UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE

APPLIQUÉES

PAR

Samira Mohammadi

Estimation de l'occupation dans un bâtiment intelligent avec le modèle de Markov caché à distribution de Dirichlet décalée et mise à l'échelle.

Septembre 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

Remerciements

Je suis profondément reconnaissante envers mon superviseur, le Dr. Mhamed Mesfioui, pour son soutien indéfectible et ses conseils. Son vaste savoir en informatique a été inestimable, et il m'a offert l'incroyable opportunité de poursuivre mon master dans une prestigieuse université internationale. J'ai toujours apprécié mes enseignants pour leur rôle dans mon progrès, et je suis particulièrement redevable au Dr. Mesfioui pour le succès académique que j'ai atteint. Cela a été un immense honneur d'être son étudiante tout au long de ce parcours.

Je tiens également à exprimer mes sincères remerciements à mon conjoint, qui a été ma source constante de soutien à travers tous les chemins de la vie.

Ma plus profonde gratitude va à mes parents et à ma sœur, Mahnaz. Malgré les kilomètres qui nous séparent, leur soutien indéfectible, leurs encouragements et leur motivation ont toujours été avec moi. Je suis certaine que cette réussite n'aurait pas été possible sans leur amour et leur soutien.

Résumé

Un modèle précis est essentiel pour extraire des motifs significatifs des énormes quantités de données générées dans l'environnement actuel, ce qui est crucial pour de nombreuses applications du monde réel. L'analyse des données est fondamentale pour exploiter les technologies intelligentes dans les bâtiments intelligents, qui réduisent la consommation d'énergie et améliorent le confort des utilisateurs. Les modèles de Markov cachés (MMC), connus pour leur efficacité en tant que modèles stochastiques finis, jouent un rôle essentiel dans la modélisation des données. Cette étude met en avant l'importance des prévisions d'occupation intérieure pour l'automatisation des contrôles environnementaux en introduisant des cadres d'estimation de l'occupation pour les bâtiments intelligents utilisant des MMC.

Un aspect clé de la modélisation des données MMC est le choix de la distribution de probabilité d'émission. Nous proposons des extensions innovantes des MMC qui utilisent la distribution de Dirichlet décalée et mise à l'échelle. Le choix de cette distribution découle de sa capacité à surmonter les contraintes des distributions conventionnelles, telles que la Dirichlet, en ce qui concerne les structures de covariance.

L'optimisation des paramètres de distribution devient cruciale une fois que les probabilités d'émission sont établies. En utilisant l'approche de l'estimation du maximum de vraisemblance (EMV), nous avons conçu un algorithme récursif d'estimation des paramètres. La complexité linéaire de cet algorithme permet une modélisation efficace des données en temps réel, élargissant ainsi les applications pratiques de ces modèles.

Table des matières

Liste des tableaux.....vi

Liste des figures.....vii

Introduction.....1

Chapitre 1 3

1.1 Dirichlet 3

1.1.1 Introduction.....3

1.1.2 La distribution de Dirichlet 3

1.1.3 Distribution de Dirichlet généralisée.....4

1.2 Dirichlet décalée-scalée 6

1.2.1 Vue d'ensemble.....6

1.2.2 Applications dans divers domaines 7

1.2.3 Formule et explication 8

1.2.3.1 Définition et formule 8

1.3 MODÈLE DE MARKOV CACHÉ.....10

1.3.1 INTRODUCTION 10

1.3.2 RECHERCHES CONNEXES 11

1.4 Chaîne de Markov 16

1.5 Vraisemblance maximale.....19

1.5.1 Introduction.....19

Chapitre 2 22

Estimation et détection de l'occupation des bâtiments 22

2.1 Introduction.....22

2.2 Estimation et détection de l'occupation avec différents types de capteurs.....23

2.3 Discussion 31

2.4 Développement d'un modèle de Markov caché pour la prédiction de l'occupation 32

2.5 Application du modèle de Markov caché (MMC) dans la prédiction de l'occupation.....33

Chapitre 3 35

Méthode récursive de SSD.....35

Chapitre 4 42

Résultats expérimentaux 42

4.1 Données réelles : Détections d'occupation dans les bâtiments intelligents 42

4.2 Ensemble de données de classification binaire (Ensemble de données de détection d'occupation) 43

4.3 Ensemble de données de classification multiclasses (Ensemble de données d'estimation d'occupation) 45

Conclusion 50

Annexe 62

Liste des tableaux

Tableau 2.1, Comparaison des différents capteurs (Chen et al., 2024).....31

Tableau 4.1, Paramètres estimés pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour l'ensemble de données de détection d'occupation.....44

Tableau 4.2, Exactitude, F-mesure, Précision et Rappel (en pourcentage) pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle appliqués à l'ensemble de données de détection d'occupation.....44

Tableau 4.3, Valeurs des paramètres pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour les données d'occupation de bureau.....46

Tableau 4.4, Exactitude, F-mesure, Précision et Rappel (en pourcentage) pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour le jeu de données d'occupation des bureau.....47

Liste des figures

Figure 1.1 La représentation graphique d'un modèle de Markov caché (MMC) de premier ordre consiste en des états Si représentant l'état caché de Markov et des états O_i représentant les observations à chaque instant t_i, i=1,2,...,T 19

Figure 2.1 La structure du MMC identifie les états cachés (Sn), la probabilité des transitions entre les états (P), et les observations (Xn) (Alfalah, Shahrestani, & Shao, 2023).....33

Figure 4.1 Visualisation des données de détection d'occupation selon différentes caractéristiques…….43

Figure 4.2 Visualization of Office-Occupancy Data According to Different Features……..…45

Introduction

Dans cette thèse, nous avons étudié les modèles de Markov cachés (MMC) et introduit une nouvelle méthode pour l'apprentissage supervisé dans la prédiction de l'occupation des bâtiments intelligents. En utilisant différentes distributions comme probabilités d'émission des (MMC), nous avons relevé les défis de la modélisation des vecteurs positifs. Malgré les recherches approfondies sur les MMC, le choix des probabilités d'émission appropriées demeure un enjeu crucial. Nous avons examiné diverses distributions sous-jacentes pour les (MMC) dans chaque chapitre. Pour fournir une estimation récursive des paramètres des (MMC) pour la modélisation des données des bâtiments intelligents, nous avons développé un modèle statistique d'apprentissage automatique. Nous nous sommes concentrés sur la prédiction de l'état d'occupation des pièces dans les bâtiments intelligents, les traitant comme des états cachés dans nos modèles. L'objectif des bâtiments intelligents est d'estimer le nombre d'occupants, en alignement avec la nature des (MMC) et en atteignant deux objectifs : économies d'énergie et confort des utilisateurs. Les attributs principaux du modèle, à savoir 1) les algorithmes récursifs et 2) l'adaptabilité dans le choix des probabilités d'émission, permettent son application à un large éventail de situations réelles. Il est crucial de souligner que la complexité linéaire des algorithmes récursifs dans nos modèles offre un avantage sur les algorithmes itératifs, nous permettant d'analyser des ensembles de données plus volumineux sans rencontrer de saturation de la mémoire ou de demandes de traitement excessives.

Dans le chapitre 1, nous offrons une vue d'ensemble complète de la modélisation probabiliste, avec un accent sur les distributions importantes et leurs utilisations. Nous examinons les caractéristiques et les applications des distributions de Dirichlet et de Dirichlet généralisé dans plusieurs domaines. Nous couvrons également les utilisations pratiques de la distribution de Dirichlet décalée et mise à l'échelle. Le modèle de Markov caché (MMC) et les études associées sont également abordés dans ce chapitre, en soulignant l'importance du modèle pour la modélisation des données séquentielles. Nous explorons les principes de l'estimation du maximum de vraisemblance (EMV) et des chaînes de Markov, offrant un fondement complet pour comprendre les modèles probabilistes complexes.

Dans le chapitre 2, nous explorons la détection et l'estimation de l'occupation des bâtiments. Nous commençons par une introduction et un résumé des études antérieures

dans le domaine, en discutant de plusieurs approches et technologies pour l'estimation de l'occupation. Cela implique l'examen des équipements en question et l'utilisation de divers types de capteurs. Ensuite, nous montrons la création d'un modèle de Markov caché (MMC) spécifique à la prédiction de l'occupation. Le chapitre se termine par un examen des utilisations des (MMC) dans la prédiction de l'occupation, en mettant l'accent sur l'utilité de l'outil et les avantages possibles dans cette situation.

Dans le chapitre 3, nous présentons une analyse approfondie de la technique de Dirichlet décalée et mise à l'échelle récursive (SSD). Ce chapitre fournit une description complète de l'approche SSD ainsi que les équations clés et les fondements théoriques. Nous disséquons systématiquement et montrons la mise en œuvre étape par étape de la technique SSD dans son application récursive. Le but du chapitre est de fournir aux lecteurs une compréhension complète des nombreux scénarios de modélisation probabiliste dans lesquels l'approche SSD peut être appliquée avec succès.

Le chapitre 4 présente nos résultats expérimentaux. Les données réelles de détection de l'occupation dans les bâtiments intelligents sont analysées dans ce chapitre. Nous discutons de l'utilisation d'un ensemble de données de classification binaire connu sous le nom de jeu de données de détection de l'occupation, ainsi que d'un ensemble de données de classification multiclasse connu sous le nom de jeu de données d'estimation de l'occupation. Nos résultats démontrent que les méthodes proposées sont efficaces pour détecter et estimer avec précision l'occupation. Un résumé des résultats et de leurs implications est présenté à la fin du chapitre, fournissant des informations sur leurs applications pratiques et les orientations futures.

Chapitre 1

Revue complète et applications dans la modélisation probabiliste

1.1 Dirichlet

1.1.1 Introduction

La distribution de Dirichlet est une distribution multivariée avec diverses applications, notamment dans les statistiques d'ordre, les modèles de programmation par contraintes probabilistes et les problèmes de livraison. Elle est également utilisée pour modéliser le comportement d'achat des clients pour des biens non durables tels que la nourriture et les articles de toilette.

Les recherches de Prékopa et Kelle (1978) et Tiao et Guttman (1965a, 1965b) discutent de ces applications. Un modèle notable pour le comportement d'achat multimarque utilise la distribution bêta multivariée pour des situations indépendantes.

Des méthodes de calcul des probabilités normales multivariées ont été proposées par Genz (1992) et Kotz et al. (2000). Cependant, moins d'études se concentrent sur les probabilités de Dirichlet. Yassaee (1974) a calculé l'intégrale de probabilité de la distribution de Dirichlet en évaluant la distribution de Dirichlet inversée, tandis que Szántai (1985) a utilisé une simulation de réduction de la variance pour estimer les probabilités de Dirichlet.

Gouda et Szántai (2004) ont introduit de nouvelles méthodes d'échantillonnage pour calculer la fonction de distribution cumulative de la distribution de Dirichlet, à savoir l'échantillonnage conditionné séquentiel (SCS) et l'échantillonnage conditionné séquentiel par importance (SCIS). Ils ont adapté SCS et SCIS en fonction des caractéristiques de la distribution de Dirichlet et ont comparé leurs performances à la méthode de simulation Monte Carlo brut (CMC).

1.1.2 La distribution de Dirichlet

La distribution de Dirichlet est une version multivariée de la distribution bêta. Elle peut modéliser des données symétriques et asymétriques. Selon Bouguila, Ziou et Vaillancourt

(2004), la distribution de Dirichlet peut être symétrique ou décalée vers la droite ou la gauche. Considérons le vecteur aléatoire X=$(x_1, x_2, \dots, x_K)$ qui suit une distribution de Dirichlet. La fonction de densité conjointe est donnée par:

$$p(x_1, x_2, \dots, x_K) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \quad , \tag{1.1}$$

où

$$\alpha_0 = \sum_{i=1}^{K} \alpha_i \quad o\text{ù} \;\; \alpha_i > 0, i = 1, \dots, K.$$

La moyenne et la variance de la distribution de Dirichlet, ainsi que la covariance entre x_i et x_j sont les suivantes:

$$E(x_i) = \frac{\alpha_i}{\alpha_0}, \tag{1.2}$$

$$Var(x_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}, \tag{1.3}$$

$$Cov(x_i, x_j) = \frac{-\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}. \tag{1.4}$$

La distribution de Dirichlet avec le vecteur de paramètres $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$ peut être représentée à la fois comme une distribution sur l'hyperplan $D_K = (x_1, x_2, \dots, x_K),\ \sum_{i=1}^{K} x_i = 1$ dans $\mathbb{R}_+^K$ ou à l'intérieur du simplexe $D_K = (x_1, x_2, \dots, x_K),\ \sum_{i=1}^{K} x_i < 1$ dans $\mathbb{R}_+^K$.

1.1.3 Distribution de Dirichlet généralisée

En raison de la structure de covariance négative de la distribution de Dirichlet et de la nécessité de décrire des données avec une structure de covariance plus générale, une généralisation de la distribution de Dirichlet a été développée, basée sur la notion statistique de neutralité. Connor et Mosimann (1969) ont présenté le concept de neutralité comme suit :

Considérons un vecteur aléatoire de proportions (P_1, P_2, ..., P_K) et les variables aléatoires Z_i, i=1,2, ...,K où $0 \leq Z_i \leq 1$, La proportion P_1 est dite neutre si Z_1=P_1 est indépendante des vecteurs Z_2=P_2/[1-P_1], Z_3=P_3/[1−P_1], ..., Z_K = P_K/[1 − P_1]. Si l'on suppose que le vecteur

aléatoire X dans la distribution de Dirichlet est entièrement neutre, alors les variables aléatoires Z, sont mutuellement indépendantes. La fonction de densité de chaque Z_i, i=1, ...,K – 1 est définie comme une distribution bêta univariée, comme suit:

$$[Beta(\alpha_i, \beta_i)]^{-1} {z_i}^{\alpha_i - 1}(1 - z_i)^{\beta_i - 1},$$

où $\alpha_i, \beta_i > 0$ et Beta(α_i, β_i) est la fonction bêta définie comme suit:

$$Beta(\alpha_i, \beta_i) = \frac{\Gamma(\alpha_i)\Gamma(\beta_i)}{\Gamma(\alpha_i + \beta_i)}.$$

Maintenant, par la transformation ci-dessous, les x_i peuvent être dérivés des Z_i:

$$x_i = Z_i\left[\prod_{m=1}^{i-1}(1 - Z_m)\right], \quad i = 1,2, \dots, K - 1.$$

La structure de covariance devient plus flexible dans cette généralisation car le vecteur des variables aléatoires correspondant à la distribution de Dirichlet généralisée n'est pas totalement neutre. Supposons que X = $(x_1, x_2, \dots, x_K)$ est un vecteur aléatoire suivant une distribution de Dirichlet généralisée avec les paramètres α

et β: GD(α;β), où α= (α_1,α_2,...,α_K) et β= (β_1, β_2, ..., β_K). La distribution de Dirichlet généralisée est définie comme suit (Boutemedjet, Ziou, et Bouguila, 2007; Epaillard et Bouguila, 2014) :

$$GD\big((X|\alpha, \beta)\big) = \prod_{i=1}^{K} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} x_i^{\alpha_i - 1}\left(1 - \sum_{r=1}^{i} x_r\right)^{\eta_i}, \tag{1.5}$$

où

$$\alpha \epsilon \mathbb{R}_+^K, \;\; \beta \epsilon \mathbb{R}_+^K, \;\; x \epsilon \mathbb{R}_+^K,$$

$$\sum_{i=1}^{K} x_i < 1,$$

$$\eta_i = \beta_i - (\alpha_{i+1} + \beta_{i+1}), \;\; i \in [1, K - 1],$$

$$\eta_K = \beta_K - 1.$$

Dans le cas où $\beta_{i-1} = \alpha_i + \beta_i$ pour $2 \leq i \leq K$ nous obtenons la distribution de Dirichlet standard.

Selon la description de Connor et Mosimann (1969), les moments de la distribution de Dirichlet généralisée sont dérivés comme suit :

$$M_{j-1} = \prod_{m=1}^{j-1} \left[(\beta_m + 1) / \left(\alpha_m + \beta_m + 1 \right) \right],$$

$$E(x_j) = \left(\alpha_j / [\alpha_j + \beta_j]\right) \prod_{m=1}^{j-1} [\beta_m / (\alpha_m + \beta_m)], \quad j = 1, \dots, K;$$

$$Var(x_j) = E(x_j)\left[\alpha_j + 1/(\alpha_j + \beta_j + 1)\right] M_{j-1} - E(x_i), \quad j = 1, \dots, K;$$

$$Cov(x_i, x_j) = E(x_i) \left[\frac{(\alpha_i)}{[\alpha_i + \beta_i + 1] M_{i-1}} - E(x_i) \right],$$

$$i = 1, \dots, K-1, \qquad j = i+1, \dots, K.$$

1.2 Dirichlet décalée-scalée

1.2.1 Vue d'ensemble

La distribution de Dirichlet décalée-scalée est une variante de la distribution de Dirichlet traditionnelle qui inclut des paramètres de décalage et de mise à l'échelle, augmentant ainsi sa polyvalence et son utilité. Cette étude de la littérature explore le travail fondamental sur la distribution de Dirichlet, ses extensions, et la création et l'implémentation de la distribution de Dirichlet décalée-scalée dans divers domaines.

La distribution de Dirichlet, une version multivariée de la distribution bêta, est largement utilisée dans l'analyse des données compositionnelles. Aitchison (1986) a mené une enquête détaillée sur la distribution de Dirichlet pour évaluer les données compositionnelles, dans lesquelles les composants sont des proportions qui s'additionnent à 1. Cette étude pionnière a ouvert la voie à un large éventail d'applications dans des domaines tels que la géologie, l'écologie et l'économie, qui utilisent tous des données compositionnelles. Johnson, Kotz et Balakrishnan (1997) ont élargi les caractéristiques et les applications de la distribution de Dirichlet dans leur livre sur les

distributions multivariées discrètes. Ils ont discuté des caractéristiques mathématiques, des stratégies d'estimation des paramètres, et des applications dans différents modèles statistiques.

Plusieurs chercheurs ont élargi la distribution de Dirichlet conventionnelle pour répondre à ses inconvénients. Connor et Mosimann (1969) ont développé une distribution de Dirichlet généralisée pour représenter des proportions dépendantes. Leur travail s'est concentré sur l'assouplissement de l'hypothèse d'indépendance proportionnelle, permettant une modélisation plus réaliste des données complexes. Gelman et al. (2013) ont examiné plusieurs extensions de la distribution de Dirichlet dans le contexte du traitement des données bayésiennes. Ils ont souligné l'utilisation de modèles hiérarchiques et de lois a priori de Dirichlet pour capturer des relations de données complexes, ce qui a conduit à la création de modèles plus avancés tels que la distribution de Dirichlet décalée-scalée.

1.2.2 Applications dans divers domaines

1. **Modélisation de sujets**

Blei, Ng et Jordan (2003) ont développé l'Allocation de Dirichlet Latente (LDA), un modèle génératif pour identifier les thèmes dans les corpus de textes. Bien que leur étude se soit concentrée sur la distribution de Dirichlet standard, elle a suscité de nouvelles recherches sur des modèles de sujets plus flexibles. La distribution de Dirichlet décalée-scalée a été étudiée pour améliorer la modélisation de sujets en reflétant mieux la distribution des thèmes dans les textes.

Griffiths et Steyvers (2004) ont utilisé la LDA pour analyser la littérature scientifique, illustrant la nécessité de modèles de sujets flexibles capables de traiter des distributions de données de plus en plus complexes. Leurs résultats ont démontré les avantages potentiels de l'utilisation de distributions de Dirichlet décalée-scalée dans l'analyse de texte.

2. **Analyse de données génétiques**

Pritchard, Stephens et Donnelly (2000) ont utilisé un modèle basé sur Dirichlet pour estimer la structure des populations à l'aide de données de génotypes multilocus. Ils ont souligné l'importance de lois a priori flexibles pour représenter efficacement la variation génétique au sein des populations. Des recherches ultérieures ont utilisé la distribution de Dirichlet décalée-scalée pour acquérir une compréhension plus sophistiquée des données génétiques.

Ewens (2004) a mis en évidence l'utilisation des distributions de Dirichlet en génétique des populations mathématiques. Il a discuté de la façon dont les extensions, telles que la distribution de Dirichlet décalée-scalée, pourraient augmenter la précision des modèles génétiques en intégrant des facteurs de décalage et d'échelle.

3. Analyse des données compositionnelles

Greenacre (2010) a étudié l'utilisation des biplots pour l'analyse des données compositionnelles, soulignant la valeur des distributions flexibles telles que la distribution de Dirichlet décalée-scalée pour modéliser des données qui ne s'inscrivent pas dans le simplexe typique. Cette méthode permet une meilleure représentation et compréhension des données compositionnelles dans de nombreux domaines.

4. Développements théoriques

Ferguson (1973) a proposé le processus de Dirichlet, une extension de la distribution de Dirichlet qui a été essentielle pour les méthodes bayésiennes non paramétriques. Son travail a ouvert la voie à divers développements futurs, notamment la distribution de Dirichlet décalée-scalée, en offrant un cadre flexible pour modéliser des données complexes.

Robert et Casella (2004) ont exploré les techniques de Monte Carlo pour l'inférence bayésienne, qui sont nécessaires pour estimer les paramètres de distributions complexes telles que la distribution de Dirichlet décalée-scalée. Leur travail fournit les outils de calcul nécessaires pour construire ces modèles sophistiqués.

1.2.3 Formule et explication

La distribution de Dirichlet décalée-scalée est une version plus complexe de la distribution de Dirichlet classique qui inclut des facteurs de décalage et d'échelle supplémentaires pour augmenter sa polyvalence. Voici une description complète de sa définition et de sa formule :

1.2.3.1 Définition et formule

La fonction de densité de probabilité (PDF) de la distribution de Dirichlet décalée-scalée pour un vecteur aléatoire de dimension k, $\mathrm{X} = (\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_k)$ est définie comme suit:

$$f(x_1, x_2, \dots, x_3; \mu, \beta, \alpha) = \frac{1}{B(\alpha \odot \beta)} \prod_{i=1}^{k} \left(\frac{x_i - \mu_i}{\beta_i}\right)^{\alpha_i \beta_i - 1}. \qquad (1.6)$$

Où :

- $x_i \geq \mu_i$.
- $\sum_{i=1}^{k} x_i - \mu_i = 1$.
- $\mu = (\mu_1, \mu_2, \ldots, \mu_k)$ est le vecteur de décalage.
- $\beta = (\beta_1, \beta_2, \ldots, \beta_k)$ est le vecteur d'échelle.
- $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ est le vecteur de concentration.
- $\odot$ désigne la multiplication élément par élément.
- $B(\alpha \odot \beta)$ est la fonction bêta multivariée, définie comme:

$$B(\alpha \odot \beta) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i \beta_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i \beta_i)}.$$

Explication des composants

1. **Vecteur de décalage (μ):** Le vecteur de décalage μ recentre la distribution sur chaque dimension. En soustrayant μ_i de chaque x_i, la distribution peut être centrée autour de μ plutôt qu'autour de l'origine. Ce changement permet à la distribution de décrire des données qui ne se centrent pas naturellement autour du simplexe conventionnel (Minka, 2000).
2. **Vecteur d'échelle (β):** Les paramètres d'échelle β modifient la concentration de la distribution. En échelonnant chaque x_i-μ_i par β_i, on modifie l'étendue de la distribution. Des valeurs de β_i plus grandes concentrent la distribution autour de μ_i tandis que des valeurs plus petites entraînent une distribution plus dispersée (Gelman et al., 2013).
3. **Vecteur de concentration (α):** Les paramètres α déterminent la forme de la distribution. Des valeurs plus élevées de α_i indiquent une plus grande concentration de x_i-μ_i autour de sa moyenne. Cela est comparable à la distribution de Dirichlet classique, où la forme et l'étendue sont dictées par les paramètres α (Aitchison, 1986).
4. **Fonction bêta multivariée:** La function $B(\alpha \odot \beta)$ garantit que la function de densité de probabilité (PDF) s'intègre à un sur le simplexe décalé et mis à l'échelle. Elle étend le facteur de normalisation utilisé dans la distribution de Dirichlet classique pour prendre en compte les paramètres de décalage et d'échelle (Johnson, Kotz, & Balakrishnan, 1997).

1.2.3.3 Propriétés

1. **Support:** Chaque x_i est limité de sorte que $x_i \geq \mu_i$ et $\sum_{i=1}^{k} x_i - \mu_i = 1$. Cette restriction garantit que la distribution est contenue dans un simplexe décalé et mis à l'échelle (Connor & Mosimann, 1969).

2. **Flexibilité:** Les paramètres de décalage et d'échelle (μ et β) permettent à la distribution de simuler une plus grande variété de données, ce qui la rend appropriée pour les situations où les données ne respectent pas les limitations typiques du simplexe (Minka, 2000).
3. **Multimodalité:** La distribution peut présenter un comportement multimodal en fonction des valeurs de α, μ, et β. Cette caractéristique est importante pour capturer des distributions de données complexes (Gelman et al., 2013).

1.3 MODÈLE DE MARKOV CACHÉ

1.3.1 INTRODUCTION

Après des décennies de reconnaissance, les modèles de Markov cachés (MMC), l'un des modèles probabilistes les plus impressionnants et efficaces, gagnent à nouveau en notoriété au sein de la communauté d'apprentissage automatique. Leur utilisation croissante pour modéliser une large gamme de données hétérogènes (images, vidéos, audio et séries temporelles) dans une multitude de scénarios du monde continue de s'accroître.

Les MMCs (Modèles de Markov Cachés) sont des modèles statistiques adaptables qui se sont révélés efficaces et précieux dans une gamme d'applications d'apprentissage automatique. Lorsque Baum et al. ont créé une technique de maximum de vraisemblance (ML) pour estimer les paramètres des MMCs à partir d'une séquence d'observations d'entraînement, ils ont été présentés pour la première fois dans leur intégralité (Baum & Petrie, 1966). Leur origine remonte à 1948, lorsque Shannon a créé un modèle de source de Markov pour la langue anglaise (Shannon, 1948). Souvent appelés techniques probabilistes dynamiques, les MMCs ont été utilisés dans plusieurs domaines, tels que le traitement de la voix (Rabiner & Juang, 1986; Epaillard & Bouguila, 2016), la reconnaissance des gestes et des textures (Aggarwal & Cai, 1999; Qiao & Weng, 2015; Epaillard et al., 2014), la vérification de signatures (Justino et al., 2000; Oliveira et al., 2005; Batista et al., 2012), et la détection d'anomalies (Rabiner & Juang, 1986; Epaillard & Bouguila, 2016). Ils ont été utilisés plus récemment pour l'estimation de l'occupation dans les bâtiments intelligents (Ai et al., 2014; Amayri et al., 2017).

Le principal avantage des MMCs réside dans leur capacité à traiter de nombreuses formes de données, qu'elles soient discrètes ou continues, et à simuler des signaux du monde réel avec précision. Ils sont excellents pour gérer des données non observables ou manquantes, fournissant des informations précieuses sans avoir besoin d'une observation directe. Cela aide à l'interprétation et à la prédiction des scénarios, ainsi qu'à la découverte de variables cachées qui pourraient être négligées par des observations directes. En améliorant la procédure d'initialisation pour une meilleure compacité et des valeurs de paramètres optimisées, la structure des MMCs a connu certaines améliorations (Lee et al., 2018; Bicego et al., 2005). Il est rare de discuter de la sélection des fonctions de distribution de probabilité des émissions, même si la procédure d'entraînement des MMCs reste standardisée. En raison de leur simplicité pratique et mathématique, les modèles de mélange gaussien (GMMs) sont souvent utilisés par défaut (Bicego et al., 2005; Cholewa & Głomb, 2013). Cependant, cette méthode ne produit pas toujours des résultats optimaux. En raison de leurs présomptions de symétrie et de support illimité, les GMMs peuvent donner des descriptions peu plausibles pour certains types de données.

Alternativement, le modèle de mélange de Dirichlet (DMM) a été proposé en raison de son support compact [0,1], ce qui lui permet de traiter des données textuelles, des images et des vidéos (Bouguila & Ziou, 2007). Le DMM peut potentiellement surpasser les GMMs dans la découverte de motifs grâce à sa polyvalence dans l'approximation des distributions symétriques et asymétriques (Bdiri & Bouguila, 2012; Chen et al., 2007).

1.3.2 RECHERCHES CONNEXES

Depuis leur réapparition notable dans les recherches révolutionnaires de Rabiner et Juang (1986), les modèles de Markov cachés (MMCs) ont subi de nombreuses modifications et expansions. Ces modifications ont été motivées par le désir d'améliorer la capacité d'un système à s'adapter à une variété de formats de données et de simplifier la modélisation des séries temporelles. Cela implique d'introduire de l'aléatoire dans les motifs réguliers, observables et apprenables, de non-stationnarité temporelle et de variabilité spatiale des événements réels (Yu & Deng, 2015).

Le traitement de la parole a fait un usage considérable des MMCs en raison de leur capacité à gérer les entrées unidimensionnelles. Cela a suscité des études similaires dans la classification d'images et la reconnaissance d'objets, utilisant continuellement les MMCs pour affiner les résultats grâce à une variété de méthodologies et de procédures. Même la reconnaissance faciale, un sujet qui est naturellement bidimensionnel, a été étudiée en utilisant des MMCs unidimensionnels. Par exemple, dans leur étude, Samaria et Fallside (1993) ont considéré les visages comme des objets 2D divisés en une série

prédéfinie de régions et ont utilisé un MMC unidimensionnel de gauche à droite. Une séquence d'observation 1D a été extraite pour l'extraction automatique des caractéristiques de chaque région. Les composants de la séquence d'observation étaient les vecteurs d'intensité des pixels. Ils ont utilisé une base de données avec de légères variations dans les positions de la tête et les expressions faciales pour entraîner un modèle double MMC 1D. Un MMC à 8 états correspondant aux huit régions différentes de l'image faciale a été le résultat de cela. Ensuite, un MMC à 5 états avec des transitions d'état limitées aux états voisins a été entraîné à l'aide de ce MMC. Cette stratégie a produit des résultats de reconnaissance réussis même avec de légers changements dans l'orientation de la tête dans la base de données d'images.

Les résultats marginaux de Samaria et Fallside (1993) ont conduit à d'autres modifications par Samaria et Harter (1994). L'application d'un véritable MMC 2D pour la catégorisation d'images a été le principal objectif des efforts ultérieurs. Par exemple, Li et al. (2000) ont amélioré la catégorisation par contexte en utilisant ce type de MMC. Ils ont proposé un modèle dans lequel les vecteurs de caractéristiques sont conditionnés par des états voisins (horizontaux et verticaux) et sont statistiquement dépendants d'un processus d'état fondamental, appelé maillage de Markov. Cela rend les dépendances visibles dans deux dimensions en même temps. Les paramètres du MMC ont été estimés à l'aide d'une technique EM, et la procédure de classification a trouvé des classes ayant collectivement la plus grande probabilité a posteriori pour tous les blocs d'états voisins. Pour atteindre la complexité en temps polynomial, le travail de Li et al. (2000) a utilisé une approche sous-optimale en conjonction avec l'entraînement Viterbi.

Bien que les MMCs aient été initialement créés pour des données discrètes et gaussiennes (Rabiner & Juang, 1986), le modèle MMC sous-jacent a dû être modifié afin d'accueillir efficacement ces nouveaux types de données en raison de ses utilisations variées dans la détection d'anomalies, le diagnostic des maladies et la prévision dynamique (Sebastian et al., 2018).

De même, les textures dynamiques suivent rarement des distributions gaussiennes. Elles peuvent être représentées à l'aide de distributions mixtes, qu'elles soient gaussiennes ou non. Les MMCs ont été utilisés dans Qiao & Weng (2015) où les variables latentes ont été traitées comme discrètes mais suivaient une distribution de probabilité d'émission arbitraire, sans définir la nature de cette distribution, en raison de l'utilité des états latents dans la classification et la segmentation des textures dynamiques. Une distribution de probabilité d'émission aléatoire combinée à une dépendance d'ordre supérieur au sein des états cachés, selon le modèle des auteurs, capturerait plus précisément la structure d'une texture dynamique et encoderait son information

d'apparence avec les variables observées. Le modèle a utilisé l'algorithme traditionnel de Baum-Welch. Pour l'estimation des paramètres, le modèle a utilisé l'approche bien connue du maximum de vraisemblance. Basée sur la notion que la texture est une caractéristique régionale, cette méthode a été récemment améliorée dans Qiao & Xing (2018), qui a utilisé un MMC multivarié pour simuler les pixels voisins changeants au fil du temps. Cet effort récent a surpassé la technique MMC conventionnelle en termes de précision de classification.

De plus, les MMCs ont été utilisés pour simuler plusieurs processus à la fois, en particulier dans les contextes où des données longitudinales sont en place. Altman (2007) a présenté une nouvelle classe de modèles connus sous le nom de MMCs mixtes (MMCMs), illustrant comment les utiliser avec des informations sur le nombre de lésions chez les patients atteints de sclérose en plaques. En permettant l'intégration de variables et l'ajout d'effets aléatoires, ce modèle étend les capacités des MMCs conventionnels pour gérer l'hétérogénéité des données provenant de nombreuses sources. En assouplissant l'exigence que les observations soient indépendantes étant donné les états cachés, ces effets aléatoires offrent une flexibilité dans la modélisation des corrélations et une estimation efficace des paramètres pour les processus typiques. Ils sont une extension naturelle des modèles. Une illustration pratique des MMCMs peut être trouvée dans Punzo & Maruotti (2016), qui examine des informations sur l'activité criminelle en Italie. L'influence des racines territoriales dans certains endroits italiens sur le crime organisé a été prise en considération par les auteurs lorsqu'ils ont modélisé de nombreux aspects concernant le moment et les types d'activités criminelles. La complexité des relations entre ces variables a été capturée en représentant différents niveaux de conditions de sécurité comme des états cachés.

La reconnaissance des expressions faciales (REF) a vu l'accomplissement d'adaptations tout aussi célèbres des MMCs. Samaria & Young (1994) ont utilisé des MMCs pour aborder un problème d'identification faciale dès 1993; ils ont été influencés par Rabiner (1989) et son travail révolutionnaire dans la reconnaissance vocale. Pour identifier un ensemble de photographies de test, ils ont utilisé des MMCs pour extraire des caractéristiques faciales à partir des données d'entraînement. Cette technique a traité les visages comme des objets bidimensionnels et a utilisé un classificateur dépendant du contexte basé sur des MMCs. Les photos de visage ont été segmentées en sections horizontales, statistiquement liées, appelées "bandes faciales" en extrayant des caractéristiques faciales statistiques (Samaria & Fallside, 1993). Pour chacune des vingt personnes différentes qui ont été prises en considération dans les études, un MMC distinct de gauche à droite a été entraîné. Chaque bande de visage dans le modèle

représentait un seul état. La stratégie basée sur les MMCs a mieux fonctionné que les autres méthodes purement statistiques, selon les résultats. L'étude d'Otsuka & Ohya (1996), où les auteurs ont utilisé un MMC de gauche à droite basé sur la densité gaussienne avec trois états pour les tâches de reconnaissance, est l'une des expériences ultérieures utilisant des MMCs pour cibler la REF. Pour chaque expression, il y avait trois états qui correspondaient à des groupes d'expressions distincts: visage neutre, visage en mouvement et visage dans son état stable. Cette approche a bien fonctionné pour quelques phrases mais pas pour l'ensemble complet, en partie parce que le modèle avait du mal avec la grande similarité de forme. Dans Wu et al. (2015), les images ont été segmentées en séquences étiquetées (sacs), chacune composée de plusieurs segments considérés comme des instances dans le sac, afin de capturer l'information temporelle dans les segments faciaux en utilisant un MMC multi-instance. Afin de maximiser la probabilité conditionnelle d'une séquence d'images donnée sa séquence de niveau associée, l'apprentissage multi-instance a été utilisé tout au long du processus de formation. Cette méthode a réussi à localiser correctement les cadres de séquences multi-pic et à étiqueter des séquences. Le MMC de Nean et al. (1999) a été utilisé par Zhou et al. (2004) pour la REF en temps réel dans le cadre d'un environnement de jeu informatique interactif. La complexité de la formation et de la reconnaissance de la tâche a été réduite par ce MMC intégré en utilisant des vecteurs d'observation composés de coefficients de transformée en cosinus discrète bidimensionnelle (2D-DCT) au lieu d'intensités de pixels. Tout bien considéré, aucune de ces techniques n'a particulièrement modifié ses modèles MMC pour traiter des vecteurs positifs, qui servent d'unité principale de représentation des images. Dans le contexte de la tâche REF, nous utilisons pour la première fois la distribution de Dirichlet inversée (ID) comme probabilité d'émission qui correspond le mieux à nos vecteurs positifs récupérés.

De même, les MMCs ont trouvé une utilisation répandue dans les travaux de détection d'occupation, en particulier dans des applications telles que la gestion de l'énergie, la sécurité, la ventilation auto-contrôlée et les environnements de bâtiments verts durables. Un MMC affiche simplement les sorties, qui sont des manifestations environnementales telles que les niveaux de température et de CO2, et n'observe pas directement l'état. En élargissant cette notion, Dong et al. (2010) ont utilisé la version traditionnelle de ce modèle statistique pour identifier les niveaux d'occupation, qui correspondent à la quantité d'états cachés. De plus, les MMCs ont été utilisés par Ai et al. (2014), Amayri et al. (2017), et Amayri et al. (2016) pour estimer l'occupation dans les bâtiments intelligents. Par exemple, les auteurs d'Ai et al. (2014) ont analysé des données d'un réseau de capteurs sans fil précédemment établi pour calculer le nombre d'habitants dans un centre de recherche en utilisant un MMC auto-régressif. Leurs résultats ont montré

que le MMC auto-régressif a surpassé la technique MMC traditionnelle, en particulier dans les situations où les niveaux d'occupation variaient souvent.

La nécessité de modéliser efficacement les durées d'état est une autre justification pour l'expansion des MMCs conventionnels. Parce que les MMCs fonctionnent par itérations où les états peuvent changer, il est crucial de modéliser la durée de ces itérations en fonction des états à chaque point. Un MMC a la possibilité de changer d'états ou de rester dans le même état lors de chaque itération. Bien que cette hypothèse de distributions géométriques constantes pour les durées d'état dans les MMCs traditionnels puisse parfois décrire avec précision la réalité physique, elle est fréquemment inappropriée pour un large éventail d'applications. Afin de résoudre cela, Ferguson (1980) a proposé le MMC à durée variable, qui permet à diverses distributions de probabilité d'être utilisées comme modèles pour les durées d'état. En élargissant cette idée, Djuric & Chun (2002) ont présenté des MMCs non stationnaires qui décrivent les durées d'état avec une fonction de masse de probabilité spécifiée, où les probabilités de transition d'état sont des fonctions du temps. Plus de précision et de flexibilité sont disponibles pour décrire la durée des états dans différentes applications avec cette méthode.

Des études se sont concentrées sur la modélisation des données proportionnelles dans une tentative d'améliorer la flexibilité des MMCs. Pour la première fois, des mélanges de Dirichlet ont été utilisés comme probabilités d'émission en utilisant des données proportionnelles synthétiques dans Bouguila & Ziou (2007). Plus tard, cette méthode a été développée et utilisée avec des données réelles dans Epaillard et al. (2014). Par la suite, Epaillard & Bouguila (2016) ont modifié les MMCs pour incorporer individuellement les mélanges de Dirichlet généralisés et la distribution de Beta-Liouville comme probabilités d'émission. Ces modifications ont produit de meilleurs résultats parce que le MMC avait une structure de covariance plus adaptable grâce à ces types de probabilités (Bouguila & Ziou, 2007).

Les travaux utilisant la distribution de Dirichlet inversée (ID) pour les applications d'apprentissage automatique sont précieux, malgré leurs limitations. Cette distribution a été explicitement utilisée pour décrire des vecteurs positifs, ce qui a conduit à un succès remarquable récent en reconnaissance de motifs (Bdiri & Bouguila, 2011a; Bdiri & Bouguila, 2011b; Tirdad et al., 2015). En plus d'être plus pratique pour décrire les types de données qui ont été discutés précédemment, la distribution ID montre également de meilleures performances lors du traitement des données symétriques et asymétriques (Tirdad et al., 2015). En utilisant une architecture variationnelle accélérée suggérée par Fan & Bouguila (2018), un modèle de mélange ID infini a été intégré et utilisé pour un défi de reconnaissance des expressions faciales (REF). D'abord proposé dans Fan & Bouguila

(2015), ce modèle a été encore affiné pour surpasser les performances du modèle de mélange gaussien et a montré une meilleure gestion des ensembles de données volumineux.

1.4 Chaîne de Markov

On peut utiliser une chaîne de Markov pour déterminer la probabilité d'une série d'occurrences observables. Cependant, les événements qui nous intéressent sont souvent cachés, ce qui signifie qu'ils ne peuvent pas être vus à l'œil nu. Par exemple, les étiquettes de parties du discours ne sont généralement pas visibles dans un texte ; à la place, nous voyons les mots et devons déduire les étiquettes à partir de l'ordre des mots. Comme ces étiquettes ne peuvent pas être vues directement, elles sont considérées comme cachées (Jurafsky & Martin, 2009).

Un modèle de Markov caché (MMC), que nous considérons comme des éléments causaux dans notre modèle probabiliste, nous permet de discuter à la fois des événements cachés (tels que les étiquettes de parties du discours) et des événements visibles (tels que les mots dans l'entrée). Les éléments suivants définissent un MMC :

$Q=q_1,q_2,\ldots,q_N$	un ensemble de N états
$A=a_{11},\ldots,a_{ij},\ldots,a_{NN}$	une **matrice de probabilité de transition A**, chaque a_{jj} représentant la probabilité de passer de l'état i à l'état j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \ \ \forall i$
$O=o_1,o_2,\ldots,o_T$	une séquence de T **observations,** chacune tirée d'un vocabulaire V= $v_1,v_2,\ldots, v_V$
$B=b_i(o_t)$	une séquence de probabilités d'observation, également appelées probabilités d'émission, chacune exprimant la probabilité qu'une observation o soit générée à partir d'un état i
$\pi=\pi_1,\pi_2,\ldots, \pi_N$	une distribution de probabilité initiale sur les états. De plus π_i est la probabilité que la chaîne de Markov commence dans l'état *i*. Certains états j peuvent avoir $\pi_j=0$, ce qui signifie qu'ils ne peuvent pas être des états initiaux.. Aussi $\sum_{i=1}^{n} \pi_i = 1$.

Un modèle de Markov caché (MMC) de premier ordre repose sur deux hypothèses simplificatrices. Tout d'abord, à l'instar d'une chaîne de Markov de premier ordre, la probabilité d'un certain état dépend uniquement de l'état immédiatement précédent :

$$\text{Hypothèse de Markov: } P(q_i|q_1, \dots, q_{i-1}) = P(q_i|q_{i-1}). \tag{1.7}$$

Deuxièmement, la probabilité d'une observation de sortie o_i dépend uniquement de l'état qui a généré l'observation q_i et non d'autres états ou d'autres observations.

L'Indépendance des sorties est define par:

$$P(o_i|q_1, \dots, q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i|q_i). \tag{1.8}$$

Il existe de nombreux phénomènes réels, appelés états, que nous cherchons à modéliser pour expliquer nos observations. Souvent, lorsqu'une séquence de symboles d'observation est donnée, il est nécessaire de découvrir ces états réels. Par exemple, les états météorologiques tels que le soleil, le nuage et la pluie (Fosler-Lussier, 1998). Imaginez que vous êtes à l'intérieur et que vous ignorez le temps qu'il fait dehors, mais que vous recevez des observations comme la vitesse du vent, la pression atmosphérique, l'humidité et la température de la part de quelqu'un d'autre. Sur la base de ces observations, vous pouvez prédire le temps à l'aide d'un modèle de Markov caché (MMC). Avant d'aborder les MMC, nous devrions d'abord passer en revue la définition d'un modèle de Markov (MM). Un MM est un modèle statistique utilisé pour représenter des processus stochastiques. Il est défini comme suit (Schmolze, 2001) : Pour une collection finie d'états S={s_1, s_2,…, s_n} avec une cardinalité n, soit Π la distribution des états initiaux où $\pi_i \in \Pi$ représente la probabilité que le processus stochastique commence dans l'état s_i. En d'autres termes, π_i est la probabilité initiale de l'état s_i, qui satisfait la condition suivante $\sum_{s_i \in S} \pi_i = 1$.

- Le processus stochastique modélisé occupe un seul état parmi S à tout moment. Ce processus stochastique est spécifié comme un vecteur fini X=(x_1,x_2,...,x_T), où chaque élément x_t est un état à l'instant t. Le processus X est connu comme le processus stochastique d'état, avec $x_t \in S$ étant un état $s_i \in S$. X est aussi appelé la séquence d'états. Les intervalles de temps peuvent être mesurés en secondes, minutes, heures, jours, mois ou années. Il est simple de conclure que la probabilité

initiale π_i=P(x_1=s_i), où x_1 est le premier état du processus stochastique. Le processus stochastique d'état X respecte la propriété de Markov. Cela signifie que, étant donné l'état précédent x_{t-1} du processus X la probabilité conditionnelle de l'état actuel x_t dépend uniquement de l'état précédent x_{t-1} et non d'un état antérieur (x_{t-2}, x_{t-3},..., x_1). En d'autres termes, $P(x_t|x_{t-1}, x_{t-2}, x_{t-3}, \dots, x_1) = P(x_t|x_{t-1})$. Ce type de processus est appelé un processus de Markov de premier ordre.

- À chaque instant, le processus passe à l'état suivant en fonction de la distribution de probabilité de transition a_{ij} qui ne dépend que de l'état précédent. Ainsi, a_{ij} est la probabilité que le processus stochastique passe de l'état actuel s_i à l'état suivant s_j. En d'autres termes, a_{ij}=$P\left(x_t = s_j \middle| x_{t-1} = s_i\right) = P\left(x_{t+1} = s_j \middle| x_t = s_i\right)$. Pour tout $s_i \in S$, $\sum_{s_j \in S} a_{ij} = 1$. Toutes les probabilités de transition a_{ij} forment la matrice de probabilité de transition A, qui est une matrice n×n car il y a n états distincts. La matrice A représente le processus stochastique d'état X. La matrice de probabilité initiale *Π* peut être considérée comme un cas dégénéré de la matrice A.

En résumé, un modèle de Markov (MM) est défini par le triplet (S, A, ∏). Dans un MM typique, les états sont directement observés par les utilisateurs, et les probabilités de transition A et Π sont des paramètres uniques. En revanche, un modèle de Markov caché (MMC) est similaire à un MM, sauf que les états sous-jacents sont cachés à l'observateur et traités comme des paramètres cachés. Un MMC incorpore des paramètres de sortie supplémentaires appelés observations. Chaque état (paramètre caché) a une distribution de probabilité conditionnelle associée à ces observations. Un MMC est chargé de découvrir les paramètres cachés (états) à partir des paramètres de sortie (observations), étant donné le processus stochastique. Un MMC a les propriétés suivantes (Schmolze, 2001) :

- Supposons qu'il existe un ensemble fini d'observations possibles Φ = {ϕ_1, ϕ_2,..., ϕ_m} de cardinalité m. Il existe un second processus stochastique qui produit des observations correspondant aux états cachés. Ce processus est appelé le processus stochastique observable et est défini comme un vecteur fini O = (o_1, o_2,.. o_T), où chaque élément O_T est une observation à l'instant t. Note that $O_T \in \Phi$ corresponds to some ϕ_k. Ce paramètre O est souvent appelé la séquence d'observation.

- Il existe une distribution de probabilité pour produire une observation donnée dans chaque état. Soit b_i(k) a probabilité d'observer ϕ_k lorsque le processus stochastique d'état est dans l'état s_i. Cela signifie que $b_i(k) = b_i(o_t = \phi_k) =$

$P(o_t = \phi_k | x_t = s_i)$. La somme des probabilités de toutes les observations dans un état particulier est égale à 1, donc pour tout $s_i \epsilon S, \sum_{\phi_k \epsilon \Phi} b_i(k) = 1$. Toutes les probabilités d'observation $b_i(k)$ forment la matrice de probabilités d'observation B. Il est commode d'utiliser la notation b_{ik} au lieu de $b_i(k)$. Notez que B est une matrice n × m car il y a n états distincts et m observations distinctes. Alors que la matrice A représente les transitions d'état, la matrice B représente le processus stochastique observable O.

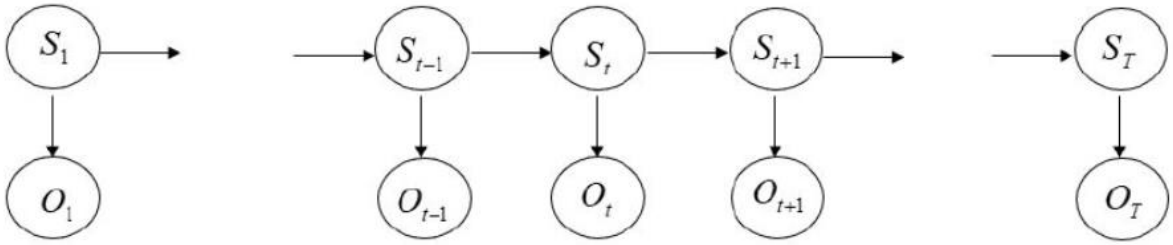

Figure 1.1: La représentation graphique d'un modèle de Markov caché (MMC) de premier ordre consiste en des états S_i représentant l'état caché de Markov et des états O_i représentant les observations à chaque instant t_i, i=1,2,...,T (Rezapoor Nikroo et al., 2022)

1.5 Vraisemblance maximale

1.5.1 Introduction

Une technique clé pour estimer les paramètres d'un modèle statistique dans diverses applications d'apprentissage automatique est l'estimation par maximum de vraisemblance (EMV). Cette méthode est largement utilisée dans les domaines où les données sont considérées comme provenant de processus aléatoires. L'EMV est largement utilisée dans plusieurs disciplines, notamment la biologie, la vision par ordinateur, le traitement du langage naturel et l'analyse d'images (Granger, 2012; Mongillo & Deneve, 2008).

L'objectif principal de la modélisation par équations linéaires multiples, ou EMV, est de maximiser la vraisemblance des données observées sous le modèle. De nombreuses stratégies sophistiquées, telles que les techniques d'estimation récursives et les algorithmes d'espérance-maximisation (EM), reposent sur cette approche. En maximisant le critère de l'EMV, ces méthodes améliorent la précision et la performance du modèle (Cappé, 2011). Maximiser la probabilité, diminuer la mesure moyenne de divergence des paramètres ou réduire l'erreur de pourcentage moyenne de sortie ou d'état sont les objectifs principaux de diverses techniques de mise en œuvre de l'EMV. Malgré sa puissance, l'efficacité de l'EMV peut être affectée par les hypothèses de distribution faites sur les données. Par exemple, la distribution en forme de cloche symétrique peut ne pas

toujours donner des estimations fiables des paramètres lorsqu'on utilise des mélanges gaussiens, en particulier si les données s'écartent des hypothèses gaussiennes (Reynolds, 2015).

D'autres distributions, telles que la distribution de Dirichlet, qui est plus flexible et peut avoir plusieurs modes symétriques et asymétriques, peuvent être envisagées dans certaines situations. Ainsi, pour que l'EMV soit appliquée avec succès dans diverses applications d'apprentissage automatique, il est essentiel de choisir la bonne distribution et la bonne technique d'optimisation (Bouguila, Ziou, & Vaillancourt, 2004; Lin, 2016).

L'estimation par maximum de vraisemblance, l'algorithme d'espérance-maximisation (EM) est la technique la plus couramment utilisée pour estimer les paramètres d'un modèle de Markov caché (MMC) et les paramètres déterminant une distribution. Lorsqu'il s'agit de MMC, l'algorithme EM est appelé algorithme de Baum-Welch. Les probabilités de transition et les paramètres de distribution dépendants de l'état sont deux exemples de paramètres du MMC qui sont estimés de manière itérative à l'aide de cette approche. Un ensemble de paramètres qui maximise la fonction de vraisemblance est l'estimation la plus probable des données échantillonnées. Cette approche présente l'avantage d'augmenter automatiquement la vraisemblance observée à chaque itération. Il a été démontré que l'algorithme EM converge vers un point de log-vraisemblance stationnaire (Cappé, 2011).

L'idée principale de cette méthode est l'estimation des paramètres avec des données manquantes ou incomplètes. Le processus itératif de l'algorithme EM est effectué jusqu'à ce qu'un critère de convergence soit satisfait. Les états non observés de la chaîne de Markov sont remplacés par leurs espérances conditionnelles et traités comme des données manquantes.

L'algorithme EM comporte plusieurs étapes :

1. Définir les paramètres à estimer.
2. L'étape E : calculer les fonctions des espérances conditionnelles des données manquantes.
3. L'étape M'optimiser la log-vraisemblance par rapport aux paramètres à estimer.
4. Évaluer la convergence et itérer les étapes E et M jusqu'à ce que la convergence soit atteinte.

L'algorithme EM n'est pas idéal pour l'adaptation aux flux de données car il nécessite que toutes les données soient fournies à chaque itération. Par conséquent, les techniques d'estimation récursive des paramètres EM gagnent en popularité. Ces techniques récursives permettent une estimation des paramètres en temps réel sans avoir besoin de stocker les données ou de perdre les informations précédemment acquises. La technique d'estimation récursive des paramètres EM considérée dans ce travail met à jour ses paramètres après le traitement d'une nouvelle observation. Nous proposons une nouvelle approche récursive EM pour estimer les paramètres du MMC de Dirichlet (Cappé & Moulines, 2007; Lagona & Barbi, 2019; Heinzl & Tutz, 2013).

Chapitre 2

Estimation et détection de l'occupation des bâtiments

2.1 Introduction

La détection précise de l'occupation est essentielle pour une gestion efficace de l'énergie des bâtiments. Alors que l'efficacité énergétique devient de plus en plus importante, la capacité à estimer et à détecter l'occupation des bâtiments a suscité un large intérêt académique et pratique. Une discussion approfondie des nombreux éléments qui influencent l'estimation et la détection de l'occupation des bâtiments est fournie. Les systèmes existants sont classés systématiquement en fonction des types de capteurs utilisés. Étant donné l'utilisation généralisée des techniques de fusion multi-capteurs dans la détection de l'occupation, les systèmes incluant plusieurs types de capteurs sont également étudiés. Enfin, une étude comparative des différents capteurs utilisés pour les estimations et la détection de l'occupation est présentée, en soulignant leurs avantages et limitations uniques.

Les bâtiments nécessitent plus de 40 % de l'énergie mondiale pour fournir aux habitants un environnement intérieur confortable et sain (D'Oca et al., 2018). En raison de la crise énergétique en cours et de l'accent mis sur le développement durable, les bâtiments économes en énergie attirent davantage l'attention. Cet objectif repose fortement sur des données fiables concernant l'occupation des bâtiments (Yan et al., 2015; Feng et al., 2015). Par exemple, la présence ou l'absence d'habitants détermine les heures de fonctionnement des systèmes CVC (Chauffage, Ventilation et Climatisation) dans les bâtiments (Hong & Lin, 2013; Oldewurtel et al., 2013). De plus, le nombre d'habitants dans une zone thermique influence la charge de refroidissement ou de chauffage dans cette région. Plusieurs systèmes de contrôle de la température basés sur l'occupation ont été proposés pour économiser de l'énergie tout en maintenant le confort.

Agarwal et al. (2010) ont développé un système à faible coût pour détecter avec précision la présence et l'absence de personnes dans les lieux de travail, entraînant une économie d'énergie de 10 % à 15 % selon les données d'occupation. Yang et Becerik-Gerber (2014) ont démontré que l'utilisation de la planification CVC basée sur l'occupation pourrait permettre de réaliser jusqu'à 9 % d'économie d'électricité. Wang et al. (2017) ont introduit une méthode de contrôle prédictif pour la gestion de l'environnement des

bâtiments basée sur les données d'occupation, qui a réduit la consommation d'énergie d'environ 40 % tout en maintenant le confort thermique et la qualité de l'air. Les informations d'occupation influencent le fonctionnement des systèmes d'éclairage (Nguyen & Aiello, 2013; Candanedo & Feldheim, 2016). Leephakpreeda (2005) a estimé que la gestion de l'éclairage basée sur l'occupation pourrait permettre d'économiser de 35 % à 75 % d'énergie. Guo et al. (2010) fournissent un aperçu détaillé des systèmes de contrôle de l'éclairage basés sur l'occupation. De plus, la présence d'occupants est cruciale pour la sécurité des bâtiments et la planification des évacuations d'urgence.

Selon l'application, de nombreux capteurs sont utilisés pour estimer et détecter correctement l'occupation d'un bâtiment. Par exemple, les capteurs infrarouges passifs (PIR) peuvent détecter la présence et l'absence de personnel dans un bureau (Dodier et al., 2006). Les caméras peuvent compter le nombre d'individus dans un espace (Zou, H., et al., 2017), tandis que les capteurs de CO2 peuvent offrir des estimations approximatives de l'occupation (Jiang et al., 2016). Chaque type de capteur a des caractéristiques et des limites distinctes pour les estimations et la détection de l'occupation. Par conséquent, cette section catégorise plusieurs stratégies d'estimation et de détection de l'occupation en fonction du type de capteur. Compte tenu de l'intérêt marqué des dernières décennies pour l'estimation et la détection de l'occupation des bâtiments, ainsi que du manque d'évaluations complètes comparant différents systèmes, cette section vise à présenter un aperçu qualitatif.

Dans ce contexte, l'estimation de l'occupation fait référence à l'estimation du nombre ou de l'étendue des habitants, tandis que la détection de l'occupation se réfère à la détermination de la présence ou de l'absence de personnes dans un lieu spécifique.

2.2 Estimation et détection de l'occupation avec différents types de capteurs

1. **Capteurs PIR**: Les capteurs PIR détectent les changements de rayonnement infrarouge induits par le mouvement des individus, ce qui leur permet de détecter les mouvements des occupants et d'indiquer leur présence. Plusieurs systèmes de détection d'occupation reposent sur des capteurs PIR.

Liu et al. (2016) ont introduit un système de détection d'occupation basé sur des capteurs PIR. Ils ont utilisé un modèle de Markov caché (MMC) avec une espérance maximisation pour identifier l'occupation.

Les capteurs PIR peuvent également déterminer le nombre d'habitants. Wahl et al. (2012) ont développé un système de comptage d'occupation qui utilise des paires de capteurs PIR pour déterminer les directions de mouvement des occupants.

Raykov et al. (2016) ont utilisé un seul capteur PIR pour estimer le nombre de personnes dans une pièce. Ils ont commencé par extraire des motifs de mouvement des occupants à partir des données brutes du capteur à l'aide d'un modèle MMC infini. Ces motifs ont ensuite été utilisés pour calculer l'état d'occupation à l'aide de modèles de régression statistique.

2. **Compteurs intelligents**: Les compteurs intelligents peuvent suivre la consommation d'énergie et refléter les interactions entre les habitants et les équipements intérieurs, permettant ainsi de déduire l'occupation des bâtiments. Cependant, le lien entre l'occupation et les statistiques de consommation d'énergie est complexe et difficile à modéliser. Les algorithmes d'apprentissage automatique sont idéaux pour simuler cette interaction complexe.

Kleiminger et al. (2013) ont utilisé des compteurs d'électricité comme capteurs d'occupation pour la détection d'occupation. Étant donné que les données brutes de consommation électrique n'indiquent pas explicitement la présence ou l'absence d'occupants, ils ont créé des caractéristiques plus représentatives. Ils ont utilisé des classificateurs tels que la machine à vecteurs de support (SVM), les K plus proches voisins (KNN), le modèle de Markov caché (MMC) et le seuillage pour la classification. Dans un travail ultérieur, Kleiminger et al. (2015) ont cherché à améliorer les performances en utilisant un ensemble de données plus important.

Becker et Kleiminger (2017) ont proposé un système de détection d'occupation basé sur les données des compteurs intelligents et les techniques d'apprentissage non supervisé. Ils ont évalué trois algorithmes d'apprentissage non supervisé : MMC, GeoMA et le test de Page-Hinkley (PHT), et les ont comparés à plusieurs méthodes d'apprentissage supervisé.

3. **Capteurs environnementaux**: Les capteurs de CO2, de température, d'humidité, de lumière et de pression sont fréquemment intégrés dans les systèmes modernes LH-VAC (éclairage, chauffage, ventilation et climatisation) des bâtiments (Dong et al., 2010). Comme les habitants ont une influence directe sur les environnements intérieurs, les données de ces capteurs peuvent être utilisées pour mesurer avec précision l'occupation des bâtiments. Plusieurs études de recherche avancées ont utilisé des capteurs environnementaux pour la prédiction et la détection de l'occupation.

Les capteurs de CO2 se sont révélés les plus efficaces pour l'estimation et la détection de l'occupation (Dong et al., 2010; Candanedoet al., 2017). Certaines études dépendent principalement des données de CO2 pour l'estimation et la détection de l'occupation,

tandis que d'autres utilisent divers capteurs environnementaux pour améliorer la precision.

3.1 **Capteurs de CO2**: Ansanay-Alex (2013) a créé un système de détection d'occupation de base qui détecte la présence et l'absence d'habitants en mesurant les concentrations de CO2. Cette approche est simple mais limitée à la détection de l'occupation sans déterminer le nombre de personnes. Les niveaux de CO2 intérieurs sont corrélés au nombre d'habitants et au taux de ventilation; par conséquent, cette relation peut être utilisée pour développer des méthodes de détection plus dynamiques.

Wang et Jin (1998) ont proposé une approche dynamique de détection de l'occupation en utilisant l'équation de bilan massique pour les concentrations de CO2 à l'intérieur. Ils l'ont comparée à une approche de détection à l'état stationnaire précédemment suggérée (ASHRAE, 1996). Cali et al. (2015) ont utilisé la même technique de détection dynamique pour améliorer la précision de la détection d'occupation dans divers environnements, y compris des pièces avec ou sans ventilation.

Un inconvénient notable de l'utilisation des capteurs de CO2 pour l'estimation et la détection de l'occupation est le retard temporel inhérent aux données. Ce retard est causé par la lente diffusion du CO2 dans les espaces intérieurs, entraînant un décalage entre les changements réels d'occupation et les niveaux de concentration de CO2 observés.

3.2 **Capteurs d'humidité**: Selon les recherches, les capteurs d'humidité peuvent identifier avec succès l'occupation dans divers scénarios. Naqi et al. (2019) ont créé un appareil polyvalent basé sur des capteurs de température et d'humidité pour détecter l'occupation dans les environnements de bureaux intelligents. Candanedo et al. (2016) ont prédit avec succès l'occupation dans un bureau en utilisant des modèles d'apprentissage statistique et des données provenant de capteurs de lumière, de température, d'humidité et de CO2. Szczurek et al. (2017) ont démontré que les données chronologiques sur la concentration de CO2, la température et l'humidité relative peuvent être utilisées pour estimer le nombre d'habitants dans des zones occasionnellement occupées. Yang et al. (2012) ont introduit une méthodologie d'estimation de l'occupation par multi-capteurs, incluant des capteurs d'humidité, pour des opérations CVC (chauffage, ventilation et climatisation) à la demande qui atteignent des taux de détection élevés lors de la surveillance en temps réel. Ces études démontrent le potentiel des capteurs d'humidité dans la détection de l'occupation.

3.3 **Capteurs de lumière**: Plusieurs études de recherche ont examiné l'utilisation des capteurs de lumière pour la détection de l'occupation dans les environnements de bureau. Tiller et al. (2010) et Guo et al. (2010) soulignent le potentiel des réseaux de capteurs pour réduire l'incertitude et améliorer le contrôle de l'éclairage, Tiller et al. (2010) mettant en avant l'importance des approches analytiques avancées. Wang et al. (2017) présentent une technique unique de contrôle de l'éclairage qui prend en compte la fiabilité des capteurs d'occupation, entraînant des économies d'énergie considérables. Yang et al. (2017) décrivent un système d'inférence d'occupation sans appareil qui utilise des luminaires à LED comme capteurs pour atteindre une haute précision dans la détection de l'occupation. Ces études mettent en évidence le potentiel des capteurs de lumière pour améliorer l'efficacité énergétique et la gestion des bâtiments.

3.4 **Capteurs de pression**: Plusieurs recherches ont étudié l'utilisation de différentes technologies de capteurs pour la détection de l'occupation, y compris les capteurs de pression. Szczurek et al. (2017) et Lam et al. (2009) ont démontré une grande précision dans la détection de l'occupation en utilisant respectivement des capteurs de gaz et environnementaux. Ekwevugbe et al. (2012) ont proposé un modèle de fusion de capteurs pour détecter l'occupation des bâtiments en combinant les données de plusieurs types de capteurs. Dodier et al. (2006) ont utilisé des réseaux de croyance des capteurs pour calculer l'occupation. Ces expériences démontrent le potentiel des capteurs de pression dans la détection de l'occupation et soulignent la nécessité de poursuivre les recherches pour améliorer leur utilisation dans ce context.

3.5 **Capteurs multiples**: En plus des capteurs de CO2, divers capteurs environnementaux, notamment la température, l'humidité, la lumière et la pression, sont utiles pour détecter et estimer l'occupation des bâtiments. Candanedo et Feldheim (2016) ont proposé un système de détection d'occupation qui utilise des capteurs environnementaux pour le CO2, la lumière, la température et l'humidité. Ils ont utilisé trois algorithmes d'apprentissage : l'analyse discriminante linéaire (LDA), les arbres de classification et de régression (CART) et la forêt aléatoire (RF) pour la detection.

Candanedo et al. (2017) ont testé la méthode MMC pour la détection de l'occupation avec le même ensemble de capteurs. Ils ont évalué le MMC avec diverses combinaisons de caractéristiques et ont constaté que le MMC avec les caractéristiques de CO2 offrait la meilleure performance en termes de détection. La caractéristique de Markov de la dynamique de l'occupation des bâtiments (Wang et al., 2011; Chen et al., 2015) a rendu le modèle MMC utile pour les estimations et la détection de l'occupation (Dong et al.,

2010; Candanedo et al., 2017). Cependant, le MMC traditionnel présente des limites, notamment une matrice de probabilité de transition fixe et un mélange gaussien de base pour les probabilités d'émission. Pour répondre à ces préoccupations, Chen et al. (2017) ont proposé un MMC inhomogène avec une régression logistique multinomiale (IMMC-MVR) pour estimer l'occupation des bâtiments à l'aide de capteurs de CO2, de température, d'humidité et de pression.

4. **Caméras:** En raison de leur grande précision, les caméras sont couramment utilisées pour estimer et détecter l'occupation des bâtiments. Fleuret et al. (2008) ont utilisé un système multi-caméras pour prédire les positions de toutes les personnes à l'intérieur, ce qui a permis d'obtenir un comptage précis de l'occupation. Benezeth et al. (2011) ont proposé un système d'estimation et de détection de l'occupation basé sur la vision qui comprend trois étapes : la suppression de l'arrière-plan, le suivi et l'identification. Erickson et al. (2011) ont utilisé des caméras pour collecter des données sur l'occupation des bâtiments, qui ont ensuite été utilisées pour créer un modèle d'occupation. Ce modèle a ensuite été intégré aux systèmes de contrôle des bâtiments pour accroître l'efficacité énergétique.
   - **4.1 Capteurs de mouvement:** Plusieurs recherches ont étudié l'utilisation de capteurs de mouvement pour la détection de l'occupation. Jiang et al. (2018) ont proposé un système qui combine les données des capteurs de dioxyde de carbone et de Pyroélectrique Infrarouge (PIR) pour réduire les retards inhérents aux capteurs de CO2, obtenant une précision allant jusqu'à 99,79 %. Andrews et al. (2020) ont créé un capteur PIR induit par le mouvement (MI-PIR) qui détecte les habitants stationnaires avec une grande précision dans la catégorisation des pièces, l'estimation du nombre d'occupants, la prédiction de la position relative et la discrimination des cibles humaines. Liu et al. (2013) ont développé un système de mesure de l'occupation via la vidéosurveillance qui combine des détecteurs statiques avec des technologies basées sur le mouvement pour garantir des estimations fiables.
5. **WiFi:** Les signaux WiFi sont largement disponibles dans les espaces intérieurs, ce qui en fait un outil viable pour l'estimation et la détection de l'occupation. Avec l'utilisation généralisée des smartphones, le nombre d'appareils connectés à un réseau WiFi peut être utilisé pour estimer le nombre de personnes dans un lieu spécifique.

Balaji et al. (2013) ont introduit une méthode d'estimation de l'occupation qui utilise l'infrastructure WiFi existante. Plutôt que d'utiliser des algorithmes de localisation basés

sur le WiFi complexes, ils ont conçu une technique grossière basée sur les informations de métadonnées et les modèles d'occupation des points d'accès associés. Lu et al. (2016) ont proposé un système d'inférence d'occupation au niveau des pièces basé sur le WiFi standard. Leur approche a initialement créé un instantané de l'occupation en comparant les mesures de la puissance du signal reçu (RSS) à celles des ancrages dans différentes zones. Les performances d'estimation du système ont ensuite été améliorées en exploitant les corrélations temporelles entre les instantanés d'occ

upation successifs et les données d'occupation précédentes. Zou, J., et al. (2017) ont développé un système de prédiction de l'occupation en utilisant une approche de localisation intérieure basée sur le WiFi. Plutôt que de collecter des données RSS WiFi à partir de smartphones, les chercheurs ont utilisé des routeurs pour rechercher des téléphones portables compatibles WiFi dans des environnements intérieurs.

6. **Bluetooth Basse Énergie**: Bluetooth Basse Énergie (BBÉ) est couramment utilisé dans les smartphones actuels et est plus économe en énergie que les technologies Bluetooth traditionnelles et le WiFi. Les chercheurs utilisent le Bluetooth Basse Énergie (BBÉ) pour estimer le nombre de téléphones portables et, par conséquent, le nombre d'habitants à l'intérieur.

Conte et al. (2014) ont proposé un système d'estimation de l'occupation basé sur les technologies iBeacon et BLE. Ils ont modifié le protocole iBeacon pour améliorer sa précision et son efficacité pour la détection de l'occupation. Deux algorithmes, KNN et arbre de décision (DT), ont été utilisés pour classer les habitants dans différentes pièces en fonction des indicateurs de puissance du signal (RSSI) collectés à partir de divers iBeacons. Dans une enquête ultérieure, Corna et al. (2015) ont amélioré la précision de l'estimation de l'occupation avec iBeacon.

7. **Autres capteurs**:Plusieurs capteurs différents ont été utilisés pour l'estimation et la détection de l'occupation. Labeodan et al. (2015) ont proposé une méthode de prédiction de l'occupation fine pour une salle de conférence en utilisant des capteurs de chaise. Une simple approche de seuillage était suffisante pour déterminer si les sièges étaient occupés ou non, et l'occupation totale a été calculée en comptant les chaises occupées.

Pan et al. (2014) ont recommandé l'utilisation de capteurs de vibration ambiante placés de manière espacée pour estimer l'occupation dans les bâtiments. Ils ont d'abord reconnu les événements de pas à partir de données de capteurs de vibration bruitées, puis ont utilisé un module de localisation pour identifier les pas, et ont ensuite suivi les pas localisés avec des informations structurelles pour mettre à jour l'occupation. Yang et al.

(2017) ont proposé un système d'inférence d'occupation basé sur des capteurs à diodes électroluminescentes (LED). La théorie était que les habitants dans le champ de vision des tableaux de LED montés au plafond induiraient des perturbations dans les réflexions diffuses sur le sol, qui étaient collectées sous forme d'images. À partir de ces images, ils ont appliqué une technique de régression par vecteurs de support (SVR) pour estimer l'occupation. Quan et al. (2014) ont proposé un système de comptage de l'occupation utilisant des capteurs ultra-large bande à impulsions radio (IR-UWB). Un système de seuillage simple a reconnu le nombre de passagers passant par une porte à l'aide de données IR-UWB, et l'occupation de la pièce a été calculée. Une autre recherche de Bartoletti et al. (2017) a développé un cadre théorique pour le comptage de l'occupation basé sur la rétro-diffusion de cibles UWB. Les auteurs ont créé deux approches pour l'estimation de l'occupation : une centrée sur l'individu et une centrée sur la foule. La technique centrée sur l'individu utilisait un algorithme de maximum a posteriori pour le comptage et la sélection de l'ordre du modèle, tandis que l'approche centrée sur la foule reposait sur des caractéristiques de bas niveau pour améliorer l'efficacité du calcul.

8. **Fusion de capteurs**: Chaque capteur a des caractéristiques et des contraintes distinctes pour l'estimation et la détection de l'occupation. La combinaison de plusieurs types de capteurs peut améliorer les performances en utilisant leurs forces tout en corrigeant leurs faiblesses.

Zhang et al. (2016) ont proposé de calculer l'occupation des bâtiments en combinant des données de consommation de CO2 et d'énergie à l'aide de la modélisation statistique. Ils ont d'abord récupéré les caractéristiques du CO2 à l'aide d'un modèle de diffusion physique. Ensuite, ils ont utilisé des modèles de distribution statistique, tels que l'estimation de la probabilité a posteriori maximale (MAP) de MMC et le test de rapport de probabilité séquentiel à hypothèses multiples (MSPRT), pour estimer l'occupation avec peu de données d'entraînement. Wang et al. (2014) ont proposé un réseau de neurones à impulsions (SNN) pour l'estimation de l'occupation par fusion de capteurs. Ils ont utilisé cinq capteurs distincts, dont le CO2, l'humidité, le PIR, le RFID et le WiFi. Le modèle SNN a été entraîné à l'aide d'une méthode Hebbienne non supervisée.

Ai et al. (2014) ont développé une méthode MMC auto-régressive pour estimer l'occupation des bâtiments en combinant PIR, CO2, température, humidité, vitesse de l'air, thermomètre global et interrupteurs à lames. Dans leurs essais, le MMC auto-régressif surpasse les MMC traditionnels car il prend en compte les corrélations des capteurs. Lam et al. (2009) ont préconisé de calculer l'occupation avec une gamme de capteurs, y compris le CO2, la température, l'humidité, la lumière, le monoxyde de carbone (CO), les composés organiques volatils totaux (TVOC), les particules fines

(PM2.5), l'acoustique et le mouvement. Tout d'abord, ils ont extrait des caractéristiques statistiques des données brutes. Ensuite, ils ont utilisé la théorie du gain d'information pour identifier les caractéristiques les plus informatives pour l'estimation. Enfin, ils ont utilisé des méthodes d'apprentissage automatique telles que les réseaux de neurones artificiels (ANN), les machines à vecteurs de support (SVM) et les modèles de Markov cachés (MMC) pour déterminer l'occupation.

9. **Capteur d'ouverture de porte**: Plusieurs recherches ont étudié l'utilisation des capteurs d'ouverture de porte pour la détection de l'occupation. Kim (2017) a découvert que l'intégration des capteurs PIR et des capteurs de porte peut améliorer considérablement la précision, obtenant un taux de détection de 99,8 % avec un algorithme d'occupation. Nasir et al. (2015) ont souligné l'importance de la fusion de capteurs pour la surveillance en temps réel de l'occupation et ont démontré un système de porte intelligente évolutif. Ang et al. (2016) ont exploré quel capteur ambiant était le plus efficace pour détecter l'occupation. Borgsen et al. (2014) ont proposé une technique de détection automatique des portes basée sur des capteurs 3D, qui a le potentiel d'améliorer les capacités des capteurs d'ouverture de porte pour la détection de l'occupation.
10. **Capteurs de consommation d'énergie**: De nombreuses recherches ont étudié l'utilisation des statistiques de consommation d'énergie pour déterminer l'occupation. Kleiminger et al. (2013) et Akbar et al. (2015) ont tous deux obtenu une excellente précision dans la détection de l'occupation avec des compteurs d'énergie numériques, Akbar ajoutant un troisième mode "veille". Gao et al. (2018) ont amélioré ces algorithmes en combinant des données de consommation d'énergie agrégées et spécifiques à l'appareil, atteignant une précision allant jusqu'à 93 %. Mikkilineni et al. (2019) ont proposé un capteur d'occupation sans fil basé sur des imageurs thermiques, indiquant une alternative viable aux capteurs de consommation d'énergie. Ces études montrent collectivement que les capteurs de consommation d'énergie ont la capacité de détecter l'occupation avec précision et à moindre coût.

2.3 Discussion

Tableau 2.1: Comparaison des différents capteurs (Chen et al., 2018).

Capteur	Infrastructure existante	Coût	Précision de détection	Précision de l'estimation	Problème de confidentialité	Limitations
PIR	Non	Bas	Élevé	Bas	Non	Manque les occupants statiques
Smart meter	Oui	Non	Moyen	N.D.	Non	Manque les occupants qui n'utilisent pas d'appareil
Environnemental	Oui	Non	Moyen	Bas	Non	Détection retardée
Caméra	Oui	Non	Élevé	Élevé	Oui	Condition d'illumination
WiFi	Oui	Non	Élevé	Élevé	Partiel	Nécessité d'activer le WiFi
BLE	Non	Moyen	Élevé	Élevé	Partiel	Nécessité d'activer le Bluetooth
Capteur de chaise	Non	Élevé	Élevé	Élevé	Non	Manque les sujets debout
Vibration	Non	Élevé	Élevé	Moyen	Non	Évolutivité : maintenance
DEL	Oui	Non	Élevé	Moyen	Non	Maintenance
RFID	Non	Moyen	Élevé	Élevé	Partiel	Inconvénient : manque les visiteurs
Zigbee	Non	Moyen	Élevé	Élevé	Partiel	Inconvénient; manque les visiteurs
UWB	Non	Élevé	Élevé	Bas	Non	Mauvaise performance dans les environements encombrés

Chen et al. (2018) ont examiné tous les capteurs indiqués dans le Tableau 2.1 en utilisant une variété de paramètres d'évaluation, y compris l'infrastructure actuelle, le coût, les préoccupations en matière de confidentialité, la précision de détection, la précision d'estimation et les contraintes. Ces mesures d'évaluation sont essentielles pour que les praticiens sélectionnent la meilleure solution en fonction de leurs besoins spécifiques. Si le type de capteur spécifié fait déjà partie de l'infrastructure actuelle d'un bâtiment, la solution n'entraînera aucun coût supplémentaire. Cependant, si de nouveaux capteurs sont nécessaires, le coût peut être un sujet de préoccupation majeur dans les applications réelles.

La précision de la détection ou de l'estimation de l'occupation, également connue sous le nom de précision de détection ou d'estimation, est cruciale car elle influence la performance globale du système, tels que les systèmes de gestion de l'éclairage et de la climatisation basés sur l'occupation (Agarwal et al., 2010; Leephakpreeda, 2005). Un autre problème critique à résoudre est celui des questions de confidentialité, qui ont été abordées dans Jia et al. (2017), où une architecture améliorée pour la confidentialité en gestion de la climatisation basée sur l'occupation a été conçue. Chaque type de capteur a des avantages et des inconvénients uniques en ce qui concerne l'estimation et la détection de l'occupation. Pour améliorer les performances, combiner plusieurs types de capteurs est une solution excellente car les divers capteurs peuvent compenser les limites des uns et des autres.

Il existe plusieurs techniques d'estimation et de détection de l'occupation pour chaque type de capteur. Cependant, comparer diverses algorithmes peut souvent être difficile. Pour permettre la comparaison des méthodes dans les domaines de la vision par ordinateur et de la surveillance non intrusive des charges, des ensembles de données de référence tels qu'Imagenet (Deng et al., 2009) et BLUED (Filip, 2011) ont été fournis. Les ensembles de données populaires pour la prédiction et la détection de l'occupation dans les applications basées sur les compteurs intelligents sont l'ECO (Kleiminger et al., 2013) et SMART* (Chen et al., 2013). Ces ensembles de données de référence permettent l'évaluation et la comparaison de diverses techniques d'estimation et de détection de l'occupation.

2.4 Développement d'un modèle de Markov caché pour la prédiction de l'occupation

Le modèle de Markov caché (MMC) est une technique statistique populaire pour la modélisation des données, la détection de motifs et la classification (Depatla et al., 2015). Ces dernières années, le MMC a prouvé sa capacité à prévoir une variété de types de données, y compris les chiffres d'occupation et la consommation d'énergie dans les

bâtiments. Des recherches antérieures ont développé le MMC de manière approfondie, démontrant sa grande précision de prédiction.

Le MMC fonctionne en considérant l'état actuel, qui dépend de l'état précédent, tout en gardant l'espace des états caché pour l'observateur. La probabilité de transition et la probabilité d'émission du modèle sont également cachées. Le MMC convertit les données d'entrée séquentielles en un score pour les séquences futures, qui constitue la sortie du modèle. La Figure 2.1 illustre la structure d'un MMC, y compris les états cachés et les observations émises par chaque état caché selon la distribution de probabilité d'émission (Baum & Eagon, 1967; Momenzadeh et al., 2019).

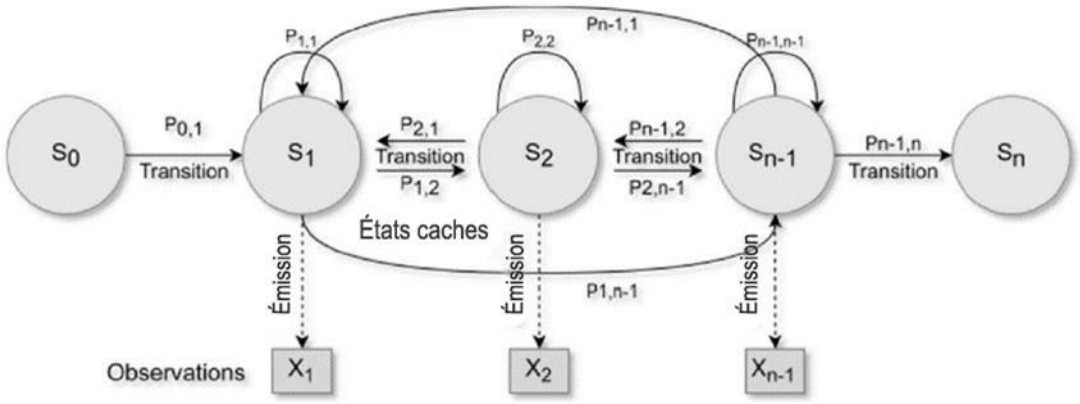

Figure 2.1: La structure du MMC identifie les états cachés (Sn), la probabilité des transitions entre les états (P), et les observations (Xn) (Alfalah, Shahrestani, & Shao, 2023).

2.5 Application du modèle de Markov caché (MMC) dans la prédiction de l'occupation

Candanedo et al. (2017) ont utilisé le MMC pour estimer le calendrier d'occupation dans une structure résidentielle. Le modèle a été créé en utilisant des données obtenues à partir de nombreux capteurs, y compris des capteurs de température, d'humidité, de dioxyde de carbone, de lumière et d'état d'occupation. Les statistiques d'occupation ont été recueillies sur une période d'un mois, en éliminant les jours où le bâtiment était vide. Le programme a prédit avec précision les schémas d'occupation typiques pour les sept jours suivants (Candanedo et al., 2017).

Dans une autre étude, Ruy et Moon (2016) ont employé le MMC pour prévoir le nombre d'occupants dans le banc d'essai de contrôle intégré des bâtiments de l'Université de Dankook (BICT). Sur une période de sept jours, des données ont été recueillies sur les schémas d'occupation, les niveaux de dioxyde de carbone, la température, la

consommation de lumière et l'utilisation des appareils. L'algorithme a réussi à prévoir les niveaux d'occupation de la semaine suivante (Ryu & Moon, 2016).

L'algorithme s'est révélé très précis pour prévoir l'état d'occupation pour les deux heures suivantes (Liu et al., 2016).

Lam et al. (2009) ont utilisé le MMC pour prévoir les taux d'occupation dans un bureau en espace ouvert. Plusieurs capteurs, y compris des capteurs de température, de dioxyde de carbone, d'humidité et de mouvement, ont collecté des données à intervalles de 1 minute pendant 55 jours. Le modèle a prédit les taux d'occupation hebdomadaires avec une précision moyenne de 68 % (Lam et al., 2009).

Plusieurs recherches ont créé des modèles de réseaux neuronaux artificiels (ANN) pour la prédiction de l'occupation des bâtiments. Zuraimi et al. (2017) ont développé un modèle ANN pour prévoir le nombre de personnes dans un amphithéâtre. Les données d'occupation, acquises sur 80 jours ouvrables à l'aide d'une caméra et de capteurs de dioxyde de carbone, ont été comptées manuellement à partir de photographies de la caméra. Le modèle a prédit le nombre d'occupants pour les cinq jours ouvrables suivants et a démontré une excellente capacité prédictive mesurée par l'erreur quadratique moyenne (RMSE) (Zuraimi et al., 2017).

Ekwevugbe et al. (2013) ont utilisé un modèle ANN pour prévoir les taux d'occupation dans un bureau en espace ouvert. Au cours d'un mois, six habitants ont été détectés à l'aide de données provenant de capteurs PIR, de capteurs de dioxyde de carbone, de capteurs de température, de détection sonore et de caméras. Le modèle a anticipé les taux d'occupation pour les 10 jours suivants (Ekwevugbe et al., 2013).

Ashouri et al. (2019) ont utilisé des signaux Wi-Fi pour collecter des données d'occupation d'un immeuble de bureaux et entraîner un modèle ANN. Les données ont été recueillies sur deux jours pour prévoir les niveaux d'occupation du lendemain. Le modèle a prédit les nombres d'occupants avec une précision de 90 % pendant les jours ouvrables (Ashouri et al., 2019).

Alam et al. (2017) ont utilisé des données de capteurs de dioxyde de carbone d'un seul lieu de travail pour créer un modèle de réseau neuronal artificiel (ANN) qui prédisait le nombre d'occupants le jour suivant. Le modèle a montré d'excellentes performances prédictives (Alam et al., 2017).

Chapitre 3

Méthode récursive de SSD

Dans cette section, nous présentons l'approche pour estimer de manière récursive les paramètres du MMC généralisé de Dirichlet en maximisant la fonction de log-vraisemblance donnée comme suit:

$$\text{SSD}(X|U) = p(x_1,x_2,\ldots, x_D) = \frac{\Gamma(\alpha_0)}{\prod_{l=1}^{n}\Gamma(\alpha_l)}\frac{1}{\boldsymbol{\Delta}^{n-1}}\frac{\prod_{l=1}^{D}\beta_l^{\frac{-\alpha_l}{\Delta}\times\left(\frac{\alpha_l}{\Delta}-1\right)}}{\left(\sum_{l=1}^{D}\left(\frac{X_l}{\beta_l}\right)^{\frac{1}{\Delta}}\right)^{\alpha_0}} \tag{3.1}$$

où

$U = (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \Delta)$ est le vecteur des paramètres positifs et

$$\alpha_0 = \sum_{l=1}^{D}\alpha_l,$$

xi > 0, i = 1...D.

Nous allons calculé la log- vraisemblance

$$\text{Log SSD}(X|U) = \log(\Gamma(\alpha_0)) - \log\prod_{l=1}^{D}\Gamma(\alpha_l) - \log\Delta^{D-1} + \log\prod_{l=1}^{D}\beta_l^{\frac{-\alpha_l}{\Delta_s}\times\left(\frac{\alpha_l}{\Delta}-1\right)} - \log\left(\sum_{l=1}^{D}\left(\frac{X_l}{\beta_l}\right)^{\frac{1}{\Delta}}\right)^{\alpha_0}. \tag{3.2}$$

Ceci peut être simplifié comme suit et D'équation finale est:

$$\log SSD(X|U) = \log\Gamma\left(\sum_{i=1}^{D}\alpha_i\right) - \log\prod_{i=1}^{D}\Gamma(\alpha_i) - (D-1)\log\Delta$$

$$-\frac{1}{\Delta}\sum_{i=1}^{D}\alpha_i\log\beta_i + \frac{1}{\Delta}\sum_{i=1}^{D}\alpha_i\log x_i - n\log x_i - \frac{1}{\Delta}\left(\sum_{i=1}^{D}\alpha_i\right)\log\sum_{i=1}^{D}\frac{x_i}{\beta_i}. \quad (3.3)$$

En considérant maintenant le MMC, la densité de probabilité logarithmique de l'observation selon chaque état du MMC est:

$$L(\pi; U)=\log[\sum_{q=1}^{N}\pi_q\, SSD(X|\ U)], \quad (3.4)$$

où π_q est la probabilité d'être dans l'état q

Comme mentionné ci-dessus, le maximum de la fonction de log-vraisemblance (3.3) doit être calculé, puis les dérivées partielles de l'expression par rapport à α_i et β_i et Δ , 1≤i≤D doivent être dérivées comme suit.

$$\frac{\partial\log SSD(X|U)}{\partial\alpha_j} = \psi\left(\sum_{i=1}^{n}\alpha_i\right) - \psi(\alpha_j) - \frac{1}{\Delta}\log\beta_j + \frac{1}{\Delta}\log x_j - \frac{1}{\Delta}\log\sum_{i=1}^{n}\frac{x_i}{\beta_i}. \quad (3.5)$$

$$\frac{\partial\log SSD(X|U)}{\partial\beta_j} = -\frac{1}{\Delta}\frac{\alpha_j}{\beta_j} + \frac{1}{\Delta}\left(\sum_{i=1}^{D}\alpha_i\right)\frac{x_j}{(\beta_j)^2}. \quad (3.6)$$

$$\frac{\partial\log SSD(X|U)}{\partial\Delta} = \frac{-(D+1)}{\Delta} + \frac{1}{\Delta^2}\sum_{i=1}^{D}\alpha_i\log\beta_i - \frac{1}{\Delta^2}\sum_{i=1}^{D}\alpha_i\log x_i + \frac{1}{\Delta^2}\left(\sum_{i=1}^{D}\alpha_i\right)\log\sum_{i=1}^{D}\frac{x_i}{\beta_i}. \quad (3.7)$$

ψ(x) est la fonction digamma, qui est définie comme la dérivée logarithmique de la fonction Gamma. Maintenant, la probabilité que le système soit dans l'état spécifique q à l'instant t étant donné l'observation x est:

$$[\ \pi_t^{<q>}\ \log SSD(x_t|\ U^{<q>})]\ \ /\ [\ \sum^{N}_{j=1} \pi_t^{<j>} \log SSD(x_t|\ U^{<j>})]\ , \qquad (3.8)$$

où

1≤q≤N et N est le nombre d'états.

$\pi_t^{<q>}$ est La probabilité d'être dans l'état q au temps t.

Par conséquent, le nombre total de fois qu'il y a une transition depuis l'état q est:

$$\sum^{T}_{t=1} \pi_t^{<q>}\ \log SSD(x_t|\ U^{<q>})\ /\ \sum^{N}_{j=1} \pi_t^{<j>} \log SSD\ (x_t|\ U^{<j>}), \qquad (3.9)$$

où

1≤q≤N et N est le nombre d'états.

Parce que la probabilité totale de chaque séquence d'observations est calculée en additionnant toutes les séquences d'états possibles, chaque observation est attribuée à chaque état en proportion de la probabilité que le modèle soit dans cet état lorsque le vecteur a été observé. Ainsi, les paramètres de la densité de probabilité du MMC peuvent être ré-estimés en sommant ces moyennes pondérées.

Sur la base de l'explication de la vraisemblance complète de chaque séquence d'observation chez Vaičiulytė et Sakalauskas (2020), les moyennes pondérées sont appliquées. Par conséquent, la formule par lots de α et β en considérant l'état spécifique q et selon les équations (3.7), (3.8) et (3.9) est:

$$\alpha_i = \psi^{-1} \sum \psi\left(\sum_{i=1}^{D} \alpha_i\right) - \frac{1}{\Delta} \log \beta_j \qquad (3.10)$$

$$+\frac{1}{\boldsymbol{\Delta}} \frac{\frac{1}{T}\sum_{t=1}^{T} \frac{Ln\left(x_t^{(s)} \prod_t^q \mathrm{logSSD(xt|}\ U\ < \mathrm{q} >)\right)}{\sum_{j=1}^{N} \prod_t^j \mathrm{logSSD(xt|}U\ < \mathrm{j} >)}}{\frac{1}{T}\sum_{t=1}^{T} \frac{\prod_t^q \mathrm{logSSD(xt|}\ U\ < \mathrm{j} >)}{\sum_{j=1}^{N} \prod_t^j \mathrm{logSSD(xt|}\ U\ < \mathrm{q} >)}} - \frac{1}{T}\sum_{t=1}^{T} \frac{\prod_t^q \frac{1}{\beta_i} log\ \mathrm{SSD(xt|}\ U\ < \mathrm{q} >)}{\sum_{j=1}^{N} \prod_t^j \frac{1}{\beta_i} log\ \mathrm{SSD(xt|}Uj)}$$

$$\beta_j = \frac{1}{\alpha_j}\left(\sum_{j=1}^{D} \alpha_j\right) x_j \ . \tag{3.11}$$

$$\Delta = \frac{1}{D+1}\left[\sum_{i=1}^{D} \alpha_i \log \beta_i + \left(\sum_{i=1}^{D} \alpha_i\right)\left(\log \sum_{i=1}^{D} \frac{x_i}{\beta_i}\right) - \sum_{i=1}^{D} \alpha_i \log x_i\right] \ , \tag{3.12}$$

où

$$1 \leq q \leq N, \quad 1 \leq s,i \leq D.$$

En supposant que les estimations des paramètres ne changent pas significativement avec de nouvelles observations, nous pouvons approximer le paramètre $\alpha_t^{(q.s)}$en utilisant les estimations précédentes du paramètre $a_{t-1}^{(q.s)}$. Cela conduit à des équations de mise à jour récursives simples et bien définies.

Dans la méthode récursive proposée, les paramètres α, β et Δ sont mis à jour en fonction de leurs valeurs estimées lors de l'observation précédente au temps t−1. Les équations mises à jour que nous avons dérivées sont les suivantes:

$$\theta_t^q = \pi_t^q \log SSD\ (x_t | U^q) \quad , \quad 1 \leq q \leq N \ . \tag{3.13}$$

$$W_t^{(q.s)} = W_{t-1}^{(q.s)} + \frac{1}{t}\left(\frac{\log(x_t^s)\theta_t^q}{\sum_{j=1}^{N} \theta_t^j} - W_{t-1}^{(q.s)}\right) \ . \tag{3.14}$$

$$y_t^q = y_{t-1}^q + \frac{1}{t}\left(\frac{\theta_t^q}{\sum_{j=1}^{N} \theta_t^j} - y_{t-1}^q\right) \ . \tag{3.15}$$

$$V_t^q = \pi_t^q \frac{\log SSD(X|\theta)}{\beta_t}(x_t | U^q) \ . \tag{3.16}$$

$$a_t = a_{t-1}^{(q.s)} + \frac{1}{t}\left(\frac{V_s^q}{\sum_{j=1}^{N} \theta_t^j} - a_{t-1}^q\right). \tag{3.17}$$

$$\alpha_t^{(q.s)} = \psi^{-1}\left(\psi \sum_{i=1}^{D} \alpha_i\right) - \frac{1}{\Delta}\log \beta_t^{(q.s)} + \frac{1}{\Delta}\frac{W_t^{(q.s)}}{y_t} \ . \tag{3.18}$$

$$\beta_t^{(q.s)} = \frac{1}{\alpha_t^{(q.s)}} \left(\sum_{i=1}^{D} \alpha_i\right) x_t \ . \tag{3.19}$$

$$\Delta_t^q = \frac{1}{D+1} \left[\sum_{i=1}^{D} \alpha_i \log \beta_i + \left(\sum_{i=1}^{D} \alpha_i\right) \left(\log \sum_{i=1}^{D} \frac{x_t}{\beta_i}\right) - \sum_{i=1}^{D} \alpha_i \log x_t\right] . \tag{3.20}$$

Prevue: Voici la démonstration de la formule expliquant comment récupérer ω et γ sont respectivement.

$$y_t^q = y_{t-1}^q + \frac{1}{t}\left(\frac{\theta_t^q}{\sum_{j=1}^{N} \theta_t^j} - y_{t-1}^q\right)$$

$$y_t^q = \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{\theta_i^q}{\sum_{j=1}^{N} \theta_i^j} + \frac{1}{t}\left(\frac{\theta_i^q}{\sum_{j=1}^{N} \theta_t^j} - \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{\theta_i^q}{\sum_{j=1}^{N} \theta_i^j}\right)$$

$$y_t^q = \frac{1}{t} \sum_{i=1}^{t} \frac{\theta_i^q}{\sum_{j=1}^{N} \theta_i^j} \ .$$

$$W_t^{(q.s)} = W_{t-1}^{(q.s)} + \frac{1}{t}\left(\frac{\log(x_t^s)\, \theta_t^q}{\sum_{j=1}^{N} \theta_t^j} - W_{t-1}^{(q.s)}\right)$$

$$= \frac{1}{t-1}\left(\sum_{i=1}^{t-1} \frac{\log(x_t^s)\, \theta_i^q}{\sum_{j=1}^{N} \theta_i^j}\right) + \frac{1}{t}\left[\frac{\log(x_t^s)\, \theta_i^q}{\sum_{j=1}^{N} \theta_i^j} - \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{\log(x_t^s)\, \theta_i^q}{\sum_{j=1}^{N} \theta_i^j}\right]$$

$$W_t^{(q.s)} = \frac{1}{t} \sum_{i=1}^{t} \frac{\log(x_t^s) \theta_i^q}{\sum_{j=1}^{N} \theta_i^j} \ .$$

$$a_t = a_{t-1}^{(q.s)} + \frac{1}{t}\left(\frac{V_s^q}{\sum_{j=1}^{N}\theta_t^j} - a_{t-1}^q\right)$$

$$\frac{1}{t-1}\sum_{i=1}^{t-1}\frac{V_i^q}{\sum_{j=1}^{N}\theta_i^j} + \frac{1}{t}\left(\frac{V_i^q}{\sum_{j=1}^{N}\theta_t^j} - \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{\theta_i^q}{\sum_{j=1}^{N}\theta_i^j}\right)$$

$$a_t = \frac{1}{t}\sum_{i=1}^{t}\frac{V_i^q}{\sum_{j=1}^{N}\theta_i^j} \ .$$

La méthode d'estimation des paramètres du modèle de Markov caché (MMC) avec échelle décalée est décrite dans l'Algorithme 1. Cette technique se compose de deux parties principales :

- Calculer les valeurs de θ, ω, y, V et a à l'aide des équations. (3.13)-(3.17).
- Mise à jour des valeurs α_t, β_t, et Δ en se basant sur les équations. (3.18) à (3.20). Les probabilités d'état initiales, les probabilités de transition, α, β, et Δ sont toutes sélectionnées aléatoirement à l'instant t = 1. L'objectif est de déterminer les valeurs optimales de α, β, et Δ afin que les états cachés soient correctement décrits par notre modèle proposé.

Avec cet algorithme de classification récursive, nous répétons le processus de manière itérative jusqu'à ce que le critère de terminaison soit satisfait (c'est-à-dire jusqu'à ce que la différence entre les valeurs des paramètres aux iterations t et t-1 soit inférieure à un seuil ϵ = 0.5). Nous procédons ainsi en recevant des données d'observation en temps réel et en mettant à jour les paramètres α, β, et Δ. Pour ce faire, nous avons divisé les données en ensembles d'entraînement et de test. Nous effectuons l'étape E et l'étape M décrites dans l'Algorithme 1 à chaque itération pendant l'entraînement jusqu'à ce que la condition de terminaison soit remplie. Une fois les paramètres α, β, et Δ mis à jour, ils peuvent être utilisés sur l'ensemble de données de test pour identifier et étiqueter les états cachés.

Nous utilisons la règle de Bayes pour déterminer la condition la plus probable pour chaque ligne de données lors des tests. En utilisant la règle de Bayes, nous pouvons déterminer la probabilité d'attribuer un certain état à une ligne de données particulière;

l'état avec la probabilité la plus élevée est considéré comme l'état caché le plus probable pour les données observées.

Algorithme 1 : Algorithme d'espérance-maximisation récursif pour l'estimation des paramètres du MMC Dirichlet décalé et mis à l'échelle

---

Résultat: $\vec{\alpha}_t^{<q,s>}, \vec{\beta}_t^{<q>}$ $\boldsymbol{et}$ $\Delta_t^{<q>}, \mathbf{1} \leq \boldsymbol{t} \leq \boldsymbol{T}, \mathbf{1} \leq \boldsymbol{s} \leq \boldsymbol{K}, \mathbf{1} \leq \boldsymbol{q} \leq \boldsymbol{N}$

Initialisation : paramètres initiaux $\vec{\alpha}, \vec{\beta}, \Delta$ **probabilité initiale de chaque état et probabilité de transition entre les états**

Entrée : chaque ligne du jeu de données; $x_t, \mathbf{1} \leq \boldsymbol{t} \leq \boldsymbol{T}$

1: tant que $\in >= \vec{\alpha}_t - \vec{\alpha}_{t-1}, \in >= \vec{\beta}_t - \vec{\beta}_{t-1}$ $\boldsymbol{et}$ $\in >= \Delta_t - \Delta_{t-1}$ **faire**

2: Normaliser le vecteur d'entrée x_t;

3: Étape E

4: Calculer les valeurs de $\theta_t^q, \omega_t^{<q,s>}, \nu_t^{<q,s>}, \gamma_t^q$;

5: Étape M:

6: Mettre à jour les valeurs de $\vec{\alpha}_t^{<q,s>}, \vec{\beta}_t^{<q>}, \Delta_t^{<q>}$

7: fin tant que

La première étape de la procédure consiste en l'initialisation aléatoire des paramètres pour le MMC et les distributions prises en compte pour les données. Ensuite, la procédure continue en boucle sur les étapes E et M jusqu'à ce que la condition de terminaison soit remplie. Ce critère, noté ϵ et fixé à 0.5, est basé sur la variation entre les valeurs précédentes et actuelles de chaque paramètre à la fin de chaque itération de boucle. Comme mentionné précédemment, chaque boucle modifie les valeurs en fonction de nouvelles lignes de données qui sont introduites dans le modèle. Nous supposons que chaque ligne du jeu de données représente une certaine période, donc les paramètres sont ajustés en fonction de l'ajout de nouvelles données.

Chapitre 4

Résultats expérimentaux

4.1 Données réelles : Détections d'occupation dans les bâtiments intelligents

Dans cette section, nous utilisons deux ensembles de données réelles différents pour la catégorisation binaire et multiclasses des occupants dans les bâtiments intelligents afin d'évaluer nos modèles proposés. Les informations recueillies au fil du temps créent une série chronologique où chaque ligne représente les caractéristiques d'une pièce, y compris le nombre de personnes l'utilisant à un moment donné. Cette configuration permet de résoudre le problème de l'apprentissage par lots avec de grands ensembles de données en permettant un entraînement incrémental du modèle, avec des paramètres ajustés à mesure que chaque nouvelle ligne de données est traitée.

Notre principal objectif en testant le modèle sur des ensembles de données d'occupation est d'améliorer l'automatisation des systèmes CVC (chauffage, ventilation et climatisation) dans les bâtiments intelligents. L'objectif est de réduire la consommation d'énergie et d'accroître le confort des résidents afin de soutenir les décisions stratégiques en gestion des installations. Le système d'automatisation d'un bâtiment intelligent utilise des capteurs pour collecter des données environnementales, telles que la température, l'humidité relative et la concentration de CO2, et intègre ces informations. Grâce à ces données de capteurs, nous sommes en mesure d'estimer le nombre d'habitants. Notre modèle de Markov caché (MMC) utilise le nombre d'occupants pour représenter les états cachés et les données des capteurs pour servir d'observations.

Étant donné qu'il existe un nombre infini d'états (n), nous initialisons uniformément les probabilités d'état (matrice π) pour chaque expérience. Les paramètres de chaque distribution et la matrice de transition sont attribués de manière aléatoire dans leurs plages respectives. Les critères de terminaison de l'algorithme sont fixés à $\epsilon = 0.5$ pour éviter une récursion infinie. Notre modèle récursif met à jour les paramètres lorsqu'un nouvel enregistrement de données est introduit, en traitant chaque enregistrement dans nos ensembles de données de séries temporelles comme se produisant à une nouvelle étape temporelle.

4.2 Ensemble de données de classification binaire (Ensemble de données de détection d'occupation)

Le premier ensemble de données, qui provient du dépôt de l'Université de Californie à Irvine (UCI) pour l'apprentissage automatique, est lié à la détection d'occupation et a été utilisé pour tester nos modèles. La distribution des cinq caractéristiques des séries temporelles qui composent cet ensemble de données est illustrée dans la Figure 4.1:

- Température (en Celsius)
- Humidité relative
- Lumière (en Lux)
- CO2 (en ppm)
- Ratio d'humidité (kg de vapeur d'eau/kg d'air, déterminé à partir de la température et de l'humidité relative)

Cette technologie permet des ajustements de paramètres en temps réel, garantissant une prédiction précise de l'occupation et une gestion efficace des ressources du bâtiment.

Pour offrir un aperçu global de l'ensemble de données, sa distribution en fonction de chaque caractéristique est présentée dans la Figure 4.1.

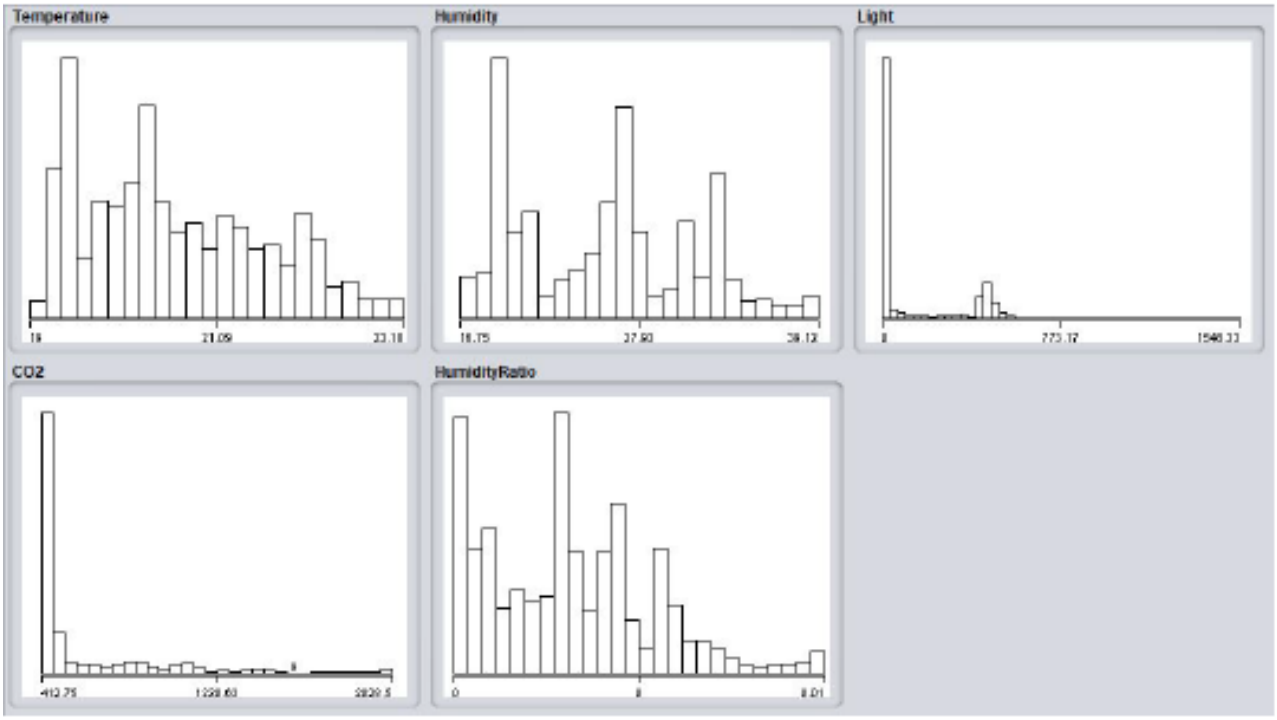

Figure 4.1.: Visualisation des données de détection d'occupation selon différentes caractéristiques

Tableau 4.1: Paramètres estimés pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour l'ensemble de données de détection d'occupation

Modèle	Nombre d'États	Paramètres
SSD-MMC	2	$\alpha = \begin{bmatrix} 0.19 & 0.19 & 0.54 & 3.04 & 0.18 \\ 0.24 & 0.97 & 0.14 & 0.95 & 0.21 \end{bmatrix}$ $\beta = \begin{bmatrix} 4.32 & 4.08 & 6.68 & 0.42 & 0.08 \\ 1.05 & 0.89 & 0.61 & 0.39 & 0.87 \end{bmatrix}$ $\Delta = \begin{bmatrix} 0.4714 & 0.1908 \end{bmatrix}$
D-MMC	2	$\alpha = \begin{bmatrix} 0.5064 & 0.51725 & 0.6047 & 75.7309 & 0.4605 \\ 0.2574 & 0.2601 & 1.5703 & 18.0087 & 0.2432 \end{bmatrix}$

Le tableau 4.1 présente les valeurs des paramètres pour deux modèles, SSD-MMC et D-MMC, tous deux fonctionnant avec deux états. Pour le modèle SSD-MMC, les paramètres α , β et Δ sont fournis sous forme de matrices, montrant différentes caractéristiques de transition et du système. Le modèle D-MMC n'inclut que le parameter α , avec une valeur significative de 75,7309 dans un état, suggérant un écart possible ou une différence majeure par rapport aux autres valeurs. Les valeurs des deux modèles mettent en évidence des variations dans le comportement des systèmes, le modèle SSD-MMC ayant plus de paramètres pour capter la complexité du système par rapport à la structure plus simple du D-MMC.

Tableau 4.2: Exactitude, F-mesure, Précision et Rappel (en pourcentage) pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle appliqués à l'ensemble de données de détection d'occupation

Modèle	**Exactitude**	**F-mesure**	**Précision**	**Rappel**
SSD-MMC	78.85	69.67	62.43	78.85
D-MMC	75.74	63.10	66.62	75.74

Le tableau 4.2 compare les métriques de performance, notamment l'exactitude, la F-mesure, la précision et le rappel, entre les modèles SSD-MMC et D-MMC. Le modèle SSD-MMC affiche une exactitude plus élevée (78,85 %) et une F-mesure supérieure (69,67 %) par rapport au modèle D-MMC, indiquant une meilleure performance globale. Bien que le D-MMC obtienne une précision légèrement supérieure (66,62 %) par rapport au SSD-MMC (62,43 %), ce dernier compense avec un meilleur taux de rappel (78,85 %). Ces résultats suggèrent que le SSD-MMC offre une performance plus équilibrée, excellant à la fois en exactitude et en rappel, ce qui en fait un choix plus adapté pour le jeu de données donné.

4.3 Ensemble de données de classification multiclasses (Ensemble de données d'estimation d'occupation)

L'objectif de cet ensemble de données est d'estimer le nombre de résidents dans les bâtiments intelligents, variant de 0 à 4 occupants, ce qui se traduit par cinq états cachés dans le modèle. L'ensemble de données provient d'une étude menée dans un bureau à l'Institut de Technologie de Grenoble en France (Nasfi et al., 2020). Les données ont été collectées à l'aide de trente capteurs mesurant divers facteurs, tels que le mouvement, la consommation d'énergie, la pression acoustique et la position de la porte (Amayri et al., 2020). Pour examiner le modèle, nous avons divisé l'ensemble de données en ensembles d'apprentissage et de test. Les données d'apprentissage sont utilisées pour estimer les paramètres du modèle, et le modèle entraîné est ensuite appliqué aux données de test pour la classification afin d'estimer le nombre d'occupants.

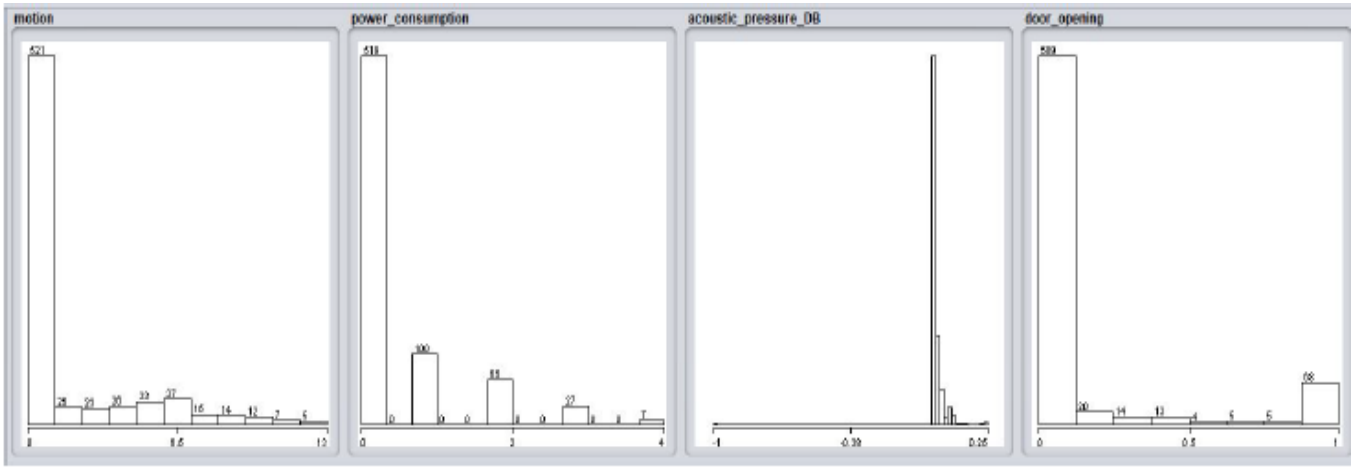

Figure 4.2: Visualisation des données d'occupation des bureaux selon différentes caractéristiques

Tableau 4.3: Valeurs des paramètres pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour les données d'occupation de bureaux

Modèle	Nombre d'États	Paramètres
SSD-MMC	5	$\alpha = \begin{bmatrix} 2.11835854 & 0.4139514 & 0.25985511 & 0.26498811 \\ 0.63830235 & 0.2799971 & 0.24187599 & 0.22238216 \\ 0.35178301 & 0.28502811 & 0.36851808 & 0.37353690 \\ 1.0619052 & 0.35015903 & 0.25926703 & 0.23820305 \\ 0.78872879 & 0.48944358 & 0.44595897 & 0.94121382 \end{bmatrix}$ $\beta = \begin{bmatrix} 4.31346452 & 0.87170831 & 0.55743195 & 0.49509734 \\ 0.78959527 & 0.65445811 & 0.76106531 & 0.55817991 \\ 0.60864214 & 0.43242496 & 0.47870529 & 0.38156932 \\ 1.83194899 & 0.74162169 & 0.58207724 & 0.48368069 \\ 0.97936623 & 0.95856731 & 0.24964398 & 0.34923551 \end{bmatrix}$ $\Delta = [17.8892 \quad 0.9219 \quad 0.2245 \quad 0.1518 \quad 0.3594]$
D-MMC	5	$\alpha = \begin{bmatrix} 2.7746 & 0.8220 & 0.8797 & 1.0617 \\ 2.5784 & 0.3424 & 0.8706 & 0.8711 \\ 0.4348 & 0.8180 & 0.0873 & 0.0590 \\ 1.8384 & 0.9840 & 0.4037 & 0.3731 \\ 0.3054 & 0.9850 & 0.4771 & 0.5966 \end{bmatrix}$

Le tableau présente les valeurs des paramètres pour les modèles SSD-MMC et D-MMC, tous deux configurés avec cinq états. Le modèle SSD-MMC inclut des matrices détaillées pour les paramètres α, β et Δ , reflétant une structure plus complexe visant à capturer diverses transitions et comportements d'états. En revanche, le modèle D-MMC ne fournit que la matrice du parameter α, ce qui suggère une structure plus simple. La présence de valeurs plus élevées dans Δ pour le SSD-MMC (par exemple, 17,8892) indique des differences significatives dans la pondération des états pouvant influencer la performance. Ces différences mettent en évidence le fait que le SSD-MMC pourrait offrir une compréhension plus nuancée des dynamiques du système, tandis que le D-MMC se concentre sur une représentation plus simple.

Tableau 4.4: Exactitude, F-mesure, Précision et Rappel (en pourcentage) pour le MMC de Dirichlet et le MMC de Dirichlet décalé et mis à l'échelle pour le jeu de données d'occupation des bureaux

Modèle	Exactitude	F-mesure	Précision	Rappel
SSD-MMC	80.53	74.19	69.31	80.62
D-MMC	75.00	67.66	73.85	75.00

Le tableau 4.4 compare les métriques de performance, notamment l'exactitude, la F-mesure, la précision et le rappel, entre les modèles SSD-MMC et D-MMC. Le modèle SSD-MMC affiche une exactitude plus élevée (80,53 %) et une F-mesure supérieure (74,19 %) par rapport au modèle D-MMC, indiquant une meilleure performance globale. Bien que le D-MMC obtienne une précision légèrement supérieure (73,85 %) par rapport au SSD-MMC (69,31 %), ce dernier compense avec un meilleur taux de rappel (80,62 %). Ces résultats suggèrent que le SSD-MMC offre une performance plus équilibrée, excellant à la fois en exactitude et en rappel, ce qui en fait un choix plus adapté pour le jeu de données donné.

Conclusion

Ce travail présente une nouvelle approche utilisant les modèles de Markov cachés (MMC) pour estimer l'occupation dans les bâtiments intelligents. Pour décrire les données d'observation de la probabilité d'émission dans les MMC, nous utilisons la distribution de Dirichlet décalée et mise à l'échelle. Nos algorithmes ont démontré leur capacité à prédire avec précision le nombre d'occupants dans une pièce, un facteur crucial pour optimiser l'utilisation de l'énergie dans les systèmes de bâtiments intelligents, en utilisant des données réelles d'occupation. Nos modèles récursifs simplifient le traitement en réduisant la complexité temporelle à un temps linéaire, ce qui entraîne des économies substantielles en termes de ressources informatiques et d'utilisation de la mémoire par rapport aux méthodes de traitement par lots standard.

Le modèle de Markov caché (MMC) de Dirichlet décalé et mis à l'échelle offre des améliorations significatives par rapport au MMC de Dirichlet conventionnel dans plusieurs domaines critiques. L'un des progrès les plus notables est sa capacité à intégrer des facteurs de décalage et de mise à l'échelle, permettant une modélisation plus précise et nuancée des données de séquence. Cette capacité améliorée permet une

représentation plus flexible des distributions sous-jacentes, ce qui est particulièrement bénéfique pour la gestion des données avec une imprévisibilité et un bruit inhérents. Par conséquent, le MMC de Dirichlet décalé et mis à l'échelle démontre une performance prédictive supérieure en identifiant mieux les connexions temporelles et les motifs dans les données.

Au-delà de son exactitude et de sa robustesse, le MMC de Dirichlet décalé et mis à l'échelle se distingue par sa plus grande adaptabilité. Il est particulièrement bien adapté à la gestion de jeux de données complexes et diversifiés en raison de sa flexibilité à s'adapter à des séquences de durées variées et à des motifs dynamiques. Cette adaptabilité garantit des performances élevées dans un large éventail d'applications, y compris l'analyse de séquences biologiques et la reconnaissance vocale. En revanche, le MMC de Dirichlet conventionnel, dépourvu de ces propriétés adaptatives, peut avoir du mal à faire des prédictions précises dans des scénarios aussi variés.

Un autre avantage notable du MMC de Dirichlet décalé et mis à l'échelle est son efficacité computationnelle. En rationalisant les processus d'estimation des paramètres et d'inférence, ce modèle réduit la complexité computationnelle globale et les besoins en ressources. Cette efficacité accélère non seulement l'analyse, mais rend également le modèle plus pratique pour les applications à grande échelle. En comparaison, le MMC de Dirichlet classique implique souvent des coûts computationnels plus élevés, ce qui peut être prohibitif dans des environnements en temps réel ou à ressources limitées. Dans l'ensemble, la précision, la flexibilité et l'efficacité du MMC de Dirichlet décalé et mis à l'échelle en font une alternative supérieure au MMC de Dirichlet conventionnel, offrant des avantages significatifs pour l'analyse des données de séquence. Par conséquent, la latence et la charge de mémoire sont considérablement réduites. En raison de sa complexité linéaire et de son besoin minimal de puissance de calcul importante, ce modèle récursif est très efficace pour diverses tâches de classification qui utilisent le modèle MMC de Dirichlet décalé et mis à l'échelle.

Une stratégie d'apprentissage basée sur la distribution de Dirichlet décalée et sur les concepts MMC a été développée pour apprendre les paramètres. Comme l'établissement d'états cachés pour chaque observation est supervisé, les résultats passés ne sont pas stockés pendant le processus d'entraînement; à la place, les valeurs des paramètres prédites sont mises à jour au fur et à mesure que de nouvelles données sont reçues. Ces valeurs mises à jour sont ensuite utilisées pour calculer les états cachés les plus probables pour chaque nouvelle observation de données en utilisant la règle de Bayes pour garantir la meilleure correspondance possible pendant la phase de test.

Nous avons démontré que la distribution de Dirichlet décalée et mise à l'échelle surpasse la distribution de Dirichlet régulière dans les tâches de modélisation impliquant des MMC. Des données réelles ont validé l'efficacité du MMC de Dirichlet décalé et mis à l'échelle par rapport au MMC de Dirichlet. De futures recherches pourraient se concentrer sur l'amélioration des méthodes d'initialisation et le développement de techniques sophistiquées de sélection des caractéristiques et d'évaluation de la qualité des données pour améliorer encore plus la précision du modèle.

Bibliographie

1. Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., & Weng, T. (2010). Occupancy-driven energy management for smart building automation. In Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-efficiency in Building (pp. 1–6). ACM.
2. Aggarwal, J. K., & Cai, Q. (1999). Human motion analysis: A review. Computer Vision and Image Understanding, 73(3), 428–440.
3. Ai, B., Fan, Z., & Gao, R. X. (2014). Occupancy estimation for smart buildings by an auto-regressive hidden Markov model. In 2014 American Control Conference (pp. 2234–2239). IEEE.
4. Aitchison, J. (1986). The Statistical Analysis of Compositional Data. Chapman & Hall.
5. Akbar, A., Nati, M., Carrez, F., & Moessner, K. (2015). Contextual occupancy detection for smart office by pattern recognition of electricity consumption data. In Proceedings of the IEEE International Conference on Communications (ICC), IEEE, pp. 561–566.
6. Alam, A.G., Rahman, H., Kim, J.K., & Han, H. (2017). Uncertainties in neural network model based on carbon dioxide concentration for occupancy estimation. Journal of Mechanical Science and Technology, 31(5), 2573–2580.
7. Alfalah, B., Shahrestani, M., & Shao, L. (2023). Developing a Hidden Markov model for occupancy prediction in high-density higher education buildings. *Journal of Building Engineering*. https://doi.org/10.1016/j.jobe.2023.106795
8. Altman, R. M. (2007). Mixed hidden Markov models: an extension of the hidden Markov model to the longitudinal data setting. Journal of the American Statistical Association, 102(477), 201–210.
9. Amayri, M., Ngo, Q.-D., & Ploix, S. (2016). Estimating occupancy from measurement and knowledge with Bayesian networks. In 2016 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 508–513). IEEE.
10. Amayri, M., Ngo, Q.-D., Ploix, S., et al. (2017). Bayesian network and hidden Markov model for estimating occupancy from measurements and knowledge. In 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Vol. 2 (pp. 690–695). IEEE.
11. Amayri, O., En-Najjary, T., Ouvrier-Buffet, J., & Farcy, R. (2020). *Occupancy Estimation in Smart Buildings Using Sensor Data*. Grenoble Institute of Technology, France.

12. Andrews, J., Kowsika, M., Vakil, A., & Li, J. (2020). A Motion Induced Passive Infrared (PIR) Sensor for Stationary Human Occupancy Detection. Electrical and Computer Engineering Department, Oakland University, Rochester, MI, USA; Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI, USA.
13. Ang, I. B. A., Salim, F. D., & Hamilton, M. (2016). Human occupancy recognition with multivariate ambient sensors. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), ACM, pp. 991–1002.
14. Ansanay-Alex, G. (2013). Estimating occupancy using indoor carbon dioxide concentrations only in an office building: A method and qualitative assessment. In Proceedings of REHVA World Congress on Energy Efficient, Smart and Healthy Buildings, CLIMA (pp. 1–8).
15. Ashouri, A., Newsham, G.R., & Gunay, H.B. (2019). Day-ahead prediction of building occupancy using Wi-Fi signals. In 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE) (pp. 1237–1242). IEEE.
16. ASHRAE. (1996). ASHRAE Standard 62-1989r, Ventilation for Acceptable Indoor Quality (Public Review Draft).
17. Balaji, B., Xu, J., Nwokafor, A., Gupta, R., & Agarwal, Y. (2013). Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, p. 17.
18. Bartoletti, S., Conti, A., & Win, M. Z. (2017). Device-free counting via wideband signals. IEEE Journal on Selected Areas in Communications, 35(5), 1163–1174.
19. Batista, L., Granger, E., & Sabourin, R. (2012). Dynamic selection of generative-discriminative ensembles for offline signature verification. Pattern Recognition, 45(4), 1326–1340.
20. Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Statist., 37(6), 1554–1563. doi:10.1214/aoms/1177699147.
21. Baum, L.E., & Eagon, J.A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bulletin of the American Mathematical Society, 73(3), 360–363.
22. Bdiri, T., & Bouguila, N. (2011a). An infinite mixture of inverted Dirichlet distributions. In International Conference on Neural Information Processing (pp. 71–78). Springer.
23. Bdiri, T., & Bouguila, N. (2011b). Learning inverted Dirichlet mixtures for positive data clustering. In International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (pp. 265–272). Springer.

24. Bdiri, T., & Bouguila, N. (2012). Positive vectors clustering using inverted Dirichlet finite mixture models. Expert Systems with Applications, 39(2), 1869–1882.
25. Becker, V., & Kleiminger, W. (2017). Exploring zero-training algorithms for occupancy detection based on smart meter measurements. Computer Science - Research and Development, 33, 1–12.
26. Benezeth, Y., Laurent, H., Emile, B., & Rosenberger, C. (2011). Towards a sensor for detecting human presence and characterizing activity. Energy and Buildings, 43(2), 305–314.
27. Bicego, M., Castellani, U., & Murino, V. (2005). A hidden Markov model approach for appearance-based 3D object recognition. Pattern Recognition Letters, 26(16), 2588–2599.
28. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3, 993-1022.
29. Borgsen, S. M. Z., Schöpfer, M., Ziegler, L., & Wachsmuth, S. (2014). Automated door detection with a 3D-Sensor. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 889–894.
30. Bouguila, N., & Ziou, D. (2007). High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(10), 1716–1731.
31. Bouguila, N., Ziou, D., & Vaillancourt, J. (2004). Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. IEEE Transactions on Image Processing,13(11), 1533-1543.
32. Boutemedjet, S., Ziou, D., & Bouguila, N. (2007). Unsupervised feature selection for accurate recommendation of high-dimensional image data. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), Advances in neural information processing systems 20, proceedings of the twenty-first annual conference on neural information processing systems, vancouver, british columbia, canada, december 3-6, 2007 (pp. 177±184). Curran Associates, Inc.
33. Cali, D., Matthes, P., Huchtemann, K., Streblow, R., & Müller, D. (2015). CO2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings. Building and Environment, 86, 39–49.
34. Candanedo, L. M., & Feldheim, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy and Buildings, 112, 28–39.
35. Candanedo, L. M., Feldheim, V., & Deramaix, D. (2017). A methodology based on hidden Markov models for occupancy detection and a case study in a low energy residential building. Energy and Buildings, 148, 327–341.

36. Cappé, O. (2011). Online EM algorithm for hidden Markov models. Journal of Computational and Graphical Statistics, 20(3), 728–749. doi:10.1198/jcgs.2011.09109
37. Cappé, O., & Moulines, E. (2007). Online EM algorithm for latent data models. CoRR. 2007;abs/0712.4.
38. Chen, D., Barker, S., Subbaswamy, A., Irwin, D., & Shenoy, P. (2013). Non-intrusive occupancy monitoring using smart meters. In Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, ACM, pp. 1–8.
39. Chen, L., Barber, D., & Odobez, J.-M. (2007). Dynamical Dirichlet mixture model. Tech. rep., IDIAP.
40. Chen, Z., Jiang, C., & Xie, L. (2018). Building occupancy estimation and detection: A review.*Energy and Buildings*, *169*, 260-270. https://doi.org/10.1016/j.enbuild.2018.03.025
41. Chen, Z., Xu, J., & Soh, Y. C. (2015). Modeling regular occupancy in commercial buildings using stochastic models. Energy and Buildings, 103, 216–223.
42. Chen, Z., Zhu, Q., Masood, M. K., & Soh, Y. C. (2017). Occupancy estimation with environmental sensing via non-iterative LRF feature learning in time and frequency domains. Energy and Buildings, 141, 125–133.
43. Cholewa, M., & Głomb, P. (2013). Estimation of the number of states for gesture recognition with hidden Markov models based on the number of critical points in time sequence. Pattern Recognition Letters, 34(5), 574–579.
44. Connor, R. J., & Mosimann, J. E. (1969). Concepts of independence for proportions with a generalization of the dirichlet distribution. Journal of the American Statistical Association, 64(325),194-206.
45. Conte, G., De Marchi, M., Nacci, A. A., Rana, V., & Sciuto, D. (2014). Bluesentinel: A first approach using iBeacon for an energy efficient occupancy detection system. In Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, ACM, pp. 11–19.
46. Corna, A., Fontana, L., Nacci, A., & Sciuto, D. (2015). Occupancy detection via iBeacon on android devices for smart building management. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, EDA Consortium, pp. 629–632.
47. D'Oca, S., Hong, T., & Langevin, J. (2018). The human dimensions of energy use in buildings: A review. Renewable and Sustainable Energy Reviews, 81, 731–742.
48. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: a large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 248–255.
49. Depatla, S., Muralidharan, A., & Mostofi, Y. (2015). Occupancy estimation using only Wi-Fi power measurements. IEEE Journal on Selected Areas in Communications, 33(7), 1381–1393.

50. Djuric, P. M., & Chun, J.-H. (2002). An MCMC sampling approach to estimation of nonstationary hidden Markov models. IEEE Transactions on Signal Processing, 50(5), 1113–1123.
51. Dodier, R. H., Henze, G. P., Tiller, D. K., & Guo, X. (2006). Building occupancy detection through sensor belief networks. Energy and Buildings, 38(9), 1033–1043.
52. Dong, B., Andrews, B., Lam, K. P., Höynck, M., Zhang, R., Chiou, Y.-S., & Benitez, D. (2010). An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network. Energy and Buildings, 42(7), 1038–1046.
53. Ekwevugbe, T., Brown, N., & Fan, D. (2012). A design model for building occupancy detection using sensor fusion. In 2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST) (pp. 1–6).
54. Ekwevugbe, T., Brown, N., Pakka, V., & Fan, D. (2013). Real-time building occupancy sensing using neural-network based sensor network. In 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST) (pp. 114–119). IEEE.
55. Epaillard, E., & Bouguila, N. (2014). Hidden markov models based on generalized dirichlet mixtures for proportional data modeling. In Artificial neural networks in pattern recognition (pp.71±82). Springer International Publishing.
56. Epaillard, E., & Bouguila, N. (2016). Proportional data modeling with hidden Markov models based on generalized Dirichlet and beta-Liouville mixtures applied to anomaly detection in public areas. Pattern Recognition, 55, 125–136.
57. Epaillard, E., Bouguila, N., & Ziou, D. (2014). Classifying textures with only 10 visual-words using hidden Markov models with Dirichlet mixtures. In Adaptive and Intelligent Systems (pp. 20–28). Springer.
58. Erickson, V. L., Carreira-Perpiñán, M. Á., & Cerpa, A. E. (2011). OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In Proceedings of the 2011 10th International Conference on Information Processing in Sensor Networks, IPSN, IEEE, pp. 258–269.
59. Ewens, W. J. (2004). Mathematical Population Genetics. Springer.
60. Fan, W., & Bouguila, N. (2015). Topic novelty detection using infinite variational inverted Dirichlet mixture models. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMVA) (pp. 70–75). IEEE.
61. Fan, W., & Bouguila, N. (2018). An accelerated variational framework for face expression recognition. 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom) (pp. 1–5). IEEE.
62. Feng, X., Yan, D., & Hong, T. (2015). Simulation of occupancy in buildings. Energy and Buildings, 87, 348–359.

63. Ferguson, J. (1980). Variable duration models for speech. In Proc. of the Symposium on the Applications of Hidden Markov Models to Text and Speech (pp. 143–179). IDA-CRD.
64. Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. The Annals of Statistics, 1(2), 209-230.
65. Filip, A. (2011). BLUED: a fully labeled public dataset for event-based non-intrusive load monitoring research. In Proceedings of the 2nd Workshop on Data Mining Applications in Sustainability (SustKDD), p. 2012.
66. Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2008). Multicamera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2), 267–282.
67. Fosler-Lussier, E. (1998). Markov Models and Hidden Markov Models: A Brief Tutorial.
68. Gao, Y., Schay, A., & Hou, D. (2018). Occupancy detection in smart housing using both aggregated and appliance-specific power consumption data. In Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm), IEEE, pp. 457–462.
69. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis. CRC Press.
70. Genz, A. (1992). Numerical computation of the multivariate normal probabilities. Journal of Computational and Graphical Statistics, 1, 141–150.
71. Gouda, A., & Szántai, T. (2004). New sampling techniques for calculation of Dirichlet probabilities. The Central European Journal of Operations Research, 12, 389–403.
72. Granger, E. (2012). A survey of techniques for incremental learning of MMC parameters. Information Sciences, 197, 105–130. doi:10.1016/j.ins.2012.02.017
73. Greenacre, M. (2010). Biplots in Practice. Fundación BBVA.
74. Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101(Suppl 1), 5228-5235.
75. Guo, X., Tiller, D., Henze, G., & Waters, C. (2010). The performance of occupancy-based lighting control systems: A review. Lighting Research and Technology, 42(4), 415–431.
76. Heinzl, F., & Tutz, G. (2013). Clustering in linear mixed models with approximate Dirichlet process mixtures using EM algorithm. Statistical Modelling, 13(1), 41–67. doi:10.1177/1471082X12471372
77. Hong, T., & Lin, H.-W. (2013). Occupant Behavior: Impact on Energy use of Private Offices. Technical Report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA, US.

78. Jia, R., Dong, R., Sastry, S.S., & Sapnos, C.J. (2017). Privacy-enhanced architecture for occupancy-based HVAC control. In Proceedings of the 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS), IEEE, pp. 177–186.
79. Jiang, C., Chen, Z., Png, L. C., Bekiroglu, K., Srinivasan, S., & Su, R. (2018). Building Occupancy Detection from Carbon-dioxide and Motion Sensors.
80. Jiang, C., Masood, M. K., Soh, Y. C., & Li, H. (2016). Indoor occupancy estimation from carbon dioxide concentration. Energy and Buildings, 131, 132–141.
81. Johnson, N. L., Kotz, S., & Balakrishnan, N. (1997). Discrete Multivariate Distributions. John Wiley & Sons.
82. Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing* (2nd ed.). Pearson. Retrieved from Stanford University.
83. Justino, E. J., El Yacoubi, A., Bortolozzi, F., & Sabourin, R. (2000). An offline signature verification system using MMC and graphometric features. In Proc. of the 4th International Workshop on Document Analysis Systems (pp. 211–222). Citeseer.
84. Kim, H. J., & Schurr, N. (2017). Integration of PIR and door sensors for accurate occupancy detection in smart buildings. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), IEEE, pp. 402–407.
85. Kleiminger, W., Beckel, C., & Santini, S. (2015). Household occupancy monitoring using electricity meters. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 975–986). ACM.
86. Kleiminger, W., Beckel, C., Staake, T., & Santini, S. (2013). Occupancy detection from electricity consumption data. In Proceedings of the 5th ACM Workshop on Embedded Systems for Energy-Efficient Buildings (pp. 1–8). ACM.
87. Kotz, S., Balakrishnan, N., & Johnson, N. L. (2000). Continuous multivariate distributions. New York: Wiley.
88. Labeodan, T., Zeiler, W., Boxem, G., & Zhao, Y. (2015). Occupancy measurement in commercial office buildings for demand-driven control applications: A survey and detection system evaluation. Energy and Buildings, 93, 303–314.
89. Lagona, F., & Barbi, E. (2019). Segmentation of mortality surfaces by hidden Markov models. Statistical Modelling, 19(3), 276–298. doi:10.1177/1471082X18777806
90. Lam, K.P., H¨oynck, M., Dong, B., Andrews, B., Chiou, Y.S., Zhang, R., Benitez, D., & Choi, J. (2009). Occupancy detection through an extensive environmental sensor network in an open-plan office building. IBPSA Building Simulation, 1452–1459.
91. Lee, H., Lee, D., & Lee, H.-J. (2018). A predictive initialization of hidden state parameters in a hidden Markov model for hand gesture recognition. In 2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia) (pp. 206–212). IEEE.

92. Leephakpreeda, T. (2005). Adaptive occupancy-based lighting control via grey prediction. Building and Environment, 40(7), 881–886.
93. Li, J., Najmi, A., & Gray, R. M. (2000). Image classification by a two-dimensional hidden Markov model. IEEE Transactions on Signal Processing, 48(2), 517–533.
94. Lin, J. (2016). On the Dirichlet distribution. Available from: https://mast.queensu.ca/communications/Papers/msc-jiayu-lin.pdf.
95. Liu, D., Guan, X., Du, Y., & Zhao, Q. (2013). Measuring indoor occupancy in intelligent buildings using the fusion of vision sensors. Measurement Science and Technology, 24(7), 074023.
96. Liu, P., Nguang, S.-K., & Partridge, A. (2016). Occupancy inference using pyroelectric infrared sensors through hidden Markov models. IEEE Sensors Journal, 16(4), 1062–1068.
97. Lu, X., Wen, H., Zou, H., Jiang, H., Xie, L., & Trigoni, N. (2016). Robust occupancy inference with commodity WiFi. In Proceedings of the 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, IEEE, pp. 1–8.
98. Mikkilineni, A. K., Dong, J., Kuruganti, T., & Fugate, D. (2019). A novel occupancy detection solution using low-power IR-FPA based wireless occupancy sensor. In Proceedings of the 2019 IEEE Sensors Applications Symposium (SAS), IEEE, pp. 1–4.
99. Minka, T. P. (2000). Estimating a Dirichlet Distribution. Technical Report, MIT.
100. Momenzadeh, M., Sehhati, M., & Rabbani, H. (2019). A novel feature selection method for microarray data classification based on hidden Markov model. Journal of Biomedical Informatics, 95, 103213.
101. Mongillo, G., & Deneve, S. (2008). Online learning with hidden Markov models. Neural Computation, 20(7), 1706–1716.
102. Naqi, M., Lee, S., Kwon, H.-J., Lee, M. G., Kim, M., Kim, T. W., Shin, H. K., Kang, S., Gandla, S., Lee, H.-S., Ahn, Y., Lee, S., & Kim, S. (2019). A fully integrated flexible heterogeneous temperature and humidity sensor-based occupancy detection device for smart office applications.
103. Nasfi, F., En-Najjary, T., Ouvrier-Buffet, J., & Farcy, R. (2020). Occupancy Estimation in Smart Buildings Using Sensor Data. Grenoble Institute of Technology, France.
104. Nasir, N., Palani, K., Chugh, A., Prakash, V. C., Arote, U., Krishnan, A. P., & Ramamritham, K. (2015). Fusing sensors for occupancy sensing in smart buildings. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, pp. 181–186.
105. Nean, A., & Hayes, M. (1999). Face recognition using an embedded MMC. In Proceedings of the IEEE Conference on Audio and Video-Based Biometric Person Authentication (pp. 19–24).

106. Nguyen, T. A., & Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. Energy and Buildings, 56, 244–257.
107. Oldewurtel, F., Sturzenegger, D., & Morari, M. (2013). Importance of occupancy information for building climate control. Applied Energy, 101, 521–532.
108. Oliveira, L. S., Justino, E., Freitas, C., & Sabourin, R. (2005). The graphology applied to signature verification. In 12th Conference of the International Graphonomics Society (pp. 286–290).
109. Otsuka, T., & Ohya, J. (1996). Recognition of facial expressions using MMC with continuous output probabilities. In Proceedings of the 5th IEEE International Workshop on Robot and Human Communication. RO-MAN'96 TSUKUBA (pp. 323–328). IEEE.
110. Pan, S., Bonde, A., Jing, J., Zhang, L., Zhang, P., & Noh, H. Y. (2014). Boes: Building Occupancy Estimation system using Sparse Ambient Vibration Monitoring. SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring, 90611O–90611O.
111. Prékopa, A., & Kelle, P. (1978). Reliability inventory models based on stochastic programming. In A. Prékopa (Ed.), Studies in applied stochastic programming, I (Vol. 80, pp. 49–74). Budapest: MTA SZTAKI.
112. Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. Genetics, 155(2), 945-959.
113. Punzo, A., & Maruotti, A. (2016). Clustering multivariate longitudinal observations: The contaminated Gaussian hidden Markov model. Journal of Computational and Graphical Statistics, 25(4), 1097–1098.
114. Qiao, Y., & Weng, L. (2015). Hidden Markov model based dynamic texture classification. IEEE Signal Processing Letters, 22(4), 509–512.
115. Qiao, Y.-L., & Xing, Z.-Y. (2018). Dynamic texture classification using multivariate hidden Markov model. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 101(1), 302–305.
116. Quan, X., Choi, J. W., & Cho, S. H. (2014). In-bound/out-bound detection of people's movements using an IR-UWB radar system. In Proceedings of the 2014 International Conference on Electronics, Information and Communications (ICEIC), IEEE, pp. 1–2.
117. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286.
118. Rabiner, L. R., & Juang, B. H. (1986). An introduction to hidden Markov models. IEEE ASSP Magazine.
119. Raykov, Y. P., Ozer, E., Dasika, G., Boukouvalas, A., & Little, M. A. (2016). Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 1016–1027). ACM.

120. Reynolds, D. (2015). Gaussian mixture models. In Springer US (pp. 827–832). doi:10.1007/978-1-4899-7488-4_196
121. Rezapoor Nikroo, F., Amayri, M., & Bouguila, N. (2022). Recursive parameter estimation generalized Dirichlet hidden Markov models: Application to occupancy estimation in smart buildings. In *Proceedings of the 2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*.IEEE.
122. Robert, C. P., & Casella, G. (2004). Monte Carlo Statistical Methods. Springer.
123. Ryu, S.H., & Moon, H.J. (2016). Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. Building and Environment, 107, 1–9.
124. Samaria, F. S., & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In Proceedings of 1994 IEEE Workshop on Applications of Computer Vision (pp. 138–142). IEEE.
125. Samaria, F., & Fallside, F. (1993). Face identification and feature extraction using hidden Markov models. Citeseer.
126. Samaria, F., & Young, S. (1994). MMC-based architecture for face identification. Image and Vision Computing, 12(8), 537–543.
127. Schmolze, J. (2001). Introduction to Hidden Markov Models.
128. Sebastian, T., Jeyaseelan, V., Jeyaseelan, L., Anandan, S., George, S., & Bangdiwala, S. I. (2018). Decoding and modelling of time series count data using poisson hidden Markov model and markov ordinal logistic regression models. Statistical Methods in Medical Research. doi:10.1177/0962280218766964.
129. Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.
130. Szántai, T. (1985). Numerical evaluation of probabilities concerning multidimensional probability distributions. Thesis, Hungarian Academy of Sciences, Budapest.
131. Szczurek, A., Maciejewska, M., & Pietrucha, T. (2017). Occupancy determination based on time series of CO2 concentration, temperature, and relative humidity. Energy and Buildings, 147, 142–154.
132. Tiao, G. G., & Guttman, I. (1965a). The inverted Dirichlet distribution with applications. Journal of the American Statistical Association, 60, 793–805.
133. Tiao, G. G., & Guttman, I. (1965b). The inverted Dirichlet distribution with applications. Journal of the American Statistical Association, 60, 1251–1252.
134. Tiller, D. K., Guo, X., Henze, G. P., & Waters, C. E. (2010). The performance of occupancy-based lighting control systems: A review. Lighting Research and Technology, 42(4), 415–431.

135. Tirdad, P., Bouguila, N., & Ziou, D. (2015). Variational learning of finite inverted Dirichlet mixture models and applications. In Artificial Intelligence Applications in Information and Communication Technologies (pp. 119–145). Springer.
136. Vaičiulytė, J., & Sakalauskas, L. (2020). Recursive parameter estimation algorithm of the Dirichlet hidden Markov model. *Journal of Statistical Computation and Simulation, 90*(2), 306-323. https://doi.org/10.1080/00949655.2019.1679144.
137. Wahl, F., Milenkovic, M., & Amft, O. (2012). A distributed PIR-based approach for estimating people count in office environments. In Proceedings of IEEE 15th International Conference on Computational Science and Engineering (pp. 640–647). IEEE.
138. Wang, C., Yan, D., & Jiang, Y. (2011). A novel approach for building occupancy simulation. Building Simulation, 4, 149–167.
139. Wang, M., Wang, X., Zhang, G., & Li, C. (2014). Occupancy detection based on spiking neural networks for green building automation systems. In Proceedings of the 2014 11th World Congress on Intelligent Control and Automation (WCICA), IEEE, pp. 2681–2686.
140. Wang, S., & Jin, X. (1998). CO2-based occupancy detection for on-line outdoor air flow control. Indoor and Built Environment, 7(3), 165–181.
141. Wang, X., Tjalkens, T., & Linnartz, J.-P. (2017). Smart Office Lighting Control Using Occupancy Sensors. Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands; Philips Lighting Research.
142. Wu, C., Wang, S., & Ji, Q. (2015). Multi-instance hidden Markov model for facial expression recognition. In 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) (pp. 1–6). doi:10.1109/FG.2015.7163116.
143. Yang, Y., Hao, J., Luo, J., & Pan, S. J. (2017). Ceilingsee: Device-free occupancy inference through lighting infrastructure based LED sensing. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, pp. 247–256.
144. Yan, D., O'Brien, W., Hong, T., Feng, X., Gunay, H. B., Tahmasebi, F., & Mahdavi, A. (2015). Occupant behavior modeling for building performance simulation: Current state and future challenges. Energy and Buildings, 107, 264–278.
145. Yang, Z., & Becerik-Gerber, B. (2014). The coupled effects of personalized occupancy profile based HVAC schedules and room reassignment on building energy use. Energy and Buildings, 78, 113–122.
146. Yassaee, H. (1974). Inverted Dirichlet distribution and multivariate logistic distributions. Canadian Journal of Statistics, 2, 99–105.
147. Yu, D., & Deng, L. (2015). Hidden Markov models and the variants. In Automatic Speech Recognition (pp. 23–54). Springer.

148. Z. Yang , N. Li , B. Becerik-Gerber , M. Orosz (2012) , A multi-sensor based occupancy estimation model for supporting demand driven hvac operations, in: Pro- ceedings of the 2012 Symposium on Simulation for Architecture and Urban Design, Society for Computer Simulation International , p. 2 .
149. Zhang, B., Shen, Y., & Zhang, L. (2016). Inferring building occupancy based on statistical modeling of multi-sensor data. In Proceedings of the 2016 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, pp. 809–814.
150. Zhou, X., Huang, X., & Wang, Y. (2004). Real-time facial expression recognition in the interactive game based on embedded hidden Markov model. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV 2004) (pp. 144–148).
151. Zou, H., Jiang, H., Yang, J., Xie, L., & Spanos, C. (2017). Non-intrusive occupancy sensing in commercial buildings. Energy and Buildings, 154, 633–643.
152. Zou, J., Zhao, Q., & Yang, W. (2017). Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation. Energy and Buildings, 152, 385–398
153. Zuraimi, M.S., Pantazaras, A., Chaturvedi, K.A., Yang, J.J., Tham, K.W., & Lee, S.E. (2017). Predicting occupancy counts using physical and statistical CO2-based modeling methodologies. Building and Environment, 123, 517–528.

Annexe

Ce code est écrit en Python. Python est un langage de programmation de haut niveau, simple et puissant, utilisé pour une variété d'applications, notamment le calcul scientifique, l'intelligence artificielle, l'apprentissage automatique, l'analyse de données, et bien plus encore.

Les bibliothèques comme NumPy, Pandas, SciPy, et Scikit-learn, utilisées dans ce code, sont toutes des bibliothèques Python bien connues pour effectuer des calculs mathématiques, de l'analyse de données et de l'apprentissage automatique.

```
import numpy as np

import math

import pandas as pd

from scipy.special import gamma

from scipy.special import psi

from scipy.stats import dirichlet

from sklearn.metrics import precision_recall_fscore_support as score

from sklearn.metrics import accuracy_score

from collections import Counter


#print(scipy.stats._file_)



def initialize_func(n_state,n_dimension):


    #transition = np.random.uniform(size=(n_state,n_state))

    #transition = transition / transition.sum(axis=1, keepdims=1)
```

```
    #transition = np.array([[0.71771497, 0.28228503], [0.38484181, 0.61515819]])
    #transition = np.array([[0.71771497, 0.28228503], [0.38484181, 0.61515819]])
    transition = np.array([[0.64639276 ,0.35360724],
     [0.37410949, 0.62589051]])
    print("this_is_transition:", transition)
    #print("trans:",transition, sum(transition[0]))

    #emission = np.random.rand(n_state, n_dimension)#instead of n_state matrix each has n_dim row, i assume D matrix
    #beta_emission = np.random.rand(n_state, n_dimension)
    emission = np.array([[ 1.2989732,  0.80458251,  0.52224264, 23.59781059 , 0.09792764],
 [ 0.15241287,  0.16904466 , 0.67130707,  2.91505114 , 0.0742454 ]])
    print("init_emiss",emission)

    val = 1/n_state
    ini_prob_state = np.full((n_state),val)
    print("init",ini_prob_state)
    return transition, emission, ini_prob_state

def data_func(kind):

    if kind == "train":
        read_file = pd.read_csv(r'E:\My_Desktop\python_proj\term 1\dirichle-hmm\occupancy_data\datatraining.txt')
        print("trainnnnn:", type(read_file))
        read_file.to_csv(r'diridchlet41.csv')
```

```
        observed_main = np.genfromtxt('diridchlet41.csv', delimiter=',')
    else:
        read_file = pd.read_csv(r'E:\My_Desktop\python_proj\term 1\dirichle-hmm\occupancy_data\datatest2.txt')
        read_file.to_csv(r'diridchlet_test14.csv')
        observed_main = np.genfromtxt('diridchlet_test14.csv', delimiter=',')

    observed = observed_main[1:]
    observed = observed[:, 2:]
    print(observed[1],'\n')#,np.sum[np.exp.observed])
    observed_shape = observed.shape
    print("indatafunc",observed_shape)
    return observed_shape,observed

def normalize_func(observed_norm,observed_shape):

    print("in_normalize:", observed_shape, sum(observed_norm), np.sum(observed_norm))
    print("in_normalize:",observed_norm[0])
    observed_norm = observed_norm / 100
    for i in range(observed_shape[0]):
        x = np.exp(observed_norm[i])
        observed_norm[i] = x / np.sum(x)#axis = 0

    return observed_norm
```

```
def dirichlet_func(output, alpha):
    return dirichlet.pdf(output, alpha)


def Generalized_Dirichlet_func(output, alpha, beta):#start at index 0 to n-1
    eta = []
    result = 0
    for i in range(n_state):
        part_1 = gamma(sum(alpha, beta)) / gamma(alpha) * gamma(beta)
        print("type_part1", type(part_1))
        part_2 = math.pow(output[i], (alpha[i] - 1))
        print("type_part1", type(part_1))
        if i == n_state - 1:
            eta [i] = beta[i-2] - 1
        else:
            eta[i] = beta[i] - alpha[i+1] - beta[i+1]
        part_3 = math.pow((1 -np.sum(alpha[:i+1])), eta[i])
        print("type_part1", type(part_1))
        result = part_1 * part_2 * part_3
        print("type_part1", type(part_1),"resss:",result)
    print("result_of_GD_IS:", result)



def update_transition_func(n_state, n_dim, data,ini_prob, tr_mat, emi_mat):
    n_col = n_state * n_state
    n_output = len(data) - 1
```

```
    compu_mat = np.ones([n_output, n_col])
    #print("identity:", n_col, n_output, n_state, n_dim, compu_mat)
    for k in range(n_output): #k represents the output sequence of row k,k+1
        for i in range(n_state):
            index = n_state * i
            for j in range(n_state):
                compu_mat[k][index + j] = ini_prob[i] * tr_mat[i][j] \
                    * dirichlet_func(data[k], emi_mat[i])\
                    * dirichlet_func(data[k+1], emi_mat[j])

    each_prob = []
    for i in range(n_col):
        each_prob.append(compu_mat[:,i].sum())
    max_prob = []
    for i in range(n_output):
        max_prob.append(np.max(compu_mat[i]))
    total_prob = sum(max_prob)
    return each_prob, total_prob

def pi_calculate_func( state_prob_pre, transition):# Ï  t = A Â· Ï  tâˆ’1.
    #print("new pi",np.dot(transition, state_prob_pre))
    '''
    if time == 0:
        return state_prob_pre
    else:
```

```
        return np.dot(transition, pi_calculate_func(time - 1, state_prob_pre, transition))
    '''
    return np.dot(state_prob_pre, transition)
def teta_calculate_func(state_prob_curr, state, alpha_time, observ_seq_time):

    first_val = np.log(gamma(np.sum(alpha_time)))
    second_val = np.sum(np.log(gamma(alpha_time)))
    third_val = np.multiply( (alpha_time[-1] -1) , np.log(observ_seq_time[-1]))
    fourth_val = np.sum(np.multiply((alpha_time[0:-1] - 1),(np.log(observ_seq_time[0:-
1]))))
    logDir = first_val - second_val + third_val + fourth_val
    #print("logdir",logDir)

    return state_prob_curr * logDir

def w_calculate_func(time, teta_max, sumation_teta, w_pre_time):
    #print("in_w_calculated")
    '''
    if time == 0:

        return
np.divide((np.multiply(np.log(observed_norm[0]),teta_max)),sumation_teta)

    else:
```

```
    return w_calculate_func(time-1, teta_max, sumation_teta)\
    + (1/time) * (np.divide((np.multiply(np.log(observed_norm[time-1]),teta_max)),sumation_teta)
        - w_calculate_func(time - 1, teta_max, sumation_teta))
  '''
  return w_pre_time + (1 / time) * (np.divide((np.multiply(np.log(observed_norm[time]), teta_max)), sumation_teta) - w_pre_time)



def y_calculate_func(time, teta_max, sumation_teta, y_pre_time):
  #print("in_y_calculated")
  '''
  if time == 0:
    return np.divide( teta_max, sumation_teta)
  else:
    return y_calculate_func(time-1, teta_max, sumation_teta) + \
        (1/time) * (np.divide(teta_max, sumation_teta) -
            y_calculate_func(time - 1, teta_max, sumation_teta))
  '''

  return y_pre_time + (1/time) * (np.divide(teta_max, sumation_teta) - y_pre_time)

def alpha_calculate_func(alpha, w, y):
  val = psi(np.sum(alpha)) + np.divide(w, y)
  print("this_is_val:", val,"hereeeee", psi(np.sum(alpha)), np.divide(w, y))
  return (1 / np.log(1 + np.exp(-val)))
```

```
def assign_class_func(prob_arr, dir_prob):

    assign_label = np.multiply(prob_arr, dir_prob)

    my_defined_label = np.argmax(assign_label, axis = 1)
    return my_defined_label



def update_alpha_func(time):
    if time == 0:
        pass
    else:
        pass

#epsilon = 0.5#0.19 #was 0.9
n_state = 2
observed_shape, observed = data_func("train")
#separate class label
class_lable = observed[:,-1]
#print("lable",class_lable)
observed = observed[:,:-1]
#print("observedd")

n_dimension = observed_shape[1] - 1 # omit the class lable column
observed_norm = normalize_func(observed, observed_shape)
```

```
print("ressL", sum(observed_norm[0]))
print("input:", n_state, n_dimension)

transition, emission, init_prob = initialize_func(n_state, n_dimension)
print("dd,",observed_norm[1], emission[1])

state_prob_curr = init_prob
time = observed_shape[0]
flag = 0#[0,0]
emission_new = np.ones([n_state, n_dimension])
teta_arr = []
sum_arr = 0
epsilon = 0.66 #8:36#0.6


   #ind = int(class_lable[j])

   #if (flag[ind]  == 1  and j > 2) or j > 500 :
for j in range(time):

   if j > 35:
      break

   print("thisistimee",j)
   ind = int(class_lable[j])
```

```
if flag == -91:# and j > 20:#j>23
  #emission_new[ind] = emission[ind]#np.ones([n_state, n_dimension])
    continue

#ind = int(class_lable[j])
#if (flag[ind]  == 1  and j > 2) or j > 500:
 #   break

else:
    state_prob_curr = pi_calculate_func(state_prob_curr, transition)
    #print("atelseee")
    teta_arr = []
    for i in range(n_state):# for each state call this function to calculate sum teta_arr
        teta_arr.append(teta_calculate_func(state_prob_curr[i], i, emission[i], observed_norm[j]))# i=ith state
        #print("jjjj:",j,observed_norm[j])
        #print("teta:")
        sum_arr = sum(teta_arr)
    if j == 0:
        w = np.divide((np.multiply(np.log(observed_norm[0]), teta_arr[ind])), sum_arr)
        y = np.divide(teta_arr[ind], sum_arr)
        #print("hereiis:::", w, y)
    else:
    #print("summmmarr:", teta_arr, n_state)
        w = w_calculate_func(j, teta_arr[ind], sum_arr, w)
```

```
        #print("w:is:", w)
        y = y_calculate_func(j, teta_arr[ind], sum_arr, y)
    #print("y:is:")
    emission_new[ind] = alpha_calculate_func(emission[ind], w,y)
    #print("emission",                emission_new[ind],emission[ind])#,emission_new,
type(emission_new))

    #print("hereee")
    emission_change = abs(emission_new[ind] - emission[ind])
    #print("emi_chane",emission_new,"**",emission,'%%%')
    emission[ind] = emission_new[ind]
    if (emission_change < epsilon).all():
        print("done!", j)
        #print("final_emission_is:",emission)
        flag = 1




    #each_prob,   total_prob   =   update_transition_func(n_state,   n_dimension,
observed_norm, init_prob, transition, emission)

    #each_prob = np.array(each_prob)/total_prob

    #for i in range(n_state):
    #    transition[i] = each_prob[i*n_state:i*n_state+n_state]
```

```
    #transition = transition / np.sum(transition, axis=1, keepdims=1)
    #print("normal_new_trans", transition)

print("test_data")
final_label = []

test_data_shape,test_data = data_func("test")
n_rows = test_data_shape[0]
new_dimension = test_data_shape[1] - 1
print("new_dimmmmmmm:", new_dimension)
#separate class label
test_lable = test_data[:,-1]

test_data = test_data[:,:-1]
test_data_norm = normalize_func(test_data, test_data_shape)
new_dimension2 = test_data_norm.shape[1] - 1
print("newwwww_dimeen",new_dimension2)

uniqie, count = np.unique(test_lable, return_counts=True)
#print("datatest", test_data_norm[0], count, type(count), uniqie)
print("number of whole datatest:")
prior_prob = count/ n_rows
dir_prob = np.ones([n_rows, n_state])
print("myprob:", prior_prob)

for i in range(n_rows):
```

```
    for j in range(n_state):
        dir_prob[i][j] = dirichlet_func(test_data_norm[i],emission[j])

final_label = assign_class_func(prior_prob, dir_prob)
#print("test_lables:", final_label)
uni_final, count_final = np.unique(final_label, return_counts=True)

confusion_mat = [0,0,0,0]
for i in range(n_rows):

    if (test_lable[i] == 0 and final_label[i] == 0) :
        confusion_mat[0] += 1
    elif  (test_lable[i] == 1 and final_label[i] == 1):
        confusion_mat[1] += 1
    elif (test_lable[i] == 0 and final_label[i] == 1) :
        confusion_mat[2] += 1
    else:#(test_lable[i] == 1 and final_label[i] == 0)
        confusion_mat[3] += 1

print("confusion", confusion_mat)
accuracy =((confusion_mat[0] + confusion_mat[1])/n_rows) * 100
#f_score = (confusion_mat[0]/(confusion_mat[0] + 0.5*(confusion_mat[2] + confusion_mat[3]))) * 100
f_score = ((confusion_mat[0]+confusion_mat[1])/((confusion_mat[1]+confusion_mat[0]) + 0.5*(confusion_mat[2] +  confusion_mat[3]))) * 100
print("result",count_final,uni_final,"gg",count, uniqie,'\n',
```

```
    "accuracy_is:",accuracy,"f-score;", f_score)
print("alpha_is:", emission)

acc = accuracy_score(test_lable, final_label)
precision, recall, fscore, support = score(test_lable, final_label, average='weighted')
print("con_mat:", acc, fscore, precision, recall, support )
```