_K)$, l'approche one-versus-all consiste à former K classifieurs binaires en affectant le label 1 à une des classes et -1 aux autres. Pendant le test, le classifieur retenu sera celui qui produit la marge la plus élevée:

$$classe\ de\ x_i = \underset{k=1..K}{\operatorname{argmax}}((w^k)^T \phi(x) + b^k) \quad (5)$$

L'approche one-versus-one consiste à former $\frac{K(K-1)}{2}$ classifieurs binaires en opposant les K classes entre elles. Pendant le test, l'échantillon x à classer est scruté par chaque modèle et un vote majoritaire permet l'identification de sa classe c_k. C'est la classe qui sera la plus souvent attribuée à x qui sera retenue. Cette stratégie de vote majoritaire est connue sous le nom de "Max Wins".

3.2.2 Les Arbres de décisions

Les arbres de décision [117] [133] [122] représentent une méthode populaire de l'apprentissage supervisé. Ils permettent de prédire les valeurs prises par une variable de sortie à partir d'un ensemble de descripteurs (variables d'entrée). Chaque nœud dans un arbre de décision représente une variable d'entrée et chaque branche représente une valeur que cette variable peut prendre. La procédure de classement d'une nouvelle instance (le vecteur test) commence par le nœud racine et se poursuit à travers les nœuds selon les valeurs des variables correspondantes. Un exemple d'arbre de décision (inspiré de [122]) ainsi que la base d'apprentissage utilisée sont montrés respectivement dans la Figure 31 et le Tableau 1. On dispose de trois variables d'entrée $x^t = (x_1, x_2, x_3)$ et chaque variable peut avoir trois valeurs

possibles, par exemple x_1 peut prendre soit a_1, b_1 ou c_1. Quant à la variable de sortie (la classe), elle peut prendre soit "oui" soit "non".

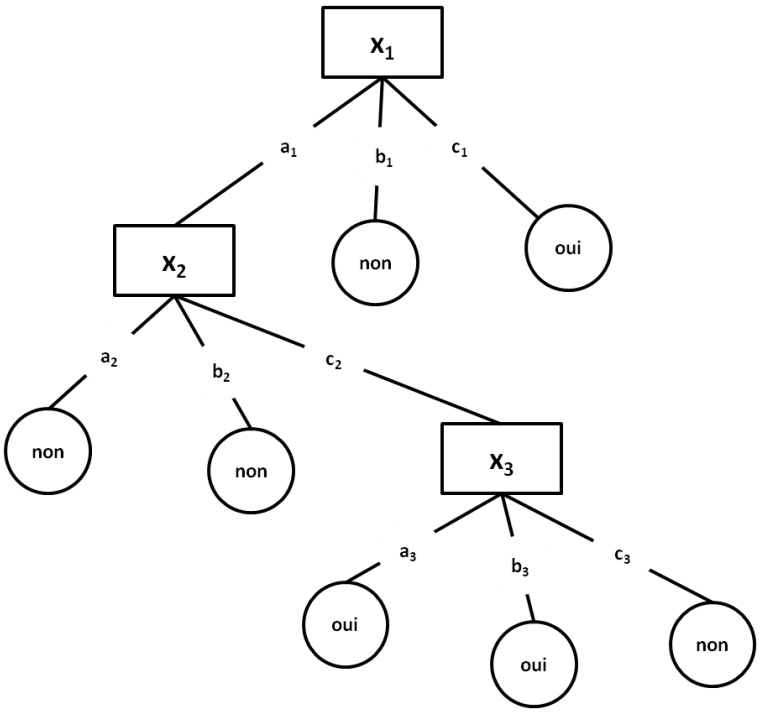


Figure 31: Exemple d'un arbre de décision

x_1	x_2	x_3	Classe
a_1	a_2	c_3	non
a_1	b_2	c_3	non
a_1	c_2	a_3	oui
a_1	c_2	b_3	oui
a_1	c_2	c_3	non
b_1	a_2	c_3	non
c_1	c_2	b_3	oui

Tableau 1: Base d'apprentissage

Pour classer une nouvelle instance, il suffit de traverser l'arbre, et l'associer à la classe attachée à la feuille à laquelle elle aboutit. Malgré la simplicité apparente, plusieurs problèmes se manifestent lors de la construction de l'arbre. On recense essentiellement trois méthodes largement utilisées par la communauté scientifique dans la construction des arbres : l'algorithme C4.5 [134] (successeur de ID3), CART [135] (Classification And Regression Tree) et la méthode CHAID [136] (CHi-squared Automatic Interaction Detector).

L’approche générale à suivre pour construire un arbre optimal consiste tout d’abord à trouver la variable qui sépare le mieux les données d'apprentissage. Cette variable sera le nœud racine de l'arbre. Il existe de nombreuses méthodes pour trouver cette variable, par exemple l'indice de Gini [135] ou aussi le gain de l’information [137] qui est utilisé dans l’algorithme C4.5. Ce processus est ensuite à réitérer pour chaque partition des données divisées, créant ainsi des sous-arbres et des feuilles qui représentent les différentes classes. Cette première phase est connue sous le nom de "phase d’expansion". La deuxième phase est une "phase d’élagage" qui consiste supprimer les branches peu représentatives. En effet, les arbres ont tendance à former un modèle assez complexe, collant excessivement aux données et causant ainsi du sur-apprentissage. Il a été démontré également que la taille des arbres croît avec la taille de la base d’apprentissage. Par conséquent, l’élagage est la solution pour réduire et déterminer la taille optimale de l’arbre. Pour la méthode CHAID, l’élagage se fait au fur et à mesure de la construction de l’arbre et non pas dans une phase postérieure comme le cas de la méthode CART ou C4.5. Notons qu’il existe d’autres solutions pour le problème de sur-apprentissage, par exemple les forêts aléatoires [138].

3.3 Application

Dans notre application, nous avons décidé de travailler avec les SVM et les arbres de décision en se servant du toolbox Weka [139]. Weka est une librairie d'algorithmes d'apprentissage statistique pour les tâches de fouille de données. En particulier, Weka contient des algorithmes pour le prétraitement des données, le classement, la régression, la segmentation, et la visualisation. Ces algorithmes peuvent être appliqués via une interface graphique ou appelés à partir du code Java. Ce toolbox est également bien adapté au développement de nouveaux algorithmes d'apprentissage statistique. Weka propose l’algorithme SMO pour les SVM et J48 pour les arbres de décision, que nous allons présenter brièvement dans la suite.

3.3.1 L’algorithme SMO

Pour l’apprentissage des SVM, plusieurs algorithmes ont été proposé, entre autres l’algorithme SMO (Sequential Minimal Optimization) développé par Platt [140] de Microsoft Research. Lors de l’apprentissage des SVM, comme déjà évoqué dans la section 3.2.1.1, on cherche la solution d’un large problème d'optimisation quadratique. L’algorithme SMO décompose ce grand problème en un ensemble de petits problèmes (résolus analytiquement) ce qui réduit le nombre d’opérations de calcul à faire. La quantité de mémoire nécessaire pour SMO est linéaire en la taille de l'ensemble d'apprentissage, ce qui permet à SMO de gérer de très grandes bases d’apprentissage. Comme le calcul matriciel est évité, la complexité temporelle du SMO est entre linéaire et quadratique en la taille des données, alors que la complexité d’un algorithme standard comme le chunking SVM [141] est entre linéaire et cubique. Sur des données réelles, SMO peut être 1000 fois plus rapide que l'algorithme chunking [140]. Le SMO implémenté dans weka transforme les attributs nominales en des attributs binaires et le problème multi-classes est résolu grâce à la méthode "one-versus-one" décrite dans la section 3.2.1.2.

3.3.2 L'algorithme J48

Pour l'apprentissage des arbres de décision, le toolbox weka propose l'algorithme J48 qui est une implémentation open source de l'algorithme C4.5, l'algorithme le plus utilisé dans la littérature des arbres de décision. Brièvement, l'implémentation s'appuie sur deux aspects essentiels de C4.5, décrits dans l'ouvrage d'origine [134]: dans la phase d'expansion, on utilise le gain d'information (gain ratio) pour sélectionner les variables de segmentation. Dans la phase d'élagage, on se base sur le principe de l'erreur pessimiste (taux de mal-classement pénalisé par les effectifs). Le C4.5 a l'avantage de gérer les variables discrètes, continues et les valeurs manquantes. L'inconvénient majeur est que cette technique a tendance à produire des arbres grands et assez profonds car, lorsque la base d'apprentissage est grande, le post élagage est moins performant.

3.3.3 Données

Notre première application concerne la première interaction intitulé "Jeu de répétition de phrases" présentée dans la section 2.2. Nous rappelons qu'il s'agit d'une interaction entre deux sujets (un sujet principal et un interlocuteur avec 10 dyades en total). Pour chaque dyade, nous disposons des activités vocales (v_1, v_2) et des regards (g_1, g_2) de chacun. Sept unités interactionnelles ont été annotées semi-automatiquement notamment "lire" , "se préparer" , "parler" , "attendre" , "écouter" , "réfléchir" et "autre".

3.3.4 Modèles

L'objectif est de concevoir un modèle pour l'interaction qui permettra d'estimer l'unité interactionnelle dans laquelle est impliquée le sujet principal, et un modèle qui permettra d'estimer la région d'intérêt fixée par son regard. Nous appelons le premier, le modèle de reconnaissance et le deuxième, le modèle de génération. Pour les SVM et les arbres de décision, un premier classifieur représentant le modèle de reconnaissance (Figure 32) est utilisé pour estimer l'IU à partir des activités vocales de chacun et le regard de l'interlocuteur. Ensuite, un deuxième classifieur représentant le modèle de génération (Figure 32) est utilisé pour estimer le regard g_1 à partir des mêmes données c'est à dire (v_1, v_2, g_2).

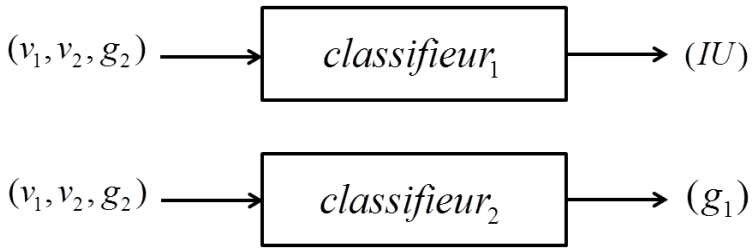


Figure 32: Les deux classifieurs proposés : le premier représente le modèle de reconnaissance et le second représente le modèle génération

3.3.5 Evaluation et distance de Levenshtein

Pour tous les modèles, nous appliquons le principe de la validation croisée (10-fold cross validation) sur nos données: 9 sujets ont été utilisés pour l'apprentissage tandis que le dixième pour le test. Le taux de reconnaissance est utilisé pour évaluer la reconnaissance des unités interactionnelles. Il représente le pourcentage des instances correctement classées par rapports à l'ensembles de données. Le modèle de génération est évalué en utilisant la distance de Levenshtein [142].

La distance de Levenshtein est une distance mathématique donnant une mesure de similarité entre deux séquences. Elle calcule le nombre minimum d'opérations élémentaires (insertions, suppressions et substitutions) requises pour passer d'une séquence à l'autre. La distance de Levenshtein satisfait bien la définition d'une distance: le résultat est un entier positif ou nul, elle est nulle si et seulement si les deux séquences comparées sont identiques et elle vérifie bien les propriétés de symétrie et de l'inégalité triangulaire. Pour comparer deux séquences (s1 de longueur t1 et s2 de longueur t2), le calcul de la distance est effectué en appliquant l'algorithme ci-dessous (on note que T = max (t1,t2), d un tableau de taille [0..T,0..T] et le passage sera de s1 à s2):

```
1:  for i=0:t1  d[i,0]=i; end //initialization
2:  for j=0:t2  d[0,j]=j; end //initialization
3:  for i=1:t1 // filling table d
4:    for j=1:t2
5:      if s1[i]==s2[i] c=0 else c=1;end //c is the cost for a substitution
6:      d[i,j]= minimum (
7:                        d[i-1,j]+1  //deletion
8:                        d[i,j-1]+1  //insertion
9:                        d[i-1,j-1]+c   //substitution
10:                       )
11: end
12:   end
13: return d[i,j]
```

En appliquant cet algorithme, par exemple, sur les deux chaînes "niche" et "chien" (on veut passer de "niche" à "chien") on obtient le tableau suivant:

		C	H	I	E	N
	0	1	2	3	4	5
N	1	1	2	3	4	4
I	2	2	2	2	3	4
C	3	2	3	3	3	4
H	4	3	2	3	4	4
E	5	4	3	3	3	4

Figure 33: Le tableau d qu'on obtient en appliquant l'algorithme de la distance de Levenshtein pour transformer "niche" à "chien", la distance est égale à la case d[5,5] (colorée en gris) c.-à.-d 4.

		C	H	I	E	N
	0	1	2	3	4	5
N	1	1	2	3	4	4
I	2	2	2	2	3	4
C	3	2	3	3	3	4
H	4	3	2	3	4	4
E	5	4	3	3	3	4

Figure 34: La séquence d'opérations à réaliser pour obtenir l'alignement souhaité.

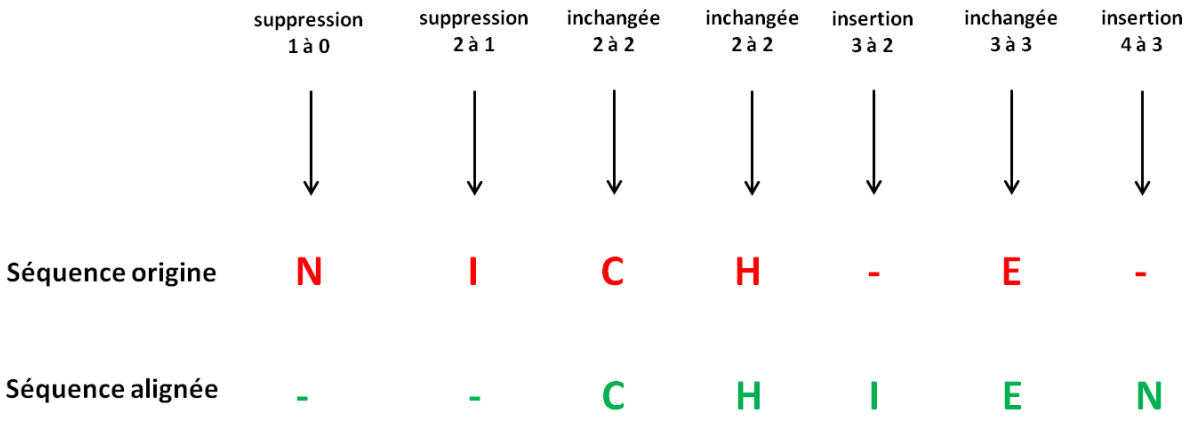


Figure 35: Le passage de la séquence d'origine à la séquence d'alignement.

Le nombre d'opérations nécessaires pour transformer "niche" à "chien" (qui est égale à 4) représente alors la distance de Levenshtein entre ces deux chaînes. Maintenant pour trouver la chaîne/séquence alignée, on applique une procédure de chaînage arrière (i.e. "backtracking") en partant de la dernière case (en bas à droite) jusqu'à la case 0 en passant par les cases qui réalisent le minimum d'opérations (cf. Figure 34). Lorsqu'on passe à une case ayant le même nombre que la case initiale, cela indique que la lettre est restée inchangée. Cependant lorsqu'on passe à une case ayant un nombre inférieur, un déplacement vertical souligne une suppression, un déplacement horizontal souligne une insertion et un déplacement diagonal souligne une substitution. Dans cette exemple, on a donc 3 lettres qui sont restées inchangées, deux lettres insérées et deux lettres retirées (cf. Figure 35). A partir de la séquence alignée, le rappel, la précision et leur moyenne harmonique (la F-mesure) peuvent être calculés. Dans cet exemple, le rappel est égal à 60%, la précision à 60% et la F-mesure également à 60%.

Dans ce chapitre, tous les taux de génération représentent des F-mesures issues de l'alignement avec la distance de Levenshtein. Nous avons adopté cette méthode pour l'évaluation de la production du regard parce qu'elle permet une comparaison moins sensible aux différences de micro-alignements et donc plus sensible aux similarités de séquençage entre les signaux générés et les signaux originaux.

3.4 Résultats

Les résultats montrent que les deux modèles proposés (SVM et arbres de décision) donnent des performances assez similaires (Figure 36). Pour les deux classifieurs, on enregistre un taux de reconnaissance de l'IU égal à 81% et un taux de génération de regard égal à 50%. Ces deux taux sont largement supérieurs aux niveaux de hasard (24% pour l'IU et 31% pour le regard) qu'on retrouve en utilisant les distributions empiriques des données. Les tests statistiques effectués de significativité (student et wilcoxon) ont démontré qu'il n'y a pas de variation significative entre les deux classifieurs pour un seuil de confiance égal à 95% (dans la suite, on garde ce seuil pour tous les tests). De même, l'évaluation de la génération du regard avec des taux exacts – sans utiliser la distance de Levenshtein – montre qu'il n'y a pas de différence significative entre les deux classifieurs étudiés (cf. Figure 37).

Lorsqu'on trace la séquence des unités interactionnelles (IU) reconnues par les classifieurs, on remarque que les deux classifieurs ne sont pas efficaces dans la détection de la structure de l'interaction. Un exemple est montré dans la Figure 38 dans lequel, pour un sujet donné, nous avons tracé la séquence des IU estimées par le modèle SVM et nous avons comparé cette séquence à la séquence de l'interaction réelle (la vérité terrain). On s'aperçoit que le modèle SVM n'arrive pas à capturer l'organisation séquentielle de la tâche et que la trajectoire estimée ne reflète pas correctement la syntaxe prédéfinie de cette tâche. Plus précisément, on remarque que le classifieur ignore des petites transitions vers des unités comme "Se préparer" ou "Réfléchir" et détecte essentiellement les transitions vers les unités plus longues comme "Parler" , "Attendre" et "Ecouter". Ceci n'est pas en contradiction avec le taux de reconnaissance de 81% parce que ces trois unités représentent à elles seules 85% de la vérité terrain.

Ce défaut d'organisation séquentielle est expliqué par la nature intrinsèque des classifieurs qui exploitent exclusivement l'information ascendante fournie par les observations à un instant t, et donc ne permettent pas de modéliser les aspects séquentiels. Pour remédier à ce manque d'information temporelle, nous allons dans la section suivante proposer de nouveaux classifieurs qui vont permettre d'appuyer les décisions sur un contexte plus large que la trame courante. Nous appellerons ces modèles les modèles avec mémoire.

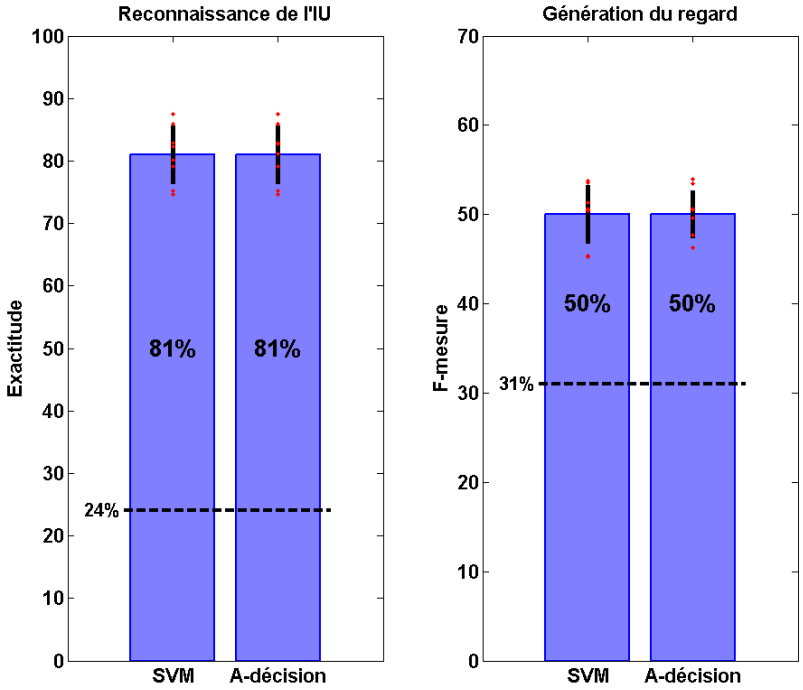


Figure 36: Les résultats des SVM et des Arbres de décision. Les lignes en pointillé montrent les niveaux du hasard.

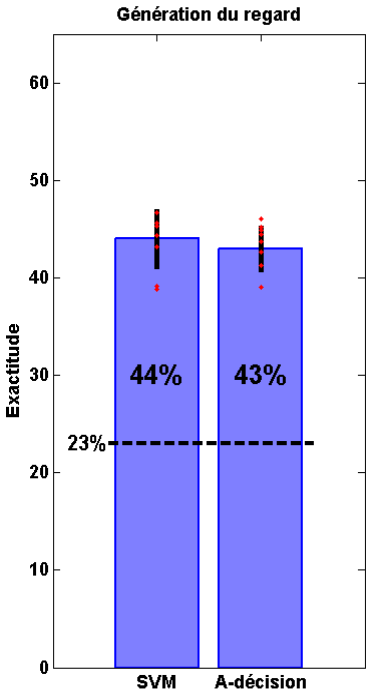


Figure 37: Taux exactes pour la génération du regard. La ligne en pointillé montre le niveau du hasard.

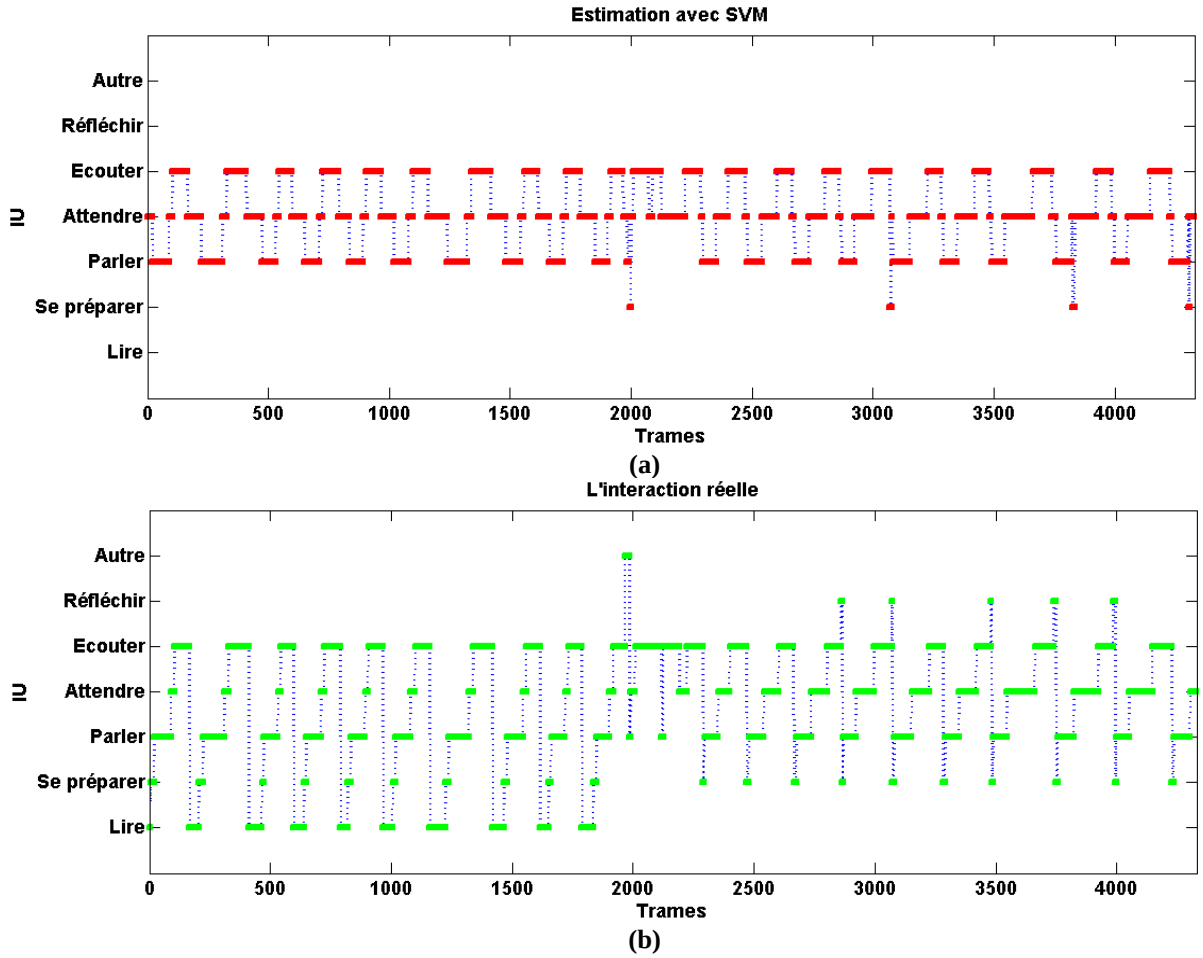


Figure 38: (a) Séquence des IU estimée par SVM (b) La vraie séquence de l'interaction réelle

3.4.1 Modèles avec mémoire

Afin de construire un modèle de génération de pronoms démonstratifs dans un dialogue relatif à un jeu de puzzle, Spanger et al. [143] ont proposé un classifieur SVM qui utilise des informations actuelles et historiques sur l'interaction. Cette idée est également utilisé par Chiu et Marsella [144] afin de générer des gestes de battement à partir d'un signal acoustique. En effet, les performances d'un classifieur peuvent être améliorées en ajoutant de la mémoire (observations historiques) au vecteur d'entrée. Dans le section précédente, les modèles proposés n'utilisent que les données de l'instant courant t. Dans la nouvelle configuration (Figure 39), nous avons ajouté les trois mêmes attributs (v_1, v_2, g_2) mais relatifs à un instant précédent $t-h$, h étant le décalage temporel propre à la mémoire que nous voulons ajouter aux classifieurs. Dans ce paragraphe, nous annotons les attributs relatifs à un instant t par (v_1^t, v_2^t, g_2^t) et les nouveaux attributs de mémoire par $\left(v_1^{t-h}, v_2^{t-h}, g_2^{t-h}\right)$. Afin de trouver la mémoire optimale que nous allons introduire, le décalage h a été varié dans un intervalle de temps allant d'une trame à 80 trames.

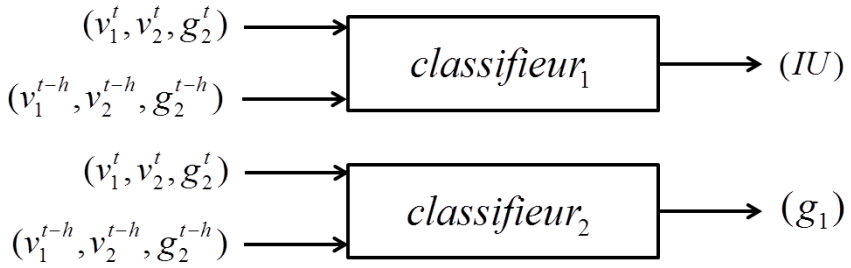


Figure 39: Les nouveaux modèles avec des attributs de mémoire

3.4.2 Résultats après l'ajout de mémoire

Nos tests ont révélé qu'il n'y a pas de différence significative entre SVM et arbres de décision. Nous allons ainsi nous concentrer sur la comparaison des modèles SVM sans mémoire d'une part, et SVM avec mémoire d'une autre part (qu'on note SVM + M). La Figure 40 ainsi que la Figure 41 montrent que, pour cette tâche, la mémoire optimale à inclure dans les observations d'entrée est $h_{optimal} = 55\ trames \sim 2\ secondes$. Nous avons obtenu le même instant pour les deux classifieurs ($classifieur_1\ et\ classifieur_2$). Ce décalage optimal est identique pour les arbres de décision.

Il est intéressant de noter que ce délai optimal correspond à une étude [145] dans laquelle les auteurs démontrent que lorsque un orateur regarde une image sur laquelle il est en train de s'exprimer, le temps le plus probable qu'un auditeur regarde la même image est exactement deux secondes après. Dans notre cas, l'information sur le regard de l'interlocuteur de 2 secondes dans le passé a permis d'attendre les meilleurs taux de classement car comme dans l'étude citée, cette information du passé est en corrélation directe avec l'IU actuelle et le regard actuel que nous voulons synthétiser pour le sujet principal.

Pour la comparaison des deux configurations (cf. Figure 42), on remarque que l'ajout d'observations du passé (~ 2 secondes) mène à des résultats significativement meilleures à la fois en ce qui concerne la reconnaissance des IU (91% vs 81%) et aussi en ce qui concerne la génération du regard (59% vs 50%). D'autre part, on remarque que cette injection de mémoire dans les classifieurs a conduit à une meilleure modélisation de la structure de l'interaction. Dans la Figure 43, nous constatons une amélioration dans l'estimation de la séquence des IU pour le modèle SVM avec mémoire. Dans cette nouvelle configuration, le modèle réussit relativement à transiter vers des IU qu'il ignorait auparavant, comme par exemple "Lire" et "Se Préparer".

Malgré cet apport, la nouvelle trajectoire estimée ne reflète pas encore correctement la structure réelle. L'IU "Réfléchir" par exemple, n'est pas détecté, et on voit aussi qu'il y a des transitions qui ne correspondent pas à la vérité terrain. Par conséquent, nous concluons que l'injection de mémoire dans les nouveaux modèles ne résolvent qu'en partie le manque d'information temporelle dans les classifieurs.

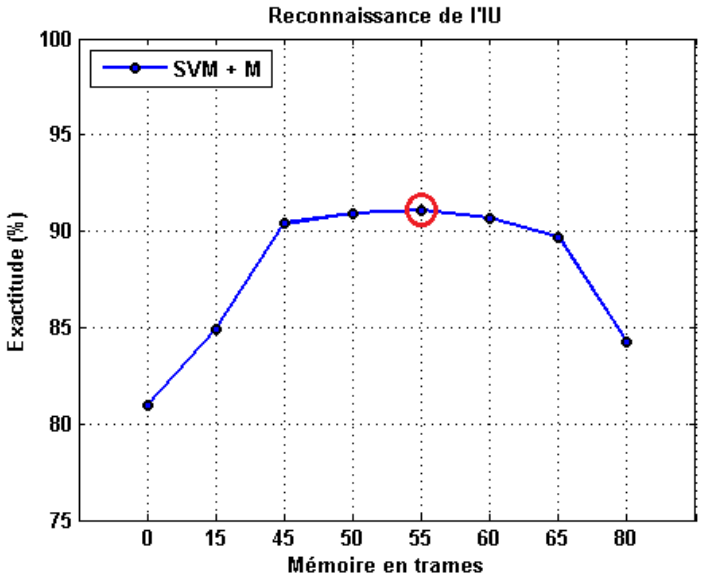


Figure 40: Meilleure mémoire à ajouter pour obtenir un taux de reconnaissance maximal (cercle rouge). Cette mémoire optimale correspond à 55 trames (~ 2 secondes).

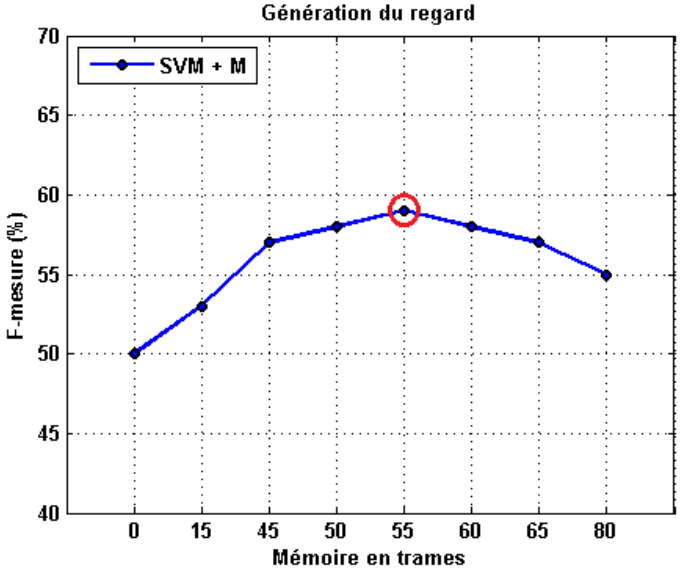


Figure 41: Meilleure mémoire à ajouter pour obtenir un taux de génération maximal (cercle rouge). Cette mémoire optimale correspond à 55 trames (~ 2 secondes).

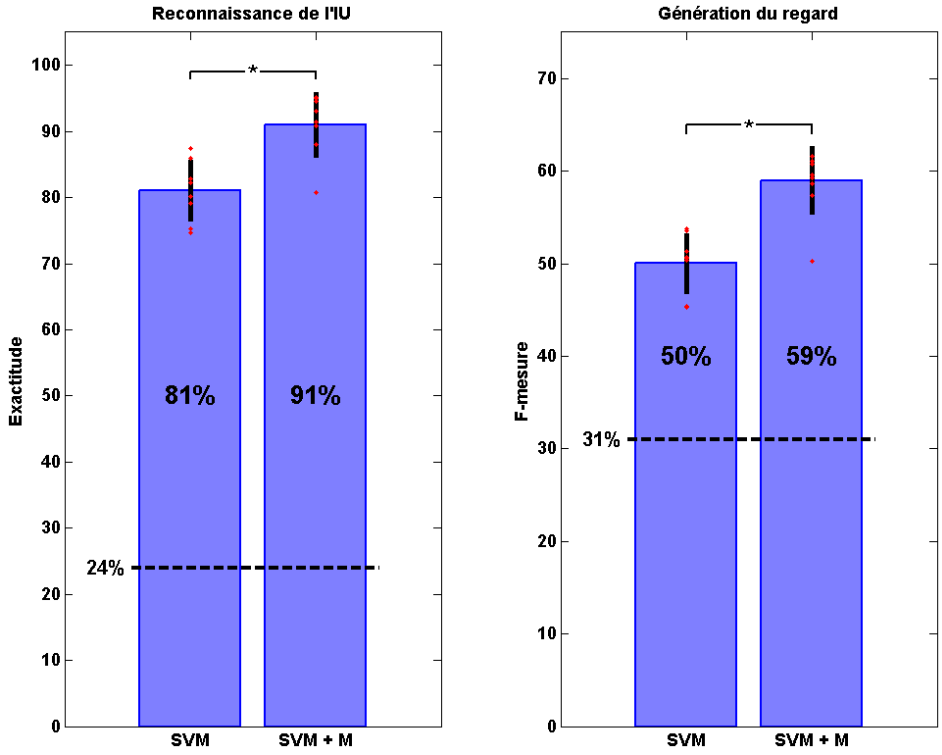
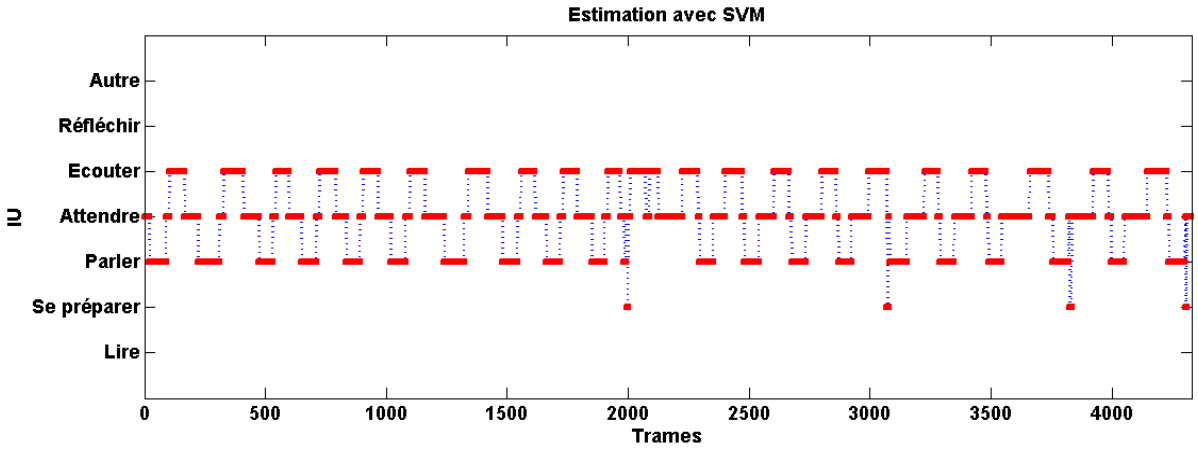
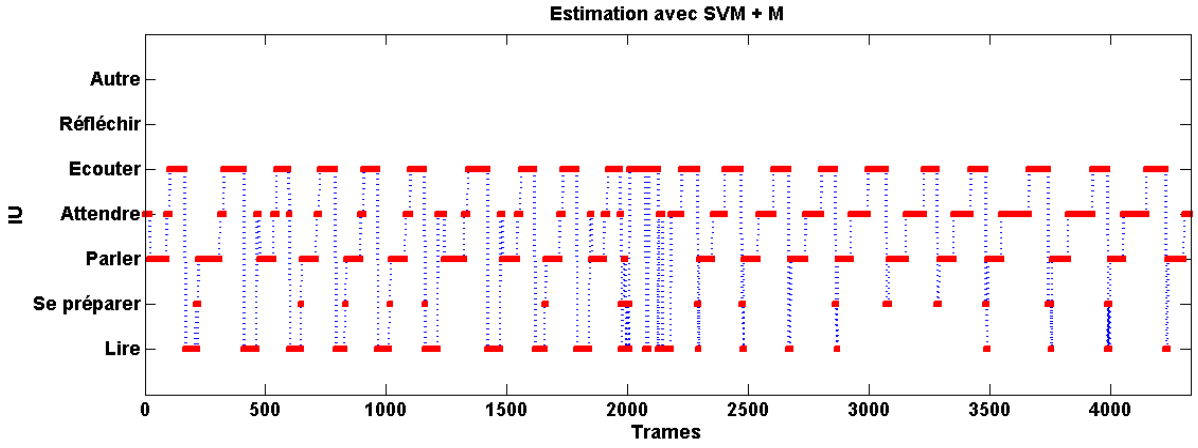


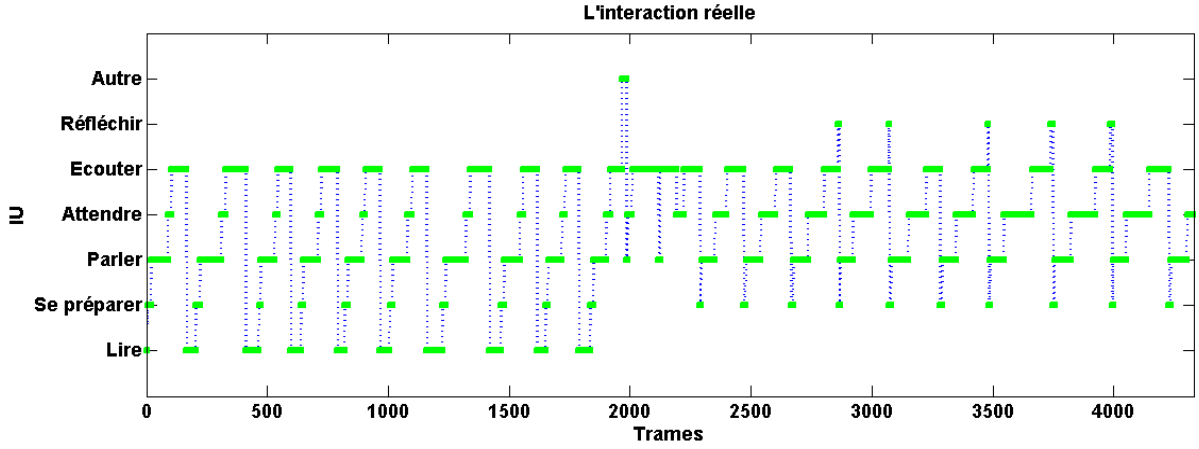
Figure 42: Résultats du modèle SVM sans mémoire (SVM) et avec mémoire (SVM + M)



(a)



(b)



(c)

Figure 43: (a) Séquence des IU estimée par SVM (b) Séquence estimée par SVM + Mémoire (c) La vraie séquence de l'interaction réelle

3.5 Conclusions

Dans ce chapitre, nous avons proposé des modèles basés sur des classifieurs, notamment les SVM et les arbres de décision afin de reconnaitre, en premier lieu, l'unité

interactionnelle dans laquelle est impliqué un sujet lors d'une interaction sociale et, en deuxième lieu, générer son regard. Les résultats ont montré qu'il n'y a pas de différence significative entre les SVM et les arbres de décision. De plus, nous avons remarqué que les premiers modèles proposés n'arrivent pas à déceler l'organisation séquentielle de la tâche par le fait qu'ils ne disposent pas de propriétés de modélisation temporelle. C'est pour cette raison que nous avons introduit les modèles avec mémoire qui ont permis d'améliorer significativement les chiffres de la reconnaissance et de la génération. Nous concluons que les classifieurs peuvent donner des bonne performances si une certaine mémoire est prise en compte dans le vecteur d'entrée (2 secondes dans notre cas). Néanmoins, il y'a d'autres pistes d'amélioration qui sont mieux adaptées à notre application, notamment l'utilisation des modèles proprement séquentielles comme les chaînes de markov cachées (HMM) que nous présenterons dans le prochain chapitre.

Chapitre 4 Modélisation par HMM & HSMM

4.1 Introduction

Dans ce chapitre, nous présentons des approches statistiques pour la modélisation temporelle des comportements joints, sensori-moteurs et multimodaux. Ces modèles devraient être en mesure pour un sujet cible d'estimer d'abord l'unité d'interaction (IU) à partir d'observations perceptuelles (par exemple, l'activité vocale des dyades / fixations de regard du partenaire) puis de générer des actions appropriées (par exemple, ses propres fixations de regard) qui reflètent l'unité interactionnelle courante et sa prise de conscience de l'évolution actuelle du plan ou de la tâche partagée.

Plus précisément, dans ce chapitre nous allons proposer plusieurs modèles qui se basent sur les chaînes de Markov cachées (HMM pour "Hidden Markov Model"). Nous avons choisi les HMM [146] [147] parce qu'ils sont par définition des modèles séquentiels. Ils disposent intrinsèquement des capacités de modélisation temporelle qui ont remarquablement manqué aux classifieurs déjà présentés. Les modèles proposés ont été appliqués sur notre première base de données (jeu de répétition de phrases). Les différentes contributions sont comme suit:

- Nous proposons un modèle sensori-moteur se basant sur les HMM et permettant la reconnaissance et la génération d'une manière incrémentale.
- Nous comparons les propriétés de ce modèle avec celles des classifieurs, notamment les SVM et les arbres de décision.
- Nous proposons également deux méthodes d'initialisation pour les HMM
- A la fin du chapitre, nous étudions la contribution des chaînes semi-Markoviennes cachées (HSMM pour "Hidden Semi-Markov Model") par rapport à des HMM standards.

4.2 Rappel sur les HMM

Un modèle de Markov caché (HMM) est un modèle de Markov statistique dans lequel le système modélisé est supposé être un processus de Markov avec des états non observés (cachés). Dans les chaînes de Markov simples, les états sont directement visibles à l'observateur, et donc les probabilités de transition entre ces états sont les seuls paramètres à estimer. Dans un modèle de Markov caché, l'état n'est pas directement visible. Mais on dispose d'une observation visible générée par cet état. L'observation générée ne dépend que de l'état actuel et, selon l'hypothèse de Markov, l'état actuel ne dépend que de l'état précédent. Chaque état dispose d'une distribution de probabilité sur les observations possibles. Par conséquent, la séquence d'observations générées par un HMM fournit des informations sur la séquence d'états. Côté applications, les HMM sont particulièrement connus et appliqués avec

succès dans la reconnaissance des structures temporelles comme la parole [148], l'écriture manuscrite [149], les gestes [150], etc.

Pour formaliser un modèle HMM λ à observations discrètes (HMM discret) (voir Figure 44), on définit les éléments suivant :

- N : le nombre des états cachés
- $Q = \{q_1, q_2, .. q_N\}$: l'ensemble des états cachés possibles
- T : la longueur de la séquence d'observations
- $O = \{o_1, o_2, .., o_T\}$: la séquence d'observations
- o_t : l'observation à l'instant t
- M : le nombre des valeurs possibles que peut prendre o_t
- $E = \{e_1, e_2, .., e_M\}$: l'ensemble des valeurs (événements discrets) possibles que peut prendre o_t
- $S = \{s_1, s_2, .., s_T\}$: la séquence des états cachés
- s_t : l'état caché à l'instant t

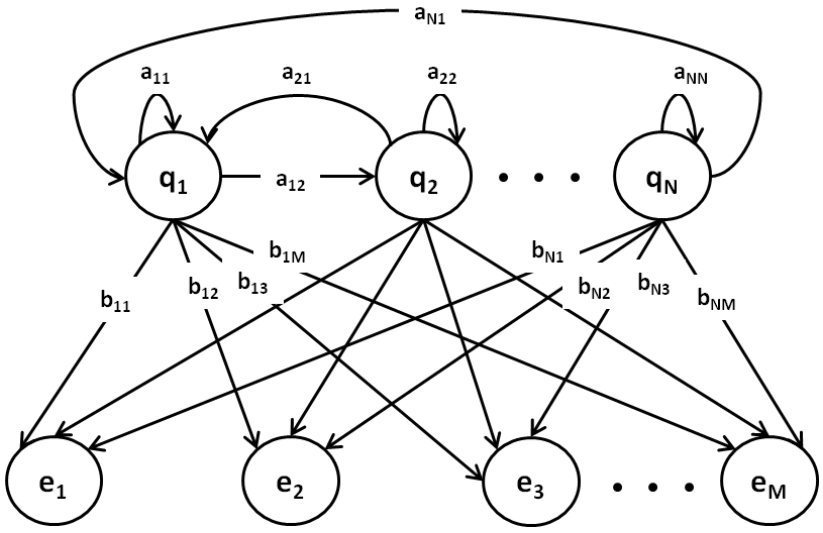


Figure 44: Un modèle HMM avec N états cachés et M observations

Un modèle HMM λ est complètement défini par trois matrices:

- La matrice de transition A qui est définie par :

$$A = \{a_{ij} | a_{ij} = P(s_t = q_j | s_{t-1} = q_i)\}_{i,j=1..N} \quad (6)$$

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

- La matrice d'émission B qui est définie par :

$$B = \{b_j(k) | b_j(k) = P(o_t = e_k | s_t = q_j)\}_{j=1..N;\ k=1..M} \quad (7)$$

$$Avec\ b_j(k) \geq 0\ et \sum_{k=1}^{M} b_j(k) = 1\ \forall j = 1..N$$

- Le vecteur d'initialisation π défini par :

$$\pi = \{\pi_i | \pi_i = P(s_1 = q_i)\}_{i=1..N} \quad (8)$$

$$Avec\ \pi_i \geq 0\ et \sum_{i=1}^{N} \pi_i = 1$$

Pour les HMM, le graphe de dépendances entre la séquence S des états cachés et la séquence O des observations est montré dans la Figure 45. Grâce à l'indépendance conditionnelle des observations sachant les états et à la propriété de Markov, la probabilité jointe peut être écrite sous la forme suivante:

$$\begin{aligned} P(O, S|\lambda) &= P(O|S, \lambda) P(S|\lambda) \\ &= \prod_{t=1}^{T} P(o_t|s_t) \prod_{t=2}^{T} P(s_t|s_{t-1})\, P(s_1) \\ &= \prod_{t=1}^{T} b_{s_t}(o_t) \prod_{t=2}^{T} a_{s_t s_{t-1}}\, \pi_{s_1} \end{aligned} \quad (9)$$

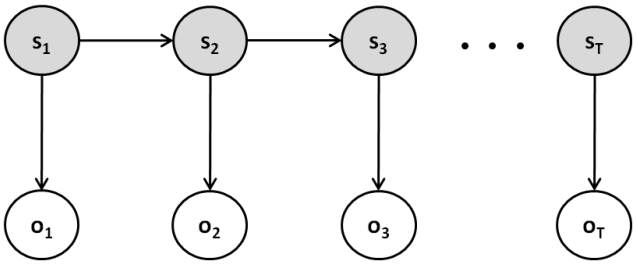


Figure 45: Le graphe de dépendances dans les HMM

Dans la théorie des HMM, on a principalement trois types de problèmes :

- Le problème d'apprentissage: étant donné une séquence d'observations O, comment apprendre les paramètres (A, B, π) du modèle λ pour maximiser $P(O|\lambda)$? Ceci est généralement résolu par l'algorithme de Baum-Welch [151] qui est en fait un cas particulier de l'algorithme EM (Expectation-Maximization) [152] bien connu et largement utilisé dans le domaine de l'apprentissage statistique.
- Le problème de l'inférence: étant donné un modèle $\lambda = (A, B, \pi)$ et une séquence d'observations O, quelle est la séquence d'états cachés la plus probable d'avoir généré cette séquence d'observations O ? Ce problème de décodage d'états cachés est résolu par l'algorithme de Viterbi [153].

- Le problème de reconnaissance ou aussi de classement: étant donnée une liste de modèles et une séquence d'observations, on cherche à sélectionner le modèle le plus optimal c.-à-d le modèle le plus probable d'avoir généré cette séquence. L'algorithme de Forward [148] peut être utilisé pour résoudre ce type de problème.

Dans la suite nous allons présenter comment nous avons conçu notre modèle de comportement à partir du paradigme des HMM.

4.3 Le modèle IDHMM

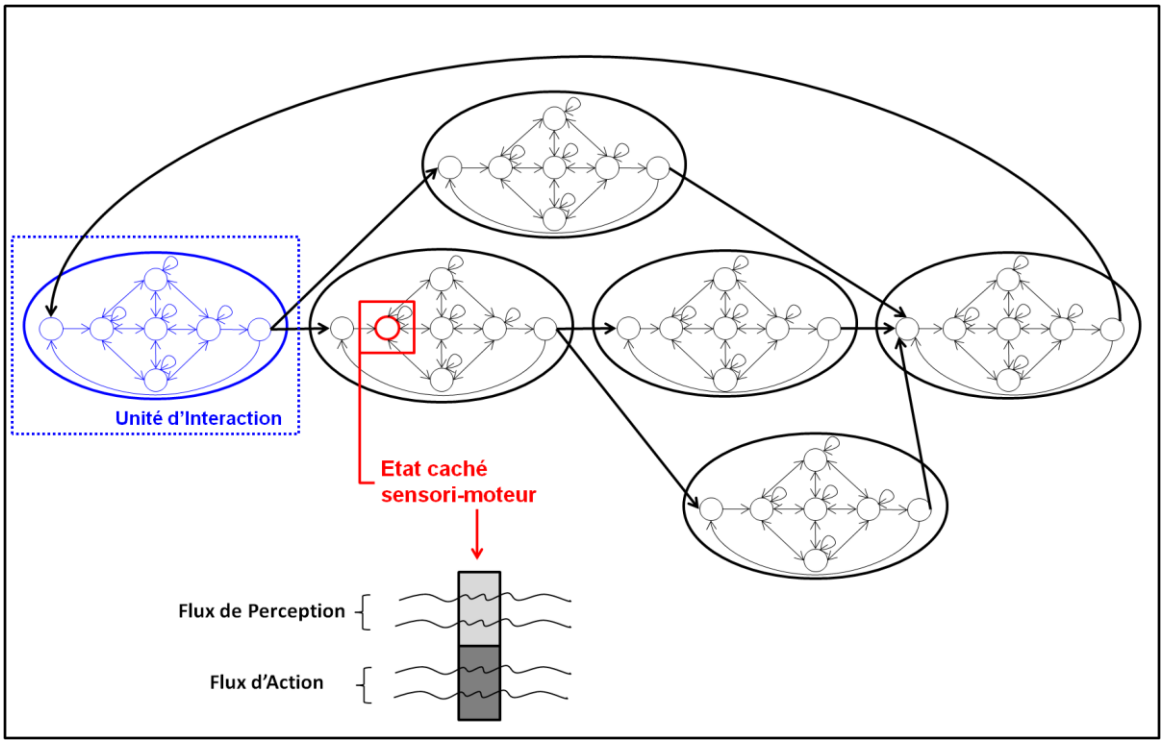


Figure 46: Gestion des boucles de perception-action dans un schéma probabiliste reliant observations, états sensori-moteurs, unité d'interaction et la syntaxe de la tâche (séquence des IU). La flèche rouge désigne la densité de probabilité, notamment les probabilités d'émission régissant la distribution des observations compte tenu de l'état caché à l'instant courant. Les observations perceptuelles sont annotées en gris clair alors que les observations d'action sont en gris foncé. Notez aussi que les observations peuvent combiner des trames actuelles ainsi que des trames du passé.

Une interaction peut être considérée comme une séquence de tâches discrètes, de sous-tâches ou d'activités [154]. Ainsi, dans ce qui suit, nous allons considérer une conversation située comme une séquence d'unités d'interaction (IU) (par exemple penser, informer, écouter, prendre le tour, etc.) qui structurent le comportement joint des partenaires. Lorsque les deux partenaires coopèrent, cette IU devrait idéalement refléter l'état mental conjoint des partenaires à ce moment de la conversation.

Dans le modèle que nous proposons, nous considérons que chaque interaction peut être segmentée en *P* unités d'interaction organisées selon une syntaxe prédéfinie. Chaque unité est modélisée par un seul modèle de Markov cachée discret (DHMM pour "Discrete Hidden

Markov Model") qu'on dénote λ_p ($avec\ p = 1..P$). Chaque modèle est caractérisé de la manière suivante :

$$\lambda_p = (A_p, B_p, \pi_p) \quad (10)$$

A_p représente la matrice de transition entre les états cachés du HMM λ_p, B_p représente la matrice d'émission alors que π_p représente la distribution d'initialisation des états. Chaque modèle λ_p dispose de n_p états cachés qui modélisent la micro-syntaxe des co-variations des comportement joints des partenaires spécifiques à cette unité d'interaction. Les états cachés sont bien des états sensori-moteurs (SMS) car ils modélisent à la fois les flux de perception et les flux d'action. L'enchaînement approprié de ces HMM (les λ_p) obéit à une syntaxe spécifique à la tâche abordée dans l'interaction et résulte de l'attention mutuelle et des actions collaboratives relatives à cette tâche. Dans notre modélisation, nous considérons les observations comme discrètes. Par exemple, les fixations du regard peuvent prendre une valeur parmi *K* régions d'intérêts possibles, un geste peut avoir plusieurs types (iconique, déictique, etc.), la parole aussi peut prendre une valeur particulière dans un vocabulaire fini, etc. Les modèles de comportement sensori-moteurs ainsi développés pilotent/déclenchent donc des contrôleurs gestuels paramétrés par le changement éventuel des cibles discrètes résultant de la transition d'un état sensori-moteur à l'autre.

Par conséquent, l'ensemble de l'interaction est modélisée par un HMM global discret λ de paramètres :

$$\lambda = (A, B, \pi) \quad (11)$$

A est la matrice de transition globale, *B* est la matrice d'émission globale et π est la matrice d'initialisation globale. Ce DHMM global λ n'est d'autre que l'enchainement et la concaténation des différents modèles HMM élémentaires λ_p, spécifiques aux différentes unités d'interaction (Figure 46). Le DHMM λ est composée de *N* états cachés avec :

$$N = \sum_{p=1}^{P} n_p \quad (12)$$

Comme mentionné précédemment, les états cachés (SMS) sont associés à des comportements sensori-moteurs conjoints et homogènes: étant donnée *T* la longueur d'une séquence d'interaction, le vecteur d'observations $O = (o_t)_{t=1..T}$ est en fait composé de deux flux:

- Le flux sensoriel $O^s = (o_t^s)_{t=1..T}$ qui recueille les observations de perception.
- Le flux moteur $O^m = (o_t^m)_{t=1..T}$ qui recueille les observations d'action.

Le vecteur d'observation est défini alors comme suit :

$$o_t = (o_t^s, o_t^m)_{t=1..T} \quad (13)$$

Dans le paragraphe suivant, nous allons présenter comment se fait l'apprentissage de ce modèle ainsi que le test.

4.3.1 Apprentissage

Afin d'apprendre le modèle HMM global λ (i.e. les trois matrices $A, B\ et\ \pi$), nous commençons par l'apprentissage séparé des modèles HMM spécifiques ($les\ (\lambda_p)_{p=1..P}$) à partir des données sensori-motrices étiquetées. Cette première étape peut être réalisée d'une manière assez directe grâce à un algorithme classique comme l'algorithme EM (Expectation-Maximisation) [152]. Une fois les modèles spécifiques appris, nous procédons comme suit:

- La matrice A du HMM globale λ contient les transitions intra-HMM et inter-HMM spécifiques. Les probabilités intra-HMM sont récupérées directement des matrices individuelles apprises $(A_p)_{p=1..P}$ et les probabilités inter-HMM sont calculées par comptage, directement à partir de la base de l'apprentissage.
- La matrice B est construite en concaténant les matrices d'émission individuelles $((B_p)_{p=1..p})$.
- La matrice π est calculée par comptage, directement à partir de la base de l'apprentissage.

Dans la pratique, à un instant t, seulement l'information perceptive est disponible et les actions doivent être générées selon ces indices perceptifs d'entrée. Pour cette raison, une fois obtenu le HMM global formé comme décrit ci-dessus, deux modèles sont extraits:

- Un modèle de reconnaissance de l'IU appelé λ_R. Ce modèle sélectionne uniquement les flux de perception c.-à-d. $o_t = o_t^s$ et effectue l'alignement optimal S^* des états sensori-moteurs (SMS) avec ces percepts. S^* est définie par :

$$S^* = \underset{S}{\operatorname{argmax}}\ P(S|O^s, \lambda_R) \quad (14)$$

- Un modèle génératif appelé λ_G dont la matrice d'émission ne contient que les observations d'action c.-à-d. $o_t = o_t^m$. Étant donné l'alignement optimal des SMS S^* effectué par λ_R, λ_G génère les actions adéquates. Plus précisément, étant donné un état SMS, λ_G va sélectionner dans chaque flux d'action l'observation ayant la probabilité maximale.

En divisant de cette manière le modèle global appris, nous disposons désormais d'un sensori-HMM (λ_R) et d'un motor-HMM (λ_G) qui répondent bien aux deux tâches cibles d'un modèle de comportement général, c.-à-d l'analyse de la scène et la génération d'un comportement pertinent et adéquat.

4.3.2 Dimension incrémentale

Pour que le modèle de comportement proposé soit pertinent, il est impératif que les deux phases (reconnaissance et génération) soient faites d'une manière incrémentale. Ceci

permettra de contrôler les latences et maintenir un temps de réponse acceptable pour un système interactif. Comme déjà évoqué, l'inférence des états cachés dans les HMM est effectuée essentiellement par l'algorithme de Viterbi [153][146].

4.3.2.1 Algorithme de Viterbi classique

L'algorithme de Viterbi estime la séquence d'état S^* la plus probable étant donné un flux sensoriel observé O et un modèle HMM λ:

$$S^* = \underset{S}{\operatorname{argmax}}\ P(S|O,\lambda) \tag{15}$$

Etant donné la nature linéaire d'un HMM (absence de cycles dans le graphe de dépendances, voir Figure 45), l'énumération de toutes les combinaisons possibles de séquences d'états n'est pas nécessaire. La meilleure séquence peut être obtenue par programmation dynamique tirant profil de la propriété Markovienne. L'alignement entre les observations et les états est effectué en deux étapes:

- Une étape "Forward" dans laquelle on calcule les vraisemblances partielles δ_t et on stocke le meilleur prédécesseur pour chaque état à chaque instant t dans une matrice ψ_t de pointeurs de backtracking. Les vraisemblances partielles sont définies par la formule suivante:

$$\delta_t(i) = \max_{S_1 S_2 .. S_{t-1}} (S_1 S_2 .. S_t = q_i, o_1 o_2 .. o_t|\lambda)$$
$$2 \le t \le T \ \ ; \ \ 1 \le i \le N \tag{16}$$

  Le calcul de cette étape se fait en trois temps:

  1. Initialisation:

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \le i \le N$$
$$\psi_1(i) = 0 \tag{17}$$

  2. Récursion:

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t) \qquad 2 \le t \le T \ \ ; \ \ 1 \le j \le N$$
$$\psi_t(j) = \underset{1 \le i \le N}{\operatorname{argmax}} [\delta_{t-1}(i) a_{ij}] \tag{18}$$

  3. Fin:

$$s_T^* = \underset{1 \le i \le N}{\operatorname{argmax}} [\delta_T(i)] \tag{19}$$

- Une étape "Backward" qui utilise ψ_t pour construire le chemin optimal à partir de la fin de la séquence d'observations:

$$s_t^* = \psi_{t+1}(s_{t+1}^*) \qquad t = T-1, T-2, .., 1 \tag{20}$$

La dernière étape implique que l'algorithme fonctionne en mode non-incrémental (ou hors-ligne), puisqu'on ne peut pas commencer le décodage des états qu'à la fin de la séquence d'observations. Afin d'effectuer le décodage d'une manière incrémentale (ou enligne), l'idée principale est d'exploiter un "backtracking" partiel au fur et à mesure de l'arrivée des données sensorielles et non pas un "backtracking" sur la totalité de la séquence. Pour résoudre cette problématique, on trouve dans la littérature essentiellement deux solutions : la fenêtre glissante et l'algorithme Short-Time Viterbi.

4.3.2.2 Fenêtre glissante

Plusieurs solutions utilisant une fenêtre glissante ont été proposées [155] [156] [157] [158]. L'approche consiste à diviser la séquence d'entrée en un ensemble de petites séquences de taille fixe et puis de les décoder de façon séparée et indépendante. Une astuce largement utilisée dans ce type d'approche est de considérer des fenêtres qui se chevauchent à 50% afin d'avoir un chemin final lisse et cohérent. Des résultats acceptables sont habituellement obtenus avec des larges fenêtres et des chevauchements importants ce qui implique un temps de latence relativement élevé.

4.3.2.3 L'algorithme Short-Time Viterbi

Une approche alternative à la méthode de la fenêtre glissante est L'algorithme connu sous le nom de Short-Time Viterbi (STV) [159] [160] [161]. L'idée centrale consiste à utiliser une fenêtre en expansion continue. A chaque nouvelle observation et en partant des différents états possibles, on examine les chemins partiels inférés par le backtracking dans cette fenêtre. Si un point de convergence (appelé aussi point de fusion) est trouvé, ce point va être inclus dans le chemin optimal et la fenêtre se rétrécit par derrière. On avance alors au point de fusion (la fenêtre s'élargit par l'avant) et on réitère la procédure. Un exemple de cet algorithme puisé dans la littérature [160] est démontré dans la Figure 47. Le principal avantage de cette méthode est que la solution est exactement équivalente à l'algorithme de Viterbi classique. L'inconvénient majeur est que lorsque le point de fusion n'est pas trouvé rapidement, on est obligé de continuer l'expansion de la fenêtre, ce qui implique un temps de latence non constant et relativement élevé.

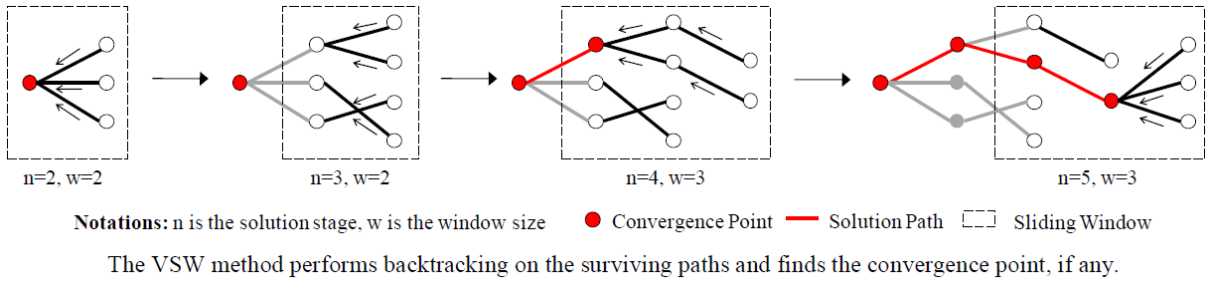


Figure 47: Un exemple du déroulement du Short-Time Viterbi, ici intitulé VSW ("Variable Switching Window") (figure reproduite de [160])

4.3.2.4 Notre approche

Dans notre modèle, nous avons adopté une version bornée de l'algorithme Short-Time Viterbi (BSTV). L'idée c'est que nous fixons un seuil au delà duquel le chemin ayant le maximum de vraisemblance est retenu lorsque il n'y a pas un point de fusion trouvé à partir de cet horizon. Bien que, la solution optimale ne soit pas toujours sélectionnée, le temps de latence est entièrement contrôlé. L'algorithme de BSTV est décrit brièvement comme suit :

```
1:  initiate δ1; ψ1; a=1;
2:  for each new frame b
3:    for each state j=1:N
4:        calculate δb(j) and ψb(j);
5:        backtracking: S*t−1 = ψt(j) with t=b:a+1;
6:        save the local path;
7:    end
8:    given all local paths find fusion point f;
9:    if (b-a<threshold and f exists)
10:       local path for t=a:f is selected; a=f+1;
11:   else if (b-a>=threshold)
12:       path with max likelihood is selected;
13:       f=b;a=b;
14:       b=b+1;
15:   else b=b+1;
16: end
```

En résumé deux cas de figures se présentent :

- La largeur de la fenêtre est encore inférieure au seuil fixé: si le point de fusion n'est pas atteint, nous continuons l'expansion de la fenêtre.
- La largeur de la fenêtre est égale au seuil fixé : si le point de fusion n'est pas encore atteint, le chemin le plus probable est retenu.

Nous allons montrer dans la pratique que, même pour des latences courtes, la performance n'est pas dégradée d'une manière significative lors du décodage des états sensori-moteurs. Avec l'algorithme BSTV, notre DHMM va être capable d'assurer les deux tâches de reconnaissance et de génération d'une manière incrémentale, tout en gardant une solution pertinente et en contrôlant efficacement les temps de latence, c'est pour cette raison que le modèle est appelé IDHMM ("Incremental Discrete Hidden Markov Model").

4.4 Le modèle IDHSMM

Une limitation majeure des HMM conventionnels est la modélisation des durées des états. Supposons que le modèle est dans un état quelconque q_i, la probabilité qu'on reste d unités temporelles dans le même état est :

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}) \qquad (21)$$

L'espérance de la durée $\overline{d_i}$ est calculée comme suit :

$$\overline{d_i} = \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}} \quad (22)$$

Par conséquent, la durée d suit implicitement une loi géométrique (ou exponentielle dans le cas continu) ce qui n'est pas approprié pour de nombreuses applications. Pour remédier à ce problème, les chaînes semi-Markoviennes cachées (HSMM) ont été introduites comme une extension aux HMM. Les HSMM présentent l'avantage de modéliser explicitement la durée de résidence dans chaque état [162]. La distribution des durées peuvent être discrète ou continue. D'autres appellations existent dans la littérature, par exemple "explicit duration HMM" [163][146], "variable-duration HMM" [164], "generalized HMM" [165] et aussi "segmental HMM" [166].

Les HSMM ont été appliqués avec succès dans plusieurs domaines tels que le traitement de la parole [167] [168], l'analyse des images et des vidéos [169] [170], la robotique [171], la sécurité des réseaux [172], la biologie [165][173] et les séries temporelles en finance [174]. Plus de détails sur la théorie des HSMM peuvent être trouvés dans [162]. En résumé, plusieurs approches ont été introduites pour résoudre les problème d'apprentissage et d'inférence dans les HSMM, notamment l'approche décrite dans le papier de Rabiner [146] qui a été initialement proposée par Ferguson [163] et améliorée ensuite par Levinson [164] et Mitchell [175]. Cette approche définit un algorithme de Forward-Backward qui évalue la probabilité jointe qu'un état se termine à un instant donné et une série d'observations jusqu'à cet instant-là. Un algorithme "Forward-Backward" plus efficace avec une complexité inférieure en temps et en mémoire a été proposé en 2003 by Yu and Kobayashi [176].

Yu and Kobayashi [176] supposent que la durée d'un état quelconque q_i est une variable aléatoire prenant la valeur d avec une probabilité égale à $p_i(d)$ ($d = \{1,2,..,L\}$ et L est la durée maximale que peut prendre un état). Un modèle semi-markovien (λ) est défini alors par 4 matrices qui sont A, B, π et D. D est la matrice des durées, elle définie comme suit:

$$D = \{p_i(d)\}_{i=1..N\ ;\ d=1..L}$$
$$avec \sum_{d=1}^{L} p_i(d) = 1 \quad \forall\, i = 1..N \quad (23)$$

On note par τ_t le temps résiduel ou de séjour pour un état s_t, le couple (q_i, d) est alors la réalisation du couple (s_t, τ_t). La chaîne va rester dans l'état q_i jusqu'à l'instant $t + d - 1$ et transitera vers un autre état à l'instant $t + d$. D'une manière générale la probabilité jointe d'un HSMM peut être écrite sous la forme suivante (qui est équivalente à l'équation 9 pour les HMM mais ayant des termes supplémentaires liés aux durées des états):

$$
\begin{aligned}
P(O,S|\lambda) &= P(O|S,\lambda)P(S|\lambda) \\
&= \prod_{r=1}^{R}\prod_{u=1}^{\tau_r} P\left(o_{f(u,r)}|s_r\right)\prod_{r=2}^{R} P(s_r|s_{r-1})\,P(\tau_r|s_r)P(s_1)P(\tau_1|s_1) \\
&= \prod_{r=1}^{R}\prod_{u=1}^{\tau_r} b_{s_r}(o_{f(u,r)})\prod_{r=2}^{R} a_{s_{r-1}s_r}\,p_{s_r}(\tau_r)\pi_{s_1}p_{s_1}(\tau_1)
\end{aligned}
\tag{24}
$$

R étant le nombre des états visités vérifiant $\sum_{r=1}^{R}\tau_r = T$ et $f(u,r) = u + \sum_{z=0}^{r-1}\tau_z$ (sachant que $\tau_0 = 0$).

Selon [176], la probabilité jointe de transiter d'un état q_i à un état q_j $(i \neq j)$ et d'une séquence d'observations O est donnée par la formule suivante:

$$
P\left(O, s_{t-1} = q_i, s_t = q_j\right) = \alpha_{t-1}(i,1)a_{ij}b_j(o_t) * \left(\sum_{d=1}^{L} p_j(d)\,\beta_t(j,d)\right) \tag{25}
$$

α est la variable "Forward" et β est la variable "Backward". Ces deux variables sont définies ci-dessous (sachant que $o_x^y = \{o_t; x \leq t \leq y\}$, avec cette notation o_1^T est égale à O):

$$
\alpha_t(i,d) = P[o_1^t, (s_t,\tau_t) = (q_i,d)] \tag{26}
$$

$$
\beta_t(i,d) = P[o_{t+1}^T|(s_t,\tau_t) = (q_i,d)] \tag{27}
$$

Les formules récursives de α et β peuvent être retrouvées dans [176]. Maintenant la probabilité jointe de transiter vers un état q_j, de rester d unités temporelles et d'observer O est donnée par la formule suivante:

$$
P\left(O, s_{t-1} \neq q_j, s_t = q_j, \tau_t = d\right) = \left(\sum_{i\neq j}\alpha_{t-1}(i,1)a_{ij}\right) * b_j(o_t)p_j(d)\beta_t(j,d) \tag{28}
$$

L'algorithme de "Forward Backward" proposé dans [176] se base sur les deux formules de probabilité jointe (25 et 28) pour faire l'estimation des états cachées (inférence) et l'estimation des paramètres (apprentissage). Dans notre travail, nous avons utilisé une version récente de l'algorithme de Yu and Kobayashi [177] et le code fourni par les auteurs. En terme de concept et structure le modèle IDHSMM est relativement similaire au modèle IDHMM.

4.5 Application

Dans cette partie, nous allons tester le modèle IDHMM sur un jeu de données particulier. Nous allons comparer les performances et les propriétés de ce modèle avec les classifieurs du Chapitre 3. Un deuxième modèle IDHMM qui dispose d'une initialisation particulière inférée des données est aussi présenté et testé (IDHMM modifié). Nous terminerons ce chapitre par le test du modèle IDHSMM (chaînes semi-markoviennes cachées) ainsi qu'une comparaison avec le modèle IDHMM.

4.5.1 Données

Les données utilisées dans cette application sont celles de la première interaction intitulée "Jeu de répétition de phrases" présentée dans la section 2.2. Nous rappelons qu'il s'agit d'une interaction entre deux sujets (un sujet principal appelé "LN" et un interlocuteur avec 10 dyades en total). Pour chaque dyade, nous disposons des activités vocales (v_1, v_2) et des regards (g_1, g_2) de chacun. Sept unités interactionnelles ont été annotées semi-automatiquement notamment "lire" , "se préparer" , "parler" , "attendre" , "écouter" , "réfléchir" et "autre". Quatre modèles sont évalués et comparés : le modèle IDHMM initial, les classifieurs, le modèle IDHMM modifié et le modèle IDHSMM.

4.5.2 Modèles et paramètres

4.5.2.1 IDHMM

Nous voulons tester la capacité du IDHMM dans l'estimation de l'IU du sujet principal "LN" et la prédiction de son propre comportement du regard g_1 compte tenu de son activité vocale v_1 ainsi que l'activité vocale v_2 et le regard g_2 de son interlocuteur. Nous utilisons pour ceci le modèle de reconnaissance λ_R pour estimer la séquence d'états sensori-moteurs s_t sachant $o_t = (v_1, v_2, g_2)$ puis le modèle de génération λ_G pour synthétiser son regard g_1 sachant s_t (Figure 48). Vu le jeu de données utilisé, notre modèle est composé de 7 DHMM spécifiques $((\lambda_p)_{p=1..7})$ correspondants aux 7 IU ("lire", "se préparer" , "parler" , "attendre" , "écouter" , "réfléchir" et "autre"). Le nombre d'états cachés par IU a été fixé à 5. Ce qui donne 35 états cachés (SMS) pour l'ensemble du modèle IDHMM. Des topologies avec un nombre fixe de 4 et 6 états cachés par IU ont également été testées, mais n'ont abouti à aucune différence significative dans les performances. Le reste des paramètres (c.-à-d (A, B, π)) ont été initialisés de façon aléatoire avant de lancer l'apprentissage. En pratique, le toolkit HTK [178] a été utilisé pour l'apprentissage et le toolkit PMTK3 (Matlab) [179] pour la mise en œuvre du modèle IDHMM et le test.

4.5.2.2 Classifieurs (Rappel)

Pour les SVM et les arbres de décisions, un premier classifieur a été utilisé pour estimer l'IU à partir de (v_1, v_2, g_2). Ensuite, un deuxième classifieur est utilisé pour estimer le regard g_1 à partir des mêmes données. Pour plus de détails voir le Chapitre 3.

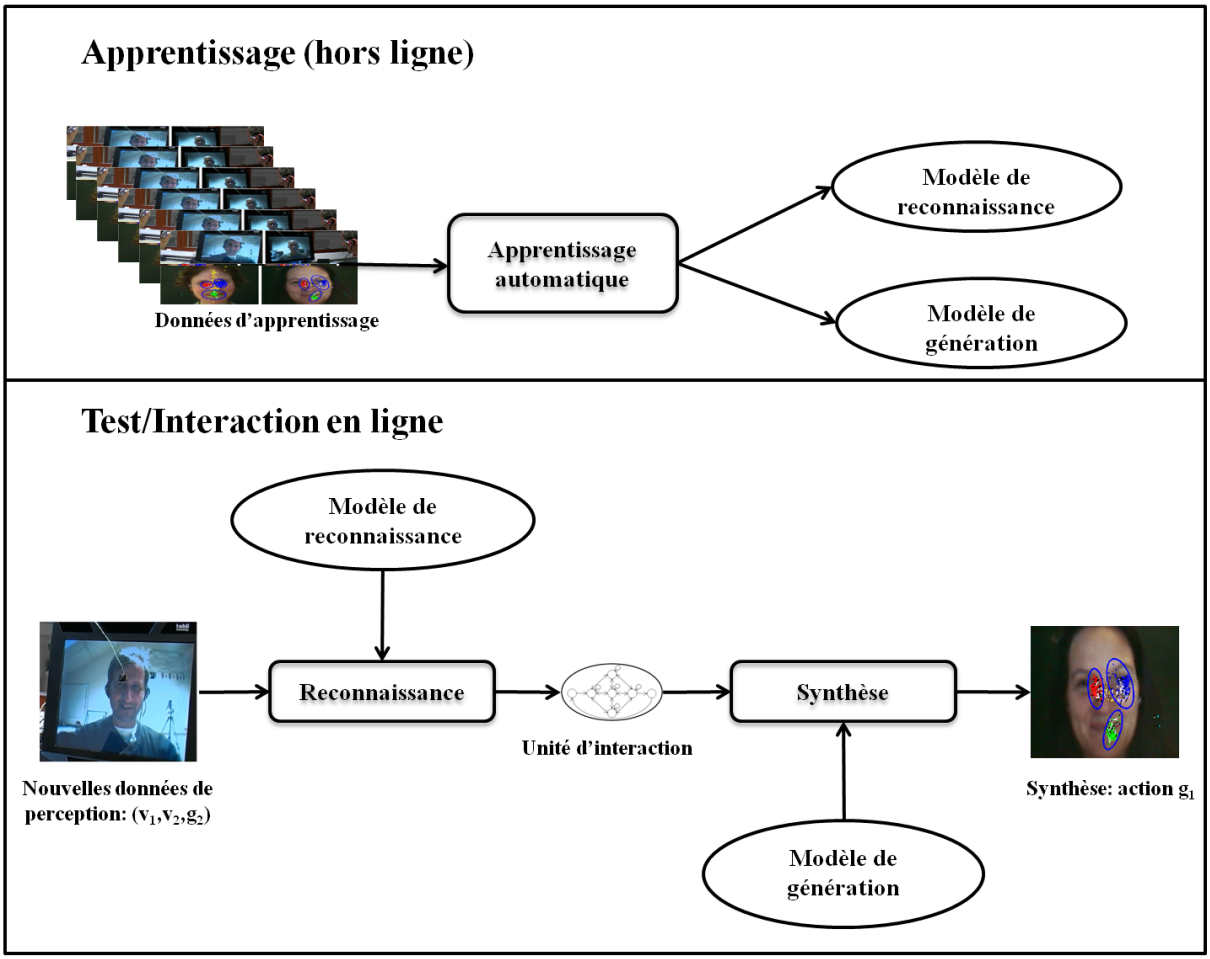


Figure 48: Le modèle IDHMM en pratique (apprentissage et test)

4.5.2.3 IDHMM modifié

Dans le modèle IDHMM initial, nous avons fixé le nombre de SMS à 5 par unité d'interaction, donc un modèle complet de 35 SMS. Or, le processus cognitif interne d'une unité d'interaction peut probablement différer d'une unité à l'autre. Par exemple, l'IU "écouter" sera sûrement plus étalée dans le temps et plus complexe en structure que l'IU "se préparer". On peut s'attendre qu'un modèle avec nombre de SMS et structure interne variables par IU (comme celui de la Figure 49) soit mieux adapté qu'un modèle avec des IU assez identiques (comme celui de la Figure 46). En partant de cette constatation, nous avons proposé une approche qui permet de faire varier le nombre de SMS cachés par IU selon un critère objectif. Un deuxième constat concerne la performance de l'algorithme de Baum-Welch qui est de nature itérative et qui dépend fortement de l'initialisation des paramètres A, B et π. Classiquement, cette initialisation est faite de manière aléatoire. Ce qui est clairement sous-optimal. Selon la distribution des observations dans la base d'apprentissage, nous proposons une méthodologie qui nous permet d'automatiquement (1) sélectionner un nombre suffisant d'états sensori-moteurs cachés représentant une IU donnée et (2) initialiser les probabilités d'émission pour chaque état caché sélectionné. Nous appelons le nouveau IDHMM modifié "IDHMM avec dimensionnement et initialisation par les données". Dans cette approche, nous considérons trois hypothèses:

1. Chaque vecteur d'observation sélectionné est associée à un état sensori-moteur unique.
2. Un vecteur d'observation n'est sélectionné que s'il a un poids significatif dans la distribution des données (un nombre significatif d'occurrences).
3. Le nombre total d'états à sélectionner doit rester dans un intervalle raisonnable.

Ces hypothèses ne sont pas strictes. Elles nous permettent d'initialiser les paramètres du modèle à des valeurs sémantiquement significatives. Cependant, durant la phase d'apprentissage assurée par l'algorithme EM, les paramètres appris peuvent évoluer en violant certaines hypothèses (notamment les hypothèses 1 et 2). Par exemple, la première hypothèse qui associe un vecteur d'observation à un SMS unique permet, entre autres, d'initialiser les probabilités d'émission de cet état. Dans la matrice d'émission, il y aura des 1 pour les événements qui figurent dans le vecteur d'observations et des 0 pour les autres événements. L'algorithme d'apprentissage EM peut ensuite revoir et réévaluer ces premières distributions d'émission et les probabilités de transition ainsi que les frontières des IU. Par conséquent, les paramètres finaux peuvent associer un état sensori-moteur à un ou plusieurs vecteurs d'observations.

Concrètement l'approche est décrite en trois étapes :

1. Nous recueillons d'abord tous les vecteurs d'observations d'une IU donnée.
2. Ensuite, nous listons ces observations dans un ordre décroissant en fonction de leurs fréquences d'occurrences.
3. Enfin, nous sélectionnons un certain nombre d'observations (qui correspondent aux futurs SMS) dont la fréquence cumulée représente au moins 90% de la vérité terrain (hypothèse 2). Le nombre de SMS à garder doit être dans l'intervalle [5,10] afin de préserver la pertinence du modèle (hypothèse 3). Avec un nombre assez faible, on risque de perdre de l'information (sous-apprentissage) alors qu'avec un nombre très élevé, le modèle peut s'avérer très compliqué et on peut tomber dans le sur-apprentissage.

Un exemple de notre méthode pour déterminer le nombre approprié des états cachés pour l'IU "écouter" est montré dans la Figure 50. La figure montre que 15 états doivent être choisis afin d'obtenir 90% de la fréquence cumulée. Cependant, nous avons fixé le nombre maximal à 10 et par conséquent seulement 10 états cachés peuvent être sélectionnés. Avec un tel nombre, nous conservons 76% de l'information. Pour cet exemple, nous montrons aussi dans le Tableau 2 les trois premiers SMS sélectionnés ainsi que les observations correspondantes. C'est ce type d'information qui va nous permettre d'initialiser les probabilités d'émission et c'est exactement ce qui est montré dans le Tableau 3 (dans lequel on voit comment se fait l'initialisation des probabilités d'émission pour l'état n° 3 de l'IU "écouter" à partir des SMS choisis).

Cette approche de dimensionnement et d'initialisation axée sur les données a été utilisée pour l'ensemble des IU. Le nombre de SMS sélectionnés par IU et leurs fréquences cumulées correspondantes sont montrés dans le Tableau 4. On voit que le nombre diffère d'une unité d'interaction à une autre (par exemple 5 pour l'IU "lire" et 10 pour l'IU "réfléchir") ce qui est

tout à fait raisonnable vu l'hétérogénéité des processus cognitifs des différentes unités d'interactions. En conséquence, le nouveau modèle IDHMM contient 59 états cachés dont les probabilités d'émission ont été initialisées à partir des distributions observées. Une fois que cette initialisation est faite, le reste de l'apprentissage est similaire au premier modèle. Dans la suite, nous appelons le nouveau modèle IDHMM modifié ou aussi IDHMM avec dimensionnement et initialisation par les données.

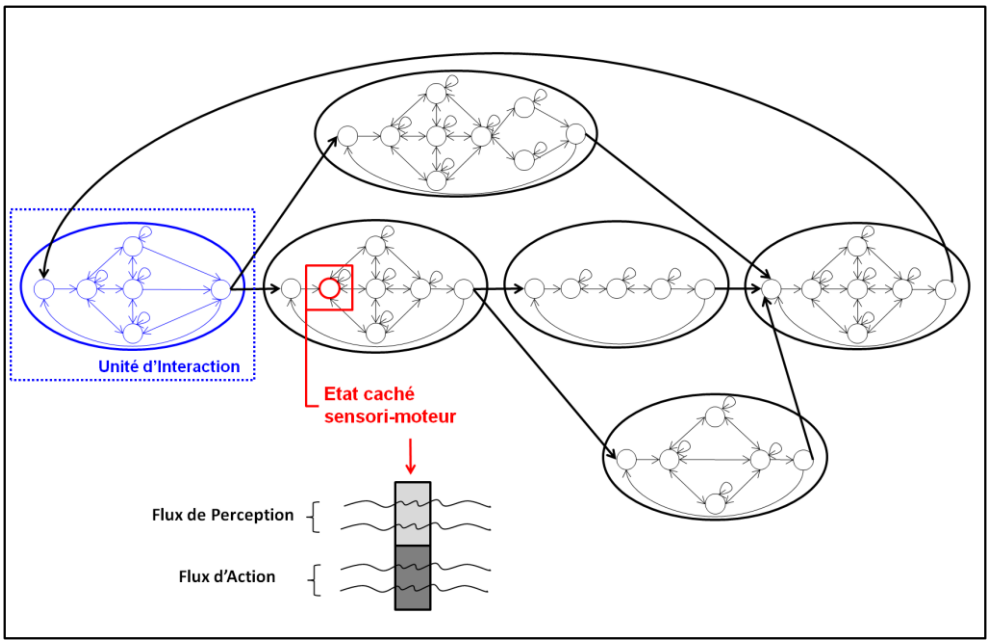


Figure 49: IDHMM avec nombre de SMS variable par IU

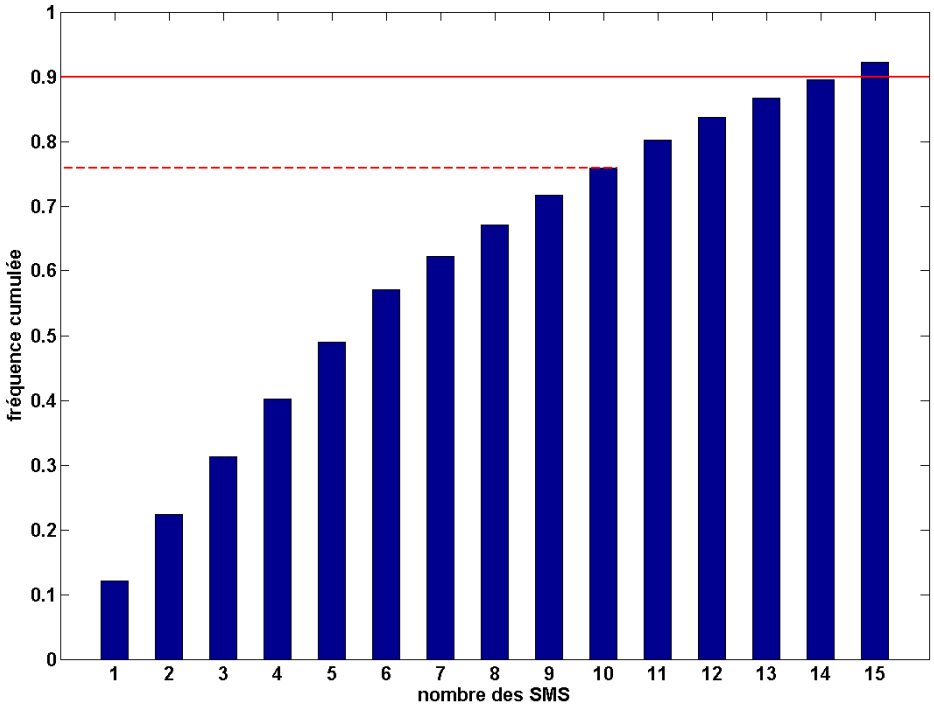


Figure 50 : Un exemple de notre méthode pour déterminer le nombre d'états sensori-moteurs (SMS) pour l'IU "écouter". Dans ce cas, le nombre maximal de 10 SMS est sélectionné.

	Etat1	Etat2	Etat3
Regard de l'interlocuteur	œil gauche	œil gauche	œil gauche
Parole de l'interlocuteur	parole	parole	parole
Regard du sujet principal	œil gauche	bouche	œil droite
Parole du sujet principal	pas de parole	pas de parole	pas de parole

Tableau 2: Les trois premiers états sensori-moteurs sélectionnés pour l'IU "écouter" et les observations dominantes avec lesquelles ils ont été initialisés.

Evénement	œil gauche	œil droite	bouche	visage	autre
Densité de probabilité	1	0	0	0	0

(a)

Evénement	pas de parole	parole
Densité de probabilité	0	1

(b)

Evénement	œil gauche	œil droite	bouche	visage	autre
Densité de probabilité	0	1	0	0	0

(c)

Evénement	pas de parole	parole
Densité de probabilité	1	0

(d)

Tableau 3: Initialisation des densités de probabilité pour l'état n°3 de l'IU "écouter" avec (a) Regard de l'interlocuteur (b) Parole de l'interlocuteur (c) Regard du sujet principal et (d) Parole du sujet principal

	Nombre	Poids
Lire	5	99 %
Se préparer	8	91 %
Parler	10	89 %
Attendre	9	91 %
Ecouter	10	76 %
Réfléchir	10	85%
Autre	7	90 %

Tableau 4: Le nombre de SMS par IU et leurs fréquences cumulées correspondantes

4.5.2.4 IDHSMM

Comme on le verra dans la section des résultats, la procédure du dimensionnement et d'initialisation axée sur les données du modèle IDHMM présente quelques propriétés intéressantes, notamment dans la capture des cycles sensori-moteurs. Pour cette raison, nous avons choisi de garder la même structure pour le IDHSMM: notre modèle contient donc 59 états cachés et a été initialisé de la même manière que celle décrite dans le paragraphe précédent. Par rapport au modèle HMM, notre modèle HSMM à besoin d'une matrice D supplémentaire, dans laquelle chaque ligne décrit une distribution discrète de la durée d'un état sensori-moteur. Nous rappelons que D est définie de la manière suivante:

$$D = \{p_i(d)\}_{i=1..N\ ;\ d=1..L}$$
$$avec \sum_{d=1}^{L} p_i(d) = 1 \quad \forall\, i = 1..N \tag{29}$$

Où N est le nombre d'états cachés (59 états sensori-moteurs) et L est la durée maximale qu'un état peut avoir. Dans notre base d'apprentissage, L est égal à 101 trames (~ 4 secondes). La matrice D est estimée à partir de données en comptant pour chaque état i le nombre de fois qu'on reste une durée d. La distribution de probabilité discrète pour chaque état i est ensuite calculée. Pour le reste des paramètres, il n'y a pas de différence dans le processus d'apprentissage par rapport à l'approche suivie dans le IDHMM modifié. Ainsi, le IDHSMM est entièrement spécifié par $\lambda = (A, B, D, \pi)$. Basé sur le concept du point de fusion [159] [160], l'algorithme d'inférence (Forward-Backward proposé par Yu [177]) a été modifié afin de pouvoir faire la reconnaissance et la génération d'une manière incrémentale.

4.5.2.5 Evaluation

Pour tous les modèles, nous appliquons le principe de la validation croisée (10-fold cross validation) sur nos données: 9 sujets ont été utilisés pour l'apprentissage tandis que le dixième pour le test.

Comme déjà évoqué dans le paragraphe 3.3.5, le taux de reconnaissance est utilisé pour évaluer la reconnaissance des unités interactionnelles, et la F-mesure issue de la distance de Levenshtein [142] est utilisée pour évaluer la génération du regard. De plus, pour les modèles à base de HMM (IDHMM initial, IDHMM modifié et IDHMM), nous ne sommes pas contenté des taux de classification et de génération mais nous avons étudié d'autres priorités comme le bouclage sensori-moteur entre les états cachés et les durées des fixations générées par les différents modèles. Dans la section suivante, nous allons discuter des résultats de tous les modèles proposés.

4.6 Résultats

Nous commençons par présenter en détails les résultats du modèle IDHMM ainsi que d'autres propriétés découvertes en utilisant ce que nous appelons "les modèles personnels" (un modèle personnel est un modèle appris à partir d'une seule interaction, la description détaillée se trouve dans la section 4.6.2). Ensuite, nous comparons les performances du

IDHMM avec celles des classifieurs. Puis, les propriétés résultantes de l'initialisation axée sur les données (IDHMM modifié) sont discutées. Enfin, en comparant les performances des HSMM à celles des HMM, nous démontrons la pertinence et l'efficacité de la modélisation des durées des états SMS.

4.6.1 Résultats du modèle IDHMM

Dans la première tâche qui consiste à reconnaitre les unités interactionnelles, nous avons enregistré un taux moyen de 92%. Ce taux a été enregistré avec la version initiale du Short-Time Viterbi (STV), c.-à-d. sans fixer un seuil lors du décodage des états sensori-moteurs. D'autant plus, comme on peut le constater dans la Figure 52, ce taux élevé est confirmé par une bonne capture de la structure de l'interaction. En comparant avec la vérité terrain, le modèle IDHMM arrive bien à estimer les transitions entres les différentes unités d'interaction. De plus, un décodage hors ligne avec l'algorithme de Viterbi classique a été effectué et les résultats trouvés étaient identiques à celle du STV ce qui confirme l'efficacité et les bonnes performances de cet algorithme.

Cependant, le problème avec le STV est la non-maîtrise du temps de latence comme nous l'avons déjà expliqué. En effet le STV cherche un point de convergence et lorsque ce dernier n'est pas trouvé la fenêtre de décodage continue à s'élargir jusqu'à le trouver. Dans notre analyse des latences, nous avons trouvé que 80% des latences sont inférieures à 5 trames (200 ms) ce qui est à première vue assez acceptable. Néanmoins, les valeurs extrêmes existent, la latence maximale enregistrée dans notre cas était de 259 trames (> 10 secondes) ce qui représente un retard inadapté pour une application de temps réel comme l'interaction sociale. Comme déjà mentionné, nous avons proposé une version bornée du STV (BSTV) afin de limiter et maitriser les retards de réponse. Les seuils testés étaient 1, 2, 5, 10 et 20 trames (voir Figure 51).

Théoriquement, un compromis optimal doit être adopté en raison de la relation inverse entre les performances et les latences. Or, en regardant la Figure 52, nous pouvons constater que notre IDHMM est capable de se rapprocher de la vraie trajectoire même avec de faibles seuils (latences). Ceci se fait au détriment d'une légère dégradation de la reconnaissance des IU, soit 89% pour un seuil égal à 1 trame (Figure 51). En outre, le rendement moyen de génération (59%) n'est pas affecté et reste pratiquement le même pour tous les seuils. Le modèle IDHMM avec une latence d'une trame dispose alors d'un taux de 89% pour la détection des unités d'interaction et d'un taux de 59% pour la production du regard. Cette performance élevée pour un tel seuil est principalement due à la proportion forte des latences très courtes: les écarts par rapport à la trajectoire optimale se rétrécissent rapidement lorsque des repères robustes sont rencontrés. Un autre facteur important est la syntaxe contrainte de la tâche: l'enchaînement des sous-tâches est relativement régulier, ce qui contraint l'alignement des unités d'interaction.

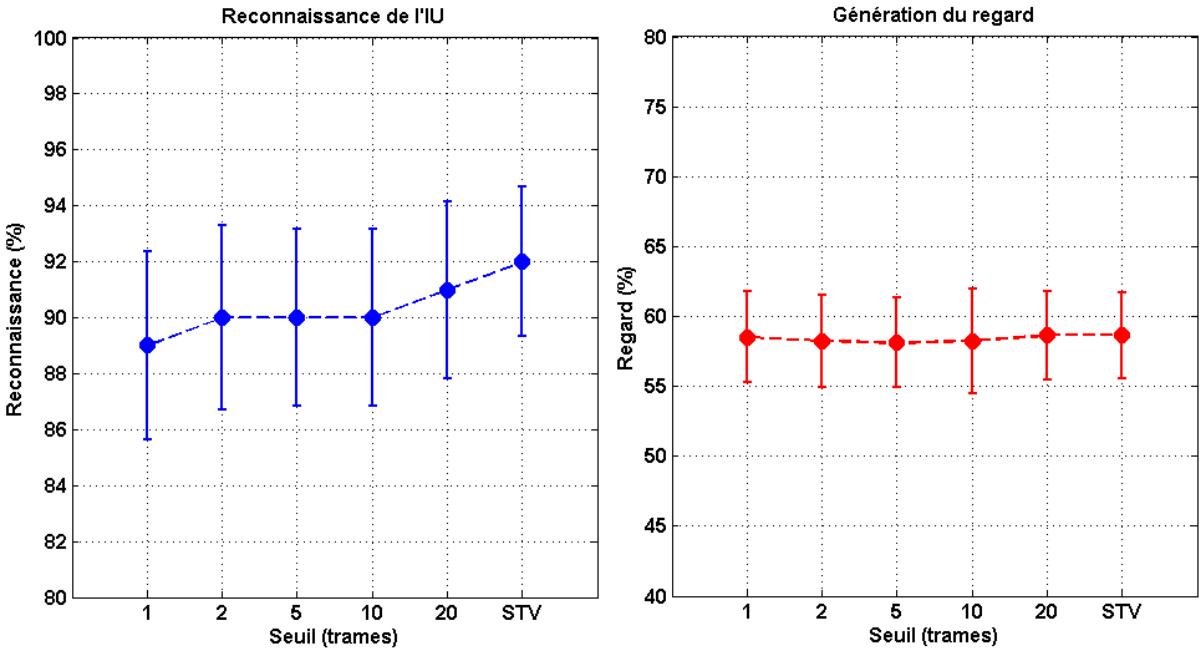
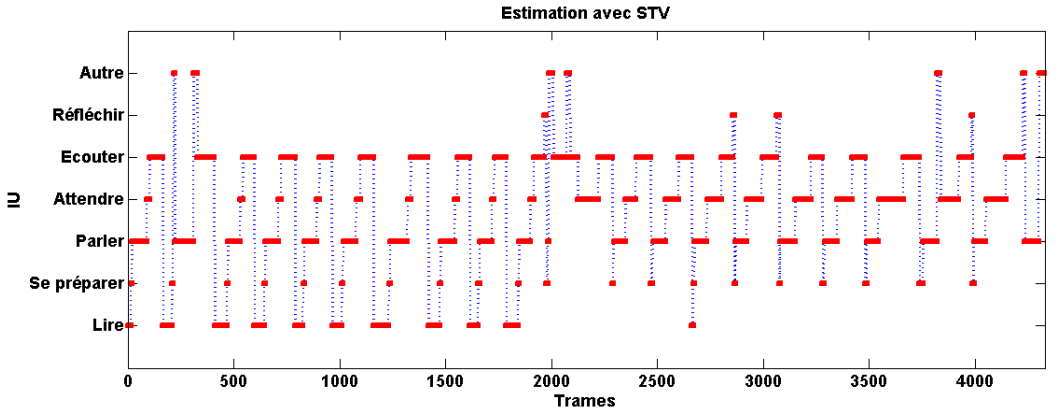
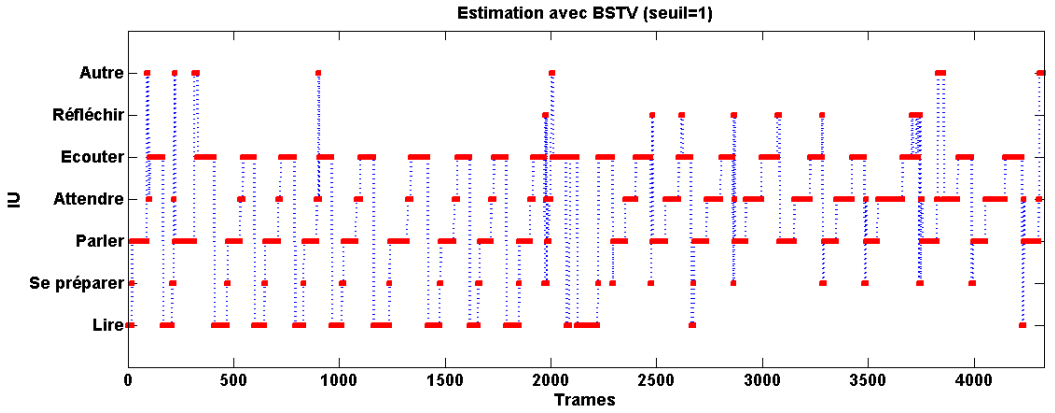


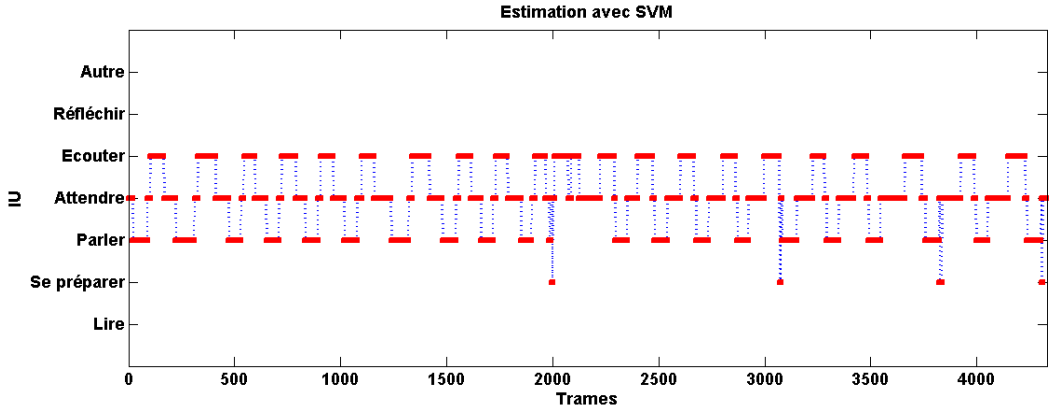
Figure 51: Résultats de reconnaissance et de génération en fonction des seuils choisis: pour les seuils 1,2,5,10,20, l'algorithme BSTV est appliqué. STV veut dire absence de seuil. Nous constatons une légère dégradation de la reconnaissance des IU, soit 89% pour un seuil égal à 1 trame vs. 92% pour la version STV. Nous remarquons également que les performances de la génération ne sont pas affectées par les seuils bas.



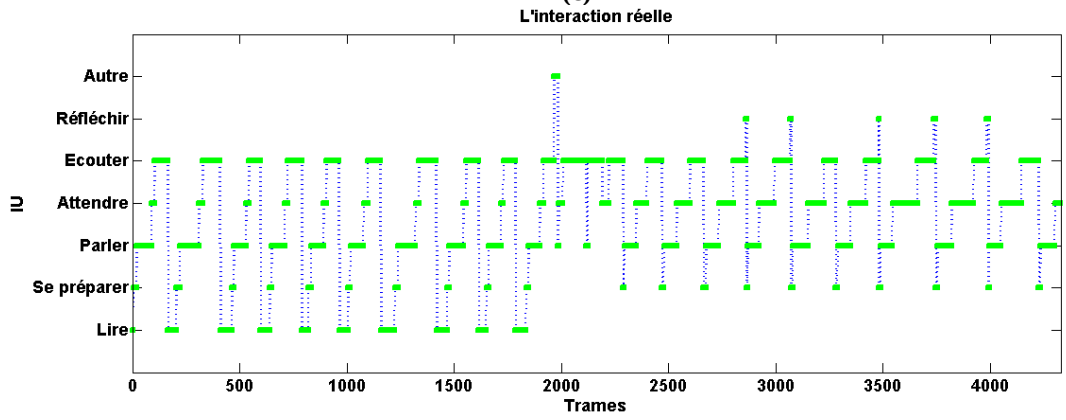
(a)



(b)



(c)



(d)

Figure 52 : Estimation des unités d'interaction (IU) pour un sujet spécifique (a) en utilisant IDHMM (pas de seuil) (b) en utilisant IDHMM (seuil = 1) (c) en utilisant SVM (d) le chemin réel des IU

4.6.2 Modèles personnels

Jusqu'à ce point, le modèle IDHMM est appris d'une manière à être indépendant d'un interlocuteur particulier et cumule les interactions effectuées par le tuteur humain avec plusieurs interlocuteurs. Puisque nous utilisons une validation croisée, le modèle testé sur un interlocuteur est appris à partir des neuf autres interactions. Nous annotons ce type de modèle par II (II pour "Interlocutor Independent"). Un modèle II peut être vu comme le modèle moyen sur neuf interlocuteurs. Or, certains travaux de la littérature [180] [80] affirment que les modèles moyens ne sont pas toujours les meilleurs et que, dans certaines applications, un modèle appris sur un seul interlocuteur peut donner des résultats beaucoup plus intéressants. L'idée consiste à dire que le modèle moyen agrège les comportements de plusieurs personnes et finit par omettre les subtilités et les spécificités du comportement humain alors qu'un modèle personnel (appris sur un seul interlocuteur) arrive à garder ces propriétés. L'enjeu est alors de sélectionner dans une bibliothèque de modèles comportements lequel sera le plus adapté au locuteur courant, soit en calculant une distance dynamique entre modèles et données observées, soit en calculant cet index avec des données indépendantes. En partant de cette idée, nous avons construit également des modèles qui sont dépendants et relatifs à un interlocuteur particulier. Nous les notons les modèles ID (ID pour "Interlocutor Dependent"). Pour chaque interlocuteur, un ensemble de neuf modèles personnels est ainsi appris en utilisant les données des autres interlocuteurs. En phase de test, un modèle ID appris sur un sujet sera testé sur les neuf autres sujets restants. Dans notre application, les résultats ne confirment pas les études citées ci-dessus, nos modèles II sont plus performants en terme de reconnaissance et génération que les modèles ID. Dans le cadre de notre étude, les modèles moyens sont meilleurs que les modèles personnels.

Néanmoins, les résultats des modèles ID ont été utilisés pour étudier la proximité sociale et les relations entre les sujets. En effet, une analyse de positionnement multidimensionnel ("Multidimensional Scaling" ou MDS [181]) a été effectuée sur l'ensemble des sujets en utilisant les taux de reconnaissance et de génération des modèles personnels correspondants à chaque sujet. Chaque sujet est représenté par deux vecteurs: un vecteur de 9 taux de reconnaissance et un vecteur de 9 taux de génération des données des autres. La MDS calcule une matrice de similarité entre les sujets et affecte à chacun une position dans un espace de deux dimensions. Dans notre cas, la MDS nous a permis de faire une analyse de proximité entre les comportements des différents interlocuteurs. Cette analyse a révélé deux groupes de sujets (étudiants et collègues, voir Figure 53), ce qui confirme une relation sociale connue a priori entre le sujet principal "LN" et ses interlocuteurs. Cette étude montre alors que "LN" se comporte avec ses collègues d'une manière significativement différente que celle avec ses étudiants. En particulier, son comportement de regard varie selon que la personne en face soit étudiante ou collègue. Cette connaissance a priori peut donc permettre de choisir parmi des modèles indexés par relation sociale.

Ce résultat consolide une fois de plus le fait que le regard est un signal très social et que les traits de personnalités et les relations de dominance se reflètent parfaitement dans les comportements du regard humain. Il est intéressant de noter l'intérêt d'accéder à des distances

entre modèles de comportements plutôt qu'à des distances basées sur des statistiques à long-terme sur les signaux bruts. En prenant en compte la structure de l'interaction, les modèles permettent une estimation plus robuste et contrastée des indices multimodaux de la relation sociale. Pentland et ses collègues [36] [37] ont déjà souligné que les modèles de comportements conjoints peuvent capturer des indices adaptatifs et subtils qui signalent des relations sociales préexistantes ou en développement. De notre part, ce genre d'exploration et de modélisation mérite de plus amples recherches.

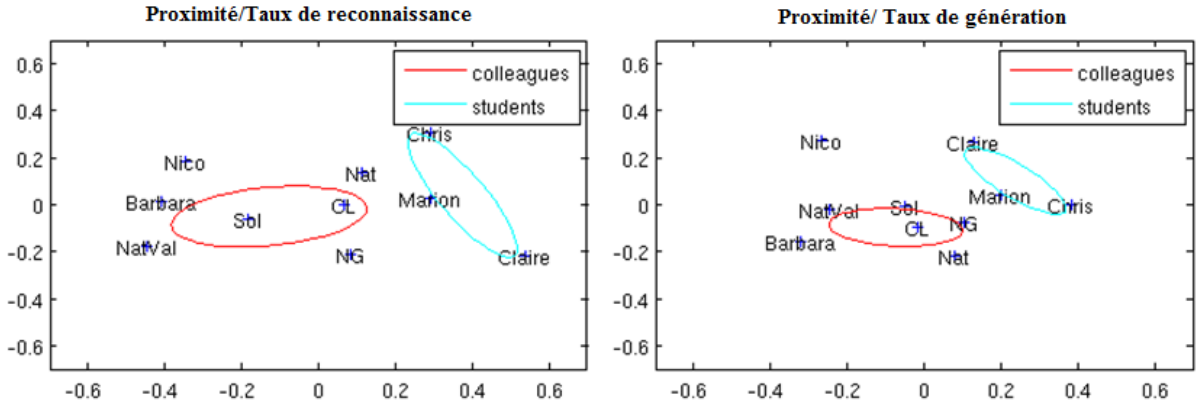


Figure 53 : Une projection MDS des performances des modèles personnels montre la proximité sociale entre les comportements des interlocuteurs. Notez bien la cohérence avec la relation sociale existante entre "LN" et ses interlocuteurs.

4.6.3 Comparaison avec les classifieurs

Dans ce paragraphe, nous comparons notre IDHMM séquentielle avec les SVM et les arbres de décision qui sont intrinsèquement non-séquentiels. La Figure 54 montre clairement qu'il n'y a pas de différence significative entre les deux classificateurs (résultat déjà vu). Cependant, le modèle IDHMM (avec l'algorithme BSTV et un seuil égal à 1) dépasse significativement les deux classifieurs dans les deux tâches ($p < 0{,}05$) . Pour la reconnaissance des IU, le taux du IDHMM est de 89% (vs. 81% pour les classifieurs) et pour génération du regard le taux de l'IDHMM est de 59% (vs. 50% pour les classifieurs). En outre, la Figure 52 montre que le modèle de IDHMM est plus efficace dans la détection de la structure de l'interaction. Nous pouvons voir que la trajectoire estimée des unités d'interaction reflète correctement la syntaxe prédéfinie de la tâche. En comparaison, le SVM a plus de difficulté à capter l'organisation réelle de la trajectoire et ignore certaines courtes unités d'interaction. Cet écart de performance est principalement dû aux contraintes séquentielles imposées par les HMM qui forcent le passage du chemin l'alignement optimal entre observations et modèle par certains états. Ces résultats confirment notre intuition que les HMM sont les modèles les mieux adaptés pour ce type d'application.

Dans le Chapitre 3, nous avons réussi à palier relativement les lacunes temporelles des SVM et à améliorer nettement leurs résultats en injectant une mémoire de deux secondes dans le modèle. Nous rappelons que le modèle SVM avec mémoire dispose ainsi de 6 variables d'entrée $v_1^t, v_2^t, g_2^t, v_1^{t-h}, v_2^{t-h}$ et g_2^{t-h}. Aux fins de test et de comparaison, nous avons utilisé

également ces variables comme entrée pour le modèle IDHMM. Les résultats nous ont montré qu'il n'y avait pas de différence significative pour la reconnaissance des IU. Par contre, nous avons enregistré une amélioration significative pour le taux de génération: 63% pour le modèle avec mémoire explicite versus 59% pour l'IDHMM disposant seulement comme entrée $v_1^t, v_2^t \; et \; g_2^t$. Les HMM gèrent implicitement une mémoire temporelle mais le fait de rajouter encore une information temporelle permet d'en améliorer les performances. En résumé, le modèle IDHMM comparé aux classifieurs est un modèle robuste et bien adapté à notre problématique.

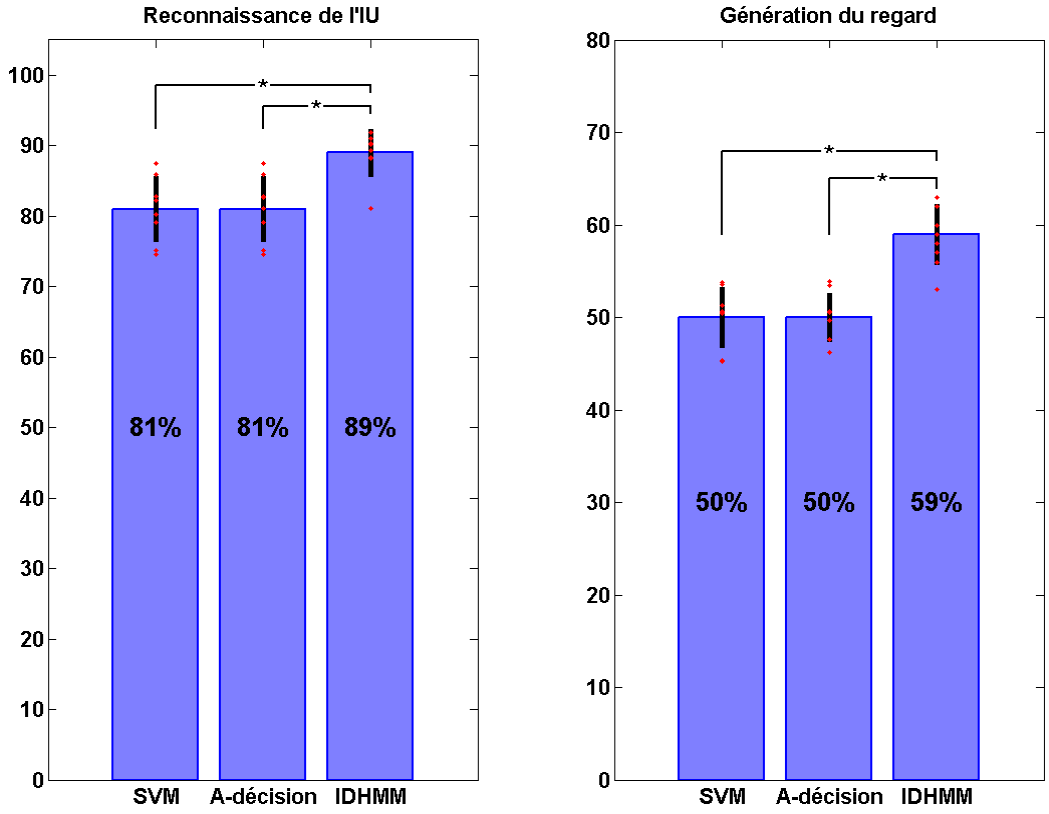


Figure 54 : Résultats des trois modèles : SVM, Arbres de décisions et IDHMM

4.6.4 Résultats du modèle IDHMM modifié

La comparaison objective entre l'initialisation aléatoire du modèle IDHMM et l'initialisation et le dimensionnement pilotés par les données a montré des résultats assez similaires en terme d'estimation des IU et de synthèse de regard, aucune différence significative n'est enregistrée. Cependant, nous remarquons des différences significatives dans les chemins d'alignement des états sensori-moteurs (SMS) estimés. Dans la Figure 55 (voir les zones sélectionnées), nous pouvons voir que, lorsque les probabilités de transition B_p sont initialisées aléatoirement (IDHMM de base), l'algorithme d'apprentissage de HTK donne à peu près un HMM de typologie gauche-droite. Toutefois, lorsque les probabilités sont initialisées avec une procédure de comptage pilotée par les données (IDHMM modifié), la

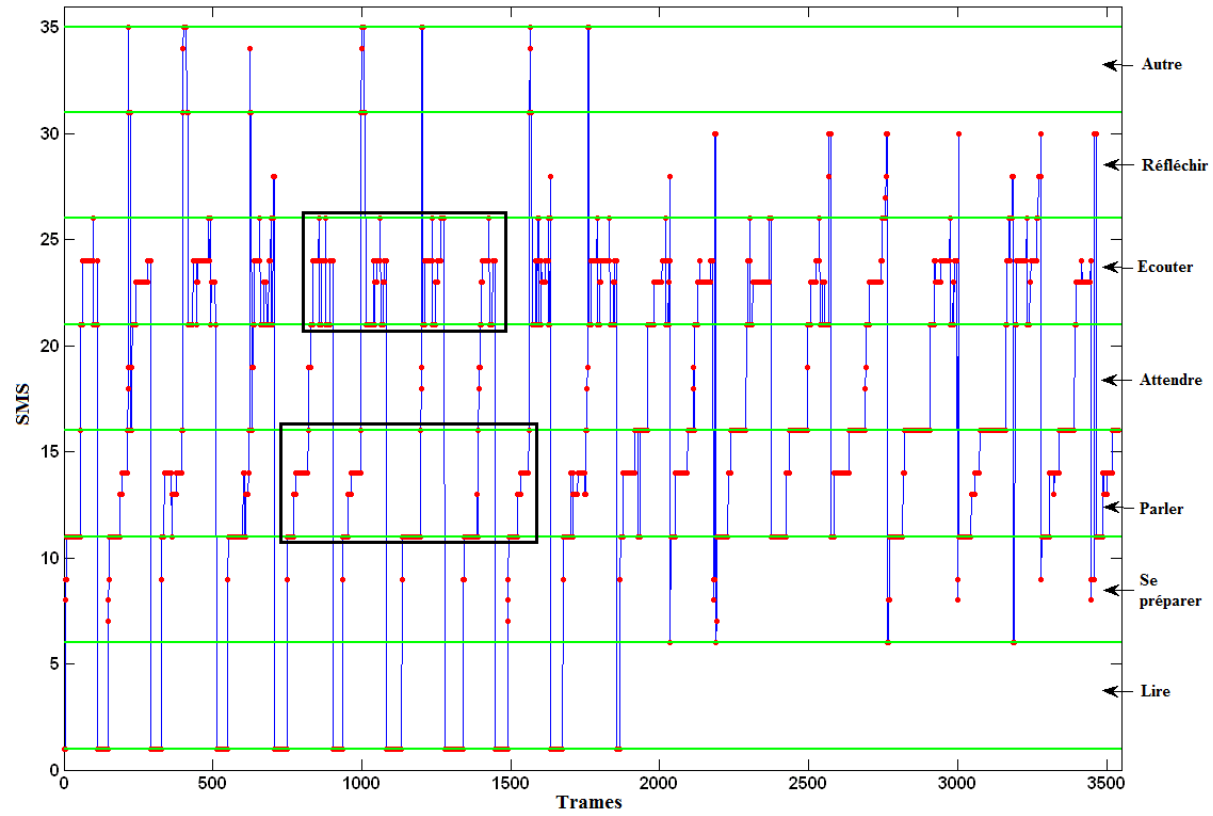
structure interne des IU devient entièrement connectée et présente des cycles entre les états sensori-moteurs. Nous appelons cette propriété intéressante le "bouclage sensori-moteur".

Afin de comparer les deux modèles en termes de "bouclage sensori-moteur", nous avons calculé le nombre de cycles détecté dans le IDHMM basique et celui détecté dans le IDHMM modifié. Nous avons comparé ensuite les résultats trouvés avec le nombre réel de cycles calculé à partir des données de la vérité terrain. Nous appelons le nombre de cycles par seconde entre SMS une "fréquence sensori-motrice". La comparaison des différentes fréquences est montrée dans la Figure 56. L'évaluation quantitative des fréquences sensori-motrices confirme bien notre évaluation qualitative du bouclage constaté dans la Figure 55. Nous voyons bien (Figure 56) que la fréquence du IDHMM modifié est plus proche à la vraie fréquence sensori-motrice que le modèle IDHMM de base. Ce résultat montre que dans l'évaluation des modèles du comportement humain, il est pertinent de ne pas se limiter aux performances pures de classement mais d'aller chercher d'autres critères qui reflètent bien les caractéristiques fines du comportement humain. En raison de cette propriété intéressante de bouclage, nous avons gardé la même structure du modèle IDHMM modifié pour le modèle IDHSMM.

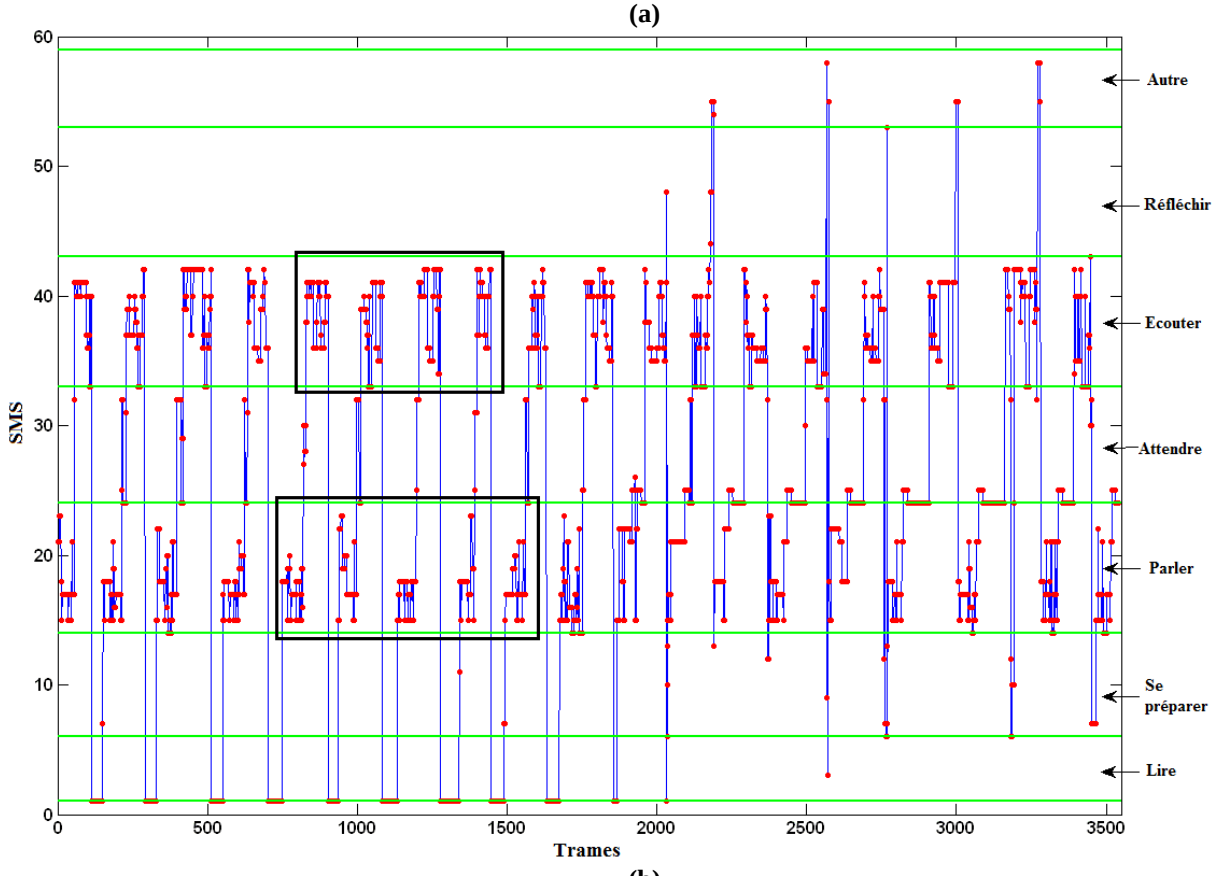
4.6.5 Résultats du modèle IDHSMM

Comme nous l'avons soutenu auparavant, les durées des états dans les HMM classiques sont modélisées par une distribution géométrique qui n'est pas adaptée à beaucoup de signaux physiques [146]. Nous étudions ici la capacité de notre IDHSMM à surmonter cette limitation majeure. Les résultats globaux de comparaison sont présentés dans la Figure 57. Comme nous pouvons le voir, les taux de reconnaissance sont presque les mêmes pour IDHSMM et IDHMM (90% vs. 89%). Toutefois, les taux de génération sont nettement meilleures pour le modèle semi-Markovien (63% vs. 59%). A noter que cette amélioration dans le taux de génération est également enregistrée avec un traitement non incrémental (64% vs. 59%).

L'amélioration significative dans le taux de synthèse du regard est expliquée par la capacité des chaînes semi-Markoviennes à bien modéliser et circonvenir les durées des fixations générées. Cette habilité à mieux gérer le temps est confirmée par la Figure 58 dans laquelle sont tracées les durées moyennes de chaque région d'intérêt (visage, œil droit, etc.) générée par IDHMM et IDHSMM. Nous remarquons que par rapport aux vraies durées calculées à partir des données réelles, les durées moyennes des fixations générées par IDHSMM sont meilleures que celles du IDHMM dans quatre régions d'intérêt sur cinq (visage, œil gauche, bouche et autre). Un autre résultat important consiste dans la capacité du IDHSMM à mieux capturer les cycles entre les SMS. Comme nous pouvons le voir dans la Figure 56, le score des chaînes semi-Markoviennes est le plus proche de la vérité terrain que les HMM classiques. En conclusion, l'IDHSMM génère des fixations de regard avec des durées plus précises et conduit à une fréquence sensori-motrice plus pertinente.



(a)



(b)

Figure 55 : Exemple de la trajectoire estimée des SMS par le (a) IDHMM de base (b) IDHMM modifié

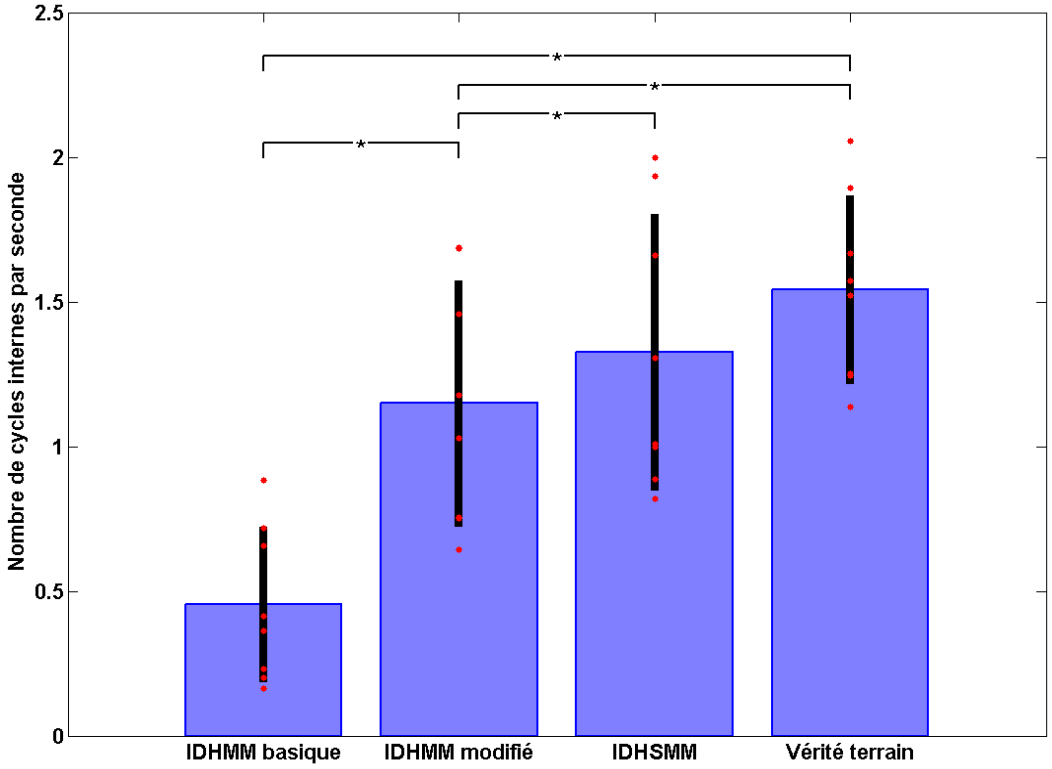


Figure 56: Les fréquences sensori-motrices de tous les modèles

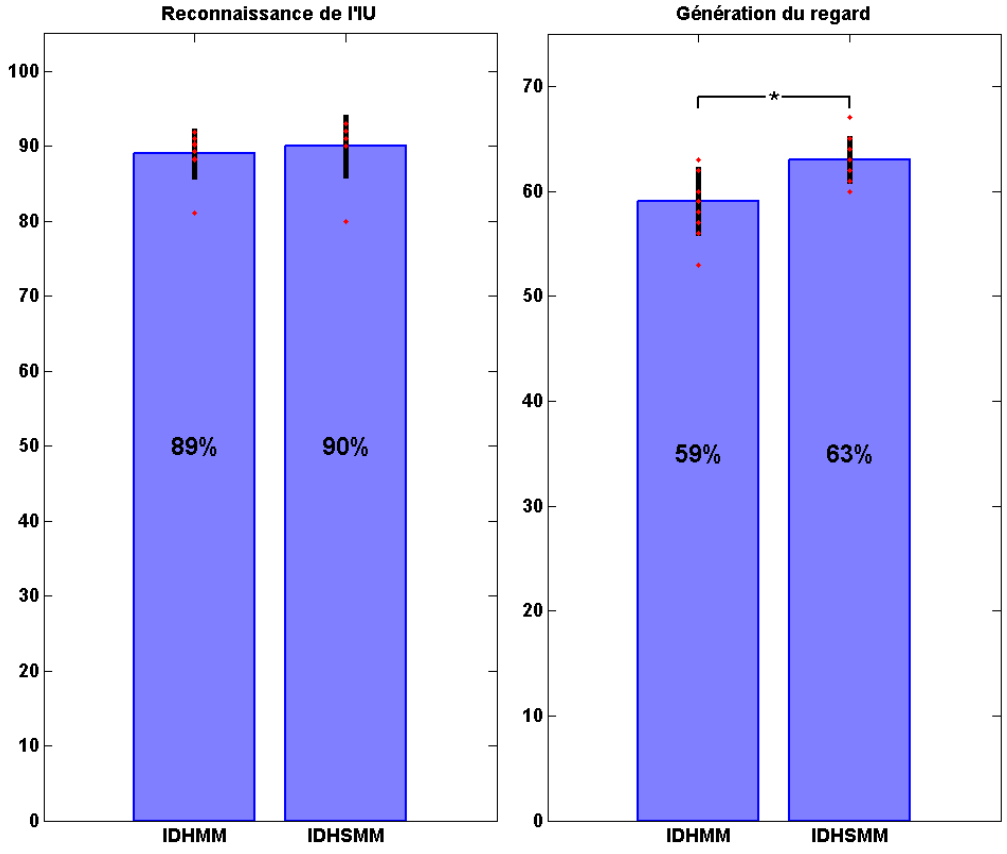


Figure 57 : Comparaison des résultats entre IDHMM et IDHSMM

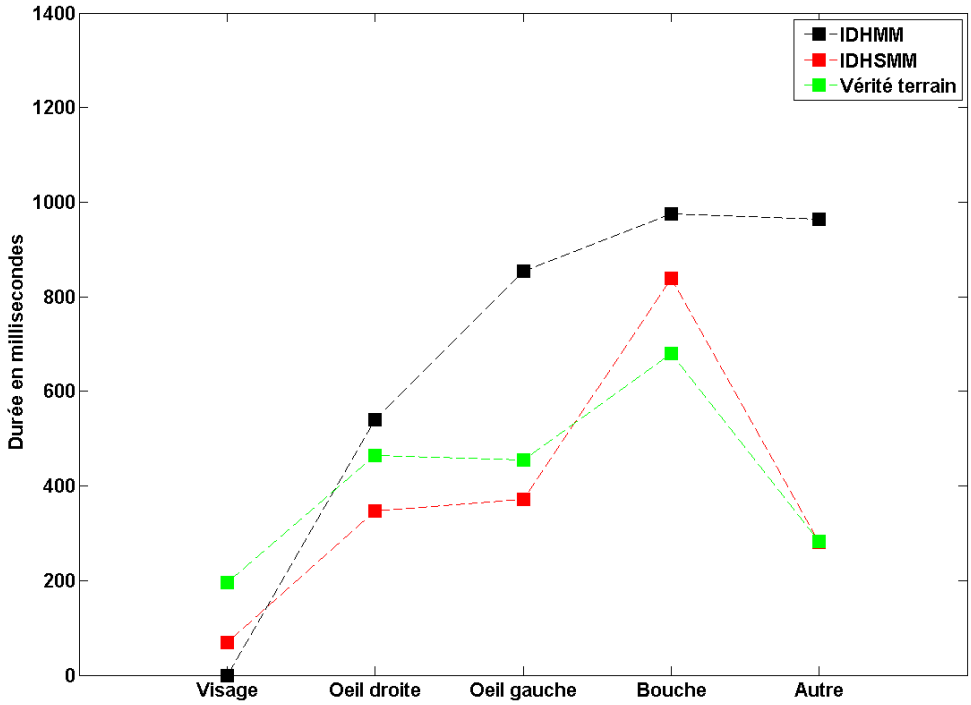


Figure 58 : Comparaison entre IDHMM et IDHSMM en terme de la durée moyenne de chaque région d'intérêt générée pour le regard. Le mouvement continu de la bouche généré par l'articulation verbale favorise la capture du regard en poursuite lente ("smooth pursuit") et donc des durées de fixation significativement plus longues sur cette région d'intérêt. Notez qu'aucune région d'intérêt correspondant au "Visage" n'a été générée par le IDHMM.

4.7 Conclusions

L'objectif de ce chapitre est de développer des modèles de comportement sensori-moteurs qui permettent d'assurer d'une manière incrémentale, une reconnaissance efficace et une génération pertinente des signaux sociaux. Les modèles proposés ont été appliqués sur notre première base de données (jeu de répétition de phrases). Le but était d'estimer l'IU dans laquelle était impliqué le sujet principal à partir du regard de l'interlocuteur et de l'activité vocale de chacun et ensuite de synthétiser son regard.

Le premier modèle proposé se base sur les chaînes de Markov cachées (HMM) à observations discrètes. Chaque HMM va modéliser une unité d'interaction. Vu que chaque interaction obéit à une syntaxe spécifique (succession d'unités interactionnelles), on a choisi de modéliser cette interaction par un seul HMM global qui concatène les HMM représentants les différentes IU, connectées par la syntaxe spécifique de la tâche. Quant au vecteur d'observations, il est conçu de manière à modéliser les boucles de perception-action. Il se décompose en deux parties, la première contient les observations de perception et la deuxième celles d'action. Les états cachés sont donc des états sensori-moteurs. Une fois l'apprentissage est fait, on extrait deux modèles: un modèle de reconnaissance qui ne prend en compte que les observations perceptuelles et un modèle de génération qui ne modélise que les observations d'actions. Le premier modèle va prédire l'IU la plus probable à partir des données perçus et le deuxième modèle va générer l'action qui correspond au mieux à cet IU estimée. Les deux

phases de décodage et de génération sont effectuées en incrémental grâce à une version modifiée de l'algorithme Short-Time Viterbi (BSTV). C'est pour cette raison que le modèle est appelé IDHMM (Incremental Discrete Hidden Markov Model). Les résultats ont montré que l'IDHMM dégage des performances aussi bonnes en mode incrémental qu'en mode différé classique.

Nous avons testé également les modèles personnels (ID) versus le modèle moyen (II) dans l'espoir de trouver un modèle personnel qui soit plus représentatif en termes de comportement que le modèle moyen. Bien que les résultats des modèles ID étaient tous moins bons, l'analyse MDS montre que ces modèles permettent de révéler les relations sociales déjà existantes entre les sujets. Ce résultats confirme que nos modèles de comportements sont capables de capturer des indices subtils (notamment de regard) qui reflètent et signalent des relations sociales entre les interlocuteurs.

Dans un deuxième temps, nous avons comparé les résultats obtenus par notre IDHMM avec les SVM et les arbres de décision. Pour chacun des modèles, un classifieur était destiné à estimer l'IU et l'autre à prédire le regard. Dans le Chapitre 3, nous avons quantifié l'apport d'une mémoire temporelle sur les performances des classifieurs. Les performances comparées des modèles montrent que l'IDHMM, grâce à ses propriétés de modélisation séquentielle, est un modèle robuste pour la reconnaissance des IU et la génération du regard, et que les classifieurs classiques comme les SVM – qui sont intrinsèquement incapables de modéliser les aspects séquentiels – peuvent donner des performances élevées si une certaine mémoire temporelle est incluse dans les observations d'entrée.

Dans un troisième temps, nous avons testé une nouvelle méthodologie pour initialiser le modèle IDHMM. Bien que les taux de reconnaissance et de génération ne soient pas significativement améliorés, nous avons pu constater une amélioration en termes de bouclage sensori-moteur. En effet, le nombre de cycles effectués par les états sensori-moteurs dans l'IDHMM modifié est plus proche de la vérité terrain que l'IDHMM de base.

Le dernier modèle testé est intitulé IDHSMM (Incremental Discrete Hidden Semi-Markov Model). Ayant la même structure que l'IDHMM modifié, le modèle semi-Markovien permet une meilleure modélisation des durées des états sensori-moteurs. En comparaison aux modèles précédents, ce modèle procure le meilleur taux de génération, le meilleur bouclage sensori-moteur et les meilleures durées pour les régions d'intérêts du regard généré. Ces résultats soulignent la pertinence de la modélisation de la durée dans les états sensori-moteurs. Dans le prochain chapitre, nous explorons des modèles graphiques plus expressifs et plus complexes que les HMM: les réseaux Bayésiens dynamiques (les DBN).

Chapitre 5 Modélisation par les DBN

5.1 Introduction

Dans ce chapitre, nous présentons des modèles de comportement multimodal basés sur le formalisme des réseaux Bayésiens dynamiques (DBN pour "Dynamic Bayesian Network"). Un réseau Bayésien dynamique est un modèle probabiliste graphique (PGM pour "Probabilistic Graphical Model") qui fournit des représentations compactes pour les relations d'indépendance conditionnelle qui existent entre des variables stochastiques [182]. Les DBN généralisent les réseaux Bayésiens statiques (BN pour "Bayesian Network") en incorporant des dépendances temporelles entre les variables aléatoires. Les DBN généralisent également les modèles de Markov cachés (HMM) et toutes leurs variantes, qui peuvent être vus tous comme des cas particuliers. En raison de leur représentation graphique intuitive, leur capacité à modéliser l'incertitude et les relations temporelles complexes entre les variables, les DBN ont été appliqués avec succès dans plusieurs domaines. Toutes ces caractéristiques font des DBN une approche particulièrement attractive et utile dans la modélisation de la dynamique des comportements multimodaux dans les interactions face-à-face.

Le modèle DBN présenté dans ce chapitre a été appris à partir des données multimodales de l'interaction face-à-face "jeu de cubes" dont l'objectif était d'étudier l'attention mutuelle et la deixis multimodale d'objets et de lieux dans une tâche de collaboration. Le défi de notre modèle de comportement est d'estimer les unités d'interaction et de générer les actions optimales d'un sujet, étant donné les actions perçues du partenaire ainsi que les objectifs joints des deux interlocuteurs. Dans notre travail, la structure du modèle DBN a été apprise à partir des données ce qui a révélé une structure de dépendance très intéressante décrivant précisément comment le comportement humain était coordonné pendant les interactions étudiées. L'utilisation de cette structure découverte a permis au modèle DBN d'avoir de meilleures performances que les modèles basés sur les HMM. Une meilleure coordination entre les modalités générées a été également démontrée. Nous commençons ce chapitre par introduire le formalisme des réseaux Bayésiens et nous enchaînons ensuite sur la conception d'un modèle sur notre jeu de données.

5.2 Les réseaux Bayésiens : description, apprentissage et inférence

5.2.1 Introduction

Les réseaux Bayésiens [183] appartiennent à la famille des modèles graphiques [182]. Ils marient au sein d'un même formalisme la théorie des probabilités et celle des graphes dans le but de fournir des outils intuitifs et efficaces pour représenter une distribution de probabilité jointe sur un ensemble de variables aléatoires. C'est un cadre très puissant qui permet une représentation compréhensible de la connaissance sur un domaine d'application donné et favorise le développement de modèles clairs et performants. Les modèles graphiques et les réseaux Bayésiens en particulier ont révolutionné le développement des systèmes intelligents

dans plusieurs domaines car ils apportent des solutions permettant de traiter deux grands problèmes bien connus en intelligence artificielle qui sont la complexité et l'incertitude. Ces dernières années, leur rôle est devenu ainsi de plus en plus important dans la conception et l'analyse d'algorithmes liés à l'inférence ou à l'apprentissage [184] [183] [185] [186].

5.2.2 Les réseaux Bayésiens

Dans la famille des modèles graphiques, deux types de graphes existent, les graphes orientés et les graphes non orientés. Les modèles à base de graphes non orientés sont souvent appelés champs de Markov [187]. Dans cette section, nous allons nous focaliser sur les réseaux Bayésiens, qui sont des graphes de dépendance orientés acycliques. Un Réseau Bayésien (BN) est un graphe acyclique orienté dont les nœuds sont des variables aléatoires de type continu ou discret et dont les arcs représentent les dépendances conditionnelles entre ces variables. L'existence d'un arc dirigé d'une variable A à une variable B, nous renseigne que A est une des "causes" possibles de B, ou encore A dispose d'une influence directe sur B. Bien sûr, la notion de causalité est un sujet controversé. Dire que la causalité est représentable mathématiquement ou affirmer qu'il est possible de la retrouver à partir des données n'est pas encore tranché [183], notamment lorsqu'il s'agit de phénomènes non physiques. La spécificité de nos signaux nous a cependant conduit à utiliser ce vocabulaire.

Formellement, un réseau Bayésien est défini par deux éléments: un graphe acyclique orienté G et une paramétrisation θ qui représente la probabilité jointe sur l'ensemble des variables. Considérons un ensemble de variables aléatoires $\mathrm{X} = (\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, \mathrm{X}_4, \mathrm{X}_5, \mathrm{X}_6, \mathrm{X}_7)$ et $\mathrm{P}(\mathrm{X})$ sa distribution de probabilité jointe:

$$\mathrm{p}(\mathrm{X}) = \mathrm{p}(\mathrm{X}_1)\mathrm{p}(\mathrm{X}_2)\mathrm{p}(\mathrm{X}_3)\mathrm{p}(\mathrm{X}_4|\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)\mathrm{p}(\mathrm{X}_5|\mathrm{X}_1, \mathrm{X}_3)\mathrm{p}(\mathrm{X}_6|\mathrm{X}_4, \mathrm{X}_5)\mathrm{p}(\mathrm{X}_7|\mathrm{X}_5) \qquad (30)$$

La Figure 59 montre un exemple d'un graphe orienté acyclique décrivant la factorisation de la distribution jointe $\mathrm{p}(\mathrm{X})$ des variables aléatoires $(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, \mathrm{X}_4, \mathrm{X}_5, \mathrm{X}_6, \mathrm{X}_7)$. La paramétrisation θ est exprimée en terme de probabilités conditionnelles des variables connaissant leurs parents. D'une manière plus générale, la probabilité jointe de K variables peut être écrite sous la forme :

$$p(\mathrm{X}) = \prod_{k=1}^{K} p(\mathrm{X}_k|pa_k) \qquad (31)$$

Où pa_k est l'ensemble des causes (parents) pour une variable X_k dans le graphe G (dans l'exemple montré K est égale à 7).

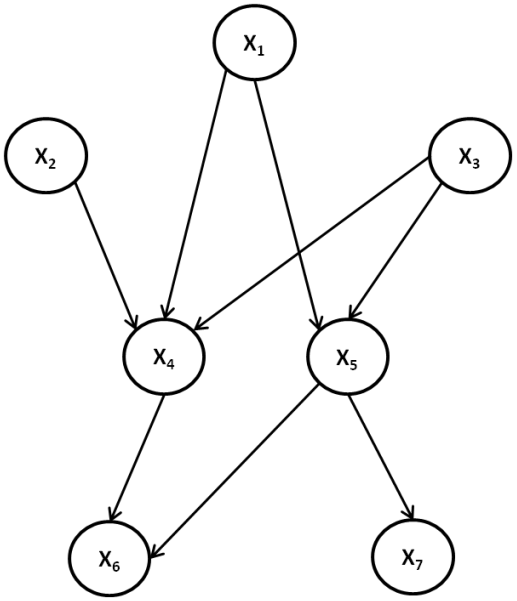


Figure 59 : graphe orienté acyclique décrivant la distribution jointe décrite par $p(\mathbf{X})$

Les Réseaux Bayésiens Dynamiques (DBN) [188] sont une extension des réseaux Bayésiens qui permet de représenter l'évolution temporelle des variables aléatoires. Soit un ensemble de variables aléatoires $\mathrm{X}_t = (\mathrm{X}_t^1, .., \mathrm{X}_t^K)$ évoluant dans l'intervalle $[0, T]$, la probabilité jointe peut être représentée par un réseau statique déroulé avec $(T + 1) * K$ variables avec la possibilité d'avoir des paramètres différents à chaque instant t [189]. Dans le cas où le processus est stationnaire, les probabilités de dépendances conditionnelles associées au réseau sont les mêmes pour tous les instants t. Dans ce cas, on peut représenter un DBN par un BN dont la structure est répétée à l'identique pour chaque instant t. On appelle tranche (i.e. "slice") la structure répétée du BN à un instant t.

5.2.3 Flux des dépendances dans les réseaux Bayésiens

5.2.3.1 Introduction au principe de d-séparation

Les indépendances conditionnelles représentent une des priorités clés d'une distribution jointe et un élément clé dans la compréhension du comportement du réseau. Ces indépendances conditionnelles peuvent être facilement déduites à partir du graph en utilisant un critère appelé la "d-séparation" [187]. Considérons trois variables X, Y et Z représentées par trois nœuds dans un graphe acyclique dirigé G . Afin de voir si X est indépendant de Y sachant Z, il faut tester si Z bloque tout chemin reliant X à Y. On dit que Z d-sépare X et Y si et seulement si Z bloque chaque chemin partant de X à Y. Un chemin est défini comme une séquence d'arcs non-dirigés consécutifs dans le graphe étudié. Un blocage peut être interprété comme un arrêt du flux d'informations entre les variables connectées. Le flux d'information est guidé par l'orientation des arcs et représente l'ordre de propagation des influences (ou des causalités) à travers le graphe. On peut interpréter cette propagation des influences comme un envoi d'information d'une variable donnée à ses variables descendantes [186]. Ainsi, un chemin entre X et Y via Z est dit actif si Z ne bloque pas le flux d'information prévenant de X. En résumé, trois cas se présentent (Figure 60):

- Chemin causal $X \rightarrow Z \rightarrow Y$: chemin actif si et seulement si Z n'est pas observée (Figure 60a).
- Cause commune $X \leftarrow Z \rightarrow Y$: chemin actif si et seulement si Z n'est pas observée (Figure 60b).
- Conséquence commune $X \rightarrow Z \leftarrow Y$: chemin actif si et seulement si Z est observée ou un descendant de Z est observé (Figure 60c). A savoir que cette structure ($X \rightarrow Z \leftarrow Y$) est connue sous le nom de "v-structure".

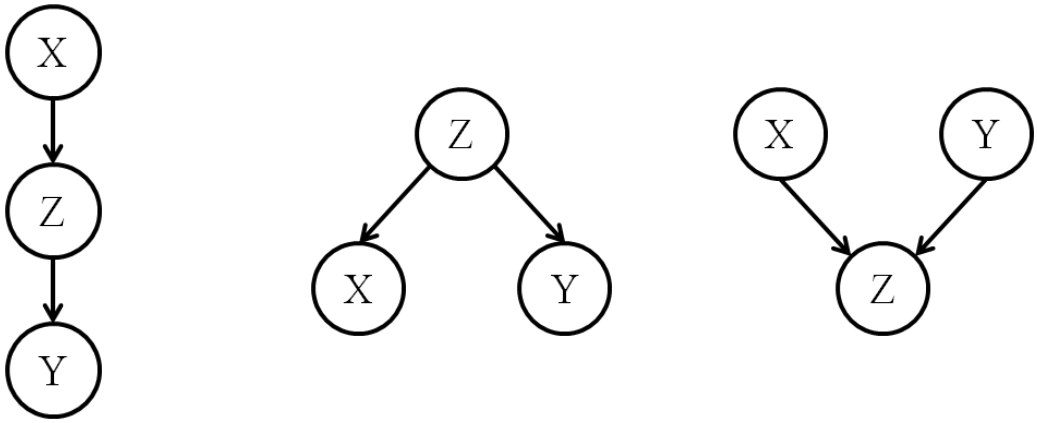


Figure 60: (a) chemin causal (b) cause commune (c) conséquence commune

Dans les deux cas du chemin causal et de la cause commune, observer Z bloque le chemin allant de X à Y. En effet, comme Z est observée, elle ne sera plus influencée par une nouvelle information sur X et par conséquent il n y aura plus d'impact sur Y. Dans le cas d'une "v-structure" ($X \rightarrow Z \leftarrow Y$) représentant deux causes ayant la même conséquence, le problème est inverse. Les deux causes $X \; et \; Y$ sont indépendantes jusqu'à la connaissance de leur conséquence commune Z. Elles deviennent ainsi dépendantes conditionnellement à Z.

Un exemple concret d'un réseau Bayésien est donnée dans la Figure 61. Il s'agit d'un réseau décrivant les relations entre les variables I (intelligence de l'étudiant), D (difficulté de la matière), T (test d'intelligence pour l'étudiant), N (note de l'étudiant dans la matière), et L (lettre de recommandation pour cet étudiant dans la matière passée). Avec cet exemple, les trois cas de figures (chemin causal, cause commune, conséquence commune) sont interprétés ci-dessous:

- Chemin causal: par exemple $D \rightarrow N \rightarrow L$, ce chemin causal est actif si N n'est pas observé, c.-à.-d, si la matière est facile, la note sera probablement bonne et l'étudiant aura une bonne lettre de recommandation. Maintenant si N est observée (mauvaise note), la lettre sera médiocre et toute information sur la difficulté de la matière n'apportera aucune information, le chemin est ainsi bloqué.

- Cause commune: par exemple $N \leftarrow I \rightarrow T$, le chemin entre T et N est actif c.-à.-d si le test d'intelligence a révélé un bon score, ceci montre que l'intelligence de l'étudiant est élevée et la note sera probablement bonne. Maintenant si I est observée (étudiant intelligent), une connaissance sur la note n'aura aucun impact sur le score d'intelligence et vice versa, le chemin devient alors bloqué.
- Conséquence commune: par exemple $D \rightarrow N \leftarrow I$, l'intelligence de l'étudiant et la difficulté de la matière sont deux variables indépendantes lorsque la note n'est pas observée. Par contre si on observe que la note est mauvaise (ou aussi une mauvaise lettre), une information sur la difficulté (par exemple matière facile) nous indique probablement que l'étudiant n'est pas intelligent. I et D deviennent alors dépendantes conditionnellement à N (ou à I).

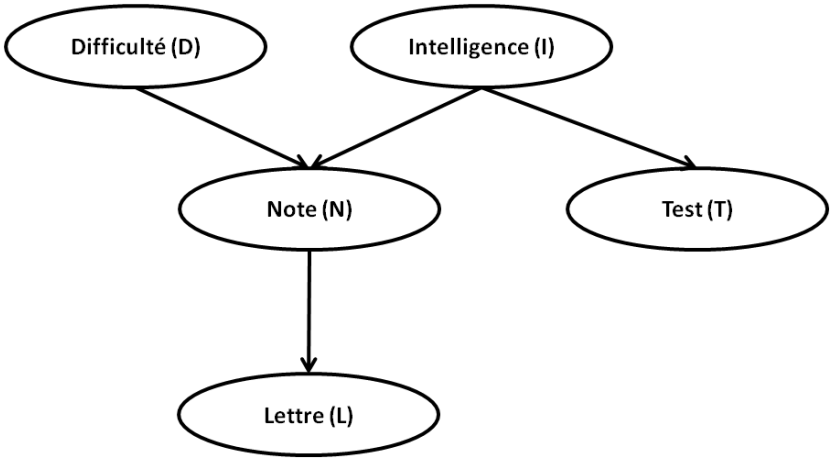


Figure 61: un exemple d'un réseau bayésien (reproduit de [182]). C'est réseau décrivant les relations entre I (intelligence de l'étudiant), D (difficulté de la matière), T (test d'intelligence pour l'étudiant), N (note de l'étudiant dans la matière), et L(lettre de recommandation pour cet étudiant dans la matière passée).

5.2.3.2 Cas général

Maintenant considérons le cas général, soit un BN de structure G et Z un ensemble de variables observées. Un chemin de G allant de X_1 à X_n est actif sachant Z si :

- A chaque fois qu'il y a une "v-structure" c.-à-d $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, alors X_i ou ses descendants appartiennent à Z.
- Il n'existe pas d'autres nœuds dans le chemin qui appartiennent à Z.

Le fait de comprendre les indépendances conditionnelles et les influences dans un BN sera assez bénéfique pour les problèmes d'inférence et d'apprentissage qui seront abordés dans les sections suivantes.

5.2.4 Inférence

5.2.4.1 Inférence exacte

Les premiers algorithmes traitant l'inférence dans les réseaux Bayésiens [190] [191] se basaient sur une procédure de passage de messages ("message passing") et ils étaient surtout

utilisés pour les arbres (graphe non orienté, acyclique et connexe). Dans cette technique, on associe à chaque nœud un processeur qui peut envoyer des messages à ses voisins jusqu'à ce qu'un certain équilibre soit atteint, après un nombre fini d'étapes. Cette méthode est connue sous le nom de "Belief Propagation" ou aussi "Sum-Product". Cet algorithme a été depuis généralisé aux réseaux quelconques pour donner l'algorithme appelé *arbre de jonction* (i.e; "junction tree") [192] [193]. Les algorithmes d'inférence exacte dans les modèles graphiques sont tous pratiquement des variantes de l'algorithme *arbre de jonction*. Pour effectuer l'inférence, l'algorithme convertit le réseau Bayésien dans une structure secondaire acyclique (appelé donc arbre de jonction) qui encapsule les relations statistiques du réseau d'origine dans un graphe acyclique, non orienté. C'est de loin l'algorithme le plus utilisé pour faire l'inférence exacte dans les BN. Pour plus de détails sur cet algorithme, veuillez consulter l'annexe B.

5.2.4.2 Inférence approximative

Plusieurs problèmes dans la pratique (par ex. graphe complètement connecté) rendent l'inférence exacte difficile voire impossible à appliquer, d'où le besoin d'exploiter des méthodes d'approximation efficaces. Une classe importante de ces approximations, est appelé les méthodes variationnelles [194] qui sont en fait une adaptation de l'algorithme EM (Expectation-Maximization). En complément de ces approches déterministes, il existe une large gamme de méthodes d'échantillonnage appelées méthodes de Monte Carlo [194]. Ces méthodes sont basées sur l'échantillonnage numérique stochastique à partir de distributions de probabilité. En particulier, elles se basent sur le parcours de chaînes de Markov qui ont pour lois stationnaires les distributions à échantillonner.

Une approche plus simple pour l'inférence approximative dans les graphes avec des boucles, consiste à appliquer simplement l'algorithme "Belief Propagation", même si il n'y a aucune garantie que cela va donner de bons résultats. Cette approche est connue sous le nom de "Loopy Belief Propagation" [195]. Ceci est possible parce que les règles de passage des messages sont purement locales. Cependant, parce que le graphe a des cycles, l'information peut circuler plusieurs fois autour du graphe. Pour certains modèles, l'algorithme converge, alors que pour d'autres non. Par conséquent, selon l'application, cet algorithme de propagation de croyances peut donner des résultats médiocres comme il peut s'avérer très efficace.

5.2.4.3 L'inférence dans les réseaux Bayésiens dynamiques avec les arbres de jonction

Puisque les DBN sont des modèles dynamiques et temporels, l'inférence peut prendre plusieurs formes, comme le montre la Figure 62 issue de [188]. On dispose essentiellement de 4 versions d'inférence, une hors ligne et trois en ligne:

- "fixed-interval-smoothing": C'est la version hors ligne de l'inférence, l'estimation n'est faite qu'après avoir la totalité des observations. Dans la suite, cette version est appelée simplement "smoothing".
- "filtering": c'est la première version en ligne de l'inférence, elle est utile pour les systèmes temps réel.

- "prediction": c'est la deuxième version en ligne, elle est utile pour les études et les analyses prévisionnelles.
- "fixed-point smoothing": c'est la troisième version en ligne. Cette version est utile pour les systèmes temps réel qui tolèrent un temps de latence court avant de faire l'inférence.

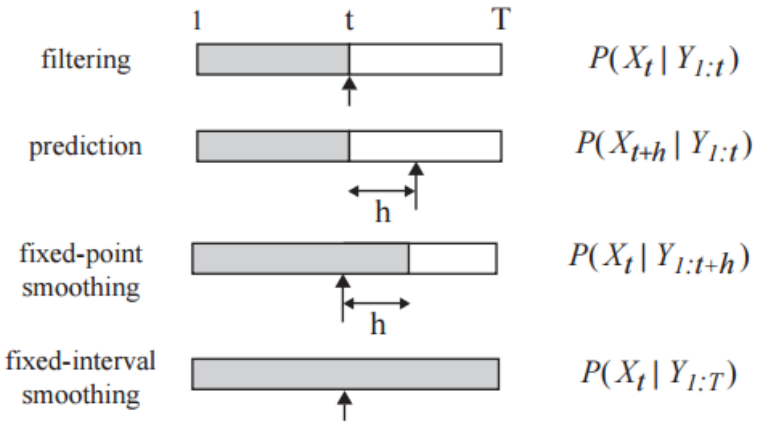


Figure 62 : Les principaux types d'inférence pour les DBN. Les parties colorées en gris montrent la quantité de temps pour laquelle on dispose de données observées (*Y* représente ces observations). Tous les types d'inférence sauf "fixed-interval-smoothing" sont en ligne (incrémental). Cette dernière est la version hors ligne de l'inférence, dans la suite cette version est appelée simplement "smoothing". Les flèches indiquent les moments t où on veut faire l'inférence. T est le temps de fin de la séquence. Pour la version "prediction" h représente l'horizon de prévision alors que pour la version "fixed-point smoothing" h représente le seuil toléré d'observations postérieures à l'instant de l'inférence (figure adaptée de [188]).

Il y a plusieurs façons d'utiliser les arbres de jonction pour faire l'inférence dans les DBN. L'approche naïve consiste à dérouler le DBN en un nombre désiré de tranches. Il suffit alors d'effectuer l'inférence sur le modèle qui en résulte comme s'il était un réseau Bayésien statique. Cependant, ceci nécessitera évidemment trop de temps et de mémoire, en particulier dans une application telle que le filtrage en ligne ("filtering"). Intuitivement, nous devrions être en mesure de faire mieux en faisant usage de l'hypothèse que tous les processus modélisés sont stationnaires (les fonctions de transition entre les slices ne dépendent pas du temps) et Markoviens (chaque fonction de transition ne dépend que de la tranche précédente). Murphy [188] a démontré que grâce à ces propriétés, l'inférence est possible en utilisant seulement une structure de deux tranches du DBN (i.e. "2-Timeslices Bayesian Network" ou "2-TBN"). L'algorithme proposé par Murphy est connu sous le nom de "interface algorithm". Un exemple [196] de 2-TBN est montré dans la Figure 63.

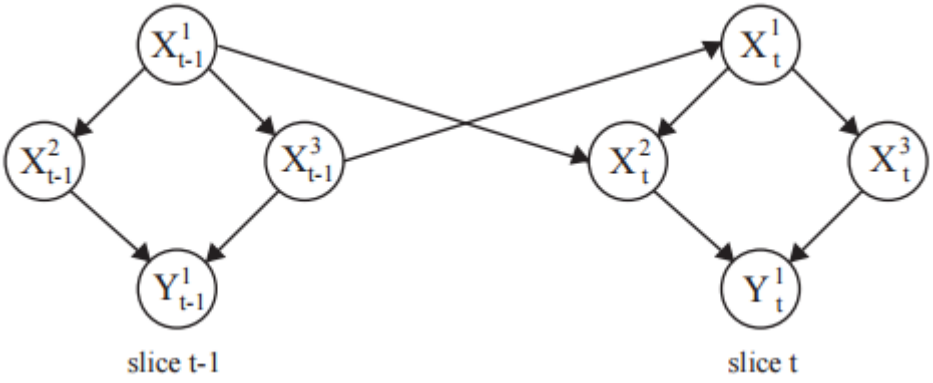


Figure 63: 2-TBN d'un réseau bayésien dynamique (figure reproduite de [196])

5.2.5 Apprentissage

Un réseau Bayésien possède deux niveaux de paramètres [197]: des paramètres quantitatifs qui sont les probabilités conditionnelles associées à chaque nœud, et des paramètres qualitatifs (structurels) qui sont les arcs entre les différents nœuds. L'ensemble de ces arcs forme la structure du réseau. Par conséquent, deux types d'apprentissage sont disponibles pour les réseaux Bayésiens : l'apprentissage des paramètres et l'apprentissage de la structure.

5.2.5.1 Apprentissage des paramètres

Le premier type, largement utilisé dans la littérature, est l'apprentissage de paramètres [183]. Si les variables sont observées, l'approche la plus simple et la mieux adaptée est l'estimation statistique qui consiste à déterminer la probabilité d'un événement en calculant sa fréquence d'apparition dans la base d'apprentissage. Cette approche est appelée la méthode de *maximum de vraisemblance* (MV). On peut procéder aussi à une estimation Bayésienne qui suit un concept un peu différent. Elle consiste à trouver les paramètres les plus probables en observant des données et en posant des a priori sur les paramètres. Dans ce cas de figure, l'approche est appelé l'approche de *maximum a posteriori* (MAP).

En pratique, les bases de données peuvent être incomplètes. Certaines variables peuvent être par définition latentes et donc non observables. D'autres, peuvent être partiellement ou totalement manquantes à cause de multiples facteurs, par exemple, une défaillance matérielle, des questions non répondues dans un sondage, etc. Dans ce cas, l'algorithme le plus utilisé est l'algorithme EM. En effet, lorsque les données sont incomplètes le calcul direct de la vraisemblance devient impossible. La méthode EM permet d'éliminer cet obstacle et fournit une solution à ce problème. L'algorithme se déroule d'une manière itérative en deux étapes: L'étape E ("Expectation") où on calcule l'espérance de la vraisemblance. Ce calcul est désormais possible vu que l'algorithme procède à l'estimation des données manquantes/latentes sachant les données observées. L'étape M ("Maximization") où on calcule le maximum du vraisemblance des paramètres en se basant la vraisemblance calculée dans l'étape E, et ensuite on met à jour les paramètres pour la prochaine itération.

5.2.5.2 Apprentissage de la structure

La structure d'un réseau Bayésien peut être fournie par un expert humain. Si ce n'est pas le cas, il est possible de l'apprendre automatiquement à partir d'un ensemble de données. Cependant, la recherche de la structure optimale n'est pas une tâche simple vu la taille exponentielle (en fonction du nombre de variables) de l'espace de recherche. Lorsqu'on procède à l'apprentissage automatique de la structure, il est recommandé de valider le réseau donné par la machine par un expert du domaine. La structure peut être également rectifiée par l'expert afin que les relations de dépendances détectées demeurent sémantiquement valides ou aussi afin de rajouter des relations jugées indispensables par l'expertise humaine.

Une première famille d'approches pour l'apprentissage automatique, consiste à rechercher les différentes indépendances conditionnelles qui existent entre les variables. L'algorithme PC [198] est l'un des premiers algorithmes dans cette direction [199]. Il utilise un test statistique pour vérifier s'il existe une indépendance conditionnelle entre deux variables. Il est alors possible de reconstruire la structure du réseau Bayésien à partir de l'ensemble des relations d'indépendances conditionnelles découvertes. En pratique, on commence par un graphe complètement connecté, et lorsqu'on identifie une indépendance conditionnelle, on retire l'arc correspondant (les "v-structure" par ex. sont exploitées dans ce type d'approches). Un algorithme avec un concept similaire (IC) a été introduit dans [200].

A l'encontre de la première famille d'approches qui visaient à retrouver des indépendances conditionnelles entre les variables, d'autres approches tentent de quantifier l'adéquation d'un réseau Bayésien au problème à résoudre, c'est-à- dire d'attribuer un score (par exemple le BIC [201]) à chaque réseau Bayésien [202]. Puis elles recherchent la structure qui donnera le score optimal dans l'espace des graphes acycliques dirigés. Un parcours exhaustif est impossible en pratique en raison de la taille de l'espace de recherche. Plusieurs algorithmes de recherche dans l'espace possible des graphes ont été proposés pour résoudre cette problématique. On cite par exemple l'algorithme qui réduit cet espace à l'espace des arbres (MWST [203]), l'algorithme qui donne un ordre aux nœuds pour limiter la recherche des parents éventuels pour chaque variable (K2 [204]), ou celui qui réalise une recherche gloutonne [205].

5.2.6 Conclusion

Dans la littérature, le formalisme des réseaux Bayésiens est bien étudié et suscite de plus en plus d'intérêt afin d'améliorer les approches d'apprentissage et d'inférence. Le succès des BN est dû principalement à leur efficacité dans l'acquisition, la représentation et l'utilisation des connaissance [183]. Leurs domaines d'application sont larges et multiples, on les trouve par exemple dans la santé, l'industrie, la défense, la finance, le marketing et notamment l'informatique. Dans la section suivante nous allons utiliser les DBN dans le cadre de notre objectif de modéliser et générer les signaux sociaux dans les interactions humaines face-à-face.

5.3 Application

Dans cette section, nous présentons des modèles de comportement multimodal basés sur le formalisme des réseaux Bayésiens dynamiques (les DBN). Pour ce but, nous utiliserons les données de notre deuxième expérimentation ("jeu de cubes"). L'objectif de cette expérimentation est d'inférer un modèle de comportement pour l'instructeur capable à partir des données perceptuelles égocentriques d'estimer l'unité interactionnelle et de générer un flux d'action pertinent. Pour les DBN, les deux phases d'apprentissage et d'inférence ont été effectuées en utilisant le Bayes Net Toolbox [206].

5.3.1 Données

Les données utilisées dans cette application sont celles de la deuxième interaction intitulée "Jeu de cubes" présentée dans la section 2.3. Nous rappelons qu'il s'agit d'une interaction entre un instructeur permanant et un manipulateur (avec 3 dyades et 30 jeux en total). Nous rappelons aussi que IU est l'unité d'interaction, MP le geste de manipulation, SP les segments de parole de l'instructeur, GT le geste déictique de l'instructeur et FX les fixations du regard de l'instructeur. Les données sont organisées en trois flux:

- Le flux de l'unité d'interaction composé de "IU".
- Le flux de perception composé de "MP" et "SP".
- Le flux d'action composé de "GT" et "FX".

Nous nous intéressons à la génération de la coverbalité ("GT" et "FX"), c'est pour cette raison que la variable SP est considérée comme une observation perceptuelle. Les sept unités interactionnelles ("s'informer", "chercher", "pointer", "indiquer", "vérifier", "valider" vs. "autre") ont été annotées semi-automatiquement . Trois modèles ont été évalués et comparés: le modèle HMM et le modèle HSMM appliqués sur ce jeu de données ainsi que le modèle DBN.

5.3.2 Le modèle DBN appris

Les réseaux Bayésiens dynamiques sont des modèles graphiques intéressants car ils généralisent de nombreuses autres approches (filtres de Kalman, HMM, etc.). Une différence majeure entre les HMM et les DBN est que, dans un DBN, les états cachés sont représentés par un ensemble de variables aléatoires. Alors que, dans un HMM, l'espace d'états consiste en une seule variable aléatoire. Une autre différence importante est que, dans les HMM, on suppose l'indépendance entre les variables observées sachant l'état alors que dans la pratique cette condition n'est pas généralement respectée. Dans les DBN, nous avons la possibilité de mettre autant d'arcs (de dépendances) entre les variables observées (et mêmes cachées) tant que c'est pertinent dans le cadre de l'application en vigueur. Donc les DBN permettent, comparé aux HMM, d'avoir un réseau avec une structure d'indépendance conditionnelle beaucoup plus puissante et beaucoup plus riche. Dans certaines situations et en fonction de l'application, cette structure de dépendance peut être donnée par un expert. Sinon, comme déjà

mentionné dans la section précédente, plusieurs méthodes ont été introduites pour apprendre automatiquement la structure du réseau.

Dans notre application, nous avons procédé à l'apprentissage de la structure la mieux adaptée à nos données sensori-motrices. La structure apprise résultante comme nous allons présenter dans la suite s'est avérée extrêmement riche et intéressante. Nous rappelons qu'un DBN n'est d'autre qu'un réseau Bayésien (BN) dont la structure est répétée à chaque instant t et qu'on peut représenter un DBN par une structure 2-TBN. Par conséquent, pour avoir une structure complète de notre modèle composé de 5 variables discrètes (IU, MP, SP, GT, FX), il va falloir trouver les relations causales intra-slice et inter-slices. La structure intra-slice est la structure de dépendance qui lie les variables d'un instant t. La structure inter-slices est la structure de dépendance qui lie les variables d'un instant t aux mêmes variables mais de l'instant $t + 1$. A cette fin, nous avons utilisé deux algorithmes largement utilisés dans la littérature, d'une part l'algorithme K2 [204] pour détecter la meilleure structure intra-slice et d'autre part, l'algorithme REVEAL [207] pour détecter la meilleure structure inter-slices conformes aux données. L'algorithme K2 nécessite un ordre prédéfini des variables afin de limiter l'espace de recherche (la recherche dans l'espace complet étant quasi-impossible). Initialement, chaque nœud n'a pas de parents et l'algorithme ajoute de manière incrémentale un parent si ce dernier améliore significativement le score de la structure résultante. L'ordre fourni à l'algorithme était IU,MP,SP,GT,FX c.-à.-d. l'unité d'interaction au premier niveau, suivi des données sensorielles puis motrices. L'IU est considérée comme le niveau le plus élevé car elle reflète les opérations cognitives qui guident les comportements sensori-moteurs [110].

La structure du réseau 2-TBN apprise est montrée dans la Figure 64. Afin d'avoir toute la boucle sensori-motrice "MP→ FX → GT→ SP → MP", une flèche allant de FX à GT a été ajoutée volontairement alors qu'elle n'avait pas été inférée par REVEAL. Les propriétés de cette boucle sont discutées dans la section 5.3.2.2. La structure résultante (apprise et rectifiée par l'expertise humaine) s'avère assez riche et révèle des propriétés logiques et très précises en ce qui concerne les relations entre les variables et le flux de l'information au sein du graphe. Nous recensons principalement 6 priorités majeures: 3 propriétés intra-slices et 3 propriétés inter-slices.

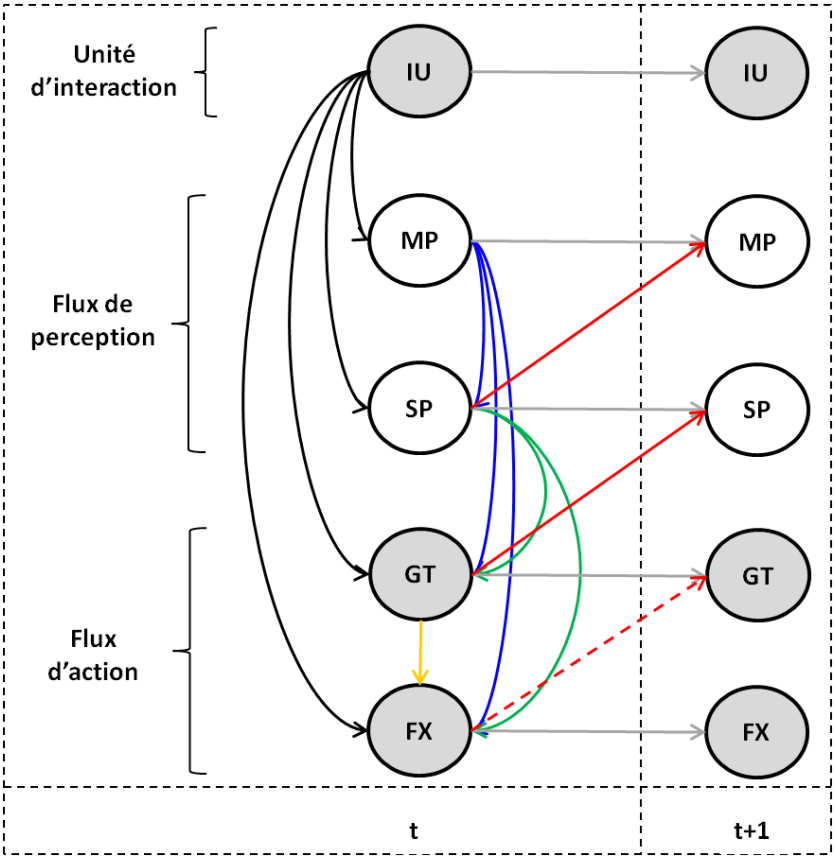


Figure 64 : La structure apprise de notre modèle DBN. La flèche pointillée en rouge a été rajoutée pour fermer la boucle sensori-motrice "MP→ FX → GT→ SP → MP". Les variables dans les cercles gris sont les variables à prédire en inférence. Pour des raisons de clarté, nous avons omis de dessiner les dépendances intra-slice t+1 (elles sont les mêmes que la slice t).

5.3.2.1 Les propriétés intra-slices

Propriété 1 : Les unités d'interactions (IU) sont des états sensori-moteurs car ils influencent à la fois les flux de perception (MP et SP) et ceux d'action (GT et FX) (Figure 65).

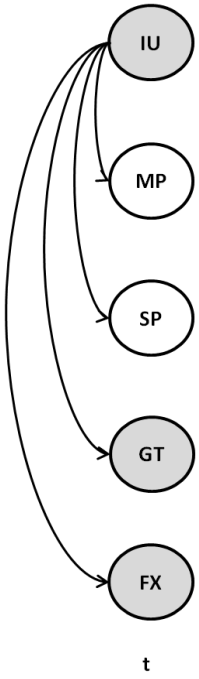


Figure 65: Propriété 1 (intra-slice)

Propriété 2 : L'instructeur réagit bien aux actions du manipulateur car on voit que MP impacte SP, GT et FX (Figure 66).

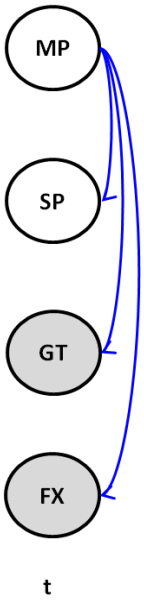


Figure 66: Propriété 2 (intra-slice)

Propriété 3 : L'activité vocale de l'instructeur (SP) influence son comportement co-verbal (GT et FX) (Figure 67). Ceci est compatible avec plusieurs travaux de la littérature qui suggèrent que le comportement co-verbal est un comportement subordonné aux énoncés verbaux [208].

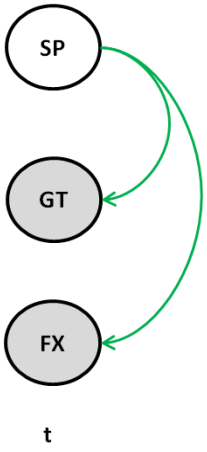


Figure 67: Propriété 3 (intra-slice)

5.3.2.2 Les propriétés inter-slices

Propriété 1 : Chaque variable aléatoire est influencée par son histoire (Figure 68).

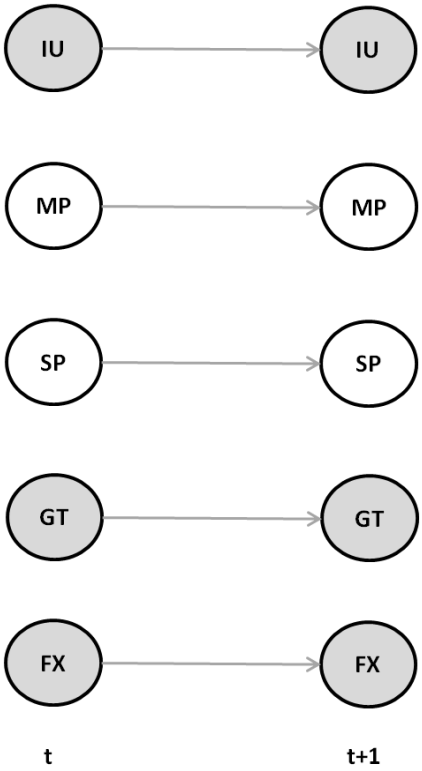


Figure 68 : Propriété 1 (inter-slices)

Propriété 2 : Les boucles perception-action peuvent être facilement identifiées comme le montre la Figure 69. En particulier, la sous chaîne "FX → GT→ SP → MP" est extrêmement intéressante car elle décrit relativement le déroulement de la tâche "mets-ça là". C'est comme-ci l'instructeur commence par fixer le cube, il le pointe, ensuite il le désigne vocalement et le manipulateur finit par appliquer l'instruction.

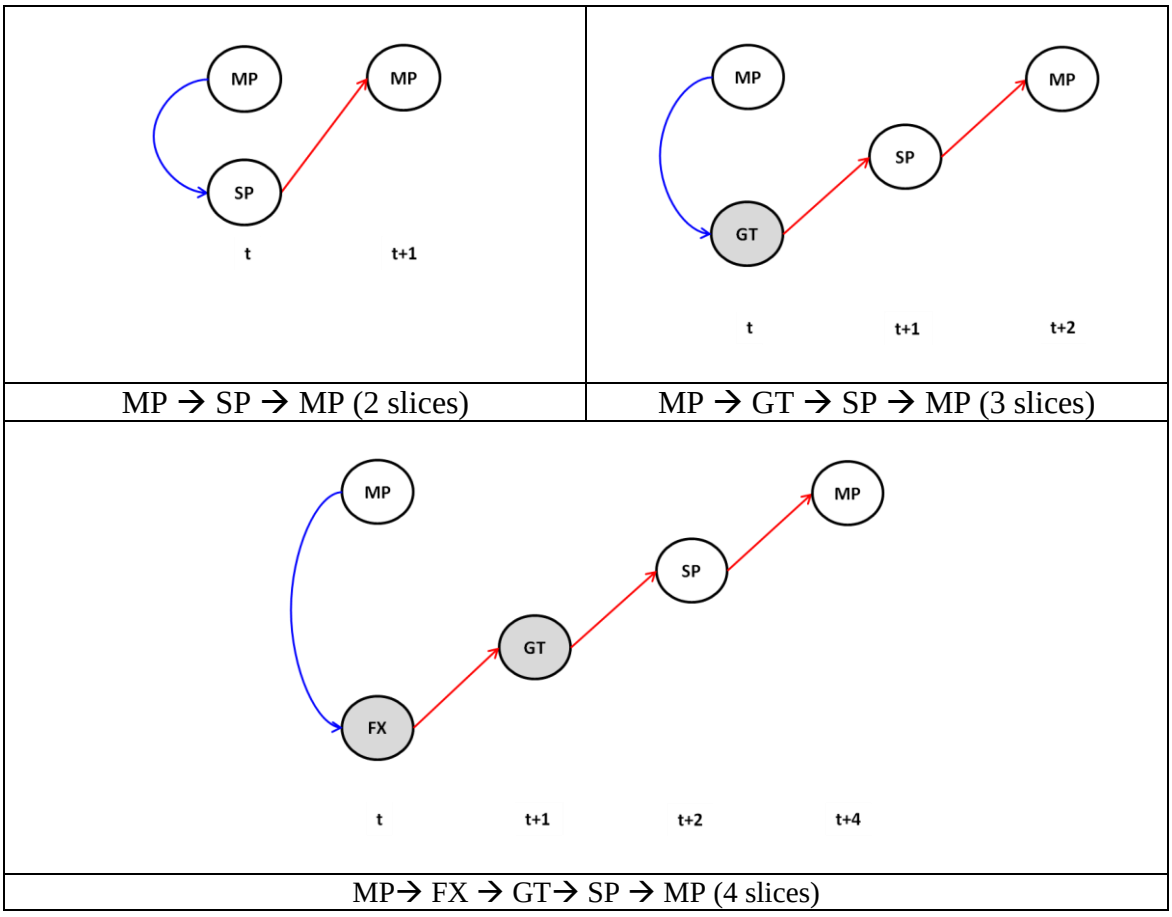


Figure 69: Propriété 2 (inter-slices)

Propriété 3 : Il existe une influence mutuelle entre les modalités du comportement de l'instructeur (Figure 70).

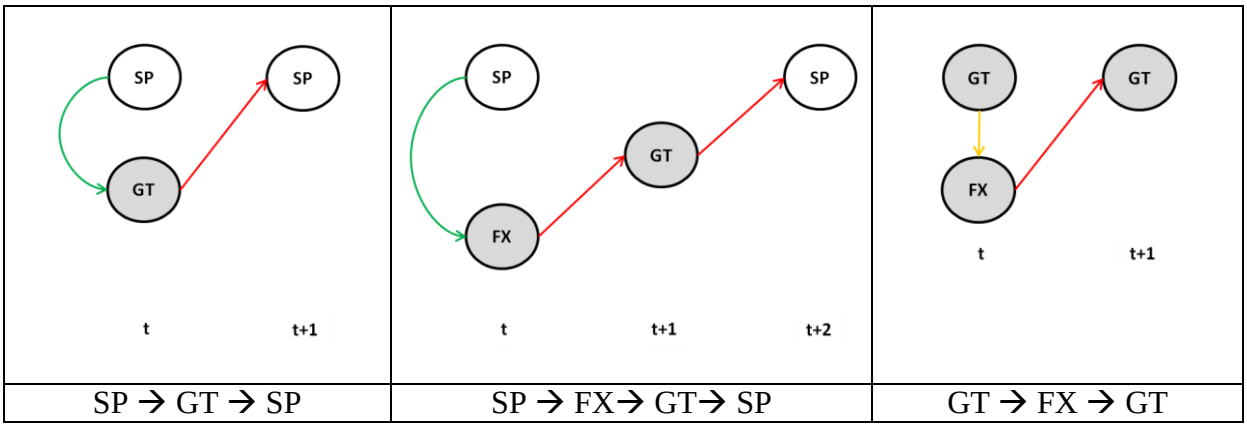


Figure 70 : Propriété 3 (inter-slices)

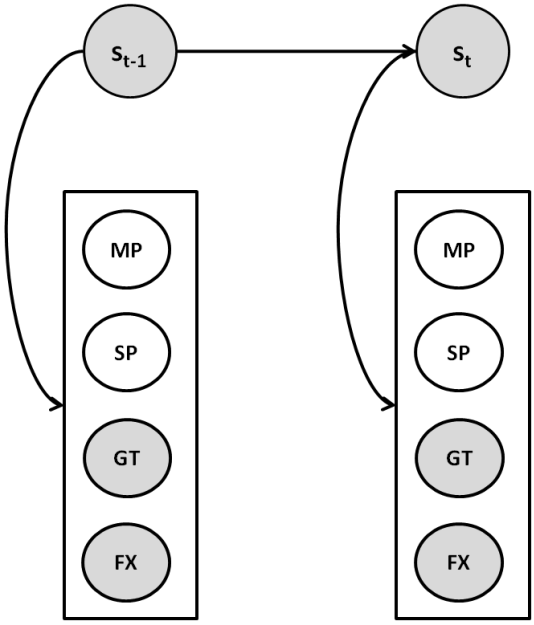


Figure 71: Le graphe du dépendance de notre modèle HMM, s_t représente l'état sensori-moteur à un instant t

Comparé au graphe de dépendance du modèle HMM (Figure 71), la structure du modèle DBN est clairement plus riche et plus remarquable. La structure du DBN réussit à capturer des relations précises et pertinentes entre les différentes variables, à l'encontre du modèle HMM, qui ne permet pas une modélisation avec une telle précision. Dans notre application, nous avons utilisé le modèle DBN appris pour estimer l'unité d'interaction et le meilleur flux d'action correspondant (gestes déictiques et fixations de regard). Les résultats quantitatifs que nous allons montrer ont confirmé la pertinence de cette structure apprise.

5.3.3 Les modèles HMM et HSMM (rappel)

Afin d'évaluer les performances du modèle DBN, nous avons appliqué sur les mêmes données les deux modèles basés sur les HMM et les HSMM qui ont été présenté dans le Chapitre 4. Pour rappel, dans les chapitres précédents, ces deux modèles ont été appliqués seulement sur la première base de données ("jeu de phrases"). Dans ce chapitre, nous les appliquons également sur notre deuxième scénario et nous comparons leurs performances avec celles du modèle DBN appris. Ceci confirme que ces deux modèles (que nous l'avons appelé IDHMM et IDHSMM dans le Chapitre 4) sont bien des modèles générique applicables et adaptables à des nombreuses applications. Pour la deuxième base de données, les mêmes approches ont été suivies dans la construction du modèle HMM et HSMM. Nous modélisons chaque unité d'interaction par un HMM et l'interaction complète par un HMM global qui concatène les modèles spécifiques aux IU. Les observations des HMM contiennent les flux de perception (MP et SP) et d'action (GT et FX). Les états cachés sont donc bien des états sensori-moteurs (SMS). Après l'apprentissage, nous extrayons deux sous-modèles: un sensori-HMM pour la reconnaissance de l'IU et un motor-HMM pour la génération du geste GT et du regard FX. Le nombre d'états cachés par IU a été fixé à 5 résultant dans 35 états cachés (SMS) pour l'ensemble du modèle HMM. Pour le modèle HSMM, qui a l'avantage de

modéliser les durées des états cachés, le dimensionnement (nombre de SMS par IU) ainsi que l'initialisation des probabilités d'émission ont été réalisés à partir des données. Le modèle résultant était ainsi composé de 48 états cachés (SMS).

5.3.4 Evaluation

Pour tous les modèles, nous appliquons le principe de la validation croisée (30-fold cross validation) sur nos données: 29 jeux ont été utilisés pour l'apprentissage tandis que le 30ème jeu a été utilisé pour le test. La F-mesure issue de la distance de Levenshtein [142] est utilisée pour évaluer la reconnaissance de l'IU ainsi que la génération du geste déictique et du regard. En outre, nous avons réaffirmé notre approche de ne pas se contenter des taux de classification et de génération mais d'étudier en plus d'autres priorités intrinsèques aux modèles. Puisque, dans ce jeu de données, nous disposons de plusieurs modalités côté instructeur (parole, geste et regard), nous avons proposé une méthodologie pour étudier la coordination entre ces différentes modalités et la capacité des modèles proposés à produire une coordination similaire. Dans la section suivante, les modèles présentés ci-dessus sont présentés et comparés entre eux en terme de performances et de propriétés de coordination.

5.4 Résultats

Les modèles proposés doivent être en mesure (1) d'estimer les unités d'interaction (IU) à partir des observations de perception (SP : activité de parole / MP : manipulation du partenaire). Lorsque les deux partenaires coopèrent, l'organisation séquentielle des unités d'interaction devrait idéalement refléter les états mentaux partagés des partenaires de conversation à un moment particulier; (2) de générer les actions appropriées (GT : gestes de la main de l'instructeur et FX : ses fixations du regard) qui reflètent sa conscience actuelle de l'évolution de la tâche partagée. La prédiction de IU, GT et FX a été faite de deux manières : une estimation hors ligne/non-incrémentale et une estimation en ligne/incrémentale. Dans les deux cas, l'inférence est assurée par l'algorithme arbre de jonction [209] et plus précisément l'algorithme interface [188] qui est une implémentation de l'arbre de jonction pour les réseaux Bayésiens dynamiques. Pour rappel, l'algorithme d'arbre de jonction donne une solution exacte, comme l'algorithme de Viterbi dans le contexte de HMM, mais il est utilisé pour les structures graphiques plus générales.

5.4.1 Résultats hors ligne et comparaison

Pour faire l'inférence hors ligne, nous avons utilisé la version "fixed-interval-smoothing" (ou simplement "smoothing") de l'algorithme interface (voir la section 5.2.4.3, Figure 62). Ceci correspond à estimer IU, GT et FX en utilisant toute la séquence d'observations concernant MP et SP. Plus précisément, l'algorithme utilisé permettait de calculer la MPE ("Most Probable Explanation") de UI, GT et FX compte tenu de la totalité de la séquence observée. Cette approche consiste à calculer les distributions marginales de chaque variable et de sélectionner ensuite l'événement le plus probable. Notons que de manière similaire aux HMM, il est possible de faire une génération stochastique, sans choisir

l'événement avec la probabilité maximale mais en faisant par exemple un tirage aléatoire dans une distribution non uniforme comme dans [85].

5.4.1.1 Performances

Une comparaison directe entre tous les modèles est montrée dans la Figure 72. Les résultats des trois modèles sont largement supérieurs aux niveaux de hasard qu'on retrouve en utilisant les distributions empiriques des données (26% pour l'IU, 39% pour le geste et 27% pour le regard). Par rapport à la base HMM, le modèle semi-Markovien (HSMM), grâce à sa capacité de modéliser les durées des états, obtient un meilleur taux de reconnaissance pour les unités interactionnelles (79% vs 72%) et un meilleur taux de génération de regard (69% vs 60%). Quant au modèle DBN, il surpasse nettement les deux modèles (HMM / HSMM) avec un niveau de confiance de 95% :

- 85% pour l'estimation des IU (vs 72% / 79%)
- 87% pour la génération du geste de la main (vs 85% / 86%)
- 71% pour la génération de regard (vs 60% / 69%)

Cet écart de performance peut être expliqué par le fait que les DBN autorisent des relations de dépendance directes entre toutes les observations (en particulier entre les observations d'entrée et de sortie). Dans le paradigme des HMM, il n'y a en effet aucune possibilité de relations directes entre les variables du vecteur d'observation (MP, SP, GT, FX), ce qui présente une limitation significative par rapport aux réseaux Bayésiens dynamiques généraux. D'autant plus que cette hypothèse d'indépendance entre les observations n'est pas valide dans la grande majorité des applications. Donc pour les HMM, cette hypothèse limite d'une part la capacité de modélisation et se trouve d'autre part ignorée par le praticien à cause des corrélations nécessairement existantes entre les données. Notez cependant que Bengio et Frasconi [147] ont proposé ce qu'on appelle les "input-output HMM" qui permettent de résoudre partiellement ce problème. Quant au DBN, ils représentent une solution plus générale et une alternative intéressante aux différents modèles se basant sur les HMM.

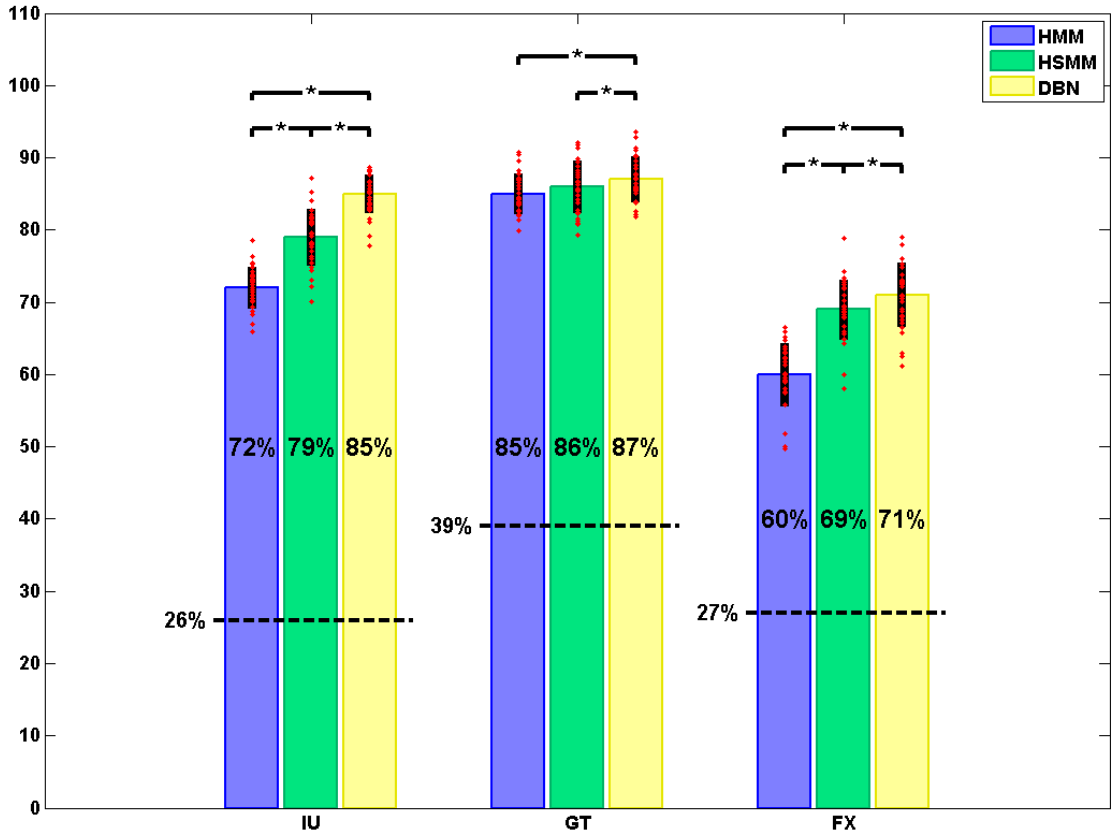


Figure 72: Les taux de prédiction de IU, GT et FX pour les trois modèles (HMM, HSMM et DBN) en mode hors ligne (smoothing). Le seuil de confiance pour tester la significativité des différences est de 95%.

5.4.1.2 Histogrammes de coordination

Au-delà de la simple comparaison entre les performances des différents modèles en termes de prédiction, nous avons évalué la capacité de chaque modèle à capturer et à reproduire la micro-coordination entre les différents flux multimodaux (c.f. parole, geste déictique et regard). À cette fin, nous avons proposé une nouvelle approche d'évaluation qui se base sur ce que nous appelons les "histogrammes de coordination" (CH pour "Coordination Histogram"). Notre approche consiste à calculer les CH issus des vraies interactions et les comparer avec les CH des flux générés par les trois modèles proposés (HMM, HSMM et DBN). Les CH sont destinés à mesurer la similitude entre les schémas de coordination des interactions générées et ceux de l'interaction réelle. Le calcul est fait de manière à avoir un CH par modalité: 1 pour la parole, 1 pour le geste et 1 CH pour le regard. Nous aurons ainsi en total 12 CH: 3 pour la vérité terrain, 3 pour le modèle HMM, 3 pour le modèle HSMM et 3 CH pour le modèle DBN.

L'histogramme de coordination d'un modèle et d'un flux particulier est calculé comme suit: d'abord, rappelons-nous que les observations de chaque modalité sont discrètes et donc caractérisées par un certain nombre d'événements de transition. Par exemple, la variable motrice GT peut prendre 5 valeurs dans le temps ("ini", "cub" , "loc", "ref" et "else"): à chaque fois qu'on transite de l'une à l'autre de ces valeurs, un évènement est émis et déclenche ainsi l'exécution d'un geste par l'organe correspondant. Le CH est construit de la manière

suivante: pour chaque événement observé dans un flux donné, nous regardons les transitions voisines des événements observés dans les autres modalités (c.-à-d. SP et FX si on étudie par exemple un événement de GT). La valeur du délai minimal entre l'évènement et les événements des autres flux est alors cumulé dans un histogramme. Pour que l'approche soit claire et plus concrète, un exemple est montré dans la Figure 73. Cette procédure est à répéter pour tous les événements d'une modalité particulière et d'un modèle donné afin d'avoir tous les délais inter-modalités utilisés dans la construction du CH final.

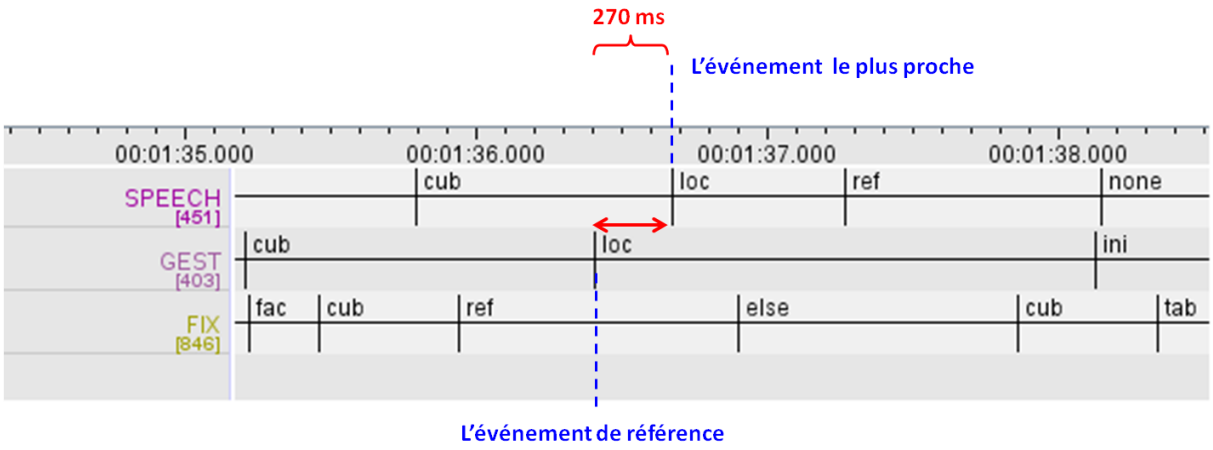


Figure 73 : Un exemple de l'approche suivi dans la construction des CH. Ici il s'agit du CH de la vérité terrain concernant la modalité geste (GT). L'événement le plus proche de "loc" (GT) est "loc" (SP), la valeur retenue pour la construction du CH de GT est positif et de 270 millisecondes.

L'approche décrite ci-dessus nous a permis de construire 12 CH qui sont montrés dans la Figure 74, Figure 75 et Figure 76. Avant d'examiner les similitudes entre les CH des interactions générées et des interactions réelles, remarquons que les CH de la vérité terrain semblent être issus d'une distribution normale. Le test de normalité de Kolmogorov-Smirnov (Tableau 5) a confirmé notre constatation, les trois histogrammes des interactions réelles sont bel et bien dérivés d'une distribution normale. Cette caractéristique montre que le comportement humain est rythmée par des motifs de dépendance temporelle. L'objectif d'un modèle de comportement est alors de modéliser et générer de manière pertinente ces motifs. Or, pour nos modèles, seuls les CH de la modalité FX des deux modèles HSMM et DBN vérifient la propriété de normalité. Ce premier résultat confirme les résultats du Chapitre 4 dans lequel nous avons démontré que le HSMM est plus performant que le HMM, et rejoint nos attentes sur la pertinence des DBN par rapports aux HMM.

En comparant visuellement les histogrammes de coordination des différents modèles, on s'aperçoit que les distributions des CH du modèle DBN sont plus proches aux distributions de la vérité terrain que les autres modèles (voir Figure 74, Figure 75 et Figure 76). Ceci est valable pour toutes les modalités étudiées c.-a.-d. SP, GT et FX. Cette proximité apparente est confirmée par la distance du Khi2 que nous avons utilisé pour calculer la distance entre les distributions des CH des interactions réelles et les distributions des interactions générées par nos trois modèles. Dans le Tableau 6, nous pouvons voir clairement que les plus petites distance par rapport aux vraies interactions sont bel et bien les distances du modèle DBN. Par

exemple, pour la modalité regard (FX), la distance du CH du modèle DBN au CH de la vérité terrain est égale à 71.34 alors qu'elle est égale à 193.36 pour le HSMM et 630.85 pour le HMM. Conformément à nos attentes, le modèle DBN, après avoir obtenu les meilleurs taux de prédiction, réalise également la coordination multimodale la plus fidèle parmi tous les modèles proposés.

	SP	GT	FX
Vérité terrain	+	+	+
DBN	-	-	+
HSMM	-	-	+
HMM	-	-	-

Tableau 5 : Tests de normalité (test de Kolmogorov-Smirnov au seuil de signification de 5%): "+" signifie que la distribution de l'histogramme provienne d'une distribution normale, sinon "-"

		DBN	HSMM	HMM
Vérité terrain	CH de SP	80.48	139.45	399.30
	CH de GT	215.47	309.27	415.27
	CH de FX	71.34	193.36	630.85

Tableau 6 : La distance de khi-deux entre l'histogramme de l'interaction réelle et les histogrammes des différents modèles en mode hors ligne (smoothing)

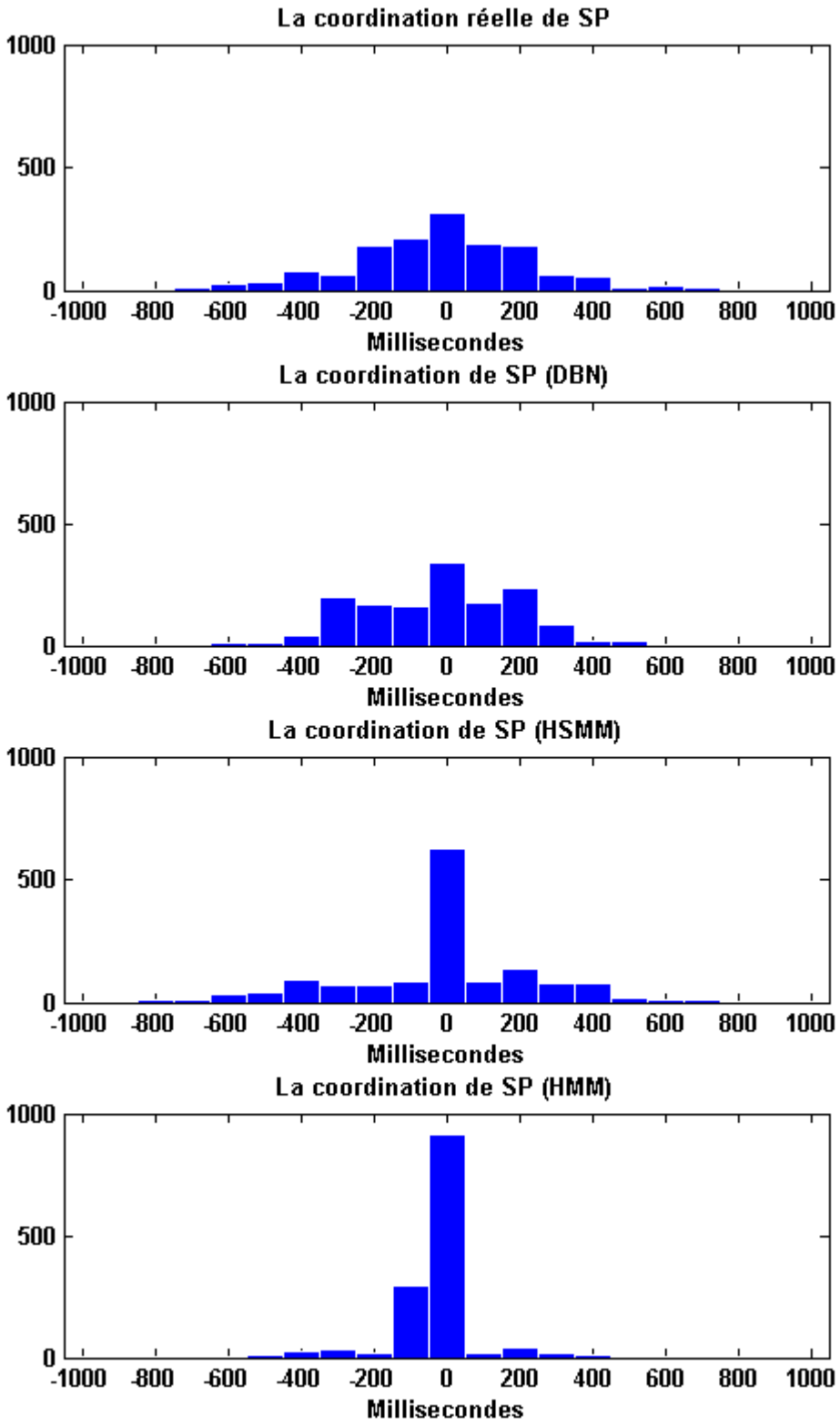


Figure 74 : Les CH de la modalité SP pour la réalité terrain et les trois modèles : DBN, HSMM et HMM

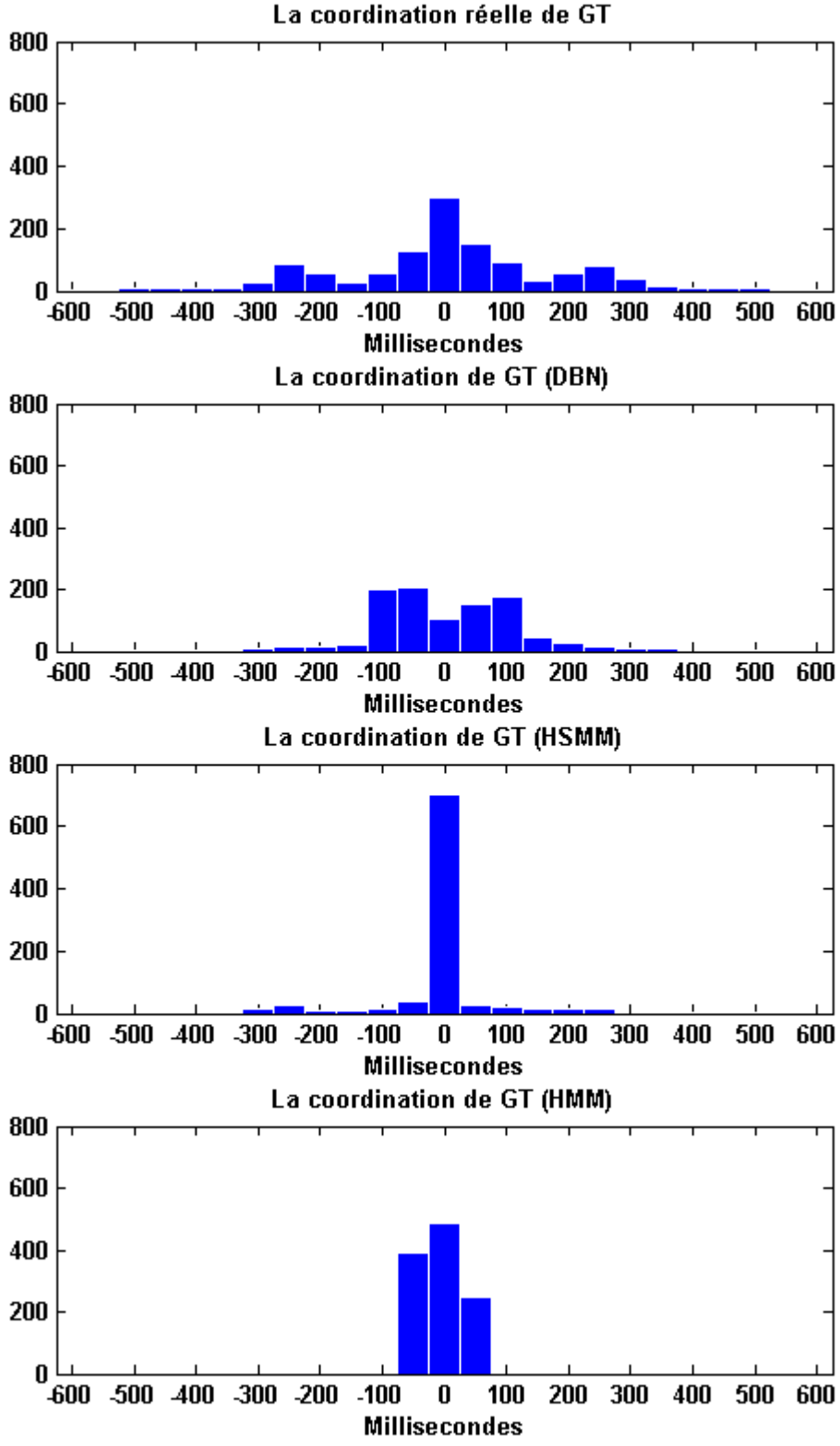


Figure 75 : Les CH de la modalité GT pour la réalité terrain et les trois modèles : DBN, HSMM et HMM

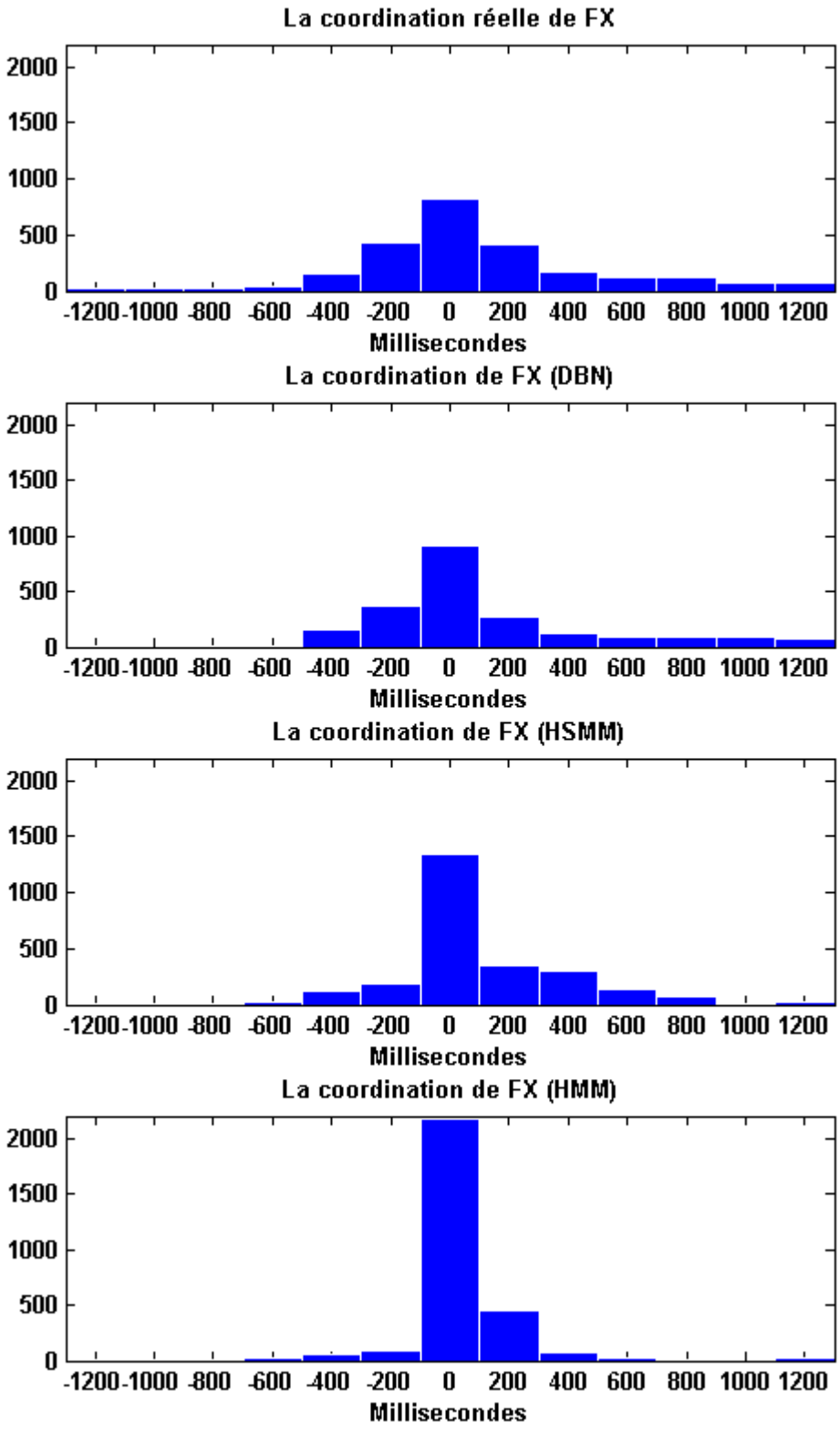


Figure 76 : Les CH de la modalité FX pour la réalité terrain et les trois modèles : DBN, HSMM et HMM

5.4.2 Résultats en ligne et comparaison

Pour faire l'inférence en ligne, nous avons généré les flux par la version "filtering" de l'algorithme interface (voir la section 5.2.4.3, Figure 62). Ceci correspond à faire l'estimation de IU, GT et FX à un instant t en utilisant la séquence d'observations concernant MP et SP jusqu'à cet instant-là. Comme pour le "smoothing", l'algorithme utilisé permettait de calculer la MPE (« Most Probable Explanation ») de IU, GT et FX, en particulier calculer les distributions marginales de chaque variable et de sélectionner ensuite l'événement le plus probable.

En regardant les chiffres de prédiction (Figure 77), nous constatons que le DBN et le HSMM sont significativement plus performants que le HMM. Ainsi, dans l'estimation des fixations (FX), le HMM obtient un taux de 61% alors que le HSMM obtient 65% et le DBN 64%. Notons que la différence entre le DBN et le HSMM n'est pas ici significative. Par contre, en examinant les distances Khi2 entre les CH réels et les CH des deux modèles (Tableau 7), on enregistre pour le DBN les plus petites distances concernant les deux modalités générées (c.-a.-d. GT et FX). Par exemple, pour le regard de l'instructeur (FX), la distance est de 391.52 pour le HSMM et 90.61 pour le DBN. De plus, les tests de normalité révèlent que seul le CH de FX du modèle DBN est issu d'une distribution normale ce qui est la cas du CH de l'interaction réelle. Avec ces deux indicateurs de plus, une légère préférence se dégage pour le modèle DBN ce qui accrédite davantage la pertinence de ce modèle.

Notons aussi que nous avons testé la version "fixed-point smoothing" qui est une autre version en ligne de l'algorithme interface et qui permet de tolérer un horizon fixe d'observations postérieures à l'instant t de l'inférence (voir la section 5.2.4.3, Figure 62). Les résultats montrent que, plus on tolère un seuil élevé d'observations futures, plus le modèle devient performant (voir Figure 78), notamment pour l'identification des IU et la génération des fixations. Bien évidement dans le cas ou on utilise ce type d'approche, il faut trouver le bon compromis entre latence et performance. La baisse significative de performance entre les deux versions "filtering" et "smoothing" est expliquée par le fait qu'une chaîne chronologique importante qui s'étale sur des intervalles supérieur à 4 slices (comme FX → GT→ SP → MP) est fortement dégradée dans l'inférence en mode "filtering". Cette chaîne est décrite dans le paragraphe 5.3.2.2 (Propriété 2). En effet, les deux variables SP et MP de cette chaîne (qui sont des variables d'entrée) sont postérieures à l'instant de la prédiction et donc ne sont plus disponibles pour l'inférence en mode "filtering". C'est ce manque d'information qui altère significativement les taux de performances en mode incrémental.

En conclusion, le modèle DBN est très sensible à la taille de l'horizon: plus grand est le nombre l'observations futures, plus il est capable d'exploiter la structure de dépendance liant les différents variables de l'application.

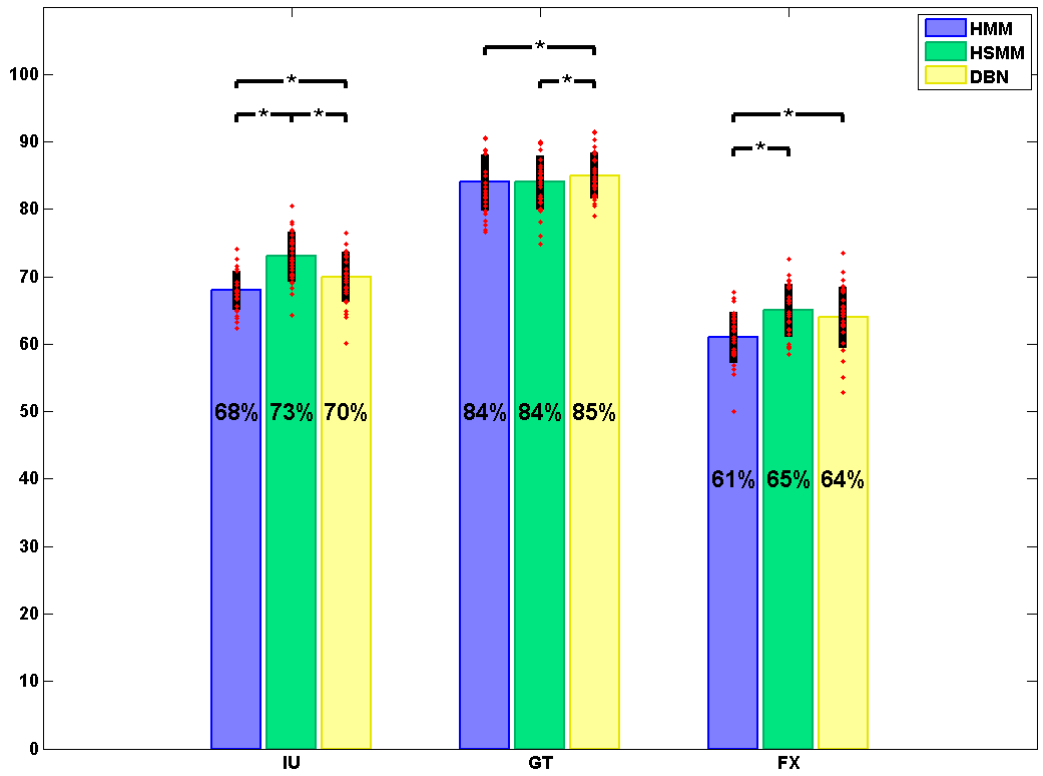


Figure 77 : Les taux de prédiction de IU, GT et FX pour les trois modèles : HMM, HSMM et DBN en mode en ligne (filtering). Le seuil de confiance pour tester la significativité des différences est de 95%.

		DBN	HSMM	HMM
Vérité terrain	CH de SP	354.84	336.37	383.78
	CH de GT	269.92	400.05	576.12
	CH de FX	90.61	391.52	481.43

Tableau 7 : La distance de khi-deux entre l'histogramme de l'interaction réelle et les histogrammes des différents modèles en mode en ligne (filtering)

	SP	GT	FX
Vérité terrain	+	+	+
DBN	-	-	+
HSMM	-	-	-
HMM	-	-	-

Tableau 8 : Tests de normalité (test de Kolmogorov-Smirnov au seuil de signification de 5%): "+" signifie que la distribution de l'histogramme provienne d'une distribution normale, sinon "-"

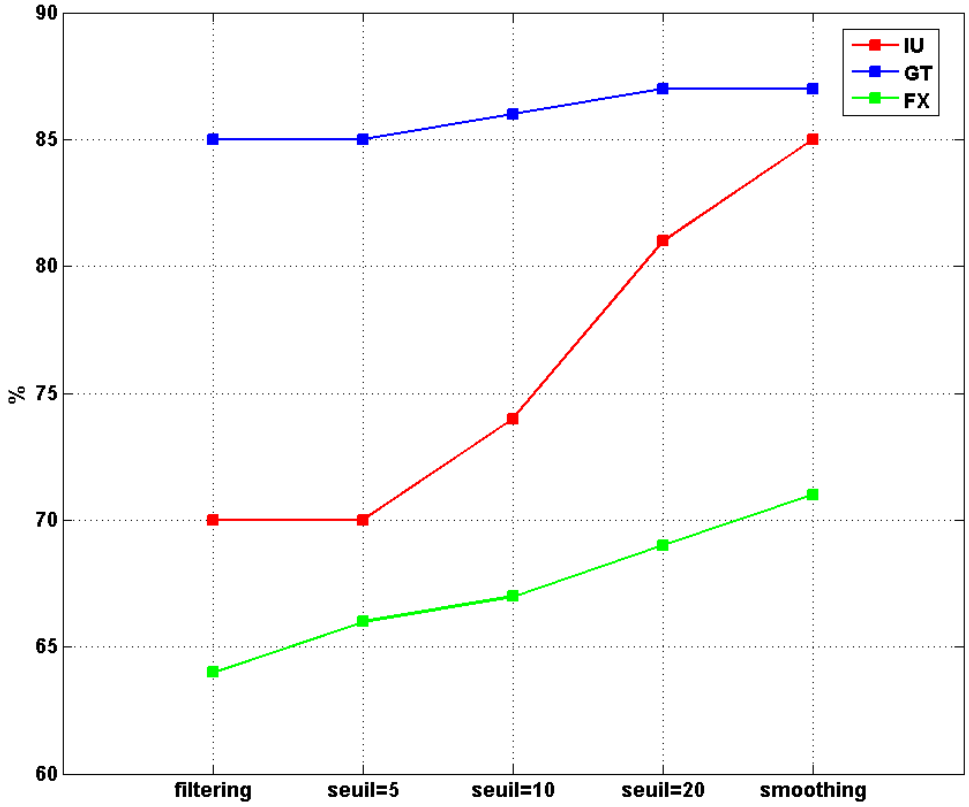


Figure 78 : Les résultats de prédiction du modèle DBN en utilisant différentes versions d'inférence (filtering, fixed-point smoothing et smoothing)

5.4.3 Résultats avec un modèle disposant d'une couche latente

D'une manière analogue au HMM, nous avons testé notre modèle DBN en rajoutant une couche latente de cardinalité 5 par unité d'interaction. Le modèle est montré dans la Figure 79. Les résultats hors ligne (cf. Figure 80) et en ligne (cf. Figure 81) montrent qu'il n'y a pas de différence avec le modèle initial qui ne dispose pas d'états latents. En effet, la structure de dépendance est suffisamment riche pour que l'ajout d'une couche latente soit sans impact sur les performances de prédiction. Néanmoins, une perspective à explorer pour améliorer ce modèle est d'initialiser le nombre et les distributions de probabilités de ces états latents non pas aléatoirement mais en procédant à une approche similaire à celle utilisée pour le modèle IDHMM modifié (cf. section 4.5.2.3).

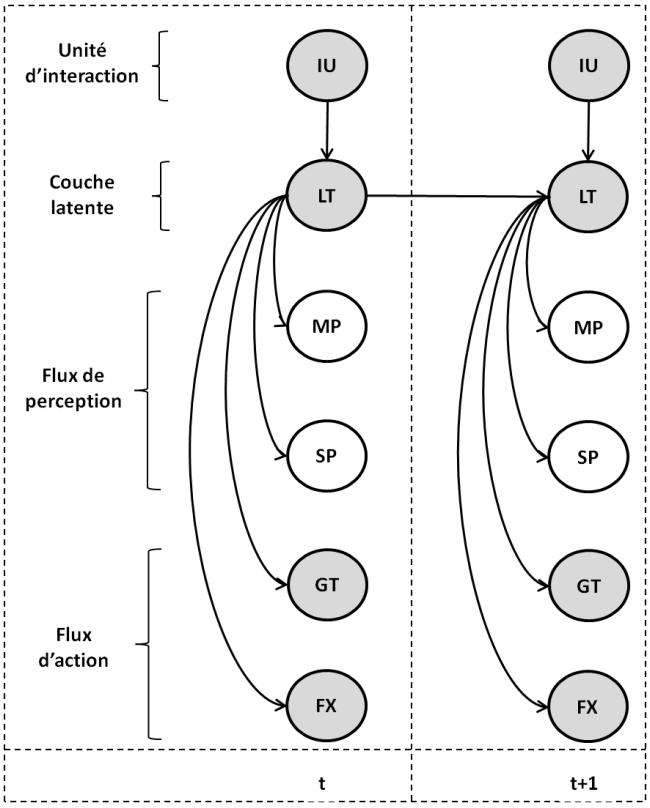


Figure 79: Modèle DBN avec une couche latente (LT). Pour des raisons de clarté, nous avons omis de dessiner les autres dépendances, elles sont similaires au modèle DBN initial.

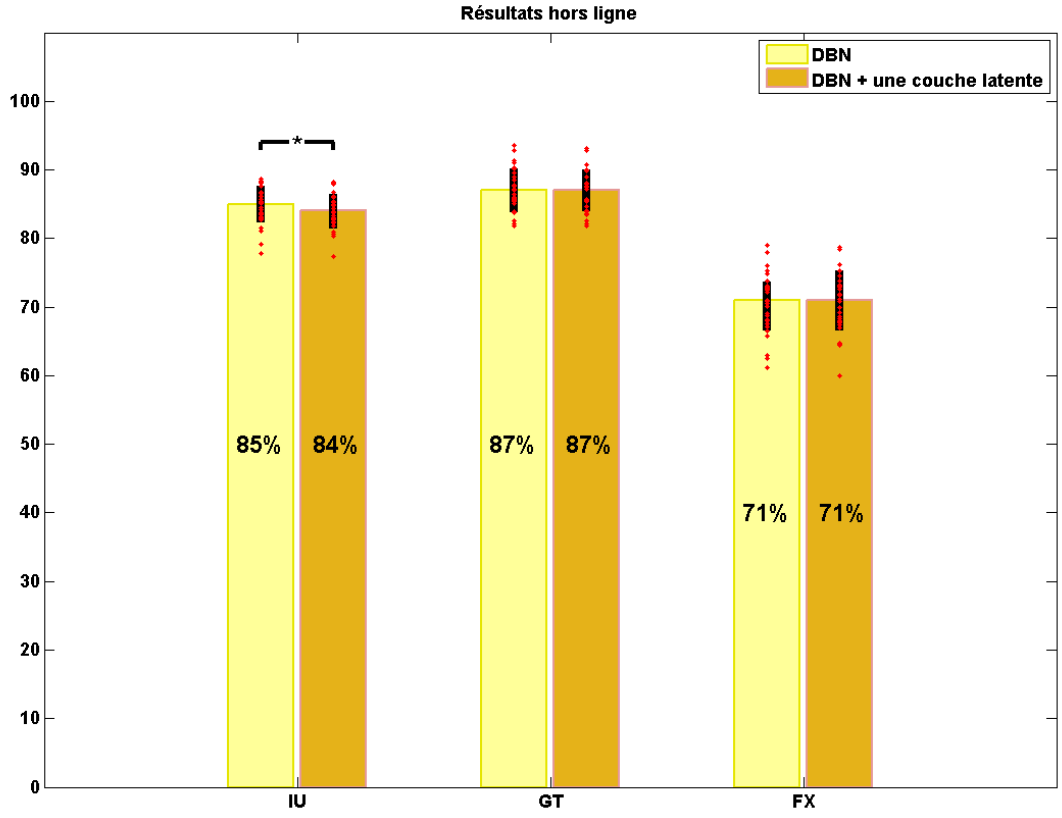


Figure 80: Résultats hors ligne: DBN initial vs. DBN avec une couche latente

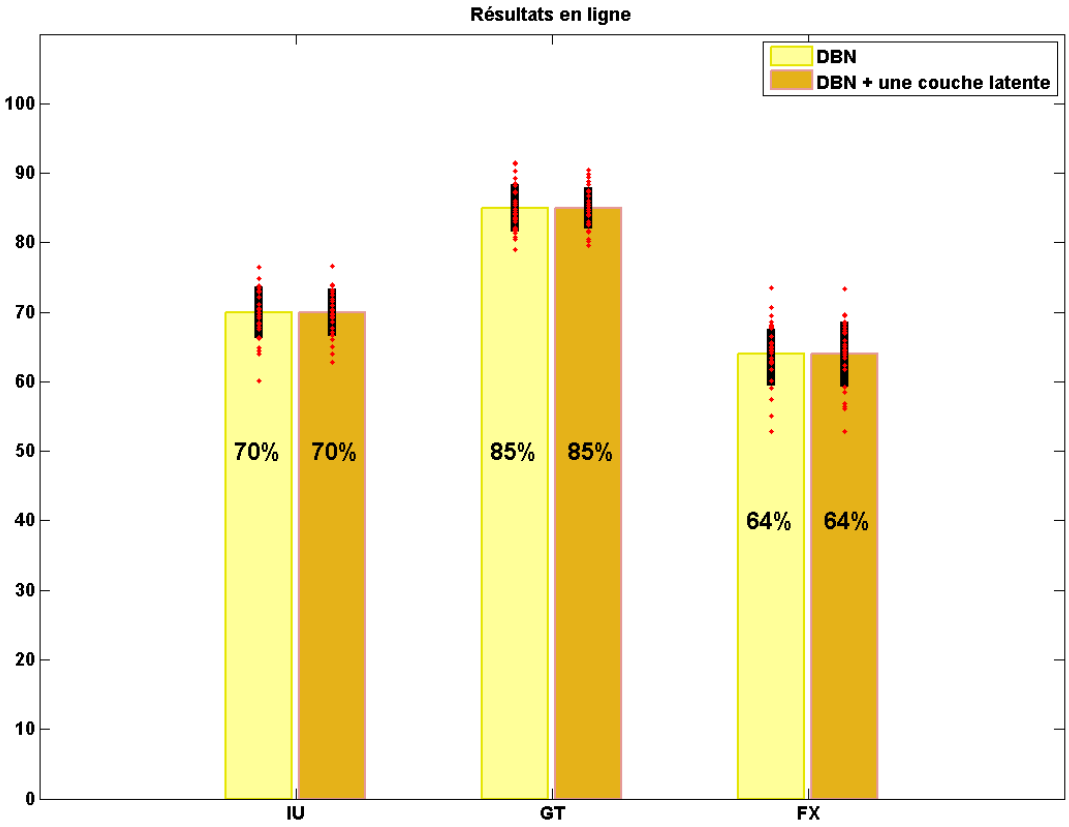


Figure 81: Résultats en ligne: DBN initial vs. DBN avec une couche latente

5.5 Conclusions

Dans ce chapitre, nous avons poursuivi notre quête d'un modèle de comportement multimodal capable d'assurer une reconnaissance efficace et une génération pertinente des signaux sociaux. Comme suite logique des améliorations (après les classifieurs et les HMM), nous avons proposé dans ce chapitre un modèle probabiliste graphique basé sur le formalisme des DBN qui présentent plusieurs avantages comparé aux HMM, notamment en terme de couplage direct des distributions conditionnelles des observations. Les données utilisées sont les données des interactions dyadiques de notre deuxième conversation face-à-face (jeu de cubes). Le but était d'estimer l'IU et de prédire le geste (GT) et le regard (FX) de l'instructeur.

Les dépendances conditionnelles entre les variables ont été apprises et la structure résultante du DBN se révèle assez pertinente: elle met en évidence les relations complexes qui régissent les relations de dépendance entre les différentes modalités du comportement humain. Afin d'évaluer le modèle appris, nous l'avons comparé à deux modèles déjà présentés (HMM et HSMM). Grâce à la structure apprise, le modèle DBN présente les meilleures performances, et ceci dans les deux tâches de reconnaissance et de génération (surtout en mode hors ligne). Afin d'aller au delà de la simple course aux taux de prédiction, nous avons introduit le concept d'histogramme de coordination (CH), qui est une nouvelle méthodologie pour évaluer la capacité des modèles à reproduire fidèlement la façon dont un humain coordonne les diverses modalités de son comportement. Les CH du DBN sont les plus

proches visuellement et analytiquement au CH réelles, en comparaison de ceux produits par les modèles HMM et HSMM. En considérant l'ensemble des indicateurs, le modèle DBN s'est avéré le plus performant à la fois dans le mode hors ligne qu'en mode en ligne. Pour terminer, rappelons qu'il sera préférable de tolérer un certain temps de latence dans l'inférence si on veut bien exploiter la richesse de la structure du modèle DBN et donc optimiser les performances de prédiction.

Conclusions et Perspectives

Rappel du contexte

Cette thèse se situe dans le contexte du traitement des signaux sociaux (SSP) dans les interactions face-à-face. Les études sur le comportement humain ont confirmé que les interactions humaines sont gérées par des boucles de perception et d'action qui permettent l'analyse et la génération continue des signaux sociaux. Ces signaux peuvent être des actions, des émotions, des attitudes ou des relations sociales. Ils sont transmis via des comportements non verbaux comme les gestes brachio-manuels, les mouvements de tête, le regard etc. Le challenge principal de cette thèse est de développer des modèles computationnels de comportement multimodal qui permettent de modéliser une interaction face-à face et de faire une reconnaissance et une génération pertinente des signaux sociaux. L'objectif à long terme de notre recherche est d'implémenter ces modèles sensori-moteurs sur des agents artificiels sociaux comme les agents conversationnels virtuels ou les robots humanoïdes. Nous visons via ces modèles, à doter ces systèmes interactifs d'une intelligence sociale qui les rende capables de mener une communication co-verbale pertinente et crédible avec un partenaire humain. L'immense partie des travaux dans le domaine de l'interaction homme-robot repose essentiellement sur des modèles à base de règles. Notre approche est différente car elle se base sur la modélisation statistique des interactions sociales afin de d'extraire les modèles souhaités.

Démarche et résultats

L'approche que nous avons suivie pour développer les modèles de comportement multimodal est une approche basée sur les données. A partir de vraies traces d'interaction, nous avons construit des modèles interactionnels par apprentissage statistique. A cette fin, nous avons utilisé des jeux de données issus de deux interactions face-à-face originales: la première est un jeu de répétition de phrases et la deuxième est un jeu de cubes. Les deux interactions étaient soigneusement pensées et enregistrées afin d'étudier et de modéliser les boucles de perception et action que l'être humain utilise dans son interaction avec un partenaire dans une tâche donnée. Avec le premier jeu, les expérimentateurs cherchaient à modéliser le comportement du regard dans une interaction située. Avec le second jeu, nous avons élargi les capacités d'actions afin de pouvoir étudier de près les stratégies de l'attention mutuelle et de la deixis multimodale d'objets et de lieux. Indépendamment de l'interaction étudiée, les modèles de comportements suggérés permettent de situer l'action du tuteur humain dans l'IU (unité interactionnelle) la plus probable et de générer d'une manière incrémentale les meilleurs actions étant donnée cette unité d'interaction, les flux perceptuels et les objectifs joints des deux partenaires. Pour l'évaluation, nous sommes allés au delà des taux de reconnaissance et de génération. Selon l'interaction et le modèle étudié, nous avons proposé des méthodes pour juger de la conformité des interactions générées aux interactions réelles.

En particulier, pour la première interaction (jeu de répétition de phrases) l'objectif était d'estimer l'IU et ensuite de générer le regard du sujet principal. Comme première démarche, nous avons proposé des modèles basés sur des classifieurs notamment les SVM et les arbres de décision. Les résultats ont montré qu'il n'y a pas de différence significative entre les SVM et les arbres de décision. De plus, nous avons remarqué que les modèles proposés n'arrivent pas à détecter l'organisation séquentielle de la tâche par le fait qu'ils ne disposent pas de propriétés de modélisation temporelle. Pour cette raison, nous avons introduit les modèles avec mémoire qui ont permis d'améliorer significativement les taux de reconnaissance et de génération. Fort de ce constat, nous avons proposé ensuite des modèles basés sur les chaînes de Markov cachés (HMM) qui ont l'avantage d'être des modèles proprement séquentiels et donc mieux adaptés à notre application. Le premier modèle proposé est appelé IDHMM ("Incremental Discrete Hidden Markov Model"). Etant donné que l'interaction étudiée obéit à une syntaxe spécifique (succession d'unités interactionnelles), nous avons modélisé l'interaction par un seul HMM global qui concatène les HMM représentant les différentes IU. Quant au vecteur d'observations, il est conçu de manière à modéliser les boucles de perception-action. Les états cachés sont donc des états sensori-moteurs. Une fois l'apprentissage fait, on extrait deux modèles: un modèle de reconnaissance et un modèle de génération. Les deux phases de décodage et de génération sont effectuées de manière incrémentale grâce à une version modifiée de l'algorithme Short-Time Viterbi (BSTV). Les résultats ont montré que l'IDHMM donne des performances meilleures que les classifieurs, et que ces performances sont aussi bonnes en mode incrémental qu'en mode hors ligne. Nous avons testé également les modèles personnels (ID) versus le modèle moyen (II). Les résultats des modèles ID étaient moins bons. Une analyse MDS a pu détecter une relation sociale hiérarchique déjà existante entre les sujets. Ce résultat confirme que les comportements du regard reflètent fidèlement les relations sociales existantes entre les interlocuteurs. En outre, une nouvelle méthodologie pour initialiser le modèle IDHMM a été explorée, donnant une amélioration en termes de bouclage sensori-moteur sans pour autant améliorer les taux de reconnaissance et de génération. Le dernier modèle testé sur cette base de données est intitulé IDHSMM ("Incremental Discrete Hidden Semi-Markov Model"). Comparé aux modèles précédents, ce modèle procure le meilleur taux de génération, le meilleur bouclage sensori-moteur et les meilleures durées pour les régions d'intérêts du regard généré. Ces résultats affirment la pertinence de la modélisation de la durée dans les états sensori-moteurs.

La deuxième base de données (jeu de cubes) étant plus riche en modalités, nous avons opté pour un modèle probabiliste graphique plus expressif et complexe que les HMM, à savoir les réseaux Bayésiens dynamiques (les DBN). Le but pour cette interaction était d'estimer l'IU et de prédire le geste (GT) et le regard (FX) de l'instructeur. Les dépendances conditionnelles entre les variables ont été apprises et la structure résultante du DBN se révèle assez pertinente et décrit avec précision les relations complexes entre les différentes modalités du comportement humain. Comparé aux modèles HMM et HSMM en mode hors ligne, le modèle DBN a conduit aux meilleures performances, dans les deux tâches de reconnaissance et de génération.

Nous avons introduit également le concept de l'histogramme de coordination (CH), qui est une nouvelle approche pour évaluer la capacité des modèles à reproduire fidèlement la façon dont un humain coordonne les modalités de son comportement. En d'autres termes, l'histogramme de coordination (CH) permet de calculer le profil moyen de la coordination inter-modalités. Comparé aux CH des modèles HMM et HSMM, les CH du DBN étaient plus proches visuellement et analytiquement aux CH réels. En mode en ligne, le modèle DBN n'exploite pas pleinement la richesse de la structure de son réseau: on constate alors une baisse des performances comparé au mode hors ligne, mais qui restent légèrement supérieures à celles des HMM et HSMM. Nous préconisons donc l'usage du modèle DBN pour continuer l'effort de développement de modèles d'interaction.

Perspectives

Dans cette thèse, nous avons pu mettre en place une interaction face-à-face dyadique avec un scénario bien précis de jeu de cubes. Comme prochaine étape, on peut envisager de nouvelles interactions dyadiques avec des scénarios plus ouverts et moins restreints comme les consultations médicales ou les jeux de quiz. Un scénario plus ouvert limitera les redondances des données et enrichira la variabilité des échanges et des comportements. Dans un tel contexte, le challenge sera certainement de trouver la solution pour aboutir à un apprentissage réussi et efficace.

Jusque là, nous n'avons traité que des interactions dyadiques, une perspective intéressante sera de passer de l'interaction dyadique à l'interaction multi-parties. Ce type d'interaction exige des compétences supplémentaires concernant l'analyse de la scène, la gestion de dialogue et le maintien de l'attention mutuelle avec plusieurs partenaires dont les intentions sont différentes.

Pour plus de richesse, il sera important aussi d'envisager des interactions, où il sera possible d'exploiter et de modéliser des données à la fois discrètes et continues. On pourrait alors envisager d'intégrer directement l'apprentissage de contrôleurs gestuels – ou de leurs paramétrages – dans le modèle interactif. Il faut également penser à remplacer toutes les procédures d'annotation manuelle par des algorithmes d'annotation automatique, notamment les algorithmes de vision par ordinateur.

Dans notre travail, nous avons proposé plusieurs modèles de comportement pour l'analyse et la génération des comportements dans les interactions face-à-face. Nous avons testé des modèles discriminatifs non séquentiels come les SVM et des modèles génératifs séquentiels comme les HMM ou aussi les DBN. Une perspective intéressante est d'explorer les modèles discriminatifs séquentiels notamment les CRF ("Conditional Random Field") qui sont largement utilisés dans le traitement des signaux sociaux [61][80]. Une autre approche intéressante est d'associer des modèles discriminatifs à des modèles génératifs comme dans [210] ou les auteurs ont combiné des CRF avec des HMM. L'idée est d'utiliser un modèle avec deux couches, une première couche discriminative de CRF qui va détecter l'état sensori-motor (état caché) et l'unité d'interaction correspondante, et une deuxième couche générative de HMM qui va faire la génération adéquate à l'état sensori-moteur estimé. C'est la puissance

discriminative des CRF qui fait la différence avec un modèle se basant simplement sur des HMM. Cette approche se déroule en 4 étapes :

- Apprentissage conjoint du HMM à partir de toutes les données de perception et d'action.
- En utilisant le HMM appris, on estime la séquence des états cachés à partir de toutes les données.
- Apprentissage du CRF en mettant en correspondance la séquence des états cachés estimée dans la deuxième étape et les données perceptuelles uniquement.
- Phase de test : à partir des données perceptuelles on utilise le CRF pour estimer l'état sensori-moteur, ensuite on utilise l'état correspondant en HMM pour faire la génération.

Nous pouvons même envisager des combinaisons un peu plus complexes notamment un CRF en couche discriminative et un DBN en couche générative pour bien exploiter la capacité des DBN à modéliser les relations complexes entre les données sensori-motrices.

Coté modèles, la notion de la perception active doit être abordée avec plus de profondeur. Les prochains modèles doivent modéliser d'une manière plus concrète ce phénomène largement utilisé par l'être humain. Les modèles appris doivent être capables de lier l'entrée (perception) à la sortie (action) et surtout la sortie à l'entrée.

L'étape la plus importante à faire dans les prochains travaux est d'implémenter les modèles proposés sur le robot humanoïde iCub (NINA) disponible au Gipsa-lab (Figure 82). Dans cette thèse, nous avons évalué nos modèles de comportements d'une manière objective, en utilisant les taux de reconnaissance et de génération, et aussi en proposant des critères pertinents (bouclage sensori-moteur, histogramme de coordination etc.) permettant d'évaluer la capacité des modèles à reproduire les interactions réelles. La mise en œuvre et l'implémentation robotique des modèles sensori-moteurs développés, permettra au robot iCub de bénéficier des démonstrations du tuteur humain et d'essayer de le remplacer dans les scénarios (par ex. l'instructeur dans l'interaction de jeu de cubes) et ainsi d'interagir en situation réelle avec un partenaire humain. Nous pourrons ainsi évaluer subjectivement les différents modèles proposés et comparer les résultats trouvés avec l'évaluation objective déjà effectuée. Cette évaluation subjective postérieure à l'interaction peut se faire par le partenaire humain ou aussi par un ensemble d'observateurs extérieurs.

Avec le développement de la plateforme de téléopération ("Beaming") de Gipsa-lab (Figure 83), on pourra également comparer les réactions et les évaluations du partenaire humain lorsqu'il interagit avec l'iCub téléopéré et l'iCub contrôlé par nos modèles de comportements. On s'attend à ce que l'interlocuteur réagisse différemment en sachant que le robot est opéré par un humain et non pas par un algorithme. On pourra éventuellement explorer plusieurs autres configurations et comparaisons, en faussant les informations données au partenaire humain.

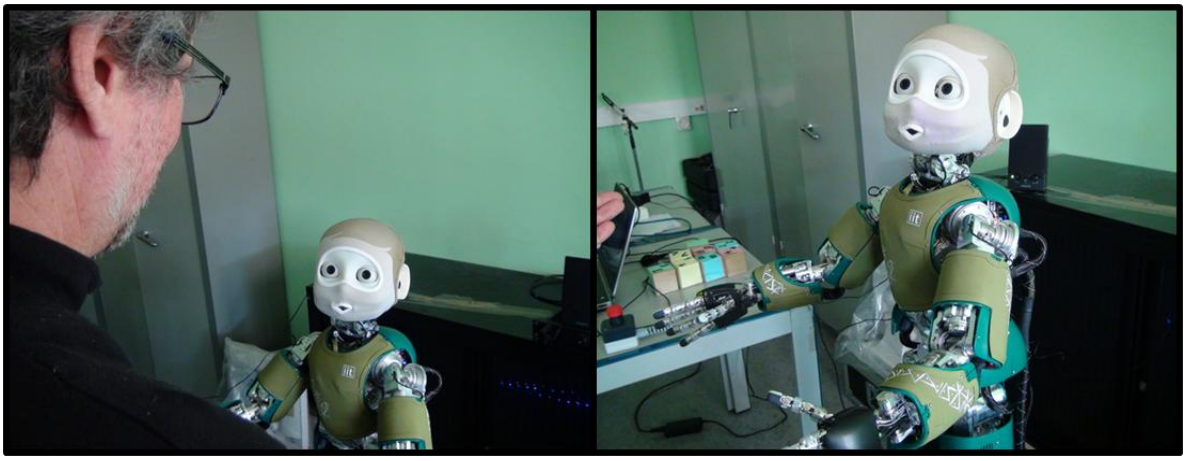


Figure 82: Interaction face-à-face avec le robot NINA

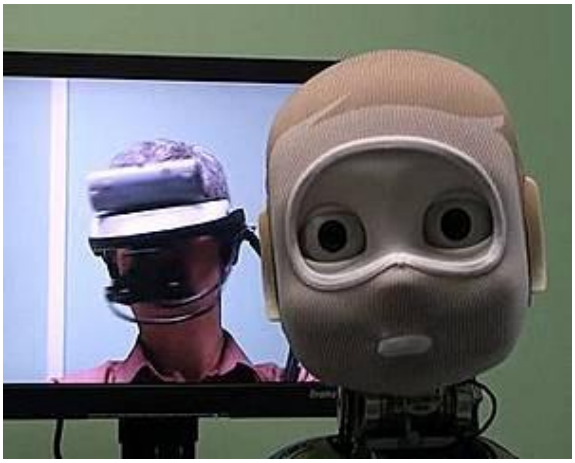


Figure 83: Téléopération du regard et des mouvements de tête dans la plateforme de "Beaming" à Gipsa-Lab

Liste des publications

2015

- Mihoub, A., G. Bailly, C. Wolf and F. Elisei. “Learning multimodal behavioral models for face-to-face social interaction.” Journal on Multimodal User Interfaces (JMUI), Volume 9, Issue 3 (2015), Page 195-210.
- Bailly, G., A. Mihoub, C. Wolf and F. Elisei. Learning joint multimodal behaviors for face-to-face interaction: performance & properties of statistical models. Human-Robot Interaction (HRI). Workshop on behavior coordination between animals, humans and robots, Portland, OR.
- Mihoub, A., G. Bailly, C. Wolf and F. Elisei (en révision). “Graphical models for social behavior modeling in face-to face interaction.” Pattern Recognition Letters.

2014

- Mihoub, A., G. Bailly and C. Wolf (2014). Modelling perception-action loops: comparing sequential models with frame-based classifiers. Human-Agent Interaction (HAI), Tsukuba, Japan, 309-314.
- Mihoub, A., G. Bailly and C. Wolf (2014). Modeling sensory-motor behaviors for social robots, Workshop Affect, Compagnon Artificiel, Interaction (WACAI 2014) , Rouen, France, 77-82.

2013

- Mihoub, A., G. Bailly & C. Wolf (2013). Social behavior modeling based on Incremental Discrete Hidden Markov Models. ACM Multimedia. International Workshop on Human Behavior Understanding (HBU), Barcelona, Spain, 172-183.

Références

[1] A. Kendon, R. M. Harris, M. R. Key, and International Congress of Anthropological and Ethnological Sciences, *Organization of behavior in face-to-face interaction*. The Hague; Chicago: Mouton ; Distributed in the USA and Canada by Aldine, 1975.

[2] S. Scherer, S. Marsella, G. Stratou, Y. Xu, F. Morbini, A. Egan, and L.-P. Morency, "Perception markup language: towards a standardized representation of perceived nonverbal behaviors," in *Intelligent Virtual Agents*, 2012, pp. 455–463.

[3] E. Shriberg, "Spontaneous speech: How people really talk and why engineers should care," in *in Proc. European Conf. on Speech Communication and Technology (Eurospeech*, 2005, pp. 1781–1784.

[4] M. L. Knapp, J. A. Hall, and T. G. Horgan, *Nonverbal Communication in Human Interaction*, 8 edition. Boston, MA: Wadsworth Publishing, 2013.

[5] V. L. Manusov and M. L. Patterson, *The Sage Handbook of Nonverbal Communication*. Thousand Oaks, Calif.: SAGE Publications Inc, 2006.

[6] R. R. Hassin, J. S. Uleman, and J. A. Bargh, Eds., *The New Unconscious*, 1 edition. Oxford: Oxford University Press, 2006.

[7] J. L. Lakin, V. E. Jefferis, C. M. Cheng, and T. L. Chartrand, "The Chameleon Effect as Social Glue: Evidence for the Evolutionary Significance of Nonconscious Mimicry," *J. Nonverbal Behav.*, vol. 27, no. 3, pp. 145–162, Sep. 2003.

[8] G. Bailly, "Boucles de perception-action et interaction face-à-face," *Rev. Fr. Linguist. Appliquée*, vol. 13, no. 2, pp. 121–131, 2009.

[9] K. R. Thórisson, "Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action," in *Multimodality in Language and Speech Systems*, B. Granström, D. House, and I. Karlsson, Eds. Springer Netherlands, 2002, pp. 173–207.

[10] A. Vinciarelli and M. Pantic, "Social Signal Processing," in *The Oxford Handbook of Affective Computing*, Oxford University Press, 2015.

[11] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D'Errico, and M. Schroeder, "Bridging the Gap between Social Animal and Unsocial Machine: A Survey of Social Signal Processing," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 69–87, 2012.

[12] I. Khaldûn and B. Lawrence, *The Muqaddimah: An Introduction to History*, Abridged edition. Princeton, N.J.: Princeton University Press, 2004.

[13] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annu. Rev. Neurosci.*, vol. 27, pp. 169–192, 2004.

[14] C. D. Frith and U. Frith, "Social cognition in humans," *Curr. Biol. CB*, vol. 17, no. 16, pp. R724–732, Aug. 2007.

[15] M. Pantic, A. Nijholt, A. Pentland, and T. S. Huanag, "Human-Centred Intelligent Human–Computer Interaction (HCI^2): how far are we from attaining it?," *Int J Auton Adapt Commun Syst*, vol. 1, no. 2, pp. 168–187, Aug. 2008.

[16] A. (Sandy) Pentland, "Socially Aware Computation and Communication," *Computer*, vol. 38, no. 3, pp. 33–40, 2005.

[17] R. W. Picard, "Affective computing: challenges," *Int. J. Hum.-Comput. Stud.*, vol. 59, no. 1–2, pp. 55–64, Jul. 2003.

[18] A. Pentland, "Social Signal Processing [Exploratory DSP]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 108–111, Jul. 2007.

[19] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image Vis. Comput.*, vol. 27, no. 12, pp. 1743–1759, Nov. 2009.

[20] K. Albrecht, *Social Intelligence: The New Science of Success*. John Wiley & Sons, 2006.

[21] I. Poggi and D. Francesca, “Cognitive Modelling of Human Social Signals,” in *Proceedings of the 2Nd International Workshop on Social Signal Processing*, New York, NY, USA, 2010, pp. 21–26.
[22] I. Poggi and F. D’Errico, “Social signals: a framework in terms of goals and beliefs,” *Cogn. Process.*, vol. 13, no. 2, pp. 427–445, Jul. 2012.
[23] N. Campbell, “On the structure of spoken language,” *Speech Prosody Dresd. Ger.*, 2006.
[24] R. Conte and C. Castelfranchi, *Cognitive and Social Action*. Psychology Press, 1995.
[25] J. Allwood, E. Ahlsén, J. Lund, and J. Sundqvist, “Multimodality in own communication management,” in *Papers from the Second Nordic Conference on Multimodal Communication*, 2006, vol. 92, p. 43.
[26] D. Heylen, “Challenges ahead: head movements and other social acts during conversations,” presented at the Joint Symposium on Virtual Social Agents, Hatfield, UK, 2005, pp. 45–52.
[27] A. J. Fridlund, A. N. Gilbert, C. E. Izard, and A. N. Burdett, “Emotions and facial expression,” *Science*, vol. 230, no. 4726, pp. 607–608, Aug. 1985.
[28] G. L., Daniel Gilbert, Susan T. Fiske, *The Handbook of Social Psychology*. Mass., 1998.
[29] K. Bousmalis, M. Mehu, and M. Pantic, “Spotting agreement and disagreement: A survey of nonverbal audiovisual cues and tools,” in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 2009, pp. 1–9.
[30] H. H. Kelley and J. W. Thibaut, *Interpersonal relations: a theory of interdependence*. Wiley, 1978.
[31] D. Gatica-Perez, “Automatic nonverbal analysis of social interaction in small groups: A review,” *Image Vis. Comput.*, vol. 27, no. 12, pp. 1775–1787, 2009.
[32] D. Gatica-Perez, “Analyzing group interactions in conversations: a review,” in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, 2006, pp. 41–46.
[33] F. De la Torre and J. F. Cohn, “Facial expression analysis,” in *Visual analysis of humans*, Springer London, 2011, pp. 377–409.
[34] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, “Paralinguistics in speech and language—State-of-the-art and the challenge,” *Comput. Speech Lang.*, vol. 27, no. 1, pp. 4–39, Jan. 2013.
[35] A. Pentland, T. Choudhury, N. Eagle, and P. Singh, “Human dynamics: computation for organizations,” *Pattern Recognit. Lett.*, vol. 26, no. 4, pp. 503–511, Mar. 2005.
[36] T. Choudhury and A. Pentland, “Characterizing Social Interactions Using the Sociometer,” in *Proceedings of NAACOS 2004*, 2004.
[37] J. R. Curhan and A. Pentland, “Thin slices of negotiation: predicting outcomes from conversational dynamics within the first 5 minutes,” *J. Appl. Psychol.*, vol. 92, no. 3, pp. 802–811, May 2007.
[38] T. Choudhury and A. Pentland, “The sociometer: A wearable device for understanding human networks,” in *CSCW’02 Workshop: Ad hoc Communications and Collaboration in Ubiquitous Computing Environments*, 2002.
[39] K. Otsuka, Y. Takemae, and J. Yamato, “A probabilistic inference of multiparty-conversation structure based on Markov-switching models of gaze patterns, head directions, and utterances,” in *Proceedings of the 7th international conference on Multimodal interfaces*, New York, NY, USA, 2005, pp. 191–198.
[40] K. Otsuka, H. Sawada, and J. Yamato, “Automatic inference of cross-modal nonverbal interactions in multiparty conversations: ‘who responds to whom, when, and how?’ from gaze, head gestures, and utterances,” in *Proceedings of the 9th international conference on Multimodal interfaces*, New York, NY, USA, 2007, pp. 255–262.

[41] R. Ishii, K. Otsuka, S. Kumano, and J. Yamato, “Analysis and modeling of next speaking start timing based on gaze behavior in multi-party meetings,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 694–698.
[42] R. Ishii, K. Otsuka, S. Kumano, and J. Yamato, “Analysis of Respiration for Prediction of ‘Who Will Be Next Speaker and When?’ in Multi-Party Meetings,” in *Proceedings of the 16th International Conference on Multimodal Interaction*, New York, NY, USA, 2014, pp. 18–25.
[43] N. Jovanović and R. op den Akker, “Towards automatic addressee identification in multi-party dialogues,” in *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, Pennsylvania, USA, 2004, pp. 89–92.
[44] N. Jovanović, R. op den Akker, and A. Nijholt, “A corpus for studying addressing behavior in multi-party dialogues,” in *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, 2005, pp. 107–116.
[45] N. Jovanović, R. op den Akker, and A. Nijholt, “Addressee Identification In Face-to-Face Meetings,” presented at the 11th Conference of the European Chapter of the ACL, EACL 2006, Pennsylvania, USA, 2006, pp. 169–176.
[46] N. Jovanovic, “To whom it may concern : adressee identification in face-to-face meetings,” Doctoral Thesis, University of Twente, Enschede, 2007.
[47] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, “The AMI Meeting Corpus,” in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.
[48] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, “Modeling individual and group actions in meetings with layered HMMs,” *Multimed. IEEE Trans. On*, vol. 8, no. 3, pp. 509–520, 2006.
[49] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, “A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 39–58, Jan. 2009.
[50] H. Gunes and M. Pantic, “Automatic, Dimensional and Continuous Emotion Recognition:,” *Int. J. Synth. Emot.*, vol. 1, no. 1, pp. 68–99, Jan. 2010.
[51] S. Petridis and M. Pantic, “Audiovisual discrimination between laughter and speech,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, 2008, pp. 5117–5120.
[52] S. Kumano, K. Otsuka, D. Mikami, M. Matsuda, and J. Yamato, “Understanding Communicative Emotions from Collective External Observations,” in *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 2201–2206.
[53] “Repository of the INTERSPEECH Computational Paralinguistics ChallengE (ComParE) Series.” [Online]. Available: http://compare.openaudio.eu/. [Accessed: 29-Jul-2015].
[54] M. S. Mast, “Dominance as Expressed and Inferred Through Speaking Time,” *Hum. Commun. Res.*, vol. 28, no. 3, pp. 420–450, Jul. 2002.
[55] R. Rienks and D. Heylen, “Dominance Detection in Meetings Using Easily Obtainable Features,” in *In Bourlard, H., & Renals, S. (Eds.), Revised Selected Papers of the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2005, pp. 76–86.
[56] D. B. Jayagopi, H. Hung, C. Yeo, and D. Gatica-Perez, “Modeling dominance in group conversations using nonverbal activity cues,” *Audio Speech Lang. Process. IEEE Trans. On*, vol. 17, no. 3, pp. 501–513, 2009.

[57] H. Hung, Y. Huang, G. Friedland, and D. Gatica-Perez, “Estimating Dominance in Multi-Party Meetings Using Speaker Diarization,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 4, pp. 847–860, May 2011.
[58] E. D’Arca, N. M. Robertson, and J. R. Hopgood, “Look who’s talking: Detecting the dominant speaker in a cluttered scenario,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1532–1536.
[59] A. Pentland, J. Curhan, R. Khilnani, M. Martin, N. Eagle, R. Caneel, and A. Madan, “Toward a Negotiation Advisor,” presented at the UIST 04, 2004.
[60] A. P. Hare, “Types of Roles in Small Groups A Bit of History and a Current Perspective,” *Small Group Res.*, vol. 25, no. 3, pp. 433–448, Jan. 1994.
[61] H. Salamin and A. Vinciarelli, “Automatic Role Recognition in Multiparty Conversations: An Approach Based on Turn Organization, Prosody, and Conditional Random Fields,” *IEEE Trans. Multimed.*, vol. 14, no. 2, pp. 338–345, April.
[62] S. Banerjee and A. I. Rudnicky, “Using simple speech–based features to detect the state of a meeting and the roles of the meeting participants,” 2004.
[63] I. de Kok and D. Heylen, “Integrating Backchannel Prediction Models into Embodied Conversational Agents,” in *Intelligent Virtual Agents*, Y. Nakano, M. Neff, A. Paiva, and M. Walker, Eds. Springer Berlin Heidelberg, 2012, pp. 268–274.
[64] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore, “BEAT: The Behavior Expression Animation Toolkit,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2001, pp. 477–486.
[65] S. Kopp, B. Jung, N. Lessmann, and I. Wachsmuth, “Max - A Multimodal Assistant in Virtual Reality Construction,” *KI*, vol. 17, no. 4, p. 11, 2003.
[66] B. Krenn, “The NECA project: Net environments for embodied emotional conversational agents,” in *Proc. of Workshop on emotionally rich virtual worlds with emotion synthesis at the 8th International Conference on 3D Web Technology (Web3D), St. Malo, France*, 2003, vol. 35.
[67] B. Krenn and H. Pirker, “Defining the gesticon: Language and gesture coordination for interacting embodied agents,” in *Proc. of the AISB-2004 Symposium on Language, Speech and Gesture for Expressive Characters*, 2004, pp. 107–115.
[68] Y. I. Nakano, M. Okamoto, and T. Nishida, “Enriching Agent Animations with Gestures and Highlighting Effects,” in *Intelligent Media Technology for Communicative Intelligence*, L. Bolc, Z. Michalewicz, and T. Nishida, Eds. Springer Berlin Heidelberg, 2005, pp. 91–98.
[69] H.-H. Kim, H.-E. Lee, Y.-H. Kim, K.-H. Park, and Z. Z. Bien, “Automatic Generation of Conversational Robot Gestures for Human-friendly Steward Robot,” in *The 16th IEEE International Symposium on Robot and Human interactive Communication, 2007. RO-MAN 2007*, 2007, pp. 1155–1160.
[70] S. Kopp, B. Krenn, S. Marsella, A. N. Marshall, C. Pelachaud, H. Pirker, K. R. Thórisson, and H. Vilhjálmsson, “Towards a common framework for multimodal generation: The behavior markup language,” in *Intelligent virtual agents*, 2006, pp. 205–217.
[71] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. Marshall, and C. Pelachaud, “The behavior markup language: Recent developments and challenges,” in *Intelligent virtual agents*, 2007, pp. 99–111.
[72] D. Heylen, S. Kopp, S. C. Marsella, C. Pelachaud, and H. Vilhjálmsson, “The Next Step towards a Function Markup Language,” in *Proceedings of the 8th international conference on Intelligent Virtual Agents*, Berlin, Heidelberg, 2008, pp. 270–280.
[73] *http://cadia.ru.is/projects/bmlr/*. .
[74] Q. A. Le and C. Pelachaud, “Generating Co-speech Gestures for the Humanoid Robot NAO through BML,” in *Gesture and Sign Language in Human-Computer Interaction and*

Embodied Communication, E. Efthimiou, G. Kouroupetroglou, and S.-E. Fotinea, Eds. Springer Berlin Heidelberg, 2012, pp. 228–237.

[75] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann, "Smartbody: Behavior realization for embodied conversational agents," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, 2008, pp. 151–158.

[76] C.-M. Huang and B. Mutlu, "Learning-based Modeling of Multimodal Behaviors for Humanlike Robots," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, New York, NY, USA, 2014, pp. 57–64.

[77] H. Cuayáhuitl, M. van Otterlo, N. Dethlefs, and L. Frommberger, "Machine Learning for Interactive Systems and Robots: A Brief Introduction," in *Proceedings of the 2Nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication*, New York, NY, USA, 2013, pp. 19–28.

[78] H. Admoni and B. Scassellati, "Data-Driven Model of Nonverbal Behavior for Socially Assistive Human-Robot Interactions," in *Proceedings of the 16th International Conference on Multimodal Interaction*, New York, NY, USA, 2014, pp. 196–199.

[79] L.-P. Morency, I. de Kok, and J. Gratch, "A probabilistic multimodal approach for predicting listener backchannels," *Auton. Agents Multi-Agent Syst.*, vol. 20, no. 1, pp. 70–84, Jan. 2010.

[80] I. de Kok, D. Heylen, and L.-P. Morency, "Speaker-adaptive Multimodal Prediction Model for Listener Responses," in *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, New York, NY, USA, 2013, pp. 51–58.

[81] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds, "The NIST 2014 speaker recognition i-vector machine learning challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[82] M. K. Neff, "Gesture modeling and animation based on a probabilistic re-creation of speaker style.," *ACM Trans Graph*, vol. 27, 2008.

[83] J. Lee and S. Marsella, "Modeling Speaker Behavior: A Comparison of Two Approaches," in *Intelligent Virtual Agents*, Y. Nakano, M. Neff, A. Paiva, and M. Walker, Eds. Springer Berlin Heidelberg, 2012, pp. 161–174.

[84] L. Morency, A. Quattoni, and T. Darrell, "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 2007, pp. 1–8.

[85] S. P. Lee, J. B. Badler, and N. I. Badler, "Eyes Alive," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2002, pp. 637–644.

[86] G. Bailly, S. Raidt, and F. Elisei, "Gaze, conversational agents and face-to-face communication," *Speech Commun.*, vol. 52, no. 6, pp. 598–612, Jun. 2010.

[87] Y. Mohammad, T. Nishida, and S. Okada, "Unsupervised simultaneous learning of gestures, actions and their associations for Human-Robot Interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, 2009, pp. 2537–2544.

[88] Y. Mohammad and T. Nishida, "Learning interaction protocols using Augmented Baysian Networks applied to guided navigation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4119–4126.

[89] J. F. Ferreira, M. Castelo-Branco, and J. Dias, "A hierarchical Bayesian framework for multimodal active perception," *Adapt. Behav.*, vol. 20, no. 3, pp. 172–190, Jun. 2012.

[90] S. Levine, P. Krähenbühl, S. Thrun, and V. Koltun, "Gesture Controllers," in *ACM SIGGRAPH 2010 Papers*, New York, NY, USA, 2010, pp. 124:1–124:11.

[91] G. Zoric, R. Forchheimer, and I. S. Pandzic, “On creating multimodal virtual humans—real time speech driven facial gesturing,” *Multimed. Tools Appl.*, vol. 54, no. 1, pp. 165–179, Aug. 2011.
[92] S. K. Kirsten Bergmann, “Modeling the Production of Coverbal Iconic Gestures by Learning Bayesian Decision Networks.,” *Appl. Artif. Intell.*, vol. 24, pp. 530–551, 2010.
[93] A. Vinciarelli, A. Esposito, E. André, F. Bonin, M. Chetouani, J. F. Cohn, M. Cristani, F. Fuhrmann, E. Gilmartin, Z. Hammal, D. Heylen, R. Kaiser, M. Koutsombogera, A. Potamianos, S. Renals, G. Riccardi, and A. A. Salah, “Open Challenges in Modelling, Analysis and Synthesis of Human Behaviour in Human–Human and Human–Machine Interactions,” *Cogn. Comput.*, pp. 1–17, Apr. 2015.
[94] H. C. Triandis, *Culture and Social Behavior*. McGraw-Hill, 1994.
[95] C. Chao and A. L. Thomaz, “Controlling Social Dynamics with a Parametrized Model of Floor Regulation,” *J. Hum.-Robot Interact.*, vol. 2, no. 1, pp. 4–29, Mar. 2013.
[96] J. Bates, “The Role of Emotion in Believable Agents,” *Commun ACM*, vol. 37, no. 7, pp. 122–125, Jul. 1994.
[97] M. Mori, K. F. MacDorman, and N. Kageki, “The Uncanny Valley [From the Field],” *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 98–100, Jun. 2012.
[98] D. Ferber, “The man who mistook his girlfriend for a robot,” *Pop. Sci.*, vol. 236, p. 60, 2003.
[99] H. Ishiguro, “Android Science,” in *Robotics Research*, D. S. Thrun, D. R. Brooks, and D. H. Durrant-Whyte, Eds. Springer Berlin Heidelberg, 2007, pp. 118–127.
[100] M. E. Foster, “Comparing Rule-based and Data-driven Selection of Facial Displays,” in *Proceedings of the Workshop on Embodied Language Processing*, Stroudsburg, PA, USA, 2007, pp. 1–8.
[101] D. Bohus and E. Horvitz, “Open-world dialog: Challenges, directions, and prototype,” in *Proceedings of IJCAI’2009 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2009.
[102] M. Tomasello, *Origins of Human Communication*. Cambridge, Mass.: A Bradford Book, 2010.
[103] B. M. Scassellati, “Foundations for a Theory of Mind for a Humanoid Robot,” Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.
[104] “wcpr14 - Workshop on Computational Personality Recognition.” [Online]. Available: https://sites.google.com/site/wcprst/home/wcpr14. [Accessed: 15-Jul-2015].
[105] S. Raidt, “Gaze and face-to-face communication between a human speaker and an animated conversational agent - Mutual attention and multimodal deixis,” PhD Thesis, Institut National Polytechnique de Grenoble - INPG, 2008.
[106] A. Kendon, “Some functions of gaze-direction in social interaction,” *Acta Psychol. (Amst.)*, vol. 26, pp. 22–63, 1967.
[107] M. Argyle and M. Cook, *Gaze and Mutual Gaze*. Cambridge University Press, 1976.
[108] C. Benoît, M. Grice, and V. Hazan, “The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences,” *Speech Commun.*, vol. 18, no. 4, pp. 381–392, Jun. 1996.
[109] C. E. Ford, “Contingency and Units in Interaction,” *Discourse Stud.*, vol. 6, no. 1, pp. 27–52, Jan. 2004.
[110] J. Lee, S. Marsella, D. Traum, J. Gratch, and B. Lance, “The Rickel Gaze Model: A Window on the Mind of a Virtual Human,” in *Proceedings of the 7th International Conference on Intelligent Virtual Agents*, Berlin, Heidelberg, 2007, pp. 296–303.
[111] J. Rickel and W. L. Johnson, “Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control,” *Appl. Artif. Intell.*, vol. 13, pp. 343–382, 1998.

[112] S. Marsella, J. Gratch, and J. Rickel, “Expressive Behaviors for Virtual Worlds,” in *Life-Like Characters*, H. Prendinger and M. Ishizuka, Eds. Springer Berlin Heidelberg, 2004, pp. 317–360.
[113] R. A. Bolt, “‘Put-That-There’: Voice and Gesture at the Graphics Interface,” in *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1980, pp. 262–270.
[114] H. Sloetjes and P. Wittenburg, “Annotation by Category: ELAN and ISO DCR.,” in *LREC*, 2010.
[115] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, 3 edition. Burlington, MA: Morgan Kaufmann, 2011.
[116] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[117] S. K. Murthy, “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,” *Data Min Knowl Discov*, vol. 2, no. 4, pp. 345–389, Dec. 1998.
[118] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[119] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, 2004.
[120] D. Wettschereck, D. W. Aha, and T. Mohri, “A Review and Empirical Evaluation of Feature Weighting Methods for aClass of Lazy Learning Algorithms,” *Artif Intell Rev*, vol. 11, no. 1–5, pp. 273–314, Feb. 1997.
[121] P. Domingos and M. Pazzani, “On the Optimality of the Simple Bayesian Classifier under Zero-One Loss,” *Mach. Learn.*, vol. 29, no. 2–3, pp. 103–130, Nov. 1997.
[122] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, 2007, pp. 3–24.
[123] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
[124] R. Polikar, “Ensemble learning,” *Scholarpedia*, vol. 4, no. 1, p. 2776, 2009.
[125] P. B. Eth, P. Bühlmann, and B. Yu, “Analyzing Bagging,” *Ann. Stat.*, vol. 30, pp. 927–961, 2001.
[126] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[127] D. H. Wolpert, “Stacked Generalization,” *Neural Netw.*, vol. 5, pp. 241–259, 1992.
[128] M. Graczyk, T. Lasota, B. Trawiński, and K. Trawiński, “Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal,” in *Intelligent Information and Database Systems*, N. T. Nguyen, M. T. Le, and J. Świątek, Eds. Springer Berlin Heidelberg, 2010, pp. 340–350.
[129] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
[130] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.
[131] J. Mercer, “Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations,” *Philos. Trans. R. Soc. Lond. Math. Phys. Eng. Sci.*, vol. 209, no. 441–458, pp. 415–446, Jan. 1909.

[132] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[133] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, Jun. 2011.
[134] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[135] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*, 1 edition. New York, N.Y.: Chapman and Hall/CRC, 1984.
[136] G. V. Kass, “An Exploratory Technique for Investigating Large Quantities of Categorical Data,” *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 29, no. 2, pp. 119–127, Jan. 1980.
[137] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in induction*. Academic Press, 1966.
[138] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[139] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *SIGKDD Explor Newsl*, vol. 11, no. 1, pp. 10–18, Nov. 2009.
[140] J. C. PLATT, “Fast training of support vector machines using sequential minimal optimization,” *Adv. Kernel Methods Support Vector Learn.*, pp. 185–208, 1999.
[141] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, New York, NY, USA, 1992, pp. 144–152.
[142] V. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions and Reversals,” *Sov. Phys. Dokl.*, vol. 10, no. 8, pp. 707–710, Feb. 1966.
[143] P. Spanger, M. Yasuhara, R. Iida, and T. Tokunaga, “Using extra linguistic information for generating demonstrative pronouns in a situated collaboration task,” *Proc. PreCogSci 2009 Prod. Referring Expr. Bridg. Gap Comput. Empir. Approaches Ref.*, 2009.
[144] C.-C. Chiu and S. Marsella, “Gesture Generation with Low-dimensional Embeddings,” in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, Richland, SC, 2014, pp. 781–788.
[145] D. C. Richardson, R. Dale, and K. Shockley, “Synchrony and swing in conversation: coordination, temporal dynamics, and communication,” in *Embodied Communication in Humans and Machines*, I. Wachsmuth, M. Lenzen, and G. Knoblich, Eds. Oxford University Press, 2008, pp. 75–94.
[146] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, 1989, pp. 257–286.
[147] Y. Bengio and P. Frasconi, “Input-output HMMs for sequence processing,” *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1231–1249, Sep. 1996.
[148] X. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition*. New York, NY, USA: Columbia University Press, 1990.
[149] J. Hu, M. K. Brown, and W. Turin, “HMM based online handwriting recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 1039–1045, Oct. 1996.
[150] H.-K. Lee and J. H. Kim, “An HMM-based threshold model approach for gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 961–973, Oct. 1999.
[151] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164–171, Feb. 1970.
[152] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” 1976.

[153] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967.
[154] D. Gatica-Perez, “Analyzing group interactions in conversations: a review,” in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, 2006, pp. 41–46.
[155] A. Seward, “Low-latency incremental speech transcription in the synface project.,” in *INTERSPEECH*, 2003.
[156] M. Ryynänen and A. Klapuri, “Automatic Bass Line Transcription from Streaming Polyphonic Audio,” in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007, pp. 1437–1440.
[157] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, “Map-matching for low-sampling-rate GPS trajectories,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA, 2009, pp. 352–361.
[158] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, “An Interactive-Voting Based Map Matching Algorithm,” 2010, pp. 43–52.
[159] J. Bloit and X. Rodet, “Short-time Viterbi for online HMM decoding: Evaluation on a real-time phone recognition task,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, 2008, pp. 2121–2124.
[160] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, “Online map-matching based on Hidden Markov model for real-time traffic sensing applications,” in *2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2012, pp. 776–781.
[161] R. Šrámek, B. Brejová, and T. Vinař, “On-line Viterbi Algorithm and Its Relationship to Random Walks,” *arXiv:0704.0062*, Mar. 2007.
[162] S. Yu, “Hidden semi-Markov models,” *Artif. Intell.*, 2010.
[163] J. D. Ferguson, “Variable Duration Models for Speech,” *Symp Appl. Hidden Markov Models Text Speech Inst. Def. Anal. Princet. NJ*, pp. 143–179, Oct. 1980.
[164] S. E. Levinson, “Continuously variable duration hidden Markov models for automatic speech recognition,” *Comput. Speech Lang.*, vol. 1, no. 1, pp. 29–45, Mar. 1986.
[165] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman, “A generalized hidden Markov model for the recognition of human genes in DNA,” *Proc. Int. Conf. Intell. Syst. Mol. Biol. ISMB Int. Conf. Intell. Syst. Mol. Biol.*, vol. 4, pp. 134–142, 1996.
[166] M. Russell, “A segmental HMM for speech pattern modelling,” in *, 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93*, 1993, vol. 2, pp. 499–502 vol.2.
[167] P. Ramesh and J. G. Wilpon, “Modeling state durations in hidden Markov models for automatic speech recognition,” in *, 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992. ICASSP-92*, 1992, vol. 1, pp. 381–384 vol.1.
[168] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Hidden Semi-Markov Model Based Speech Synthesis,” in *in Proc. of ICSLP, 2004*, 2004.
[169] P. Lanchantin and W. Pieczynski, “Unsupervised non stationary image segmentation using triplet Markov chains,” *Adv. Concepts Intell. Vis. Syst. ACVIS 04*, 2004.
[170] S. Hongeng and R. Nevatia, “Large-scale event detection using semi-hidden Markov models,” in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, 2003, pp. 1455–1462 vol.2.
[171] K. Squire, “HMM-based semantic learning for a mobile robot, Ph.D. dissertation,” Ph.D. dissertation, University of Illinois at Urbana-Champaign.

[172] S. Yu, “Multiple tracking based anomaly detection of mobile nodes,” in *2005 2nd International Conference on Mobile Technology, Applications and Systems*, 2005, p. 5 pp.–5.
[173] S. C. Schmidler, J. S. Liu, and D. L. Brutlag, “Bayesian segmentation of protein secondary structure,” *J. Comput. Biol. J. Comput. Mol. Cell Biol.*, vol. 7, no. 1–2, pp. 233–248, Apr. 2000.
[174] J. Bulla and I. Bulla, “Stylized facts of financial time series and hidden semi-Markov models,” *Comput. Stat. Data Anal.*, vol. 51, no. 4, pp. 2192–2209, Dec. 2006.
[175] C. Mitchell, M. Harper, L. Jamieson, and C. T. M, “On the Complexity of Explicit Duration HMMs,” in *IEEE Transactions on Speech and Audio Processing*, 1995, pp. 213–217.
[176] S. Yu and H. Kobayashi, “An efficient forward-backward algorithm for an explicit-duration hidden Markov model,” *IEEE Signal Process. Lett.*, vol. 10, no. 1, pp. 11–14, Jan. 2003.
[177] H. K. Shun-Zheng Yu, “Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden Markov model.,” *IEEE Trans. Signal Process.*, vol. 54, pp. 1947–1951, 2006.
[178] HTK, The Hidden Markov Model Toolkit, *http://htk.eng.cam.ac.uk/*. .
[179] M. Dunham and K. Murphy, *PMTK3: Probabilistic modeling toolkit for Matlab/Octave, http://code.google.com/p/pmtk3/*. .
[180] A. Mihoub, G. Bailly, and C. Wolf, “Social behavior modeling based on incremental discrete hidden Markov models,” in *Human Behavior Understanding*, Barcelona, Spain: Springer, 2013, pp. 172–183.
[181] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling, Second Edition*, 2 edition. Boca Raton, Fla.: Chapman and Hall/CRC, 2000.
[182] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
[183] P. Naïm, P.-H. Wuillemin, P. Leray, O. Pourret, and A. Becker, *Réseaux bayésiens*, 3e édition. Paris: Eyrolles, 2007.
[184] M. I. Jordan, Ed., *Learning in Graphical Models*, 1st edition. Cambridge, Mass.: A Bradford Book, 1998.
[185] A. P. Dawid, “Applications of a general propagation algorithm for probabilistic expert systems,” *Stat. Comput.*, vol. 2, no. 1, pp. 25–36, Mar. 1992.
[186] D. Bellot, “Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine,” phdthesis, Université Henri Poincaré - Nancy I, 2002.
[187] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1 edition. San Francisco, Calif: Morgan Kaufmann, 1988.
[188] K. P. Murphy, “Dynamic bayesian networks: representation, inference and learning,” University of California, Berkeley, 2002.
[189] K. Hallouli, L. Likforman-Sulem, and M. Sigelle, “Réseaux Bayésiens Dynamiques pour la reconnaissance des caractères imprimés dégradés,” in *19° Colloque sur le traitement du signal et des images, FRA, 2003*, 2003.
[190] J. Pearl, “Reverend Bayes on inference engines: a distributed hierarchical approach,” in *in Proceedings of the National Conference on Artificial Intelligence*, 1982, pp. 133–136.
[191] J. P. Jin H. Kim, “A Computational Model for Causal and Diagnostic Reasoning in Inference Systems.,” pp. 190–193, 1983.
[192] S. L. Lauritzen and D. J. Spiegelhalter, “Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems,” *J. R. Stat. Soc. Ser. B Methodol.*, vol. 50, no. 2, pp. 157–224, Jan. 1988.
[193] S. L. L. F. V. Jensen, “Bayesian updating in recursive graphical models by local computations,” *Comput. Stat. Q.*, vol. 4, no. 1, pp. 269–282, 1990.

[194] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2007.
[195] B. Frey and D. MacKay, "A Revolution: Belief Propagation in Graphs With Cycles," in *In Neural Information Processing Systems*, 1998, pp. 479–485.
[196] K. Gimpel, D. Rudoy, L. Laboratory, U. S. M. D. Agency, and U. S. A. Force, *Statistical Inference in Graphical Models*. Massachusetts Institute of Technology, Lincoln Laboratory, 2008.
[197] S. Baghdadi, C.-H. Demarty, G. Gravier, and P. Gros, "Apprentissage de structure dans les réseaux bayésiens pour la détection d'événements vidéo," presented at the Traitement et analyse de l'information : méthodes et applications, 2009.
[198] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search, Second Edition*, Second edition edition. Cambridge, Mass.: A Bradford Book, 2001.
[199] P. L. Olivier François, "Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens," presented at the Rencontres des Jeunes Chercheurs en IA, 2003.
[200] J. Pearl and T. S. Verma, *A Theory Of Inferred Causation*. 1991.
[201] G. Schwarz, "Estimating the Dimension of a Model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, Mar. 1978.
[202] P. Leray, "Réseaux bayésiens : Apprentissage et diagnostic de systemes complexes," thesis, Université de Rouen, 2006.
[203] C. L. C. Chow, "Approximating discrete probability distributions with dependence trees," *Inf. Theory IEEE Trans. On*, no. 3, pp. 462 – 467, 1968.
[204] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, Oct. 1992.
[205] D. M. Chickering, "Learning Equivalence Classes of Bayesian-network Structures," *J Mach Learn Res*, vol. 2, pp. 445–498, Mar. 2002.
[206] K. P. Murphy, "The Bayes Net Toolbox for MATLAB," *Comput. Sci. Stat.*, vol. 33, p. 2001, 2001.
[207] S. Liang, S. Fuhrman, and R. Somogyi, *Reveal, A General Reverse Engineering Algorithm For Inference Of Genetic Network Architectures*. 1998.
[208] D. McNeill, *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, 1992.
[209] R. G. Cowell, P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*. Springer New York, 2003.
[210] Y. Ding, M. Radenen, T. Artieres, and C. Pelachaud, "Speech-driven eyebrow motion synthesis with contextual Markovian models," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3756–3760.

Annexes

Annexe A : Un exemple d'une entrée "Gesticon"

```
<gesticonEntry key ="g001"
               identifier="Thinking"
               modality="arms">
  <verbatim>
    Thinking: adaptor:  Tina: adaptor:
    moves right hand to chin but in
    addition left hand moves to
    shoulder-hight +
    palm up
  </verbatim>
  <function type="adaptor"
            alignto="sentence"
            aligntype="par" start="-200"
            meaning="think"/>
  </function>
  <form>
    <position>
      <!-- starts with D(own) O(ut) -->
      <start left="DO" right="DO"/>
      <!-- ends with T(op) C(enter) -->
      <end   left="TO" right="TC"/>
    </position>
    <components>
      <stroke>
        <dur min="1000" default="1300"
             max="2000"/>
      </stroke>
      <hold>
        <dur min="500" default="1000"
             max="50000"/>
    </components>
  </form>
  <restrictions>
    <and>
      <constraint name="gender"
                  val="female"/>
      <constraint name="speaker"
                  val="tina"/>
      <constraint name="occ_emotion"
                  val="anger"/>
    </and>
  </restrictions>
  <playercode type="charactor"
     id="tina/char/motions/gs_thinking'
</gesticonEntry>
```

Figure 84: Un exemple d'une entrée "Gesticon" (figure reproduite de [67])

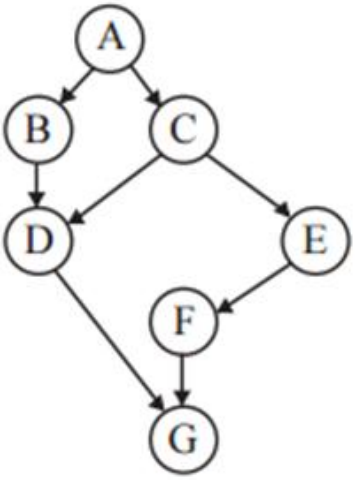
Annexe B : Arbre de jonction

L'algorithme de l'arbre de jonction se déroule sur deux phases : une phase de construction et une phase de propagation. Ces deux phases sont décrites en détails ci-dessous et un exemple [196] est montré dans la Figure 85.

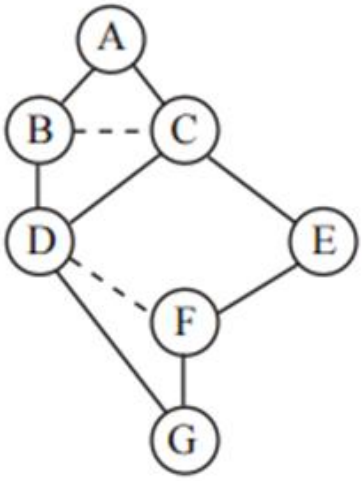
- La phase de construction: elle se déroule sur plusieurs étapes dont l'objectif est de transformer le graphe initial en un arbre de jonction. Les nœuds de cet arbre représente des regroupements (clusters) de nœuds du graphe initial. Cette transformation permet d'éliminer les boucles du graphe et de réduire le temps de calcul nécessaire à l'inférence. Elle s'effectue en trois étapes:

1. La moralisation du graphe (voir Figure 85): elle consiste à marier les parents de chaque variable en les reliant par un arc non-dirigé. Après la moralisation, on transforme la totalité du graphe en un graphe non-dirigé en enlevant les directions des arcs. L'idée de base est que la nouvelle distribution de probabilité satisfasse aux contraintes d'indépendances conditionnelles définies par le graphe initial. Cowell et al. [209] présentent une justification de l'équivalence de la distribution de probabilités jointe issue du graphe initial et de la distribution issue du graphe moral.
2. La triangulation du graphe et la détermination des cliques maximales qui construiront les nœuds du futur arbre de jonction (voir Figure 85). Un graphe est triangulé si tout cycle de longueur supérieure à 3 admet une corde (c'est-à-dire une arête reliant deux sommets non-adjacents du cycle). Une fois le graphe est triangulé on extrait les cliques maximales qui formeront les nœuds de l'arbre de jonction utilisé pour l'inférence. Une clique d'un graphe non orienté est, en théorie des graphes, un ensemble de sommets deux-à-deux adjacents. La clique maximale d'un graphe est la clique dont le cardinal est le plus grand c'est-à-dire qu'elle possède le plus grand nombre de sommets. On applique la triangulation car du graphe moralisé on ne peut pas systématiquement déduire un arbre de jonction. En effet, le graphe moralisé peut encore contenir des cycles et l'algorithme d'inférence ne fonctionne que lorsque le graphe résultant est un arbre. C'est pourquoi, après la moralisation, on effectue une phase de triangulation. Cowell et al. [209] ont démontré que pour un graphe G donné, il existe un arbre de jonction correspondant si et seulement si G est triangulé.
3. La création de l'arbre de jonction en utilisant les cliques maximales (voir Figure 85).

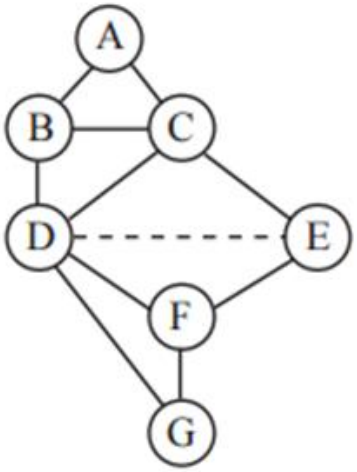
- La phase de propagation : c'est dans cette phase qu'on applique la méthode "Belief propagation". C'est la phase du calcul probabiliste où s'effectue la mise à jour de l'ensemble des distributions de probabilités du réseau en passant des messages entre les nœuds de l'arbre de jonction construit.



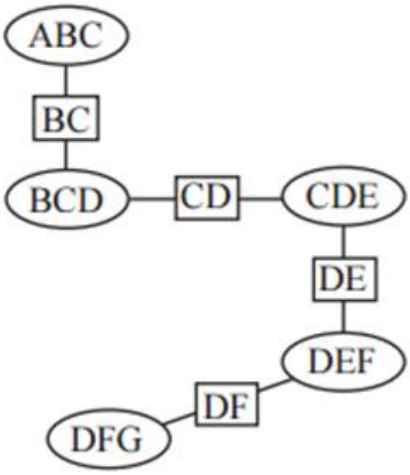
(a)



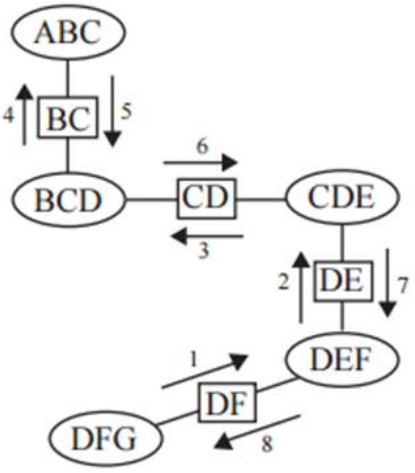
(b)



(c)



(d)



(e)

Figure 85: (a) Graphe initial (b) Phase de construction: moralisation (c) Phase de construction: triangulation (d) Phase de construction: construction de l'arbre de jonction en utilisant les cliques maximales du graphe triangulé (e) Phase de propagation ou on applique l'algorithme " Belief propagation " sur l'arbre construit (figures reproduites de [196]).