

Méthodes SVM et MLN pour la reconnaissance automatique d'activités humaines dans les habitats perceptifs: tests et perspectives

Pedro Chahuara, Anthony Fleury, Michel Vacher, François Portet

► **To cite this version:**

Pedro Chahuara, Anthony Fleury, Michel Vacher, François Portet. Méthodes SVM et MLN pour la reconnaissance automatique d'activités humaines dans les habitats perceptifs: tests et perspectives. RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle), Jan 2012, Lyon, France. pp.978-2-9539515-2-3, 2012. <hal-00656557>

HAL Id: hal-00656557

<https://hal.archives-ouvertes.fr/hal-00656557>

Submitted on 17 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méthodes SVM et MLN pour la reconnaissance automatique d'activités humaines dans les habitats perceptifs : tests et perspectives

P. Chahuara¹

A. Fleury²

M. Vacher¹

F. Portet¹

¹ Laboratoire d'Informatique de Grenoble, UJF/Grenoble-INP/CNRS 5217, 38041 Grenoble

² Univ. Lille Nord de France, F-59000 Lille, France. EMDouai, IA, F-59508 Douai Cedex, France.

Résumé

Cet article compare deux approches pour la reconnaissance des Activités de la Vie Quotidiennes (AVQ) dans un habitat intelligent : - une approche SVM utilisée pour son excellente capacité de classification, - ainsi qu'une approche à base de réseaux logiques de Markov pour leur capacité à allier une excellente expressivité et un raisonnement dans l'incertain. Ces méthodes ont été testées sur des données réelles acquises lors d'expérimentations impliquant 21 personnes effectuant des AVQ. Les résultats montrent la difficulté de la tâche et permettent de dégager des pistes d'amélioration.

Mots Clef

reconnaissance d'activité, réseaux logiques de Markov, SVM, habitat intelligent, assistance à domicile.

Abstract

This paper reports the comparison of two methods for the recognition of Activities of Daily Living (ADL) in a smart home. A SVM based approach was developed due to its excellent classification capacity as well as a Markov logic network one because of its high expressibility and its capacity to reason with uncertainty. These methods were evaluated on real data acquired during an experiment that involved 21 participants performing ADLs. Results show the difficulty of the task but make it possible to draw ways of improving the method.

Keywords

Activity Recognition, Markov Logic Network, Support Vector Machine, Smart Home, Ambient Assisted Living.

1 Introduction

La reconnaissance d'activité humaine à partir de capteurs est un domaine scientifique en pleine effervescence avec de nombreuses applications dans le domaine de la sécurité et de la vidéo-surveillance de lieux publics [7]. Cette recherche s'est récemment étendue à l'assistance à domicile [16, 15, 19], à l'assistance et à la surveillance dans les unités de soins spécialisés, ainsi qu'à la surveillance et à l'évaluation de procédés dans l'industrie. La reconnaissance d'activité humaine est aussi actuellement très étudiée

pour la fourniture de services en contexte (*context-aware*) dans les smartphones et autre objets intelligents.

Dans cet article, nous nous intéressons à la reconnaissance d'activité dans le cadre de l'assistance à domicile et de la domotique. L'environnement considéré est donc un espace privé dans lequel plusieurs capteurs, activateurs et matériels domotique coexistent. Ce type d'environnement impose des contraintes au niveau des capteurs ainsi qu'au niveau des techniques employées pour la reconnaissance. La plupart des progrès apportés dans le domaine de la reconnaissance d'activité proviennent de la vision par ordinateur [1]. Cependant, non seulement l'installation de caméras chez l'habitant soulève des questions éthiques [16], mais elle entraîne aussi un rejet de la part des utilisateurs [13] comme nous l'avons mis en évidence. La solution développée dans le cadre du projet SWEET-HOME¹ consiste à compléter les capteurs conventionnels de la domotique (détecteurs de présence infra-rouge, interrupteurs, etc.) par des microphones pour permettre à l'utilisateur de maîtriser son environnement grâce à une reconnaissance vocale et un contrôleur intelligent. Le contrôleur intelligent doit prendre les décisions en fonction du contexte. Par exemple, l'allumage d'une lampe se fera de manière différente selon que l'utilisateur interrompt son sommeil ou qu'il est en train de manger. La reconnaissance d'activité constitue donc un élément essentiel du système.

La reconnaissance automatique de l'activité humaine (manger, discuter, regarder la télévision, etc.) peut être définie comme l'identification d'une séquence d'actions atomiques (prendre un ustensile de cuisine, s'allonger, etc.). Ceci implique d'abstraire les signaux bruts en symboles (propositions) étiquetés temporellement (p.ex. : claquement de porte à 11h32), des signatures de situations particulières d'événements atomiques sont détectées à travers un processus d'abstraction hiérarchique. Par exemple, les mouvements détectés dans la chambre à coucher peuvent faire partie de l'activité « se lever » qui elle-même peut faire partie d'un plan de journée (p.ex. : matinée du dimanche). La reconnaissance automatique d'activité est l'un des secteurs de recherche les plus actifs et les plus ambi-

¹. Ce travail a été réalisé dans le cadre du projet Sweet-Home qui bénéficie d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-09-VERS-011

teux en raison de la quantité importante de bruit dans les données et de la difficulté de modélisation des situations ; pour une même personne, une activité peut se dérouler de nombreuses façons.

Un nombre très important de méthodes ont été proposées pour la reconnaissance d'activité à partir de capteurs non visuels. La reconnaissance à base de règles logiques expertes, qui est très répandue [15, 19], prend difficilement en compte l'incertitude, souvent à travers une pondération experte alimentant un système d'inférence floue ou un raisonnement de Dempster-Shafer. Une autre approche consiste à modéliser les activités par modèle de Markov caché (*Hidden Markov Model* - HMM) [12, 5]. Plusieurs variantes permettent plus de souplesse dans l'ordre des séquences (p.e : les HMM hiérarchiques [12]) ou la modélisation de la durée des activités [5]. Cependant, malgré une expressivité améliorée, ces modèles nécessitent beaucoup de données d'apprentissage, données qui sont très rares dans le domaine de l'assistance à domicile. De plus, il reste difficile d'intégrer des connaissances *a priori* dans ces modèles probabilistes. Une nouvelle méthode utilisant les réseaux logiques de Markov (MLN) a été appliquée à ce domaine [17]; elle combine la puissance d'expressivité de la logique du premier ordre et la capacité à traiter l'information incertaine des méthodes probabilistes. Ceci présente de multiples avantages : le modèle est défini dans un langage logique formel qui le rend facilement vérifiable et réutilisable [2], l'incertitude est prise en compte de manière systématique, et enfin l'apprentissage du modèle permet l'inclusion aisée de connaissances *a priori*.

Dans cet article, nous présentons les méthodes à base de MLN et de SVM que nous avons employées pour effectuer une reconnaissance d'activité de la vie quotidienne dans un environnement perceptif. Nous présentons également les résultats préliminaires obtenus et discutons des limites des approches avant de dégager des pistes d'amélioration pour les modèles.

2 Méthode

La méthode employée pour reconnaître les activités au fur et à mesure de l'arrivée des données (c.-à-d. en-ligne) suit le schéma de la Figure 1. Les données brutes extraites des capteurs sont composées de valeurs symboliques instantanées (p.ex. : détecteur infra-rouge) ou persistantes (p.ex. : interrupteur), de série temporelles (p.ex. : température) et de signaux continus (p.ex. : microphone). Un prétraitement permet d'extraire des informations de haut niveau comme par exemple la parole et la localisation de l'habitant. Toutes les données sont abstraites en vecteurs de description, chaque vecteur résumant une fenêtre temporelle. Le traitement de ces vecteurs par les classificateurs permet de décider du type d'activité en cours le plus probable. Les modèles de classification sont appris de manière supervisée. Cette section détaille les prétraitements réalisés, les attributs considérés ainsi que les modèles de classification et leur apprentissage.

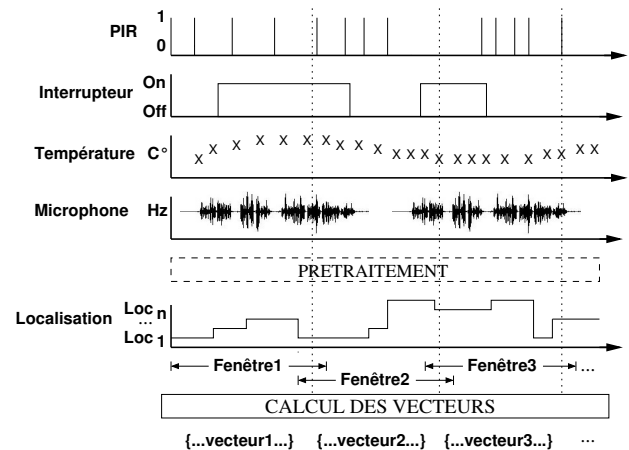


FIGURE 1 – Fenêtrage temporel utilisé pour l'obtention des attributs à partir des données fournies par les capteurs.

2.1 Activités pour l'application considérée

Le projet SWEET-HOME a pour but principal de contrôler les éléments domotiques notamment à travers un système réactif à la voix. De ce fait, des commandes telles que, l'allumage de la lumière ou la génération de messages d'alertes sont générées suite à un ordre explicite (ou implicite dans le cas de détection de situations de détresse). Cependant, la pertinence de ces commandes dépend de la situation courante de l'utilisateur, autrement dit du contexte dans lequel l'ordre a été généré. Nous définissons le contexte ainsi :

Définition 1 (Contexte) *Le contexte est composé d'informations dynamiques qui permettent de caractériser la situation courante de l'utilisateur, la situation courante de l'environnement ainsi que l'état de la communication entre le système et l'utilisateur.*

La définition du contexte est extrêmement dépendante de l'application mais typiquement la localisation de l'utilisateur, la date et l'heure de la journée sont les composantes les plus classiques des systèmes ubiquitaires *context-aware*. Dans notre cas, nous souhaitons également connaître les activités de la personne afin de générer des commandes plus fines. Par exemple, si la personne dit "allume la lumière", la commande peut être différente selon que la personne se réveille ou que la personne s'habille (p.ex., intensité de la lumière). Nous définissons une activité ainsi :

Définition 2 (Activité) *Une activité est une tâche de la vie quotidienne, telle que dormir, s'habiller, se nourrir, communiquer, que la personne effectue sur un intervalle de temps donné. Ces activités sont un sous-ensemble de la catégorisation de Katz [9].*

Les activités considérées ont donc une granularité complètement différente de celle d'autres études où, par

exemple, l'usage d'un tournevis doit être détecté. Dans notre cas, les informations disponibles sont très pauvres, indirectes (pas de capteurs portés ou fixés aux objets) et asynchrones (pas de mesure en continue de l'environnement). Cependant, chaque activité est composée d'un ensemble d'événements qui laisse une ou des signatures caractéristiques de l'activité en cours qui, couplées au contexte, peuvent permettre d'inférer l'activité courante. Notre hypothèse est que ces signatures peuvent être représentées par des statistiques sur les informations fournies par les capteurs et le contexte. Étant donné le caractère dynamique des activités, ces statistiques doivent être calculées sur des fenêtres temporelles inférieures à la taille minimale des activités visées.

2.2 Prétraitements

Les données brutes provenant de l'habitat intelligent contiennent de l'information qu'il est nécessaire d'extraire en vue de la reconnaissance d'activité. Deux types de traitements sont considérés ici : la localisation de la personne — qui est une information primordiale pour la reconnaissance d'activité — et la détection de parole qui permet de détecter les activités de communication.

Détection de parole. Dans notre approche, les événements audio sont détectés, en temps réel, par un algorithme à seuil adaptatif basé sur la transformée en ondelettes [18]. Les événements sont ensuite classés en parole ou son de la vie courante (p.ex. : chute d'objet) par un classificateur utilisant un modèle de mélange gaussien (24 gaussiennes) appris par algorithme *Expectation Maximisation* (EM). Les microphones étant omnidirectionnels, un son peut être enregistré en même temps par plusieurs microphones placés dans plusieurs pièces ; ces observations simultanées sont repérées. Pour chaque observation sonore, le rapport signal sur bruit (RSB) est calculé.

Localisation. Dans les habitats intelligents la localisation des personnes est typiquement faite en utilisant des détecteurs de mouvement infrarouge. Afin d'améliorer la localisation, la méthode que nous avons mise en place fusionne l'information provenant de plusieurs sources de données (ici, contacts de porte, microphones et détecteurs infrarouge). Le modèle de fusion est constitué d'un réseau dynamique à deux niveaux [4] dont les nœuds représentent des concepts (ici les différentes localisations) et les arcs valués représentent la force de la relation entre concepts.

2.3 Calcul des vecteurs d'attributs

Les traces provoquées par une activité humaine et générées par les capteurs sont difficilement généralisables étant donné la très grande variété de réalisations d'une même tâche inter- et intra-personne. Nous considérons donc des fenêtres temporelles à l'intérieur desquelles les données brutes inférées sont résumées. De cette manière, l'ordre des événements est perdu, mais cette abstraction permet justement d'éliminer des variations pouvant être assimilées à du bruit pour une plus grande généralisation.

Afin d'obtenir la meilleure résolution possible plusieurs tailles de fenêtres temporelles (1, 2, 3 et 5 minutes) ont été considérées avec différents recouvrement (de 0% à 50% de la taille de la fenêtre). Par ailleurs, les attributs ont été filtrés afin de ne retenir que ceux qui avaient une corrélation suffisante avec les classes à détecter.

2.4 Les Séparateurs à Vaste Marge (SVM)

Les séparateurs à vaste marge [3] consistent en la construction d'un hyperplan de séparation $\vec{w}^T \vec{x} + w_0 = 0$ qui sera ensuite utilisé par une fonction $f()$ qui permet de savoir à laquelle des 2 classes appartient un vecteur \vec{y} :

$$f(\vec{w}^T \vec{y} + w_0) = \text{sign}(\vec{w}^T \vec{y} + w_0) \quad (1)$$

Étant donné un ensemble d'exemples $\{(\vec{x}_1, e_1), \dots, (\vec{x}_N, e_N)\}$ constituant le jeu d'apprentissage, avec \vec{x}_i les vecteurs de caractéristiques de dimension D , et $e_i \in \{-1, +1\}$ les classes, l'hyperplan choisi doit maximiser la distance entre les points les plus proches de chaque classe tout en restant un séparateur.

Cela revient à minimiser $\frac{1}{2} \|\vec{w}\|^2$ sous les contraintes $e_i(\vec{w}^T \vec{x}_i + w_0) \geq 1$, pour $i \in 1 \dots N$. Ceci est classiquement résolu par la méthode des multiplicateurs de Lagrange, où le lagrangien est donné par :

$$L(\vec{w}, w_0, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{k=1}^N \alpha_k [e_k(\vec{w}^T \vec{x}_k + w_0) - 1] \quad (2)$$

où les coefficients α_k sont les multiplicateurs de Lagrange. Le lagrangien doit être minimisé par rapport à \vec{w} et w_0 , et maximisé par rapport aux coefficients α_k . La résolution du problème montre que l'hyperplan optimal ne dépend que des n_s vecteurs supports \vec{x}_k du problème :

$$\vec{w} = \sum_{k=1}^{n_s} \alpha_k e_k \vec{x}_k \quad (3)$$

En notant $\langle \cdot, \cdot \rangle$ le produit scalaire de deux vecteurs, le système d'équations 1 devient :

$$f(\vec{y}) = \text{sign}(\vec{w}^T \vec{y} + w_0) = \text{sign}\left(\sum_{k=1}^{n_s} \alpha_k \langle \vec{x}_k, \vec{y} \rangle + w_0\right) \quad (4)$$

Ce cadre décrit la méthode de résolution des SVM dans un cas linéairement séparable. Cependant, le cas non-linéairement séparable peut être traité en projetant l'ensemble des points à l'aide d'une fonction noyau K . Dans ce nouvel espace, les données seront alors linéairement séparables. Dans notre cas, un noyau RBF est utilisé pour projeter l'ensemble des points du problème de départ dans ce nouvel espace :

$$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right)$$

car il permet de meilleurs résultats qu'un noyau polynomial [6]. Nous utiliserons une méthode « un-contre-tous » pour gérer le cas multiclasse en utilisant un ensemble de classificateurs permettant de séparer chaque paire de classes.

À la différence d'autres méthodes et notamment de la méthode présentée par la suite, les SVM sont des classificateurs purs, qui ne vont pas tenter d'inférer des règles mais juste de construire un ensemble de séparations optimales entre les différentes classes et ceci uniquement à partir d'exemples fournis. De plus, le classificateur construit sera dépendant de l'appartenance (du fait de cette construction), il sera difficile d'exporter les exemples acquis d'une configuration vers une autre. Cependant, les résultats obtenus lors d'une précédente expérimentation ont montré l'intérêt d'une telle approche [6].

2.5 Les Réseaux Logiques de Markov (MLN)

Un réseau logique de Markov, *Markov Logic Network (MLN)* [14] est un modèle statistique qui intègre la logique de premier ordre et les réseaux de Markov. Un MLN est composé d'un ensemble de formules logiques pondérées par des poids $w \in \mathbf{R}$ permettant d'exprimer leur degré de vérité. Cette approche relaxe la contrainte de la logique de premier ordre qui considère qu'une formule est exclusivement vraie ou fausse. Dans un MLN, quand une règle est violée, elle devient moins probable mais demeure quand même possible. Succinctement, considérons F un ensemble de formules logiques du premier ordre, $w_i \in \mathbf{R}$ le poids de la formule $f_i \in F$, C un ensemble de constantes, X un ensemble de variables aléatoires tel qu'une assignation $x_i \in C$ de $X_i \in X$ est une instance atomique (un prédicat avec toutes ses variables instanciées). Lors du processus d'inférence, le réseau logique de Markov $M_{L,C}$ construit un réseau de Markov (dans lequel tous les prédicats sont instanciés) pour obtenir la distribution $P(X = x)$. Dans la représentation graphique du réseau de Markov résultant, chaque sommet est un prédicat instancié, et il existe un arc entre deux sommets si leurs prédicats se trouvent dans une même formule. La probabilité d'une assignation spécifique est donnée par la formule :

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{f_i \in F} w_i n_i(x)\right) \quad (5)$$

où $Z = \sum_{x' \in \chi} \exp\left(\sum_{f_i \in F} w_i n_i(x')\right)$ est un facteur de normalisation, χ l'ensemble des états possibles, et $n(x)$ est le nombre de formules qui sont satisfaites par l'assignation de valeur à des variables dans X . L'inférence exacte dans un MLN complexe est un problème insoluble, donc des méthodes d'approximation de type MCMC sont appliquées. Dans la littérature, la méthode la plus utilisée est l'algorithme MC-SAT qui se base sur la résolution de problèmes de satisfaisabilité booléens. L'apprentissage d'un MLN consiste en 2 parties indépendantes : l'apprentissage des règles logiques, c.-à-d la structure du réseau ; et celui des poids pour les règles. Kok et Domingos [10] ont proposé un algorithme d'apprentissage de structure qui génère des règles candidates et mesure leur pertinence par une estimation du maximum de log-vraisemblance. Cependant, ce score n'est pas une mesure de pouvoir de discrimi-

nation et par conséquent il n'est pas adapté à un modèle discriminant tel que la reconnaissance d'activité. Dans cet article, nous présentons trois approches d'apprentissage d'un MLN pour la reconnaissance d'activités :

1) Le **MLN naïf** avec les règles $attribut_i(X, V_i) \rightarrow classe(X, V_c)$ où les variables X , V_i et V_c représentent respectivement l'instance à classifier, la valeur du i^e attribut, et la valeur pour la classe cible. Il est nécessaire de générer une règle pour chaque combinaison de variables V_i et V_c . Il faut noter que le graphe résultant de ce MLN correspond à une structure de modèle de Markov naïf, ou les prédicats d'attributs sont connectés uniquement avec le prédicat cible ;

2) Le **MLN+PLI1** où l'apprentissage de la structure s'effectue à travers la programmation logique inductive (PLI). L'algorithme MDIE proposé par Muggleton [11], implémenté dans le logiciel PROGOL, génère une hypothèse sous forme de programme logique pour expliquer des prédicats cibles à partir des connaissances introduites sous forme de règles et une liste d'exemples positifs et négatifs. L'application de la PLI pour apprendre la structure d'un MLN a été étudiée par Huynh [8] sur des données artificielles.

3) Le **MLN+PLI2** qui implique une variation dans l'algorithme d'apprentissage MDIE. Dans cette variation, chaque exemple positif est utilisé indépendamment des autres pour générer des règles. Ceci conduit à un plus grand nombre des règles apprises dont les pertinences peuvent être pondérées lors de la phase d'apprentissage des poids.

L'apprentissage des poids est un problème d'optimisation qui se traite par des algorithmes de gradient pour maximiser la fonction de vraisemblance conditionnelle. La méthode appliquée est *Scaled Conjugate Gradient*. Comme la structure d'un MLN est basée sur la logique de premier ordre, ce modèle offre une meilleure expressivité par rapport aux méthodes statistiques traditionnelles. De plus, il permet d'ajouter facilement des connaissances *a priori* sous forme de règles logiques.

3 Expérimentations et Résultats

3.1 Présentation de la plateforme Domus

La plateforme d'expérimentation Domus a été conçue et mise en place par l'équipe Multicom du Laboratoire d'Informatique de Grenoble [13]. Cet appartement fonctionnel de 34 m² est équipé de capteurs et d'actionneurs afin d'agir sur l'environnement (éclairage, volets, systèmes de sécurité, chauffage, ventilation, contrôle audio-vidéo ...). Plus de 150 capteurs, actionneurs et sources d'informations sont gérés dans l'appartement. 7 microphones sans fil placés dans le faux plafond équipent également l'habitat. La figure 2 décrit l'emplacement des différents capteurs.

Les différents types de capteurs considérés dans cet article sont les suivants :

- des microphones intégrés au plafond. Ces microphones peuvent être utilisés pour détecter des sons anormaux,

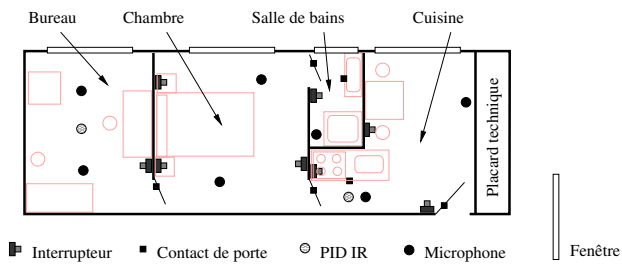


FIGURE 2 – Plan de l'appartement DOMUS et disposition des capteurs.

- retranscrire la parole, ou plus généralement classer les sons et les paroles qui sont entendus dans l'appartement ;
- un grand nombre de capteurs sur bus KNX (norme ISO/IEC 14543) permettant d'identifier l'utilisation d'une porte (de communication entre des pièces ou une porte d'une commodité - réfrigérateurs, placards, etc.), d'une prise électrique, d'un interrupteur pour la lumière, d'un volet, etc.
 - des capteurs « environnementaux », avec notamment des capteurs de luminosité, des capteurs de température et d'hygrométrie, ou encore un capteur de CO₂ ;
 - un compteur électrique et un compteur d'eau ;
 - un ensemble de caméras de surveillance qui servent uniquement à vérifier les expérimentations et à annoter celles-ci.

Le tableau 1 montre les 31 attributs obtenus à partir des informations de capteurs.

Attribut	Nombre	Information
PourcentageLoc _x	4	Pourcentage de temps dans la fenêtre temporelle ayant x comme localisation
PredominantLoc	1	Localisation avec le plus grand pourcentage
DerniereLoc	1	Dernière localisation avant le début de la fenêtre temporelle
TempsDansLoc	1	Temps en seconds dans la localisation prédominant
ActDesactFenetre _y	2	Nombre d'activations/désactivations du contact de la fenêtre y
ActDesactPorte _w	3	Nombre d'activations/désactivations du contact de la porte w
ActDesactLumiere _z	6	Nombre d'activations/désactivations de l'interrupteur de la lumière z
AmbientSensor	11	Variation de la valeur entre les fenêtres de : CO ₂ , température, Humidité, luminosité, débit d'eau et électricité.
Power.LastUse	1	Identifiant de la dernière utilisation d'une prise électrique ou d'un interrupteur.
Power.LastLastUse	1	Identifiant de l'avant-dernière utilisation d'une prise électrique ou d'un interrupteur.
Classe	1	Repos, préparer et prendre un repas, rangement, hygiène, s'habiller/se déshabiller, se détendre

TABLE 1 – Attributs utilisés pour la classification

Le fait que cette plateforme soit réalisée comme un appartement classique permet d'organiser des expérimentations dans lesquelles les personnes vont réaliser leurs activités de la vie quotidienne de la même manière qu'à leur domicile.

3.2 Données d'expérimentation et activités

Le but de l'expérimentation qui a été menée était d'obtenir des résultats dans l'analyse des activités d'une personne dans des conditions les plus proches possibles de la réalité. Pour ce faire, 21 personnes ont été recrutées et ont eu à charge de réaliser, le plus naturellement possible, des scénarios prédéfinis qui comportaient différentes activités (habillage, hygiène, petit-déjeuner, vaisselle, sortie pour les courses, ménage, écoute de la radio, ...) dont deux conversations téléphoniques. La moyenne d'âge des 21 participants était de 38.5 ans et s'échelonnait entre 22 et 63 ans. Chaque session d'enregistrement a duré environ 2 heures pour une durée totale de plus de 26 heures et 7 minutes. Ces activités ont été annotées avec le logiciel ADVENE. À partir des enregistrements, les données correspondant aux six activités énoncées précédemment et appartenant aux scénarios ont été extraites automatiquement, et ce pour différentes tailles de fenêtres temporelles avec différents recouvrements. L'ensemble des classes à classifier est composé des activités plus la classe rejet qui contient principalement les temps entre chacune des activités du scénario. Il y a un déséquilibre entre les cardinalités de chacune des classes. Par exemple, la répartition des échantillons pour un fenêtrage d'une minute avec un recouvrement de 25% est respectivement (cf. classes du tableau 1) : 210, 160, 27, 232, 42, 88, 151. Ceci s'explique par la différence entre les durées moyennes de chacune des activités (p.ex., manger et se laver les mains).

3.3 Paramétrisation

Pour ces données de localisation, l'apprentissage de la structure du réseau s'est fait sur 5 des 21 sujets considérés. Ainsi, ces sujets sont sortis de l'étude de classification des Activités de la Vie Quotidienne étant donnée que leurs données de localisation sont supposées connues (ayant servi de référence).

Avant classification les données sont normalisées. Le noyau utilisé est un noyau RBF pour lequel l'hyperparamètre ainsi que les paramètres de réglage des SVM sont optimisés au préalable sur un ensemble de test [6].

3.4 Résultats

Le tableau 2 présente les résultats pour la classification avec les SVM et avec les trois implémentations de MLN avec et sans la classe rejet (dans le cas sans classe rejet, l'apprentissage/test se fait sur six classes). Le modèle qui donne les meilleurs résultats est le MLN naïf. Clairement les activités à reconnaître dans notre corpus sont souvent liées à un ensemble très réduit d'attributs. Ainsi, par exemple, la présence dans les toilettes suffit pour expliquer l'activité hygiène. L'apprentissage dans le cas MLN naïf mesure l'importance de tous les attributs et leurs possibles

Rejet	Sans						Avec					
	Fenêtrage Recouvrement			Sans			60 secs			120 secs		
	0	25%	50%	0	25%	50%	0	25%	50%	0	25%	50%
SVM	64,9	67,3	68,6	64,8	63,8	65	51,5	54,9	59,3	55,4	54,8	56,4
MLN naïf	65,9	70,8	72,8	76,2	65,4	67,2	51,8	55,9	61,2	54,8	57,7	57,5
MLN+PLI1	40,2	59,8	61,3	47,8	40,8	49,2	45,5	40,8	51,1	31,8	30,4	45,6
MLN+PLI2	64,9	66,8	64,2	60,6	59,8	64,2	51,1	53,3	57,9	50,8	52,0	53,7

TABLE 2 – Taux de bonne classification avec SVM et MLN (avec et sans classe rejet)

valeurs pour inférer chaque activité. Malgré sa simplicité le MLN naïf couvre exhaustivement l’influence de chaque capteur pour reconnaître des activités. De plus, il considère la corrélation inverse des attributs et activités : par exemple, il va donner une probabilité très basse à l’activité hygiène si la présence dans le bureau est significative. Par contre, les règles apprises par PLI risquent de ne pas considérer certaines relations importantes entre attributs et activités. Pourtant, la PLI a le potentiel de modéliser des relations plus complexes entre attributs et cela pourrait être facilement complété avec des règles introduites comme connaissance *a priori*.

Le tableau 3 présente les résultats pour la classification avec les SVM avec et sans la classe rejet. Ce tableau montre un fort écart de reconnaissance entre les différentes classes, qui peut s’expliquer également par le déséquilibre entre celles-ci. L’augmentation du pourcentage de recouvrement améliore quelque peu les résultats. Cependant, les résultats restent faibles pour la classe habillage/déshabillage et pour la classe hygiène. Ces deux classes ont des fenêtres relativement courtes et des résultats qui se répartissent entre les différentes classes dans la matrice de confusion. Lorsque la classe rejet est présente, elle fait partie de ces classes qui sont mal reconnues. L’ajout de la classe rejet détériore les résultats sans spécialement significativement changer les résultats pour chacune des classes. On remarque que beaucoup d’éléments de la classe rejet sont confondus avec les autres classes (mais peu d’exemples se retrouvent classés comme rejet).

Concernant MLN+PLI1, cette méthode n’a pas assez d’exemples positifs et n’offre donc pas de règles utiles pour expliquer une activité. Lorsque le réseau de Markov est instancié, il ne considère pas l’existence des activités qui n’ont pas été apprises. La méthode MLN+PLI2, dont le temps d’apprentissage est beaucoup plus long, offre un ensemble plus conséquent de règles couvrant toutes les activités possibles. Cette augmentation du nombre de règles explique la différence de performance entre ces deux méthodes. Cependant, le MLN naïf reste la méthode la plus performante. Le tableau 4 présente les performances pour chaque activité avec cette méthode. Les performances pour les classes rejet et habillage sont aussi les plus basses. Les résultats pour la classe habillage sont meilleurs à cause du poids positif important associé aux règles relatives aux présences de courtes durées associées à cette classe, un poids négatif

étant associés aux présences longues (contre-exemples). De plus amples expérimentations sont nécessaires pour trancher entre deux explications possibles. La première explication serait une très forte ambiguïté entre les classes mal détectées et les autres, du fait du peu d’utilisation de portes ou d’appareils électriques dans le cas de l’habillage ou du rejet par exemple et du fait aussi que certaines activités sont dissociées d’un lieu unique d’occurrence (l’habillage partage son lieu avec d’autres activités). Les résultats de sélection de variables montrent que les attributs les plus importants sont : la localisation présente, la dernière localisation, le temps de présence dans la pièce, et les pourcentages de présence dans chaque pièce. La localisation est donc essentielle pour la reconnaissance des activités. Les classes rejet et habillage sont distribuées dans plusieurs localisations, par conséquent la localisation ne suffit pas à les reconnaître. D’ailleurs, aucun capteur ne permet de reconnaître facilement l’habillage, (par exemple, il n’y a pas des capteurs dans le tiroir).

La seconde explication serait en rapport avec le découpage temporel. En effet, le fenêtrage est fait de manière périodique et donc rien n’empêche un vecteur d’être à cheval sur plusieurs activités en même temps. Ceci est encore plus probable avec la classe rejet qui est composée de transitions courtes entre activités.

4 Discussion

Le tableau 2 nous montre les résultats dans différentes conditions (de recouvrement et de fenêtrage), et ce pour les SVM puis les MLN (apprentissage version naïve, PLI1 et PLI2). Les données sont acquises dans un environnement représentant un appartement réel avec un minimum de contraintes sur le scénario.

Les résultats les plus bas par classe (présentés dans les tableaux 3 et 4) se retrouvent surtout dans les classes à faible nombre d’éléments (habillage, ranger, hygiène), mais aussi dans la dernière classe, appelée rejet et qui contient tout ce qui n’est pas annoté. Pour les premières classes citées, il est assez difficile d’obtenir des bases de données équilibrées, de par la nature des activités mesurées. Cependant, augmenter encore la taille de notre base de données permettra d’avoir une meilleure description de chacune de ces classes. Les activités décrites par ces classes sont relativement courtes. Ces classes sont donc aussi sensibles à des petits décalages dans l’annotation. Pour la classe rejet,

Rejet	Sans						Avec					
	60 secs			120 secs			60 secs			120 secs		
Fenêtrage Recouvrement	0	25%	50%	0	25%	50%	0	25%	50%	0	25%	50%
Repas	75,4	74,3	72,9	75,9	75,3	73,6	73,3	74,3	75	74,1	71,8	75,3
Ranger	50,3	51,9	52,1	56,8	55,6	55	42,2	46,9	49,5	49,2	50	49,6
Hygiène	44	40,7	35,5	40,9	40,9	40,9	32	25,9	25,8	40,9	40,9	40,9
Détente	79,9	80,6	82	81,6	80,1	82,1	76,3	78	81,4	80	79,4	80,8
Habillage	4,8	4,8	4,8	4,8	2,4	9,5	2,4	0	4,8	4,8	4,8	2,4
Dormir	68,1	81,8	83,3	63,8	60,8	63,2	68,1	78,4	82,5	55,3	62,7	61,4
Rejet	-	-	-	-	-	-	9,3	7,9	7,3	17,1	9,8	7,3

TABLE 3 – Rappel de la classification des activités avec les SVM (validation croisée à 5 partitions)

Rejet	Sans						Avec					
	60 secs			120 secs			60 secs			120 secs		
Fenêtrage Recouvrement	0	25%	50%	0	25%	50%	0	25%	50%	0	25%	50%
Repas	71,3	78,1	78,4	84,2	77,0	75,8	70,8	77,1	75,0	73,6	75,9	77,5
Ranger	53,7	57,5	57,4	63,1	67,7	53,5	42,2	36,9	53,7	44,1	52,4	48,8
Hygiène	52,0	44,4	51,6	66,7	54,5	45,5	40,0	14,8	32,3	50,0	50,0	45,5
Détente	80,9	83,6	85,6	85,3	58,1	84,0	74,2	83,2	84,3	77,6	80,1	79,5
Habillage	19,0	19,0	21,4	62,0	19,0	28,6	11,9	2,4	14,3	33,3	16,7	28,6
Dormir	68,1	76,1	77,2	71,9	82,4	61,4	50,0	43,2	71,9	46,8	56,9	50,9
Rejet	-	-	-	-	-	-	21,9	18,5	13,2	12,2	13,4	7,3

TABLE 4 – Rappel de la classification des activités avec les MLN Naïve (validation croisée à 5 partitions)

celle-ci est difficile à modéliser. En effet, elle doit contenir tout ce qui n'est pas annoté, et donc principalement des transitions entre les activités. Cependant, selon l'annotation (il est également complexe de savoir à quel moment une activité démarre réellement), ces transitions peuvent contenir quelques actions qui sont associées à l'une ou l'autre des classes. Cette classe rejet a donc de très mauvais résultats quelque soit le fenêtrage ou le recouvrement, du fait de son ambiguïté.

En ce qui concerne le fenêtrage et le recouvrement, un recouvrement plus important améliore les résultats. Ceci s'explique par le fait que l'on va de mieux en mieux décrire chacun des instants de l'activité (et également augmenter le nombre de fenêtres), et ainsi améliorer les modèles réalisés. En revanche un fenêtrage plus important diminue les résultats, pour les mêmes raisons (en décrivant les activités avec une granularité moins fine, on obtient des recouvrements plus importants entre les classes).

Le découpage temporel des données doit aussi être repensé. En effet, il n'existe aucune détection d'activité dans notre approche. Ainsi, les fenêtres ne sont pas placées au début et à la fin d'une activité mais de manière périodique. Il serait sûrement très pertinent d'essayer d'estimer les changements d'activités pour éviter de trop grande confusions. Par ailleurs, une activité serait aussi plus facilement classifiable si un contexte était pris en compte tel que la fenêtre précédente et/ou suivante.

Les résultats entre les MLN et les SVM sont relativement bas et semblables (autour des 65% sans classe rejet et 55% avec classe rejet), cependant une légère amélioration

est à noter dans le cas des MLN naïfs. Cette deuxième méthode permet en plus de mieux comprendre la manière dont la séparation est faite entre les classes. Pour les MLN avec apprentissage basé sur la PLI, le nombre d'exemples de certaines classes et le déséquilibre entre classes peut également être responsable de ces mauvais résultats. Ces résultats sont en partie inférieurs à d'autres résultats de la littérature, cependant ce fait peut s'expliquer par les capteurs utilisés, qui sont plus facilement installables (car tous sur bus KNX) mais aussi beaucoup moins informatifs que certains capteurs portés (contraignants). Par ailleurs, ces résultats sont même inférieurs à de précédents résultats que nous avons eu dans des conditions différentes mais comparables [6]. Cependant, les données acquises dans le cadre de cette étude montrent un plus fort réalisme des conditions d'enregistrement (ambiguïté, activités interrompues, activité impliquant plusieurs pièces, etc.) que celui de la plupart des autres travaux de la littérature. De plus amples recherches sur cette base de données permettra très certainement d'obtenir des modèles plus robustes et plus réalistes. Les résultats obtenus avec MLN+PLI1 et MLN+PLI2 sont assez décevants, mais ces méthodes permettent d'obtenir un ensemble de règles vérifiables telles que celle obtenues ci-dessous :

$PercentagePresenceCuisine(90 - 100\%) \wedge$
 $TempsDansLocalisationPresente(8 - 19min) \wedge$
 $VariationTemperatureChambre(0 - 0.05C^\circ) \rightarrow$
 $Activite(PreparerRepas)$
 $PercentagePresenceChambre(90 - 100\%) \wedge$
 $VariationCO2Chambre(3 - 9ppm) \rightarrow Activite(Dormir)$
 $PercentagePresenceBureau(90 - 100\%) \rightarrow Activite(Detente)$

Ces règles pourraient être réutilisées pour l'apprentissage

dans d'autres habitats de façon à constituer une base de règles *a priori* génériques voir même être transférée directement dans un classificateur. Le développement de méthodes MLN+PLI semble être un axe prometteur pour acquérir des classificateurs réutilisables dans d'autres environnements que ceux dans lesquels ils ont été appris. Ce thème de recherche est très important étant donné le coût d'acquisition des corpus nécessaires et le frein que constituent les méthodes probabilistes au développement des bâtiments intelligents du fait de la grande taille des corpus qu'elles réclament.

5 Conclusion

Dans ces travaux, nous avons présenté deux méthodes de classification d'activité appliquées sur des données acquises en conditions réalistes dans un appartement équipé de capteurs. Les résultats obtenus, bien que modestes, sont encourageants en ce qui concerne les méthodes à base de MLN qui donnent des résultats comparables à la méthode SVM, et fournissent des résultats plus facilement interprétables. De meilleurs résultats pourraient être obtenus en incluant l'information audio. Dans de précédents travaux, la localisation et la parole étaient des données très importantes pour la classification, notamment en ce qui concerne les activités de communications et d'interaction avec des objets. Enfin, l'augmentation continue de nos bases de données nous permettra, avec de meilleurs modèles, et de vérifier à quel point la variabilité entre les personnes et entre les différentes façons de réaliser une tâche est bien modélisée dans les algorithmes.

Références

- [1] J. Aggarwal and M. Ryoo. Human activity analysis : A review. *ACM Comput. Surv.*, 43 :1–43, April 2011.
- [2] A. Artikis, A. Skarlatidis, F. Portet, and P. G. Logic-based event recognition. *Knowledge Engineering Review*, to appear.
- [3] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, *ACM*, pages 144 – 152, Pittsburgh, 1992.
- [4] P. Chahua, F. Portet, and M. Vacher. Fusion of audio and temporal multimodal data by spreading activation for dweller localisation in a smart home. In *STAMI, Space, Time and Ambient Intelligence*, pages 17–22, Barcelona, Spain, Jul. 16-22 2011.
- [5] T. Duong, D. Phung, H. Bui, and S. Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. *Artificial Intelligence*, 173(7-8) :830–856, May 2009.
- [6] A. Fleury, M. Vacher, and N. Noury. SVM-based multi-modal classification of activities of daily living in health smart homes : Sensors, algorithms and first experimental results. *IEEE Transactions on Information Technology in Biomedicine*, 14(2) :274 –283, march 2010.
- [7] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman. Video understanding for complex activity recognition. *Machine Vision and Applications*, 18(3-4) :167–188, 2007.
- [8] T. N. Huynh and R. J. Mooney. Discriminative structure and parameter learning for markov logic networks. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, Helsinki, Finland, July 2008.
- [9] S. Katz. Assessing self-maintenance : activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society*, 31(12) :721–727, 1983.
- [10] S. Kok and P. Domingos. Learning markov logic networks using structural motifs. In J. Fürnkranz and T. Joachims, editors, *ICML*, pages 551–558. Omnipress, 2010.
- [11] S. Muggleton. Inverse Entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4) :245–286, 1995.
- [12] U. Naeem and J. Bigham. Activity recognition using a hierarchical framework. *Pervasive Computing Technologies for Healthcare*, pages 24 – 27, 2008.
- [13] F. Portet, M. Vacher, C. Golanski, C. Roux, and B. Meillon. Design and evaluation of a smart home voice interface for the elderly — acceptability and objection aspects. *Personal and Ubiquitous Computing*, to appear.
- [14] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2) :107–136, 2006.
- [15] H. Storf, M. Becker, and M. Riedl. Rule-based activity recognition framework : Challenges, technique and learning. In *Pervasive Computing Technologies for Healthcare*, pages 1–7, London, UK, 2009.
- [16] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, 2 :158–175, 2004.
- [17] S. D. Tran and L. S. Davis. Event modeling and recognition using markov logic networks. In *Proceedings of the 10th European Conference on Computer Vision : Part II*, pages 610–623, Marseille, France, 2008. Springer-Verlag.
- [18] M. Vacher, A. Fleury, F. Portet, J.-F. Serignat, and N. Noury. *Complete Sound and Speech Recognition System for Health Smart Homes : Application to the Recognition of Activities of Daily Living*, pages 645 – 673. Intech Book, 2010.
- [19] N. Zouba, F. Bremond, M. Thonnat, A. Anfosso, E. Pascual, P. Mallea, V. Mailland, and O. Guerin. A computer system to monitor older adults at home : Preliminary results. *Gerontechnology Journal*, 8(3) :129–139, July 2009.