Titre: Title:	Modèle hybride combinant réseau de neurones convolutifs et modèle basé sur le choix pour la recommandation de sièges
Auteur: Author:	Théo Moins
Date:	2020
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Moins, T. (2020). Modèle hybride combinant réseau de neurones convolutifs et modèle basé sur le choix pour la recommandation de sièges [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/5336/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/5336/
Directeurs de recherche: Advisors:	Daniel Aloise
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Modèle hybride combinant réseau de neurones convolutifs et modèle basé sur le choix pour la recommandation de sièges

THÉO MOINS
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Juillet 2020

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Modèle hybride combinant réseau de neurones convolutifs et modèle basé sur le choix pour la recommandation de sièges

présenté par **Théo MOINS**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Michel DESMARAIS, président
Daniel ALOISE, membre et directeur de recherche
Quentin CAPPART, membre

REMERCIEMENTS

Je tiens à commencer ce mémoire par de sincères remerciements envers mon directeur de recherche Daniel Aloise, pour ses conseils, sa disponibilité, sa sympathie, et surtout sa confiance tout au long de ma maîtrise. Son soutien moral et financier a été indispensable pour surmonter toutes les difficultés et les imprévues pendant ces deux années, et j'en suis particulièrement reconnaissant. Je souhaite également remercier Simon Blanchard avec qui nous avons collaboré, pour sa disponibilité et pour m'avoir assisté lorsque j'en avais besoin.

Ma maîtrise ne pouvant se dissocier avec ma vie d'expatrié à Montréal qui s'achève également, je me dois également de remercier l'ensemble des personnes qui ont été présentes et importantes pour moi durant ces 2 ans, qui ont suivi mon anxiété, mes sauts d'humeurs et j'en passe.

À tous mes amis de France, qui à un moment (voire plusieurs fois !) ont traversé l'atlantique et avec qui j'ai partagé des moments géniaux : Clément, David, Stenzel, Justine, Romain, Paul, et Théo.

À tous mes amis de Montréal qui vont me manquer, ceux du labo fraîchement baptisé Macadamia, pour leurs conseils, leur soutien, et surtout leur bonne humeur au quotidien : Kim, Clara (mãinha !), Laurent, Thiago, Rodrigo et Leandro. Merci aussi aux autres, Rémi, Irving, Quentin, Amine, Sanae et Khalifa, que j'espère revoir au plus vite.

À toute ma famille en France pour le soutien sans failles, et en particulier à mes parents, à qui la pousse de cheveux blancs continue d'accélérer par ma faute.

Enfin, à ma famille de Montréal, Émilie et Donatien que je ne remercierai jamais assez, et Jeff et Alice, mes colocs préférés.

RÉSUMÉ

Avec la vente de billets en ligne, les consommateurs souhaitant réserver un ticket pour un concert, une pièce de théâtre ou un film ont désormais la possibilité de choisir leur emplacement. Ce choix influence l'expérience vécue : différents facteurs sont à considérer, et chaque client fait son propre raisonnement (plus ou moins consciemment) pour prendre cette décision. Par exemple, dans un cinéma, certaines personnes vont privilégier les sièges au centre pour avoir la meilleure vision possible de l'écran, tandis que d'autres pourront préférer les sièges latéraux pour être moins dérangés par la présence d'autrui, en particulier si beaucoup de sièges au centre sont déjà réservés. Cet exemple illustre l'hétérogénéité de raisonnement d'un consommateur dans cette situation, et met en valeur deux catégories de facteurs influant sur la prise de décision : la position dans la salle, et la proximité aux autres.

La réservation en ligne a ainsi permis de collecter ces choix dans des bases de données, et pour l'industrie culturelle (dans notre exemple le gérant de cinéma), cette information peut être cruciale. D'abord, connaître les sièges les plus attractifs à un instant donné peut permettre de modifier la tarification et ainsi augmenter l'affluence dans les salles et donc les recettes. De plus, si cette connaissance se fait spécifiquement pour chaque utilisateur ayant déjà effectué des réservations par le passé, cela peut également permettre d'améliorer les stratégies marketing par la mise en place d'un système de recommandation personnalisé de sièges.

Un premier objectif du mémoire est la revue de méthodes permettant l'estimation de l'attractivité d'un siège dans une salle partiellement remplie. Deux stratégies sont possibles : la première consiste à traiter chaque client individuellement afin d'assurer une modélisation personnelle de la prise de décision, mais qui est limité par la quantité de données disponible par clients. L'autre stratégie consiste à regrouper l'ensemble des données pour pouvoir appliquer des modèles avec plus de capacité comme de l'apprentissage profond, mais qui perd l'information du comportement individuel. Une hypothèse de ce mémoire est que malgré une performance plus faible pour la deuxième stratégie, cette dernière apporte de l'information utile, et une combinaison des deux permet d'améliorer la performance globale et de pallier au problème de la stratégie individualisée du possible manque de données.

Le deuxième objectif du mémoire est l'étude d'une base de données ne provenant pas d'une collecte expérimentale, mais bien de vrais choix provenant d'une salle de concert nord-américaine. Nous explorons les difficultés qui s'ajoutent pour de telles données, et nous suggérons que notre structure hybride combinant une approche individuelle et une profonde

permet aussi de résoudre ces problèmes.

Les recherches effectuées jusqu'ici pour cette problématique étant très récentes, beaucoup d'explorations sont possibles à l'avenir, dans le croisement de domaines variés comme les facteurs psychologiques influant la prise de décision, les modélisations traditionnelles par modèles de choix discret, l'apprentissage automatique et profond, ou encore les systèmes de recommandation.

ABSTRACT

With online ticket sales, consumers wishing to book a ticket for a concert or a movie now have the opportunity to choose their location. This choice influences the lived experience: different factors have to be considered, and each client makes his own reasoning (more or less consciously) to make this decision. For example, in a movie theatre, some people may prefer centre seats to get the best possible view of the screen, while others may prefer side seats to be less disturbed by the presence of others, especially if many centre seats are already reserved. This example illustrates the heterogeneity of reasoning of a consumer in this situation, and highlights two categories of factors influencing decision making: position in the room, and proximity to others.

Online booking has thus made it possible to collect these choices in databases, and for the cultural industry, this information can be crucial. Firstly, knowing the most attractive seats at a given time can help to modify the pricing and thus increase attendance in halls and thus revenues. Moreover, if this knowledge is done specifically for each user who has made reservations in the past, it can also help improve marketing strategies by implementing a personalized seat recommendation system.

A first objective here is the review of methods for estimating the attractiveness of a seat in a partially-filled room. Two strategies are possible: the first one is to treat each client individually to ensure personal modeling of decision making, but this is limited by the amount of data available per client. The other strategy is to aggregate the data to be able to apply models with more capacity such as deep learning, but lose the information about individual behaviour. One hypothesis of this paper is that despite weaker performance for the second strategy, the latter provides useful information, and a combination of the two can improve overall performance and overcome the problem of the individualized strategy of the possible lack of data.

The second objective is the study of a database that does not come from an experimental collection, but real choices from a North American concert hall. We explore the added difficulties for such data, and suggest that our hybrid structure combining an individual approach with a deep one also helps to solve these problems.

As the research carried out so far on this issue is very recent, many explorations are possible in the future, in the intersection of various fields such as psychological factors influencing decision making, traditional discrete choice models, machine learning, or recommendation systems.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

LR	Régression logistique (*Logistic Regression*)
SVM	Machine à vecteur de support (*Support Vector Machine*)
GBT	*Gradient Boosted Trees*
RF	Forêts aléatoires (*Random Forest*)
CNN	Réseau de neurones convolutifs (*Convolutional Neural Network*)
CDNN	CNN avec déconvolution (*Convolutional Deconvolutional Neural Network*)
MLP	Perceptron multicouche (*Multilayer Perceptron*)
MNL	Modèle logit multinomial (*MultiNomial Logit*)
LC-MNL	Modèle MNL à classe latente (*Latent Class MultiNomial Logit*)
ML	Modèle logit mixte (*Mixed Logit*)

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Après une introduction de l'ensemble des concepts utilisés dans ce mémoire, la section 1.2 présente les motivations liées à la recommandation de sièges et les différentes approches possibles pour notre problème. Ensuite, la section 1.3 présente les contributions apportées et la section 1.4 le plan du mémoire pour les chapitres suivants.

1.1 Définitions et concepts de base

Dans cette section, nous présentons succinctement l'ensemble des domaines abordés dans ce mémoire, que l'on peut diviser en trois grandes catégories : classification supervisée, apprentissage profond, et systèmes de recommandation.

1.1.1 Classification supervisée

Parmi l'ensemble des algorithmes de *machine learning*, nous n'aborderons ici que ceux dont l'apprentissage est supervisé, c'est à dire pour lesquels nous disposons d'un ensemble de N données $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$ formées de labels y_i que l'on souhaite déduire à partir des valeurs $\mathbf{x}_i$ que l'on nommera attributs ici. On parle de classification lorsque l'ensemble des valeurs que peut prendre y_i est discret (ces valeurs sont appelées classes). Lorsque seules deux classes sont possibles pour la prédiction, la classification est dite binaire ($y_i \in \{0, 1\}$), et multi classe s'il y en a plus de deux.

Régression logistique

Souvent désignée comme le moyen le plus simple de faire de la classification binaire, la régression logistique vise à transformer une combinaison linéaire d'attributs pour la ramener entre 0 et 1, et ainsi l'interpréter en une valeur de probabilité :

$$P(y_i = 1|\mathbf{x}_i, \mathbf{w}) = f(\mathbf{w}^\top \mathbf{x}_i) = f\left(\sum_{j=1}^{d} w_j x_{i,j}\right) \tag{1.1}$$

avec $f : x \mapsto \frac{1}{1+e^{-x}}$ la fonction couramment nommée sigmoïde ou logit. Les poids $\mathbf{w}$ sont des paramètres qui sont optimisés pour minimiser l'erreur empirique entre les prédictions et les labels. Pour cet algorithme de classification binaire, l'entropie croisée est utilisée pour modéliser cette erreur, et s'exprime ainsi :

$$J(\mathbf{w}) = -\frac{1}{n}\sum_{i=1}^{N} y_i \log\Big(f(\mathbf{w}^\top \mathbf{x}_i)\Big) + (1-y_i)\log\Big(1 - f(\mathbf{w}^\top \mathbf{x}_i)\Big) \tag{1.2}$$

Une autre façon pour définir $P(y_i = 1|\mathbf{x}_i, \mathbf{w})$ aurait pu se faire en utilisant la fonction softmax :

$$\sigma : \mathbf{z} \mapsto \left(\frac{e^{z_1}}{\sum_{\ell=1}^{n} e^{z_\ell}}, \ldots, \frac{e^{z_n}}{\sum_{\ell=1}^{n} e^{z_\ell}}\right) \tag{1.3}$$

Ainsi, on peut définir $P(y_i = 0|\mathbf{x}_i, \mathbf{w})$ et $P(y_i = 1|\mathbf{x}_i, \mathbf{w})$ comme le softmax $\sigma(z_0, z_1)$, avec $z_0 = \mathbf{w}_0^\top \mathbf{x}_i$ et $z_1 = \mathbf{w}_1^\top \mathbf{x}_i$. En observant que $\sigma(z_0 + C, z_1 + C) = \sigma(z_0, z_1)$, on peut se ramener à une modélisation par $\sigma(0, \mathbf{w}^\top \mathbf{x}_i)$, c'est-à-dire à la première définition à l'équation 1.1.

Cependant, cette définition permet une généralisation à un problème à K classes avec $K > 2$, en définissant $P(y_i = k) = \sigma(\mathbf{z})_k$, avec $\mathbf{z} = (0, \mathbf{w}_0^\top \mathbf{x}_i, \ldots, \mathbf{w}_{K-1}^\top \mathbf{x}_i)$.

Ce modèle est nommé régression logistique multinomiale.

Machines à vecteurs de support

Les *support vector machines* (SVM), est un algorithme de classification binaire dont le but est de maximiser la marge qui sépare des points appartenant à deux classes. Ici, $y_i \in \{-1, 1\}$, et on définit le séparateur linéaire $f_{\mathbf{w},w_0}(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + w_0$. Ainsi, l'hyperplan séparateur est $f_{\mathbf{w},w_0}(\mathbf{x}_i) = 0$, et les deux hyperplans $f_{\mathbf{w},w_0}(\mathbf{x}_i) = -1$ et $f_{\mathbf{w},w_0}(\mathbf{x}_i) = 1$ équidistant de l'hyperplan séparateur, sont ceux définissant la marge entre les deux classes.

L'objectif est de trouver $\mathbf{w}$ et w_0 tel que $y_i = -1$ si $f_{\mathbf{w},w_0}(\mathbf{x}_i) < -1$, $y_i = 1$ si $f_{\mathbf{w},w_0}(\mathbf{x}_i) > 1$ (autrement dit, tel que $y_i.f_{\mathbf{w},w_0}(\mathbf{x}_i) > 1$), et qui maximise la distance entre ces deux hyperplans.

En observant que cette distance vaut $\frac{2}{||w||^2}$, le SVM linéaire peut s'écrire :

$$\begin{aligned} \min \quad & \frac{||w||^2}{2} \\ \text{tel que} \quad & y_i.f_{\mathbf{w},w_0}(\mathbf{x}_i) > 1 \quad \forall i = 1, \ldots, N \end{aligned} \tag{1.4}$$

Les SVMs non linéaires sont obtenus en redéfinissant le produit scalaire de base par une fonction noyau $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, qui changerait la façon dont la similarité entre deux points est modélisée. Cette méthode se nomme l'astuce du noyau (en anglais *kernel trick*), et il existe différentes familles de fonctions noyaux, comme les gaussiens, polynomiaux, ou sigmoïdes.

Arbres de décision

Les arbres de décision sont une famille de méthodes pouvant être utilisée comme algorithme de classification ou de régression. L'idée générale est de construire un ou plusieurs arbres dont les feuilles seraient des valeurs prédites, et où les noeuds représenteraient une combinaison d'attributs.

Lorsque plusieurs arbres sont construits, on parle de méthodes d'ensembles. Plusieurs méthodes d'ensemble existent pour faire de la classification binaire, les deux versions les plus fréquentes étant les forêts aléatoires et les arbres boostés par gradient (*Gradient Boosted Trees*, GBT).

Proposé par Breiman [4], le principe des forêts aléatoires est de construire des arbres en se restreignant à un nombre aléatoire d'attributs à partir de données auquel on a appliqué un *bagging*. Le *bagging* consiste en la création de jeux de données par échantillonnage avec remplacement à partir de l'initial, puis par l'agrégation des prédictions pour chacun de ses ensembles par un système de vote pour obtenir la prédiction finale. Les *gradient boosted trees* sont quant à eux une méthode dans laquelle des arbres avec une capacité d'apprentissage faible sont mis à jour itérativement pour diminuer la valeur de résidus (cela peut être l'erreur quadratique moyenne par exemple), puis rassemblés pour obtenir le classificateur final.

1.1.2 Apprentissage profond

Perceptron multicouche (MLP)

La version la plus standard d'une architecture de réseaux de neurones artificiels est le perceptron multicouche (MLP), qui correspond à un empilement de couches de neurones, aussi appelés perceptrons. Chacun de ses neurones effectue une combinaison linéaire d'un vecteur $\mathbf{x}$ et applique ensuite une fonction d'activation pour obtenir une sortie non linéaire : ainsi, la sortie peut s'écrire $h(\mathbf{x}) = g(b + \mathbf{w}^\top \mathbf{x})$, avec g la fonction d'activation. Le choix le plus populaire pour g est $g : x \mapsto \max(0, x)$, qui est la fonction couramment nommée ReLU (Rectified Linear Unit), mais d'autres fonction comme la tangente hyperbolique ou la sigmoïde sont parfois utilisées.

Ainsi, les données $\mathbf{x}$ en entrée passent successivement chaque couche du MLP. Avec une notation matricielle, si $\mathbf{W}^{(k)}$ est l'ensemble des poids de la couche k, c'est-à-dire tel que $\mathbf{W}^{(k)}_{i,j}$ correspond au poids que le perceptron i de la couche k donne à la cellule j de la couche $k-1$, et $\mathbf{b}^{(k)}$ est de la même manière le vecteur contenant chaque biais, chaque couche applique la fonction :

$$\begin{aligned} \mathbf{h}^{(0)}(\mathbf{x}) &= \mathbf{x}, \\ \text{et } \mathbf{h}^{(k)}(\mathbf{x}) &= g(\mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}(\mathbf{x})) \end{aligned} \tag{1.5}$$

En sortie du réseau est appliqué une autre fonction, qui dépend de la tâche effectuée : par exemple, dans le cas d'une classification multi classe, une fonction softmax est appliquée pour obtenir un vecteur de probabilité en sortie.

L'estimation de l'ensemble des poids s'effectue via un algorithme d'optimisation comme la descente de gradient, grâce à une méthode de rétro-propagation de gradient. Celui-ci a beaucoup été amélioré et plusieurs algorithmes sont possibles, la version utilisée par défaut aujourd'hui étant l'algorithme ADAM [5].

Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN, pour *Convolutional Neural Network*) sont une catégorie de réseaux de neurones ayant des résultats remarquables dans le traitement de données visuelles. Grâce à ces réseaux, des progrès significatifs ont été effectués sur des tâches comme la classification d'image [6], la segmentation [7], ou encore la détection d'objets en temps réel [8].

L'architecture d'un CNN dépend de notre problème et de la tâche visuelle que l'on souhaite effectuer (quantité et dimension des données, invariances à considérer, etc.). L'idée générale d'un CNN est l'utilisation de l'opération de convolution, souvent couplée à une opération de *pooling*, afin d'avoir un réseau avec les trois propriétés suivantes :

1. **Connectivité locale** : l'opération de convolution est effectuée entre une image et des noyaux de plus petite taille, ce qui fait que chaque pixel en entrée est uniquement relié à ces voisins. Cette propriété permet de réduire le nombre de paramètres du réseau et la quantité de calculs nécessaire.
2. **Partage des paramètres** : puisque chaque noyau effectue la convolution avec les mêmes valeurs pour l'ensemble de l'image, cela permet de réduire le nombre de paramètres tout ayant des propriétés d'invariances spatiales. En effet, les caractéristiques que vont détecter les noyaux le seront peu importe où elles seront placées en entrée.
3. **Regroupement des pixels voisins** : l'ajout d'une couche de *pooling* ou de *subsampling* (sous-échantillonnage) permet de réduire la taille des couches intermédiaires et d'ajouter de l'invariance locale.

Pour plus de détails sur l'opération de convolution et ses variantes possibles pour l'appren-

tissage profond (taille de noyau, *padding*, *stride*, etc.), un guide complet a été rédigé par Dumoulin et al. [9] [1].

Une autre méthode pour augmenter la prise en compte d'invariances est l'augmentation de données, qui consiste à dupliquer les données d'entraînement en y appliquant des transformations pour aider l'algorithme à ne pas prendre en compte des invariances comme les symétries, les rotations, ou les zooms.

Ainsi, un CNN typique consiste en un empilement de blocs de couches de convolutions/activation/sous-échantillonnage, comme l'illustre la figure 1.1. La profondeur du réseau lui permet de détecter des caractéristiques plus abstraites et plus complexes à partir des représentations des premières couches.

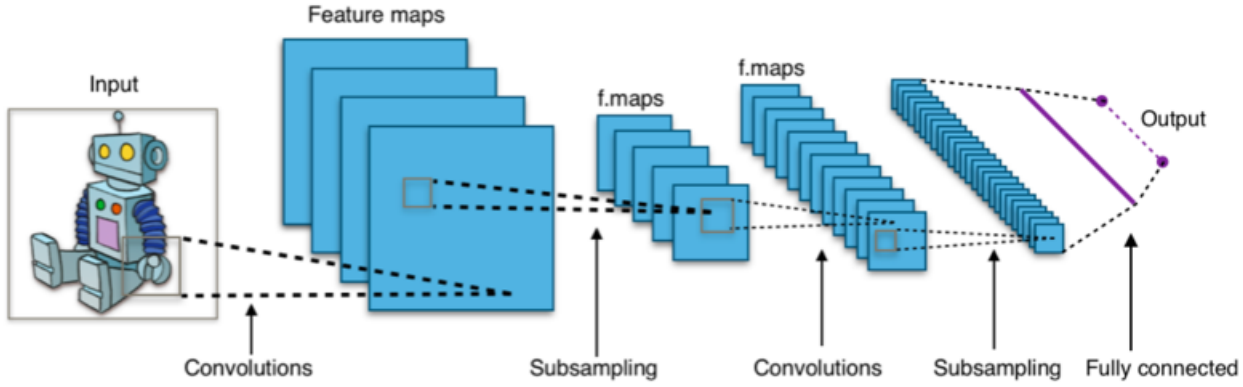

Figure 1.1 Illustration d'une architecture typique de CNN, tirée de Wikipédia Commons [1]. Les couches de convolutions et de sous-échantillonnage sont alternées, puis une couche linéaire permet d'obtenir la sortie souhaitée.

1.1.3 Système de recommandation

Un système de recommandation a pour but de prédire si un utilisateur va aimer ou non un item d'un ensemble donné, basé sur des actions qu'il a fait par le passé. On parle alors de filtrage, car on souhaite restreindre (ou filtrer) l'ensemble des combinaisons entre utilisateurs et items pour ne conserver que celles qui sont les plus pertinentes.

Pour les modèles les plus basiques, on distingue deux stratégies de filtrage : les systèmes de filtrage collaboratif, qui utilisent les interactions recueillies entre les utilisateurs et les items, et les systèmes de filtrage *content-based*, (basés sur le contenu), qui utilisent quant à

1. Une version plus synthétique de ce guide a été réalisée pour le tutoriel de la librairie Theano : `http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html`

eux l'information contenu dans les items et/ou utilisateurs. Une troisième catégorie, nommée hybride, regroupe l'ensemble des systèmes de recommandation qui combinent un système de filtrage collaboratif avec un basé sur le contenu. [2]

Un exemple d'un livre qui traite de façon globale et complète les systèmes de recommandation est celui d'Aggarwal et al. [10].

Filtrage collaboratif

Pour fonctionner, les systèmes de recommandation à filtrage collaboratif font l'hypothèse que deux utilisateurs qui auront évalué un sous-ensemble d'items de manière similaire auront un comportement similaire [11], et que donc l'on peut généraliser ce comportement sur l'ensemble des items.

À partir de la liste des utilisateurs et de celle des items, la modélisation usuelle est de construire une matrice *user-item* qui regroupe l'ensemble des interactions recueillies entre les utilisateurs sur les items. Alors que l'utilisation des systèmes de recommandation sont pour la plupart destinés à des entreprises comme *Amazon*, *Netflix* ou *Spotify* qui disposent d'un très grand nombre de clients, mais aussi d'items, les deux principales difficultés liées à l'utilisation d'une matrice *user-item* [3] sont les suivantes :

1. *Scalability* (dimension de la matrice) : avec des dimensions pouvant aller jusque plusieurs millions d'utilisateurs et items, même un algorithme avec une complexité linéaire ne serait pas raisonnable. C'est pourquoi beaucoup d'articles qui traitent de filtrage collaboratif ont pour objectif d'améliorer la représentation des données en utilisant des méthodes de factorisation de matrice ou de réduction de dimension (voir les articles de Koren et al. [12] ou Sedhain et al. [13] pour des exemples concrets).
2. *Data sparsity* (matrice creuse) : chaque client n'interagissant qu'avec une proportion très faible d'items, la matrice *user-item* va se retrouver extrêmement creuse, ce qui est problématique pour appliquer certains algorithmes de prédiction. Une nouvelle fois, la réduction de dimension est une alternative, cependant cela n'aide pas dans la situation extrême (mais commune) d'un "démarrage à froid" (*cold start problem*), qui est le cas où l'on ne dispose d'aucune information au préalable pour un utilisateur ou un item. L'une des solutions courantes pour pallier ce problème est l'utilisation d'un modèle hybride.

2. Notons que bien que nous désignons notre méthode comme hybride car combinant deux approches, elle ne l'est pas avec cette définition, car aucune des deux approches n'est du filtrage collaboratif dans notre cas.
3. Notamment relevé par Su et al. [11] dans leur revue de littérature

Filtrage basé sur le contenu (*content-based*)

La deuxième grande catégorie des systèmes de recommandation est le filtrage basé sur le contenu (*content-based filtering* en anglais). Cette approche consiste en l'utilisation de l'information intrinsèque des utilisateurs et des items pour faire la prédiction. Cette information est souvent représentée par des attributs, par exemple un film peut être représenté par son titre, son réalisateur, son genre, son année, sa notation sur des sites de critiques, etc.

Les applications les plus communes étant celles avec du texte descriptif en guise d'attributs, un exemple typique de filtrage basé sur le contenu est avec une utilisation de méthodes de traitement du langage (NLP, *Natural Language Processing*), comme par exemple l'utilisation de matrices TF-IDF. [4]

Un avantage à utiliser de telles méthodes est la résolution du problème du démarrage à froid qu'ont les méthodes collaboratives, dans le cas où un nouvel item apparaît et que l'on a aucun historique le concernant. En revanche, si l'on se base uniquement sur le contenu des items, le *cold start problem* demeure à l'ajout d'un nouvel utilisateur. Ces modèles ont souvent l'avantage d'être plus transparent, car il est plus simple d'expliquer comment s'est fait une recommandation à partir d'attributs qu'à partir de notation d'utilisateurs jugés similaire, mais inconnu pour l'utilisateur cible. Néanmoins, les deux principaux inconvénients de ces méthodes sont les suivants :

1. Le manque d'information : si on ne dispose pas d'informations suffisantes qui permettraient de suffisamment distinguer des items et/ou des utilisateurs, la recommandation peut s'avérer compliquée. En guise d'illustration, Pazzani et al. [15] donne l'exemple de recommandation de blagues : un modèle basé sur le contenu sera probablement en mesure de distinguer le thème de la blague, mais aura bien plus de difficulté pour différencier une très bonne blague, ce que peut faire un système collaboratif.
2. La sur-spécialisation (*over-specialization*) : les modèles vont souvent se restreindre à de la recommandation d'items très similaires, ce qui peut créer un biais élevé dans la recommandation.

1.2 Éléments de la problématique

Avec la démocratisation de plateformes d'achat de billets en ligne pour différents domaines comme le transport aérien, les matchs sportifs ou les évènements culturels, les consommateurs bénéficient désormais de la possibilité de choisir leur emplacement au moment de la réserva-

4. Pour plus d'information, voir l'article de blog [14] pour un exemple d'une introduction au filtrage à base de contenu avec une méthode TF-IDF.

tion. Ces plateformes comme *Stubhub* ou *Ticketmaster* peuvent être un atout majeur pour le développement économique d'entreprises. Historiquement, des méthodes de tarification dynamique et de gestion des revenus (*revenue management*) ont rapidement été développées dans des compagnies aériennes. Pour les centres culturels comme les théâtres, les salles de concert ou les cinémas, optimiser la fréquentation peut s'avérer crucial pour maintenir un équilibre économique : son augmentation induirait une augmentation des revenus tout en conservant des prix accessibles. De plus, dans le cas des cinémas, cette stratégie est particulièrement efficace car la vente sur place de nourritures et de boisson est une source importante de revenus [16]. De ce fait, la recommandation de sièges pourrait servir à améliorer les stratégies de ventes grâce à des campagnes marketing ciblées ou à de la tarification dynamique, et ainsi augmenter l'occupation des salles.

Pour identifier les sièges les plus attractifs à un instant donné, deux points de vue sont possibles : soit l'historique agrégé de l'ensemble des billets achetés est considéré, et le problème est traité comme un problème de classification, soit l'hétérogénéité de comportement des individus est prise en compte, auquel cas est adopté un point de vue des systèmes de recommandation, avec pour items l'ensemble des sièges de la salle étudiée. Le premier a pour avantage de considérer l'ensemble des données disponibles, et permet donc d'appliquer des modèles avec beaucoup de capacité comme les réseaux de neurones pour modéliser les interactions d'ordre élevé entre les sièges de la salle. En revanche, une telle méthode ne pourra que se contenter d'une prédiction de tendances globales, à l'inverse du second point de vue qui sera en mesure de considérer les préférences individuelles de chaque client, mais qui sera une approche moins efficace lorsqu'il n'y a que trop peu de données par client disponible.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est l'exploration de modèles pouvant servir à la recommandation de sièges dans des salles de théâtres, concerts ou cinéma, et la proposition d'une architecture hybride combinant une méthode basée sur l'historique de chaque utilisateur avec un réseau de neurones convolutifs, ce qui permet d'améliorer les performances et d'être plus flexibles sur la quantité de données disponibles par client. Contrairement aux recherches effectuées par le passé se basant sur des données expérimentales, un autre objectif de ce mémoire est l'étude des difficultés d'une application à des données réelles, avec l'étude d'une salle de concert nord-américaine.

Ainsi, les contributions détaillées dans ce mémoire sont les suivantes :

1. Nous proposons une architecture hybride pour la recommandation de siège, qui a l'avantage de s'adapter à la quantité de données disponibles, mais aussi au nombre de

sièges réservés au sein d'une même commande.

2. Nous améliorons les performances d'une récente étude de Blanchard et al. [17] sur leurs données.
3. Nous étudions des transactions provenant d'une vraie salle de concert, et montrons que notre architecture reste la plus efficace et la plus adapté dans ce cas.

Les contributions effectuées durant ce mémoire ont été synthétisées pour un article soumis à la conférence ACM RecSys 2020 traitant des systèmes de recommandations (les résultats de la soumission seront dévoilés après le dépôt).

1.4 Plan du mémoire

Ce mémoire se poursuit au chapitre 2 par une revue de littérature de l'ensemble des domaines concernés par notre problème. Ensuite, le chapitre 3 détaille l'approche proposée, puis le chapitre 4 décrit l'ensemble des expériences réalisées ainsi que les résultats pour chacune d'entre elle. Enfin, ce mémoire se conclut en chapitre 5 par une conclusion résumant l'approche et ses limitations, puis en suggérant plusieurs explorations futures possibles.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre présente l'état de l'art sur les notions qui sont abordées dans ce mémoire. La section 2.1 s'attarde sur l'étude du comportement de placement d'un consommateur, ce qui nous permet d'établir les facteurs à prendre en compte pour notre prédiction. Ensuite, nous verrons que différents points de vue sont possibles pour notre problème : en section 2.2, nous traiterons des dernières avancées en apprentissage profond sur des données spatiales, puis nous verrons en section 2.3 les modèles de choix discret, le qui est le point de vue classique pour un tel problème. Enfin, la section 2.4 porte sur les systèmes de recommandation, qui est un point de vue que nous pouvons traiter car nous disposons de l'historique des choix par utilisateurs.

2.1 Facteurs influençant le comportement de placement dans une salle de spectacle

Cette section s'attarde sur les différents facteurs à considérer dans la décision. Liée principalement au domaine de la psychologie et du marketing, elle est celle qui va guider l'ensemble de nos travaux, car elle permet de trouver une représentation convenable pour notre système de recommandation. Ces facteurs peuvent se diviser en deux : la géométrie de la pièce, et la proximité aux autres. Nous verrons cependant à la sous-section 2.1.4 que la seule prise en compte de facteurs explicatifs peut s'avérer insuffisante.

2.1.1 Influence de la géométrie de la pièce

Pour choisir un siège, un consommateur doit évaluer les options qui lui sont disponibles, afin de discriminer les possibilités qui lui sont offertes. Certains critères de discrimination sont raisonnés, et vont dépendre de l'environnement, des caractéristiques du lieu. Par exemple, une scène ou un écran comme dans un théâtre va influencer le choix du positionnement, et cet élément peut rendre les places au centre plus attirantes que les latérales. D'autres éléments de la salle comme la localisation des ailes, l'entrée/sortie, etc., peuvent également influencer la prise de décision. Les consommateurs utilisent ces critères liés à la géométrie de la pièce dans le cadre d'un raisonnement conscient, mais propre à chaque individu (par exemple, certaines personnes aiment se placer sur les ailes ou à l'arrière dans un cinéma, et choisissent leurs sièges en conséquence).

2.1.2 Influence de la proxémie

En plus de ces critères spatiaux, un autre facteur doit également être pris en compte pour améliorer l'estimation : les sièges déjà occupés lors du choix, autrement dit la proximité aux inconnus, qui est étudiée dans une branche de la psychologie appelée proxémie. Ces motivations sont plus complexes à modéliser, car souvent inconscientes pour le consommateur, mais interagissent avec les premières.

Introduite par l'anthropologue Edward T. Hall en 1963 [18], la proxémie (ou proxémique) consiste en l'étude de la façon dont un humain occupe l'espace en fonction de la présence d'autrui. Dans ce livre, Hall démontre l'existence de quatre surfaces de tailles croissantes, souvent appelées familièrement «bulles», qui correspondent à des intensités émotionnelles différentes lors d'une interaction. Ces distances physiques vont de l'intime (pour la zone la plus proche) au public (pour la plus éloignée), mais Hall souligne que les dimensions de ces différentes bulles dépendent de notre culture (par exemple, les pays occidentaux diffèrent des pays méditerranéens), et varie même pour chaque individu.

Par la suite, beaucoup de recherche en psychologie et marketing ont confirmé l'hypothèse que les consommateurs ont tendance à être sensibles à leur proximité aux autres, ce qui en fait un critère à prendre à compte pour notre problème. Pour mieux comprendre comment l'espace personnel interagit avec la prise de décision, Harrell et al. [19] montre que seule la proximité immédiate de l'individu et non la densité générale de la zone étudiée est importante. Ceci explique pourquoi l'objectif de la majorité des études consiste en la modélisation de cette bulle, locale autour de chaque individu. Par exemple, Argyle [20] montre que cette région est généralement circulaire, mais aussi particulièrement sensible aux invasions frontales. De plus, Blanchard et al. [17] ont récemment ajouté que venir seul ou accompagné modifie également la géométrie de la bulle.

Proxémie et transports en commun

Dans le but d'améliorer l'agencement de l'espace et d'optimiser le degré d'utilisation de l'ensemble des sièges, les études de comportement de placement sont plus souvent appliquées aux transports en commun (bus, train, avion, etc.). Ce sont en effet des lieux idéaux pour analyser l'invasion de l'espace personnel, qui peut s'avérer très prononcé si le taux de remplissage est élevé.

Par exemple, Evans et al. [21] étudient la corrélation entre la densité d'individus au sein d'une voiture de métro, et le stress engendré pour les passagers, notamment due à l'intrusion de l'espace personnel. Les auteurs montrent à l'aide d'une étude empirique que ce stress est

corrélé à une densité locale (et non globale) autour du siège choisi, confirmant ainsi l'analyse de Harrell [19]. Ils observent également que le stress décuple lorsqu'un individu s'installe au milieu d'un emplacement à trois sièges, dus à l'intrusion de l'espace personnel doublé, couplé à la possibilité de la contrecarrer qui est réduite. Cette étude, bien que pouvant servir dans le cadre d'une réflexion sur le design des voitures de métro, a l'avantage d'être généralisable à d'autres contextes comme celui des centres culturels. En effet, on peut supposer que le phénomène demeure similaire dans un cinéma, et que les sièges entourés par des sièges occupées seront significativement moins attractifs.

Une étude plus générale du comportement de placement dans les transports en commun a été faite par Schöttl et al. [22], avec des données de trains collectées à Munich. Malgré l'impact que cela peut avoir sur l'aménagement intérieur ou la capacité par exemple, les auteurs soulignent que la plupart des articles qui ont besoin de modéliser le placement dans un train font l'approximation d'une distribution uniforme, et que très peu de recherches sont effectuées pour des analyses plus précises. Les auteurs obtiennent dans leur article la confirmation empirique que cette approximation ne peut être valable ici. Pour ce faire, un test d'hypothèse a été effectué sur les fréquences de placement dans différents cas (sièges voisins occupés ou non, position du siège, etc.). Cette étude prouve ainsi la nécessité de s'intéresser plus en détail à la distribution du choix de placement dès que l'on souhaite modéliser le remplissage d'une salle ou d'un transport en commun, car l'hypothèse d'uniformité n'est pas raisonnable. L'objectif des auteurs n'était donc pas de faire une quelconque prédiction de placement, simplement d'avoir une confirmation empirique que le choix n'est pas uniforme, et plutôt régi par des lois comme celle de la proxémie [18].

2.1.3 Influence du contexte

L'ajout de métadonnées n'a pas été étudié lors de cette maîtrise, mais serait un approfondissement futur (voir 5). En effet, en dehors de la salle en elle-même et de son remplissage, d'autres informations peuvent influencer la prise de décision. Par exemple, comme le souligne l'étude de Weyers et al. [23], les tendances de placement sont différentes dans un théâtre (où le centre sera plus privilégié) et dans un cinéma (où ça sera plus le fond). Pour compléter cela, Blanchard et al. [17] montrent que l'influence de la proximité aux autres dépend aussi du contexte, et qu'elle va être différente dans une salle de cinéma et de concert.

De plus, pour une même salle, Weyers et al. [23] ajoutent que d'autres caractéristiques peuvent influencer le placement, comme le genre d'un film pour les cinémas. Ainsi, même pour des études qui se focalisent sur une seule salle, il peut être intéressant de collecter des métadonnées pour distinguer le type de représentation.

Dans le cas d'une prédiction de l'attractivité d'un siège pour un match de NBA, Huang et al. [24] prennent en compte des informations comme la performance récente des équipes, le nombre de joueurs stars, etc., ce qui leur permet de modifier la tarification des sièges en conséquence (plus de détails sur cet article sont donnés en section 2.2.2).

2.1.4 Biais dans le choix du siège

Dans l'ensemble des modèles qui seront décrits à la section 2.3, le choix d'un individu sera modélisé par une valeur d'utilité qui sera la plus grande de tous les choix disponibles. Ainsi, même avec une composante aléatoire (décrite dans la partie 2.3.1), cela suppose que pour un individu ayant deux items avec une valeur d'utilité égale, la probabilité que l'un ou l'autre soit choisi est la même. En d'autres termes, des variables caractéristiques et quantifiables permettent d'expliciter le raisonnement, et pour deux items non discriminés, la probabilité que l'un soit pris par rapport à l'autre vaut 0.5.

Des recherches réfutent cette hypothèse dans le cadre du choix d'un siège dans un cinéma ou dans un théâtre : c'est un exemple d'interaction qui peut mettre en évidence des biais dans les préférences de choix, mais qui semblent encore difficiles à caractériser. Plus exactement, plusieurs études [23, 25–28] montrent qu'au-delà même de toute stratégie, le côté droit de la salle aura tendance à être privilégié par rapport au gauche. Une hypothèse de cette asymétrie dans la prise de décision est les différences fonctionnelles des hémisphères de notre cerveau, qui est un organe asymétrique, et dont l'hémisphère droit est plus réactif lors d'une expérience émotionnelle subjective [29].

Ainsi, des études de comportement de placement pour des centres culturels existent dans la littérature, mais elles ont pour but principal de comprendre des mécanismes cognitifs lors d'une prise de décision. Elles permettent cependant de mettre en avant la difficulté de modéliser certains mécanismes qui influencent la stratégie de placement.

La première étude de la sorte a été menée en 2000 par Karev [25]. Cinq salles différentes sont présentées à un panel de 870 étudiants, qui choisissent un siège dans chacune des salles, avec les sièges au centre occupé. Chacune des salles diffère dans la position de l'entrée/sortie, des ailes, etc., afin de ne pas considérer ces facteurs dans la prise de décision. Ayant également collecté l'information de si les étudiants étaient gauchers, droitiers ou mixtes, l'auteur confirme son hypothèse sur ces 3 groupes, mais avec un biais plus prononcé sur les droitiers.

Dans la continuité du travail de Karev [25], Weyers et al. [23] prolongent l'expérience pour des cinémas, des théâtres et des restaurants. Pour autant, les auteurs relèvent que ce biais directionnel n'est pas toujours vrai, et dépend du placement de l'écran pour les cinémas ou de

la scène pour le théâtre. Les auteurs expliquent ce constat par la différence entre le montage expérimental et la réalité : le biais serait ainsi présent sur la feuille utilisée pour représenter la salle lors du sondage (quelque soit l'emplacement de l'écran ou de la scène), et non la salle en elle-même. Cette explication sera cependant remise en question par la suite [26], car le fait d'avoir tourné le plan rend l'expérience moins spontanée pour le cerveau, ce qui empêche de conclure.

Par la suite, Okubo [26] ajoute également l'engouement lié au fait de voir un film comme influençant ce biais : ce comportement ne serait ainsi valable que dans les situations où les personnes sont enthousiastes à l'idée de voir le film.

Une limitation de ces articles [23, 25, 26] est la réalisation des expériences en laboratoire, et donc malgré les efforts fournis pour reproduire les conditions d'un choix naturel, un biais peut exister entre l'expérience et le monde réel, en particulier sur de telles études portées sur les biais comportementaux. Cependant, la démocratisation de la vente des billets en ligne va permettre de pallier ce problème. En effet, des sites internet comme *Ticketmaster* ou *Stubhub*, rendent possible la réservation de son emplacement précis, en plus de la simple réservation de ticket. Ainsi, l'utilisation de ces données collectées permet de manipuler des choix dans des situations réelles, et d'éviter des biais expérimentaux. C'est ce qu'ont fait Nicholls et al. [27] dans leur étude sur l'asymétrie dans le comportement de placement dans des théâtres et des avions. Les données analysées contiennent 100 vols des compagnies Air Canada et American Airlines, et 37 performances de différents théâtres en Australie. Un autre avantage relevé par les auteurs d'utiliser de «vrais» données est la variation du taux de remplissage, ce qui n'était pas le cas pour les études précédentes. Ainsi, cette étude montre que le biais vers la droite se produit seulement lorsque la salle est suffisamment vide, c'est-à-dire lorsque le client a plus de degrés de libertés dans son choix. Même si les auteurs ne le mentionnent pas, une explication possible est que lorsque la salle est plus remplie, l'invasion de l'espace personnel doit être à considérer, et de ce fait, la prise de décision est changée.

Enfin, le comportement peut différer si l'on choisit en avance ou directement à l'intérieur de la salle. Pour ce qui est du biais vers la droite, Harms et al. [28] confirment que le biais se reproduit lors d'un placement sans réservation, en utilisant des données collectées en filmant un théâtre lors de son remplissage.

Pour résumer, même pour des sièges qui seraient évalués comme identiquement attractifs d'un point de vue de la position et de l'espace personnel, la distribution dans le choix de ces sièges ne sera pas uniforme. De plus, les facteurs conduisant à une non-uniformité sont encore assez mal connus ; la prise en compte du fait d'être droitier ou gaucher [25] a été discuté par la suite [23, 26], et d'autres facteurs comme la motivation pour voir un film [26] et le taux de

remplissage global [27] seraient aussi à considérer. De telles données (qui sont pour certaines personnelles) n'étant jamais relevées par les sites de réservations en ligne, il est impossible de prendre en compte de tels paramètres. Néanmoins, ces études nous permettent de confirmer l'intuition que ces décisions sont complexes à comprendre et à modéliser, et que l'hypothèse d'une distribution uniforme peut être trop grossière, quels que soient la disposition et les sièges restants.

2.2 Traitement de données visuelles

Pour prendre en compte la géométrie de la pièce dans notre prédiction, une méthode simple serait de voir le plan de la salle comme un tableau bi-dimensionnel, et de ce fait l'assimiler à une image binaire qui distingue les sièges disponibles et indisponibles. Ce point de vue nous permet ainsi de voir notre problématique comme étant de la classification d'image, avec pour classes les sièges disponibles. De ce fait, les dernières avancées dans ce domaine, en particulier en apprentissage profond, peuvent être appliquées.

2.2.1 Réseau de neurones convolutifs

Les réseaux de neurones convolutifs (CNN, introduit à la section 1.1.2) sont aujourd'hui la référence pour la majorité des tâches visuelles.

Beaucoup d'améliorations ont été effectuées depuis le succès de Krizhevsky et al. et de leur réseau AlexNet [6] à la compétition ImageNet [30], qui consiste en un problème de classification avec 1000 classes, à partir d'une base de données de 1,2 million d'images.

D'abord, Simonyan et al. [31] ont montré avec leur réseau VGG-Net l'intérêt d'avoir des réseaux très profond tout en gardant des petits noyaux de convolution (de taille 3×3 dans toute l'architecture). Ensuite, les résultats ont été améliorés par He et al. [32] grâce au réseau ResNet, qui introduit des connexions "raccourci", qui sautent au-dessus de quelques couches. Par la suite, malgré des résultats ayant atteint la précision d'un humain sur ImageNet [30] (94.9%), d'autres améliorations ont été trouvées, grâce à d'autres architectures [1] ou à l'introduction de méthodes comme la *Batch Normalization* [34].

Pour des problèmes comme la segmentation ou la génération d'images où une image en entrée produit une image en sortie, il est possible d'ajouter un réseau de déconvolution au réseau de convolution, qui aurait pour but de reconstruire une sortie structurée de haute dimension. Un exemple d'architecture est représenté figure 2.1, tirée de l'article de Noh et al. [2] pour le problème de segmentation. Ici, la partie de droite est effectuée grâce à de l'*unpooling* et des

1. Pour n'en citer qu'une, voir par exemple DenseNet par Huang et al. [33]

opérations de déconvolutions. L'*unpooling* est effectué en retenant la position du maximum où le pooling avait été effectué, et en affectant les autres valeurs à 0. La déconvolution, quant à elle, est effectuée comme une convolution, mais avec des valeurs de paramètres (*padding*, *stride*, etc.) qui rendent la sortie plus grande que l'entrée.

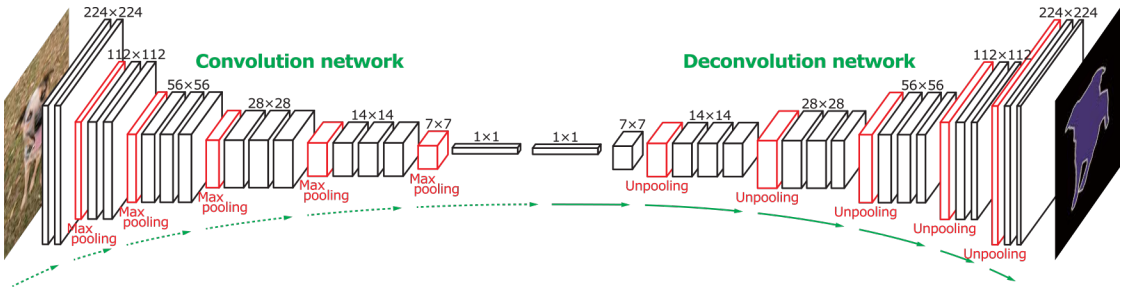

Figure 2.1 Architecture formée d'un réseau de convolution et d'un réseau de déconvolution, proposé par Noh et al. [2] pour de la segmentation d'image (image tirée de l'article).

Un autre moyen pour sur-échantillonner les données est par interpolation, par exemple de l'interpolation au plus proche voisin ou de l'interpolation bilinéaire. Proposée par Odena et al. [35], cette méthode est aujourd'hui privilégiée, car elle permet d'éviter la génération d'artefacts (voir les exemples de [35]).

2.2.2 Réseau de neurones sur des données spatio-temporelles

Nos données étant des choix de clients dans une salle à un instant précis, le point de vue spatio-temporel peut également être adopté dans notre problème.

L'aspect temporel n'a pas été exploré dans cette maîtrise et serait une amélioration future possible. Récemment, il a été abordé dans un article de Huang et al. [24], qui traite d'un sujet semblable au nôtre, puisqu'il s'agit de quantifier les sièges les plus attractifs dans un centre sportif ou culturel, afin de modifier la tarification en conséquence. Pour ce faire, le prix de chaque siège acheté par un client est collecté, et associé à une date, une position, et un type de représentation, ce qui résulte en des données particulièrement creuse. Pour pallier cela, une "couche de grossissement" (*coarsening layer*) est ajouté afin d'avoir un rôle semblable à une couche de pooling, pour les dimensions temporelles et spatiales. Ainsi, cette couche rend difficile la prise en compte de l'occupation de la salle dans la prise de décision, et en particulier de la proximité aux autres (voir section 2.1.2), puisque les sièges proches sont mis en commun au début de l'architecture. Ensuite, un module spatial contenant un CNN et un module temporel contenant un RNN sont combinés.

Dans ce problème, les auteurs ne prennent pas en compte l'hétérogénéité des comportements

de client, et traitent le problème d'une manière globale. Cependant, notre recherche et celle de Blanchard et al. [17] montrent que ne pas distinguer les clients rend la prédiction plus difficile pour ce problème.

2.3 Modèles de choix discret

Cette section présente les modèles de choix discret, une approche traditionnelle pour modéliser le choix d'un consommateur à partir d'un ensemble d'alternatives. Les conditions d'application sont le nombre fini d'alternatives, l'exhaustivité (tout choix être compris dans un ensemble défini), et l'exclusivité (un seul choix est effectué). Ces modèles sont basés sur la théorie de l'utilité aléatoire, décrits à la section 2.3.1. Pour autant, des avancées récentes décrites en section 2.3.4 ont été effectuées pour avoir une approche plus *data-driven* que l'approche initialement *theory-driven*.

2.3.1 Théorie de l'utilité aléatoire

Développée par McFadden [36] à partir de 1974, la théorie de l'utilité aléatoire a pour but de modéliser des choix discrets de consommateurs dans une situation donnée. Dans cette approche, on dispose d'une population de taille N et d'un ensemble d'items de taille J disponible pour le choix de chaque individu. Cet ensemble d'items est identique pour l'ensemble des individus, pour autant dans le cadre des salles de spectacles, le nombre d'items disponibles sera inférieur ou égal au nombre total J (puisque les items seront les sièges disponibles), et il faudra donc restreindre les calculs à un sous-ensemble des items lors du choix de chaque client.

On suppose ensuite que chaque individu i alloue à chaque item j une valeur d'utilité $U_{i,j}$. L'objectif du consommateur est de maximiser cette valeur en choisissant parmi toutes les possibilités qui lui sont offertes, c'est-à-dire choisir :

$$\hat{j} = \underset{1 \leq j \leq J}{\operatorname{argmax}} \left(U_{i,j} \right) \tag{2.1}$$

Cette valeur $U_{i,j}$ peut dépendre à la fois des caractéristiques observables de l'item j, et de celles du client i. De ce point de vue, cela présuppose que chaque consommateur a un comportement déterministe, c'est-à-dire qui peut s'expliquer uniquement par des attributs observables. En pratique, cette valeur est impossible à déterminer avec précision, pour plusieurs raisons : d'abord, la prise de décision ne s'effectue pas toujours après l'observation de toutes les caractéristiques, et l'individu peut choisir avec une vision partielle de tout les attri-

buts. Ensuite, des caractéristiques non observables peuvent être à prendre en compte, comme des facteurs psychologiques, difficilement quantifiables. Enfin, le consommateur lors de son choix peut faire une erreur d'évaluation, ce qui peut modifier les valeurs données aux attributs et ensuite la décision. Tout ceci rend l'hypothèse d'une utilité totalement déterministe peu raisonnable.

De cette observation découle la théorie de l'utilité aléatoire, qui consiste en la décomposition de la valeur d'utilité $U_{i,j}$ de la manière suivante :

$$U_{i,j} = V_{i,j} + \epsilon_{i,j} \tag{2.2}$$

avec $V_{i,j}$ la composante déterministe de l'utilité, et $\epsilon_{i,j}$ une variable aléatoire qui représente l'incertitude quant au choix. C'est cette composante qui modélise l'incapacité de prédire avec exactitude le choix d'un consommateur, puisqu'on ne connaît pas la réalisation de $\epsilon_{i,j}$.

Étant donné le caractère aléatoire de l'utilité, l'estimation du choix ne peut se faire qu'en termes de probabilité, c'est-à-dire calculer les valeurs de $p_{i,j} = P(U_{i,j} = \max_{1 \leq \ell \leq J} (U_{i,\ell}))$ pour chaque item j.

2.3.2 Modèles logits

L'ensemble des modèles que nous décrivons ici sont des modèles de choix multinomiaux, car notre problème a plus que deux classes. À partir d'hypothèses faites sur les valeurs d'utilité, il est possible de retrouver une famille de méthodes de classification qui sont les modèles logits, que nous décrivons ici.

Modèle logit multinomial (MNL)

Une pratique courante est de supposer que la composante déterministe $V_{i,j}$ est une combinaison linéaire d'un nombre T d'attributs observables :

$$V_{i,j} = \mathbf{x}_{i,j}^\top \boldsymbol{\beta} = x_1\beta_1 + x_2\beta_2 + \ldots + x_T\beta_T, \tag{2.3}$$

avec $\mathbf{x}_{i,j}$ l'ensemble des attributs (pouvant être continus, binaires, etc.) observés par le client i pour l'item j. Bien souvent, il arrive qu'aucune distinction ne soit faite sur les clients, auquel cas les attributs ne seront que des caractéristiques des items : $\mathbf{x}_{i,j} = \mathbf{x}_j$.

Les coefficients $\boldsymbol{\beta}$ sont ceux qui vont être estimés pour lier ces attributs.

En ce qui concerne $\epsilon_{i,j}$, si l'on suppose que cette variable suit une loi de Gumbel (aussi

appelé loi des valeurs extrêmes de type I), c'est-à-dire de densité de probabilité $f(x) = e^{e^{-x}}$, cela permet de simplifier les calculs pour les valeurs de $p_{i,j}$. En effet, avec les hypothèses de dépendance linéaire et de distribution de la composante aléatoire, McFadden [36] prouve que l'on obtient :

$$p_{i,j} = P(U_{i,j} = \max_{1 \leq \ell \leq J} (U_{i,\ell})) = \frac{\exp(V_{i,j})}{\sum_{\ell=1}^{J} \exp(V_{i,\ell})} = \frac{\exp(\mathbf{x}_{i,j}^{\top} \boldsymbol{\beta})}{\sum_{\ell=1}^{J} \exp(\mathbf{x}_{i,\ell}^{\top} \boldsymbol{\beta})} \tag{2.4}$$

ce qui correspond exactement à un modèle de régression logistique multinomiale (introduit en 1.1.1), avec pour classes les J items intervenant dans le choix.

Une manière d'estimer les paramètres $\boldsymbol{\beta}$ est par maximum de log-vraisemblance, pour obtenir :

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \mathcal{L} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \sum_{i} \sum_{j} y_{i,j} \ln(p_{i,j}) \tag{2.5}$$

avec $y_{i,j}$ une variable binaire indiquant si l'individu i a effectué le choix j.

Dans la littérature, ce modèle est également couramment nommé MNL, pour *Multinomial Logit*.

Une conséquence de cette modélisation est l'hypothèse d'indépendance des alternatives non pertinentes (IIA), qui est que la préférence d'une alternative par rapport à une autre ne dépend que de ces deux possibilités. En d'autres termes, pour deux alternatives j_1 et j_2, le ratio $\frac{p_{i,j_1}}{p_{i,j_2}} = \exp(V_{i,j_1} - V_{i,j_2})$ ne dépendant que de j_1 et j_2, l'ajout d'un troisième item j_3 ne pourra jamais influencer cet ordre de préférence. Souvent cité comme une faiblesse pour ce modèle, cette hypothèse est aussi probablement trop stricte pour notre recherche, où l'influence des sièges disponibles ou non est à prendre en compte à cause de la proxémie (2.1.2).

Modèle logit multinomial avec classe latente (LC-MNL)

Pour prendre en compte l'hétérogénéité des individus, une amélioration possible du modèle MNL est d'ajouter Q classes latentes [37], Q étant un hyperparamètre :

$$p_{i,j} = \sum_{q=1}^{Q} P(C(i) = q) P(U_{i,j} = \max_{1 \leq k \leq K} (U_{i,k}) \,|C(i) = q) \tag{2.6}$$

avec C(i) la classe de l'individu i, et les probabilités sachant la classe de i étant celle d'un modèle MNL.

L'ensemble peut être ensuite estimé par maximum de vraisemblance, de la même manière

que le modèle MNL.

Modèle logit mixte (ML)

Le modèle logit mixte (en anglais *Mixed Logit*) est une version généralisée du modèle MNL, introduit pour l'utilité aléatoire par McFadden et Train [38]. Une manière de retrouver ce modèle à partir d'une maximisation d'utilité est de supposer que les coefficients $\boldsymbol{\beta}$ diffèrent pour chaque individu, et deviennent donc des variables aléatoires [39] :

$$U_{i,j} = \mathbf{x}_{i,j}^{\top}\boldsymbol{\beta}_i + \epsilon_{i,j}, \quad \text{avec } \boldsymbol{\beta}_i \sim f(\boldsymbol{\beta}|\boldsymbol{\theta}) \tag{2.7}$$

où $\boldsymbol{\theta}$ sont les paramètres de la distribution de $\boldsymbol{\beta}$. En pratique, si l'on était en mesure d'observer les coefficients $\boldsymbol{\beta}_i$ pour chaque individu, alors on retrouverait le modèle MNL, et l'on pourrait calculer les probabilités $p_{i,j}$ à l'aide de l'équation 2.4. Néanmoins, la plupart du temps, seuls les attributs $\mathbf{x}_{i,j}^{\top}$ sont observés, ce qui oblige à conditionner sur l'ensemble des valeurs de $\boldsymbol{\beta}_i$ possibles :

$$p_{i,j} = \int \left(\frac{\exp(\mathbf{x}_{i,j}^{\top}\boldsymbol{\beta})}{\sum_{k=1}^{T} \exp(\mathbf{x}_{i,k}^{\top}\boldsymbol{\beta})} \right) f(\boldsymbol{\beta}|\boldsymbol{\theta}) d\boldsymbol{\beta} \tag{2.8}$$

Cette probabilité correspond à celle d'un modèle logit mixte. Le choix le plus courant pour la distribution des $\boldsymbol{\beta}_i$ est la distribution normale multidimensionnelle, même si d'autres distributions comme la lognormale ont également été étudiées (plus de détails sont dans le chapitre 6.2. du livre de Train [39]).

2.3.3 Applications courantes des modèles logits

À notre connaissance, il existe peu d'articles de recherches dont l'objectif est de prédire l'emplacement que va choisir un client dans un cinéma, un théâtre ou une salle de concert. Hormis celui de Blanchard et al. [17], les exemples de modélisation doivent donc se trouver dans des sujets différents, comme le *revenue management* (gestion de revenus), ou la prédiction de demande. Dans tous les cas, la majorité des articles modélisant le comportement d'un client utilisent un modèle provenant de l'utilité aléatoire introduit dans cette section.

Revenue management

On retrouve cette modélisation dans beaucoup de problèmes de *revenue management* (gestion de revenus). Le *revenue management* est une branche de la théorie de la décision qui consiste en l'étude de comportement de consommateurs afin d'optimiser la disponibilité d'un produit

pour maximiser les recettes. De la même manière que pour les compagnies aériennes, qui est le domaine d'application principal du *revenue management*, les théâtres ont deux moyens directs pour augmenter leurs recettes : par l'augmentation du prix des tickets, ou par l'augmentation du nombre de tickets vendus. Dans les deux cas, ces stratégies requièrent souvent d'estimer quelles sont les places les plus attractives, dans le but de construire une tarification optimale, notamment combien de sièges affecter à quelle classe et lesquels (des exemples sont les travaux d'Hetrakul et al. [40] et de Wang et al. [41], tous deux des modèles MNL appliqués à des trains). Ainsi, l'objectif de ces articles est différent du nôtre, car le but est d'assigner une classe à chaque siège, et non un siège à chaque utilisateur. Il est cependant proche, et notre objectif peut être vu comme un problème en amont à la gestion de revenus ; en effet, faire un modèle de prédiction peut permettre de quantifier l'attractivité d'un siège, ce qui sert d'information pour modifier la tarification dans un second temps.

Récemment Baldin et al. [42], proposent de traiter ce problème avec un modèle d'optimisation bi-objectif des revenus et de l'affluence. Pour ce faire, les auteurs utilisent deux modèles de prédiction : un pour la demande (le nombre de clients qui achèteront un ticket), et un pour le choix de la zone de prix pour chaque client, pour pouvoir déduire une estimation du revenu total et de l'affluence. Pour le choix de siège, les auteurs appliquent un modèle d'utilité aléatoire, avec les hypothèses permettant d'obtenir un modèle MNL. Pour autant, les items ici ne sont pas les sièges spécifiquement, mais plutôt les «catégories de prix», et les clients ne sont distingués qu'en fonction de s'ils sont jeunes ou non. Ainsi, le modèle n'est pas spécifique à chaque utilisateur et ne permet donc pas de distinguer les préférences personnelles, et n'est pas conçu pour une localisation précise du siège, ce qui rend l'estimation partielle.

Prédiction de demande

Un autre exemple de problématique analogue est celui de la prédiction de demande, c'est-à-dire de la popularité d'une représentation d'une pièce de théâtre, d'un film ou d'un concert. Estimer le nombre de places vendues pour un évènement donné a un intérêt pour toute organisation culturelle, que ce soit pour améliorer la programmation ou la stratégie de vente : ceci explique l'intérêt porté à de telles recherches d'un point de vue marketing. Cela s'éloigne cependant de notre problématique initiale, car ces études s'effectuent à l'échelle d'une séance et non d'un siège. Pour autant, les difficultés sont similaires : l'un des problèmes relevées par Willis et al. [43] est le fait de concilier à la fois l'hétérogénéité du spectacle (catégorie, genre, qualité, etc.) avec l'hétérogénéité des consommateurs, dont l'exigence, les critères ou les intérêts peuvent considérablement varier. Un exemple d'estimation de la demande pour une pièce de théâtre est donné par Grisolia et al. [44], qui applique un modèle à classe

latente (LC-MNL) qui permet de relier des observations à une variable latente, ici la classe du siège. Grâce à ce modèle, les auteurs ont pu ainsi distinguer trois classes différentes de consommateurs dans leurs données de théâtres en Angleterre : une classe aisée, une populaire, et une intellectuelle (chacune ayant des caractéristiques et des comportements qui leur sont propres). L'intérêt de cette étude est qu'elle distingue et modélise les comportements des clients, ce qui est une information importante d'un point de vue de la politique culturelle. Néanmoins ce modèle, bien qu'utile pour regrouper des comportements similaires, n'est pas fait pour une analyse spécifique à chaque client en fonction de leurs données. De plus, ce qui est mesuré est la volonté de payer une place, et non le choix d'un siège dans la salle. Toutefois, même avec un objectif différent du notre, ces méthodes peuvent s'adapter à un système de prédiction par sièges, ou alors être combinées à notre modèle, comme c'est le cas dans [42]. On pourrait également imaginer un problème où l'objectif serait de modéliser un remplissage de salle sans avoir connaissance à l'avance du nombre de tickets vendus.

Problème du choix de position

Cette maîtrise a été effectuée en parallèle et dans le prolongement d'un article de Blanchard et al. [17], qui apporte plusieurs contributions à notre sujet.

Dans l'article, les auteurs ont effectué une revue de littérature qui regroupe des facteurs potentiels pour la prise de décision qui n'avait, selon eux, jamais été considérés pour une telle tâche. Ensuite, ces facteurs vont être utilisé pour construire une représentation basé sur des attributs théorique, qui servira pour un modèle logit mixte permettant de prédire le siège qu'un client va choisir dans une configuration donnée.

Les améliorations ajoutées pour la prédiction sont les suivantes :

- *Prise en compte de la proxémie (2.1.2) dans la prédiction* : en plus de la position dans la salle : lorsqu'un client choisit un siège, il considère entre autres la position des sièges déjà occupés dans la salle, afin de limiter l'invasion de l'espace personnel.
- *Prise du compte du renoncement (du non-choix) des clients* : si un consommateur peut encore annuler sa réservation lorsqu'il découvre qu'aucun siège ne lui convient, cela ajoute un biais à prendre en compte dans la prédiction (malgré la difficulté de collecter de telles données).
- *Étude et prise en compte du nombre de sièges par commande* : le plus souvent, dans des centres culturels comme des cinémas ou des théâtres, les spectateurs ne s'y rendent pas seuls : plusieurs sièges peuvent être choisis dans une seule réservation. Cela a ensuite un impact dans la prédiction, mais aussi sur la psychologie dans le choix : deux personnes peuvent par exemple confronter leur point de vue pour discriminer les possibilités, ce

qui change complètement le raisonnement et qui peut donc également influer sur la prise de décision. L'article étudie ainsi des choix de une ou deux personnes.

— *Prise en compte individuel du comportement de chaque client grâce au modèle logit mixte* : enfin et surtout, ce modèle est le premier (à notre connaissance et à celle des auteurs de [17]) à proposer un modèle spécifique à chaque individu (par une estimation individuelle des coefficients) pour ce problème, montrant de ce fait l'hétérogénéité de raisonnement de chaque client.

Pour ce faire, les auteurs appliquent un modèle d'utilité aléatoire introduit dans la section 2.3, avec les hypothèses permettant de retrouver un modèle logit mixte avec distribution normale des coefficients. Un modèle bayésien hiérarchique est ensuite utilisé pour l'estimation.

Ici, la partie déterministe $V_{i,j}$ est décomposée de la manière suivante :

$$V_{i,j} = V_{L_{i,j}} + V_{P_{i,j}} + \beta_{nc} x_{nc}, \tag{2.9}$$

avec $V_{L_{i,j}}$ l'ensemble des attributs pour la position dans la salle, $V_{P_{i,j}}$ ceux pour la proximité aux autres, x_{nc} un attribut binaire ajouté pour modéliser la possibilité ou non d'annuler la réservation sans faire de choix.

L'avantage d'un tel modèle (par rapport au MNL par exemple) est la prise en compte individuel des comportements, que les auteurs montrent indispensable pour avoir une prédiction correcte. Pour ce faire, une comparaison est faite avec un modèle d'apprentissage profond qui ne distingue pas les clients et qui, malgré sa capacité bien plus élevée, ne parvient pas à avoir des résultats de prédiction aussi bons. Pour autant, l'une des hypothèses de cette maîtrise consiste en l'affirmation que ces deux modèles peuvent être complémentaires dans l'apport d'information pour la prédiction.

Enfin, cette étude a été réalisée en intégralité avec des bases de données générées par le service *Amazon Mechanical Turk*, dans des contextes de salles de cinémas et de concert. On peut alors se demander ce qu'il en est de ces résultats sur des données réelles, avec des préférences révélées. Il est ainsi possible qu'un biais existe, entre les choix provenant de réels consommateurs et ceux provenant de données générées.

2.3.4 Utilisation de l'apprentissage automatique pour les modèles de choix discrets

Les modèles de choix discrets peuvent être vus comme des problèmes de classification. En effet, si l'objectif est de prédire l'item j qui va être choisi par le client i, il suffit de créer un

label $y_{i,j}$ pour l'ensemble des attributs observables, indiquant si l'individu i a choisi l'item j, et ensuite entraîner un modèle de classification à partir des paires $(\mathbf{x}_{i,j}, y_{i,j})$.

De ce fait, il est possible d'utiliser les dernières avancées en apprentissage automatique pour remplacer les modèles logits traditionnels. Néanmoins, la majorité des algorithmes de classification font l'hypothèse de traiter des données i.i.d. (indépendantes et identiquement distribués) qui ne s'applique pas ici. En effet, puisque l'on sait qu'un seul choix est effectué, lorsque l'un des $y_{i,j}$ est choisi, nous pouvons déduire que les autres ne le seront pas, et donc modifier leurs probabilités. En d'autres termes, dans notre problème, si l'on connaît le siège que va choisir un client, cela influe sur l'information que l'on a des autres sièges, car on peut déduire que la probabilité qu'ils soient choisis sera nulle.

La majorité de la littérature qui existe sur l'utilisation de l'apprentissage automatique pour les choix discrets est appliquée à la prédiction de choix de mode de transport pour un usager. Dans ce domaine, la précision de prédiction est tout aussi importante que son interprétabilité, car il est important de comprendre pourquoi un item a été choisi et pas un autre, afin de pouvoir prendre des mesures adaptées par la suite. C'est pourquoi les modèles logits d'utilité aléatoire ont pendant longtemps été privilégiés, malgré l'essor de l'apprentissage automatique : leur modélisation permet de déduire très facilement quel est l'impact de chaque attribut, en observant leurs coefficients associés. Ce n'est pas le cas de la plupart des modèles d'apprentissage supervisé, dont les plus complexes sont parfois qualifiés de boîtes noires, car leur seul objectif est l'optimisation de la performance.

Machine Learning et modèle de choix discret

Malgré tout, beaucoup de recherches sont aujourd'hui effectuées pour lier ces deux domaines. Récemment, Zhao et al. [45] ont fait une étude comparative complète entre les modèles logits et les modèles d'apprentissage pour le choix du mode de transport. Dans cet article, en plus d'une comparaison des performances de prédiction, une analyse du comportement des algorithmes est effectuée, pour savoir à quel point les résultats peuvent être expliqués et analysés (on parle "d'interprétabilité" des résultats). Entre autres, les modèles *Naive Bayes*, arbres de décision, SVM (utilisés par Zhang et al. [46] et plus récemment par Pirra et al. [47]), forêts aléatoires (voir un article de Cheng et al. [48]) et réseaux de neurones sont comparés dans cette étude, ainsi que les modèles MNL et logit mixte pour les choix discrets. La comparaison avec ce dernier est une contribution des auteurs, car la plupart des articles se limitent au modèle MNL classique. Il est donc intéressant de voir comment l'apprentissage machine se situe face à un modèle logit plus complexe comme le logit mixte, qui devrait prendre en compte l'hétérogénéité du comportement des individus. Les auteurs constatent alors que le

modèle logit mixte ne donne pas de résultats satisfaisants, probablement parce qu'il souffre de surapprentissage pour ce problème. Ils montrent alors que le meilleur algorithme est celui avec des forêts aléatoires, ce qui confirme l'hypothèse de Cheng et al. [48], qui donnait les mêmes conclusions en comparaison avec les modèles SVM, Adaboost et MNL.

Dans un autre domaine, Lhéritier et al. [49] ont aussi récemment appliqué des forêts aléatoires dans la prédiction du choix d'itinéraire aérien. Les auteurs montrent que l'intérêt d'un tel algorithme se trouve dans sa capacité à prendre en compte l'hétérogénéité des individus, de la même manière qu'un modèle à classes latentes (LC-MNL) le ferait.

Deep Learning et modèle de choix discret

En parallèle, des tentatives dès les années 2000 ont été réalisées notamment par Bentz et al. [50] ou Hruschka et al. [51] pour modéliser le modèle MNL par un réseau de neurones. Pour des exemples de modélisation récente, voir les travaux de Wang et al. [52], Sifringer et al. [53], ou encore Han et al. [54]. Malgré un succès fulgurant dans d'autres domaines, il se fait plus tardif pour la modélisation de choix discret. Plusieurs raisons à cela : d'abord, le défaut d'interprétabilité dont souffrent la plupart des réseaux traditionnels est très contraignant dans un tel domaine.[2] De plus, même les performances de ces algorithmes peuvent être mitigées [54] : ceci est souvent dû à un souci de surapprentissage [45], de manque de données, ou de difficulté à trouver des hyperparamètres convenables (difficulté relevée par Wang et al. [52]), plutôt que d'un problème de capacité du réseau. Pour le choix de mode de transport, Sifringer et al. [53] et Han et al. [54] proposent dans leurs articles respectifs une architecture pour modéliser un choix discret par réseaux de neurones, avec une composante assurant l'interprétabilité, et une autre pour augmentant la capacité du modèle. Dans la version de Sifringer et al. [53], la deuxième composante utilise les attributs non utilisés par la première, qui sont ceux dont il n'est pas nécessaire de connaître la façon dont ils impactent la prise de décision.

Pour la prédiction d'itinéraire aérien, Mottini et al. [56] proposent de reprendre une architecture de réseaux de neurones récurrents (RNN) couplés à un module d'attention, que sont les *Pointer Network* [57]. Plus précisément, l'architecture de ce réseau était initialement *Sequence-to-sequence* (*Seq2Seq*), c'est-à-dire adapté pour avoir une séquence en entrée et en sortie, ce qui est usuellement utilisé pour des tâches de traduction. Ici, le décodeur ne sert pas, il est remplacé par une sortie de taille 1 qui indique simplement l'indice de l'item choisi.

Enfin, Otsuka et al. [58] proposent une adaptation des machines de Boltzmann restreintes

2. Plus de détails sont donnés dans un papier de Wang et al. [55], qui utilisent l'apprentissage statistique pour relever la difficulté de concilier décision et réseaux de neurones.

aux modèles de choix discrets, et appliquent leur modèle à des choix d'agents qui choisiraient des images provenant de la base de données MNIST [59].

Il est important de noter que l'ensemble de ces papiers utilisent des bases de données de choix de clients anonymisés, c'est-à-dire qu'il n'est pas possible de reconstituer un historique, et c'est pour cette raison que l'on ne peut pas parler de systèmes de recommandation. Pour autant, certains modèles (comme celui de Lhérithier et al. [49]) tentent de prendre en compte l'hétérogénéité de comportement des individus, malgré une agrégation par client des données.

2.4 Système de recommandation

Les données dont nous disposons pouvant être regroupées par client, nous pouvons adopter le point de vue d'un système de recommandation. Le paradigme y est similaire à celui présenté à la section précédente : on dispose d'un ensemble d'utilisateurs et d'un ensemble d'objets que l'on nommera items. [3] L'objectif ici est de restreindre l'ensemble des possibilités d'association entre utilisateurs et items, pour ne conserver que les plus pertinentes.

2.4.1 Système de recommandation et apprentissage profond

L'apprentissage profond est devenu en quelques années la méthode privilégiée pour les systèmes de recommandation. Zhang et al. [60] propose une revue de littérature qui synthétise les méthodes qui allient ces deux domaines.

Système de recommandation et contenu visuel

Parmi elles, les CNN ont beaucoup été utilisés pour le traitement de données à structure complexe, comme pour des données spatiales par exemple. Ces données peuvent être les items que l'on souhaite recommander, ou alors des attributs de ces items ou des utilisateurs. Selon Zhang et al. [60], les CNNs sont majoritairement utilisés pour l'extraction de caractéristiques par l'apprentissage de représentations, le plus souvent en appliquant une méthode de factorisation de matrices.

Un exemple de recommandation basé sur le contenu est le modèle ConTagNet proposé par Rawat et al. [61], qui permet la génération de *tag* pour une image à partir de cette dernière et de son contexte d'acquisition. Le modèle concatène ainsi un CNN pour les images avec un MLP pour le contexte, pour ensuite ajouter une couche linéaire avec une sortie sigmoïde pour obtenir un vecteur de probabilité de tags. L'idée de rendre personnelle la recommandation de

3. Habituellement, les items sont des produits d'un site de e-commerce, des films, des vidéos, des musiques, etc. Ici, les items seront les sièges dans la salle de théâtre ou le cinéma.

tag pour une image a aussi été reprise par Nguyen et al. [62], mais qui utilise une méthode bayésienne de filtrage collaboratif.

Pour la recommandation d'image à un utilisateur, Lei et al. [63] proposent un réseau de neurones divisé en trois : deux CNNs, un entraîné sur les images qu'un utilisateur a aimées et l'autre sur celles qu'il n'a pas aimé, et un MLP contenant les informations de l'utilisateur en question. Le réseau est ensuite entraîné pour discriminer les images positives des images négatives.

Prise en compte des interactions faibles et fortes dans un système de recommandation

En 2016, Cheng et al. [3] proposent une architecture *Wide & Deep* (large et profonde), pour combiner les avantages de deux approches différentes. La première, une combinaison linéaire d'attributs, assure la prise en compte des items les plus fréquemment liés historiquement, tandis que la deuxième, un MLP, assure la prise en compte des liens plus complexes entre les items. Pour la combinaison des deux modèles, les auteurs font la somme pondérée des deux sorties, puis appliquent la fonction logit pour retrouver un nombre entre 0 et 1 en sortie. Ils précisent également que l'ensemble a été entraîné simultanément, ce qui permet d'avoir une partie *wide* avec moins de dimensions que si les deux modèles étaient entraînés chacun de leurs côtés.

Cette architecture met en avant la nécessité pour un système de recommandation d'avoir à la fois les propriétés de mémorisation (prise en compte des liens directement présents dans les attributs) et de généralisation (apprentissage d'une représentation pour généraliser les résultats), respectivement assurées par les deux parties du modèle *Wide & Deep*. Ainsi, elle permet d'avoir une précision correcte tout en ayant des recommandations diversifiées, c'est-à-dire en évitant les problèmes de surspécialisation (voir section 1.1.3). L'article a ainsi introduit une nouvelle famille d'algorithmes, que l'on pourrait nommer *wide and deep learning*[4], et qui consiste en l'amélioration du principe général proposé par Cheng et al. [48].

Cette idée a ensuite été reprise par Guo et al. [64], en l'appliquant à une prédiction de taux de clic. Ils améliorent ainsi la version *Wide & Deep* [3] en faisant une architecture *end-to-end*, c'est à dire qui peut s'entraîner de bout en bout sans avoir besoin de faire de *feature engineering*. Pour ce faire, la partie *Wide* de Cheng et al. [3] est remplacée par une machine de factorisation, qui a pour but de détecter les liens entre les attributs. Par la suite, une autre amélioration de cette composante sera proposée par Lian et al. [65], toujours dans

4. Nom donné par Zhang et al. [60] dans leur revue de littérature

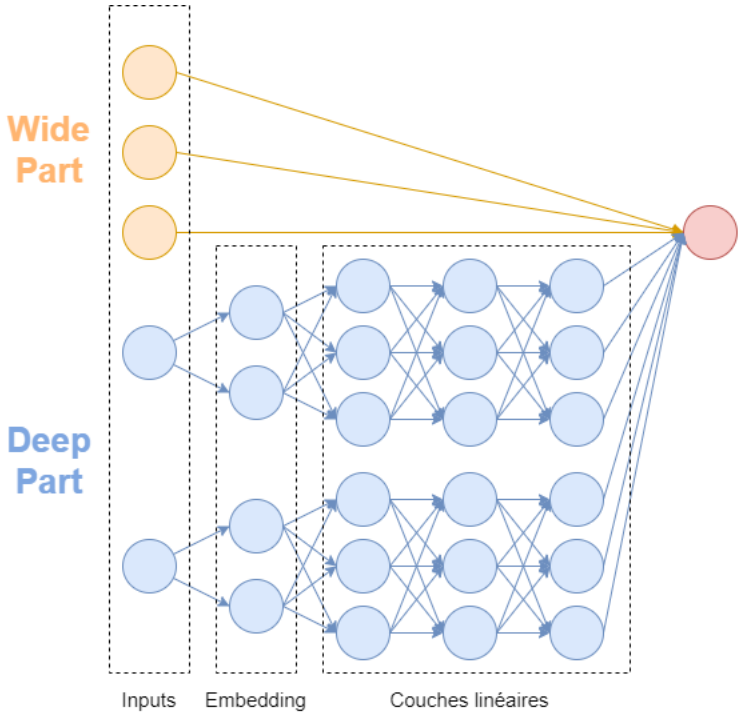

Figure 2.2 Schéma de l'architecture *Wide & Deep* [3] (inspiré du schéma de l'article)

la prédiction du taux de clic, afin que soient prises en compte les interactions explicites et implicites.

Limitations de l'approche par réseaux de neurones

Malgré un succès indéniable, l'utilisation de l'apprentissage profond pour la recommandation peut avoir des inconvénients. Zhang et al. [60] en citent trois :

1. Interprétabilité : la difficulté d'interpréter des prédictions provenant de réseaux de neurones fait qu'ils sont souvent qualifiés de boîte noire, ce qui rend impossible la possibilité d'expliquer comment a été faite une recommandation.
2. La quantité de données requise : la plupart des modèles requièrent beaucoup de données pour pouvoir fonctionner convenablement, ce qui peut poser problème pour des applications à plus petite échelle. Par exemple dans notre cas, un cinéma indépendant peut ne pas recueillir la même quantité de données qu'une entreprise qui en possède une centaine, ce qui peut limiter l'utilisation de modèles avec beaucoup de paramètres.
3. Le nombre d'hyperparamètres à régler : ce problème est aussi valable pour certains algorithmes d'apprentissage machine, mais les réseaux de neurones sont en général particulièrement confronté à cette difficulté.

Récemment, Dacrema et al. [66] se sont interrogés sur le réel progrès qu'ont amené ces approches. En collectant l'ensemble des articles récents provenant de quatre conférences sur un sujet précis (la recommandation top-N), les auteurs mettent en avant l'impossibilité de reproduire les résultats de la majorité des papiers, à cause d'un code ou d'une base de données non accessible. De plus, certains articles qui prétendaient avoir atteint l'état de l'art en termes de performance se retrouvent (après reproduction) être moins bons que des algorithmes références plus basiques. Au-delà du manque de rigueur, les auteurs mettent également en avant la nécessité d'avoir des bases de données et des algorithmes de références plus globaux, connues et accessibles par tous, comme c'est le cas dans la reconnaissance d'image par exemple (avec la base de données MNIST [59] et les algorithmes AlexNet [6], ResNet [32], etc.)

2.4.2 Systèmes de recommandation et modèles de choix discret

Dans leur article, Chen et al. [67] relèvent que la majorité des systèmes de recommandation ne sont pas axés sur les processus de prise de décision. En effet, l'ensemble des méthodes usuelles en recommandation ne font aucunes hypothèses sur la façon dont est prise la décision, et les liens entre ces deux domaines sont encore trop faibles. La plupart des systèmes de recommandation, malgré leur efficacité, ne permettent pas d'expliquer pourquoi la recommandation plairait à l'utilisateur. Par exemple, la seule explication possible que serait en mesure d'apporter un algorithme de filtrage collaboratif serait de donner les individus jugés similaires par le modèle, ce qui ne permet pas d'expliquer pourquoi un utilisateur préfère un item à un autre. Ainsi, la grande majorité du domaine de la théorie de la décision reste à explorer pour les systèmes de recommandation.

À notre connaissance et soulignée récemment par Mottini et al. [68], la littérature concernant la combinaison de modèles de choix discrets pour la recommandation est encore faible. Pourtant comme le souligne Saavedra et al. [69], le rôle d'un système de recommandation pouvant être interprété comme de la prédiction de choix, les modèles de choix discrets peuvent s'appliquer très naturellement à de tels systèmes. L'idée est apparue la première fois dans une thèse de Chaptini [70], puis a été reprise plus récemment par Saavedra et al. [69] puis par Mottini et al. [68].

L'article de Saavedra et al. [69] part de deux constats :

1. les systèmes de recommandation manquent de fondement théorique qui permettraient d'expliquer pourquoi un algorithme est plus efficace qu'un autre
2. les méthodes collaboratives, qui sont les plus populaires, ne déduisent la préférence qu'à partir d'anciennes notes qu'ont données des utilisateurs. Pourtant, en pratique,

la préférence peut aussi dépendre de la qualité intrinsèque des items, du contexte dans lequel le choix est fait, etc.

Pour pallier ces deux limites, les auteurs proposent d'utiliser des modèles de choix décrit à la section 2.3. Appliqués à des choix de snacks, les modèles MNL et Logit Mixtes donnent des résultats plus satisfaisants que deux algorithmes de filtrage collaboratif (un modèle *user-based* et un modèle de factorisation de matrice).

En reprenant cette idée, Mottini et al. [68] établit aussi le lien entre les modèles de choix et les systèmes de recommandation. L'objectif ici est la recommandation de trajet en avion : après avoir montré que choix discrets et recommandation étaient deux problèmes équivalents, les auteurs reprennent l'idée de Saavedra et al. [69], mais en l'appliquant avec des modèles de choix utilisant des algorithmes d'apprentissage automatique (voir section 2.3.4). Ainsi, les auteurs établissent la comparaison entre le modèle MNL, LC-MNL, la version de Lhérithier et al. [49] utilisant des forêts aléatoires, et la version de Mottini et al. [56] utilisant de l'apprentissage profond. Les auteurs montrent alors que les forêts aléatoires sont la meilleure alternative. En effet, bien qu'utilisant un modèle d'apprentissage, ce modèle reste plus interprétable qu'un système de recommandation classique, sans pour autant perdre en performance de prédiction, comme ça peut être le cas avec un simple MNL. Un compromis entre explicabilité et performance est ainsi mis en avant ici.

2.5 Conclusion de la revue de littérature

Pour notre problématique, la prédiction de localisation d'un client dans une salle de spectacle à partir de son historique, deux catégories de critères sont à prendre en compte : la position des sièges dans la salle (2.1.1), mais aussi son interaction avec le remplissage à un instant donné, pour prendre en compte la proximité aux autres (2.1.2). De plus, il sera essentiel de faire un modèle qui prenne en compte à la fois l'hétérogénéité du comportement de chaque individu [17], mais qui a aussi la capacité nécessaire pour prendre en compte des biais plus généraux qui interviennent lors de la prise de décision (2.1.4).

À partir de ce constat, il est possible d'aborder notre problème avec trois angles différents :

1. un point de vue purement *data-driven* qui utiliserait des méthodes de classification en *Machine Learning/Deep Learning*,
2. un traitement des données plus fin par utilisateur qui utiliserait des systèmes de recommandation,
3. un point de vue plus théorique basé sur la théorie de choix de consommateur en utilisant des modèles de choix discrets.

Chacun de ses modèles possèdent des forces et des faiblesses (décrit tout au long de ce chapitre), pour autant ils ne sont pas disjoints. Il existe des méthodes qui permettent de combiner deux approches parmi les trois cités : la combinaison de l'apprentissage machine avec des systèmes de recommandation est aujourd'hui pratique courante (2.4.1), et celle entre apprentissage machine et modèles de choix discrets commence aussi à se démocratiser (2.3.4). Pour la troisième combinaison, la littérature concernant l'alliance de modèle de choix discret et de recommandation est plus pauvre (2.4.2), pour autant le lien entre les deux domaines est naturel, et semble offrir de nouvelles perspectives.

Il est aussi intéressant de constater que pour combiner modèle de choix discrets ou système de recommandation avec l'apprentissage profond, les architectures qui fusionnent deux sous-modèles donnent des résultats encourageant dans les deux cas. Même si l'application et les données sont différentes, un parallèle peut être fait entre la démarche de Sifringer et al. [53] et celle de Cheng et al. [3] : les deux utilisent des réseaux de neurones pour gagner en capacité et en généralisation, mais combinent leur réseau avec un autre pour s'assurer d'un compromis (l'interprétabilité dans un cas, et la mémorisation dans l'autre, mais ces deux propriétés semblent assez équivalentes).

Très peu de recherches sont directement appliquées à notre problème initial. Les plus proches (dans l'application) sont Huang et al. [24], dont l'approche est purement *Deep Learning*, et Blanchard et al. [17] qui adopte plutôt celle des choix discrets. [5] C'est pourquoi il a été nécessaire d'étudier divers problèmes analogues, pour nous aider dans le positionnement de notre problème.

L'intersection de ces trois domaines n'est pas impossible, et le seul exemple existant à notre connaissance est l'article de Mottini et al. [68] appliqué au transport aérien. Un objectif de ce mémoire est de montrer que le problème de prédiction de choix de sièges peut aussi être à l'intersection de ces trois domaines.

5. Ce modèle pourrait aussi être interprété comme étant un système de recommandation, car il prend en compte l'hétérogénéité des utilisateurs.

CHAPITRE 3 DÉTAILS DE LA SOLUTION

On suppose que nous avons un ensemble de N clients ayant déjà réservé un siège par le passé. Pour chaque client $i = 1, \ldots, N$, on note K_i le nombre de choix présents dans la base de données, ce qui correspond à K_i configurations de sièges disponibles dans la salle, avec les positions des sièges choisis correspondants. L'objectif est de construire une méthode capable de généraliser ces choix, c'est-à-dire de prédire la position d'un siège choisi en fonction d'une liste de sièges disponibles.

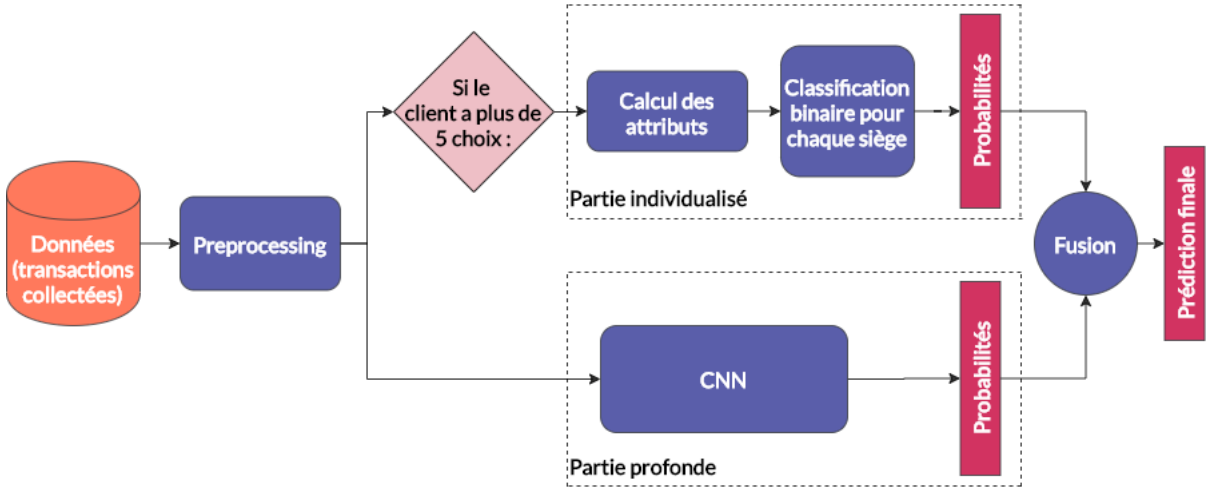

Figure 3.1 L'architecture proposée pour la recommandation de sièges.

Pour ce faire, nous adoptons une approche qui combine une partie basée sur une prédiction individualisée avec une autre profonde, schématisée Figure 3.1. Nous commençons par décrire successivement les deux parties qui composent notre architecture et qui pourraient être des méthodes de prédiction à part entière, avant de s'intéresser à la combinaison. Enfin, la section 3.4 abordera la façon dont on peut traiter les réservations de plus d'un siège, lorsqu'un couple ou un groupe achète une place.

3.1 Composante individuelle

Cette composante a pour objectif de considérer les différences de comportements qui existent entre les consommateurs lorsqu'ils choisissent leurs emplacements. Nous pouvons donc adopter ici le point de vue de systèmes de recommandation, néanmoins deux contraintes nous empêchent de considérer les approches les plus courantes. D'abord, tous les sièges ne sont pas disponibles en tout temps, et chaque choix est fait à partir d'un sous-ensemble des items

disponibles. De plus, ce sous-ensemble peut dépendre du nombre de sièges que le client souhaite réserver : par exemple, pour des réservations de deux sièges, le choix se limite aux paires de sièges disponibles. Ensuite, les sièges (qui sont les items du système de recommandation) interagissent de par leur disponibilité. Avec l'influence de la proximité aux autres, l'attractivité d'un siège dépend de la disponibilité de ceux qui sont autour. De ce fait, toute modélisation par une matrice *user-item*, comme elle n'indique que les interactions entre client et items sans considérer la disponibilité des autres sièges, ne peut prendre en compte que les facteurs liés à la géométrie et non à la proxémie.

Ainsi, les modèles de choix discrets décrits en Section 2.3 ont l'avantage de fournir un système de recommandation basé sur le contenu, qu'il sera possible de contraindre aux sièges disponibles et qui pourra modéliser l'influence de la proximité aux autres grâce à des attributs adaptés.

3.1.1 Espace d'attributs

Pour chaque client i et chaque siège j disponibles dans les salles dans laquelle i a fait un choix, nous calculons un vecteur d'attributs $\mathbf{f}_{i,j}$ en reprenant l'espace théorique suggéré par Blanchard et al. [17], qui prend en compte les facteurs détaillés en Section 2.1. Ces attributs sont regroupés en trois catégories :

1. **Attributs liés à la position (POS)** : cinq variables normalisées indiquant la position globale dans la salle. Pour chaque siège j dans une salle de taille $w \times h$, la position est modélisée par une forme quadratique de ces deux coordonnées (x_j, y_j) :

$$\mathbf{f}_{i,j}^{(POS)} = \left(\frac{x_j}{w}, \frac{y_j}{h}, \frac{x_j^2}{w^2}, \frac{y_j^2}{h^2}, \frac{x_j y_j}{hw} \right) \tag{3.1}$$

2. **Attributs liés à l'espace personnel (EP)** : sept variables binaires indiquant si les sièges autour du siège j sont occupés ou non : à gauche, à droite, à gauche et à droite simultanément, devant, derrière, les deux sièges diagonaux avant et les deux sièges diagonaux arrières.

3. **Attributs liés à la densité (R_p)** : la proportion des sièges à une distance de plus d'un siège. Pour $p = 2$ et $p = 3$, le ratio des sièges disponibles distants de p sièges sur l'ensemble des sièges distant de p sièges est calculé pour chaque position. En terme mathématique, si $P^{(k)}$ désigne l'ensemble des sièges disponibles d'une salle k et $S_p(x, y) = \{(r, s) | \max(|x - r|, |y - s|) = p\}$ [1] désigne les sièges distants de p sièges

1. la dépendance en k est intentionnellement omise ici pour plus de clarté

d'une position (x, y) avec la norme infinie, alors ces attributs s'écrivent :

$$\mathbf{f}_{i,j}^{(R_p)} = \frac{|S_p(x_j, y_j) \cap P^{(k)}|}{|S_p(x_j, y_j)|} , \quad p = 2, 3 \tag{3.2}$$

On obtient ainsi 15 attributs pour chaque siège disponible de chaque salle dans laquelle un choix a été effectué.

3.1.2 Entraînement et évaluation

Pour chaque siège dont les attributs ont été calculés, on introduit une variable binaire $y_{i,j}$ indiquant si le client i choisi ou non le siège j, afin d'entraîner un modèle de classification binaire et obtenir une estimation de $p_{i,j} = p(y_{i,j}|\mathbf{f}_{i,j})$, conformément à un modèle de choix basé sur de l'apprentissage comme introduit dans la revue de littérature (2.3.4). Les méthodes de classification qui seront testées sont la régression logistique, les SVMs, les forêts aléatoires, et les GBTs. Les détails pour chaque algorithme sont donnés dans la section sur les détails expérimentaux (4.1). Chaque client sera ici traité de façon indépendante, c'est-à-dire que le modèle sera entraîné séparément sur chacun d'entre eux. Ainsi, cela nous assure de prendre en compte les différences de comportement de chaque client, en d'autre terme l'importance accordée à chacun des attributs.

Pour l'évaluation des modèles, la méthode usuelle qui consisterait en la définition d'une valeur seuil de $p_{i,j}$ serait inefficace ici, car l'on sait d'avance combien de sièges vont être choisis dans la salle, c'est-à-dire le nombre de sièges dans la classe positive. En d'autres termes, les données ne sont pas indépendantes, et la prise en compte de la dépendance des labels va se faire lors du calcul des prédictions. Pour chaque salle k, les prédictions sont déduites des valeurs $p_{i,j}$:

$$\hat{j} = \underset{j}{\operatorname{argmax}}\, p_{i,j} \tag{3.3}$$

Plus généralement, pour calculer la précision Top-N sur une salle, il suffit de considérer les positions j correspondantes aux N plus grandes valeurs de $p_{i,j}$.

Il est important de noter que chacune des salles doit être considérée séparément lors de l'évaluation pour déduire les prédictions top-N, contrairement à l'entraînement qui se fait sur les $\mathbf{f}_{i,j}$ de l'ensemble des salles pour un client i donné. De plus, toutes les salles en entrée ne sont pas représentées de manière égale lors de l'entraînement, car elles ne fournissent pas le même nombre de données, celui-ci correspondant au nombre de sièges disponible. Ainsi, plus une salle aura de sièges disponibles, plus elle fournira de données pour la classification.

Également, il peut être intéressant d'interpréter la valeur $\max_j p_{i,j}$ comme un indicateur d'at-

tractivité des sièges restants. En effet, pour un même classificateur prédisant le siège $\hat{\jmath}$ pour une salle donnée, il est possible que $p_{i,\hat{\jmath}} = 0.1$ ou $p_{i,\hat{\jmath}} = 0.7$: dans les deux cas, le siège prédit sera le même, mais l'on peut supposer les choix disponibles étaient moins attractifs dans le premier cas que dans le deuxième.

3.2 Composante générale profonde

Bien que le modèle basé sur le choix décrit précédemment soit conçu pour estimer les préférences individuelles, elle requiert une quantité de données suffisante par client. À l'inverse, agréger les données de l'ensemble des clients permet d'utiliser des modèles avec plus de capacité comme de l'apprentissage profond, qui permettrait de modéliser des interactions d'ordre plus élevé qui serait impossible d'anticiper à partir d'attributs théoriques.

3.2.1 Description de l'architecture

Ici, les données en entrée seront directement les salles, que l'on peut modéliser comme un tableau avec deux dimensions, formé de coefficients indiquant si chacun des sièges est occupé ou non. On note $\mathbf{x}^{(k)} = \left(x^{(k)}_{r,c}\right)_{\substack{1<r<h \\ 1<c<w}}$ la représentation de la salle associée au choix k [2], avec $x^{(k)}_{r,c} = 1$ si le siège de la ligne r et colonne c est disponible et 0 sinon.

En d'autres termes, nos données en entrée ont une structure similaire à celle d'images binaires, d'où la possibilité d'adapter aisément un CNN pour modéliser ces interactions spatiales. Ainsi, notre architecture alterne entre des couches de convolutions, d'activation et de *pooling*, avant de terminer par une couche linéaire avec une fonction softmax en sortie. Ici, les propriétés de connectivité locale et de partage des paramètres décrites en introduction (1.1.2) semblent adaptées à la modélisation de la proximité aux autres, qui est une propriété supposée d'invariance spatiale (identique pour chaque siège). Ensuite, la couche linéaire permet de modéliser les facteurs concernant la salle plus généralement, comme la position du siège.

Pour la dimension de la couche de softmax en sortie, celle-ci doit être égale à l'ensemble des choix possibles pour un client, autrement dit au nombre de sièges dans la salle $h \times w$. La sortie a donc la même dimension et la même structure que l'entrée : de ce fait, il est possible d'ajouter une seconde partie de "déconvolution" à notre réseau, dédiée à reconstituer la structure de la salle en entrée. Le schéma global de l'architecture est représenté figure 3.2 : l'architecture de la partie convolution est reproduite avec une structure en miroir, et en remplaçant le pooling par du sur-échantillonage, comme suggéré par Odena et al. [35].

2. Ici, les clients n'étant pas distingués, on a $k = 1, \ldots, K = \sum_{i=1}^{N} K_i$ (le nombre total de choix dont on dispose).

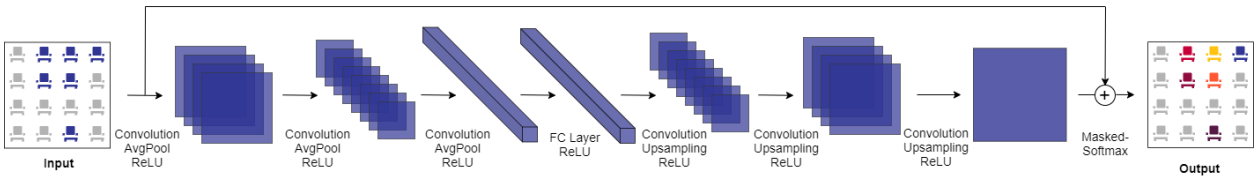

Figure 3.2 Schéma de notre modèle avec déconvolution pour reformer la salle en sortie.

3.2.2 Softmax masqué

Une observation pour améliorer la prédiction est le fait que le modèle devrait attribuer une probabilité nulle aux sièges occupés de chaque entrée $\mathbf{x}^{(k)}$. Cette restriction va s'effectuer lors de la couche de softmax, où le calcul ne sera fait qu'à partir des sièges disponibles. Habituellement, la fonction de perte utilisée pour l'optimisation est la log-vraisemblance de la couche de softmax, qui peut s'écrire :

$$\mathbf{o}_j = -\log\left(\frac{\exp(\mathbf{u}_j)}{\sum_{\ell=1}^{\bar{J}}\exp(\mathbf{u}_\ell)}\right) = -\mathbf{u}_j + \log\left(\sum_{\ell=1}^{\bar{J}}\exp(\mathbf{u}_\ell)\right), \quad j = 1,\ldots,\bar{J} \tag{3.4}$$

avec $\mathbf{u}$ le vecteur correspondant à la sortie de la couche linéaire, et $\bar{J} = h \times w$ la dimension de la sortie. Avec une telle sortie, on observe que tout les sièges de la salle se verront octroyer une probabilité non nulle, alors que l'on sait que certains sont déjà réservés et ne peuvent pas être choisis.

Ici, on remarque que chaque tableau binaire $\mathbf{x}$ en entrée peut servir de masque ici, pour que chaque siège avec les coordonnées (r, c) ne soit considérée que si $x_{r,c} = 1$. Pour ce faire il suffit de modifier le vecteur $\mathbf{u}$ avant le calcul du softmax de la façon suivante :

$$\tilde{\mathbf{u}}_j = \mathbf{u}_j + \log(\mathbf{x}_j + \epsilon), \quad \epsilon \ll 1, \tag{3.5}$$

avec ϵ utilisé pour rester dans le domaine de définition de la fonction logarithme. Avec ce changement, la sortie devient :

$$\tilde{\mathbf{o}}_j = -\log\left(\frac{\exp(\mathbf{u}_j)\cdot(\mathbf{x}_j+\epsilon)}{\sum_{\ell=1}^{\bar{J}}\exp(\mathbf{u}_\ell)\cdot(\mathbf{x}_\ell+\epsilon)}\right). \tag{3.6}$$

De cette façon, on obtient $\tilde{\mathbf{o}}_j \approx 0$ si $\mathbf{x}_j = 0$ et pour un ϵ suffisamment petit, et le softmax n'est calculé qu'à partir des sièges disponibles.

3.3 Modèle combiné

Les deux sections précédentes décrivent deux méthodes très différentes pour prédire quel siège va être choisi. Une comparaison des propriétés attendues pour les deux modèles montre qu'elles peuvent être complémentaires dans l'information apportée. Inspirés par des avancées récentes dans les systèmes de recommandations avec des modèles *Wide & Deep* [3, 64, 65] mais aussi les avancées dans l'utilisation de l'apprentissage profond pour les modèles de choix discrets [53, 54], nous verrons qu'une combinaison pourrait s'avérer ici performante.

3.3.1 Comparaison des deux composantes

Un résumé de la comparaison du modèle individualisé avec le modèle profond est représenté Tableau 3.1. La plupart des caractéristiques de la composante individuelle s'expliquent par la construction basée sur des recherches théoriques en psychologie et proxémie pour modéliser la prise de décision. C'est ce qui rend ce modèle *theory-driven*, qui lui octroie des propriétés d'interprétabilité et de mémorisation [3] , mais également qui le restreint à des interactions d'ordre plus faible que la composante profonde. Cependant, ces propriétés sont plus nuancées à cause de l'utilisation de méthodes *data-driven* provenant de l'apprentissage automatique, et tout ceci va également dépendre de l'algorithme de classification qui sera choisi à la sous-section 4.1. Par exemple, la régression logistique va en effet modéliser des interactions d'ordre faible et être interprétable, tandis que les forêts aléatoires auront plus de capacité, mais seront plus complexes à interpréter, malgré des progrès récents dans le domaine [45, 48].

À l'inverse, la composante profonde est purement *data-driven*, et n'utilise aucuns prérequis sur les facteurs qui impactent la prise de décision. Cependant, cette propriété est acquise au prix d'une agrégation de données qui lisse les comportements des clients.

Composante individuelle	Composante profonde
Theory-driven	*Data-driven*
Nécessite des attributs théoriques	Aucun espace d'attribut requis
Utilise uniquement l'historique d'un client	Utilise toutes les données
Considère les différences de comportements	Données de clients agrégées
Interactions d'ordre faible	Interactions d'ordre élevé
Propriété de mémorisation	Propriété de généralisation
Interprétabilité [48]	Non interprétable

Tableau 3.1 Tableau récapitulatif des propriétés des deux composantes.

Ainsi, ces deux composantes, de par la différence des données fournies et la différence de

3. Terme utilisé pour les systèmes de recommandation avec apprentissage profond (voir 2.4.1).

capacité d'apprentissage, vont utiliser des caractéristiques différentes pour la prédiction : la partie profonde va se concentrer sur une attractivité intrinsèque à chaque siège, c'est-à-dire uniquement en fonction des disponibilités et au-delà des préférences individuelles. L'autre composante va utiliser des attributs dont certains sont aussi liés à la géométrie de la pièce, mais pour une estimation biaisée sur le comportement individuel.

Enfin, l'objectif de combiner ces deux modèles n'est pas uniquement dans le but d'améliorer la performance de prédiction, mais aussi d'être plus flexible à la quantité de données par client. En effet, on peut s'attendre à ce que la composante individuelle soit suffisamment performante lorsqu'elle dispose d'un historique suffisamment grand, en revanche elle est confrontée au problème du démarrage à froid (*cold-start problem*), et donc la composante profonde sera plus pertinente pour les clients avec très peu voire aucun choix enregistré par le passé. De ce fait, il est important que la combinaison soit différente pour chaque client, car la performance des deux modèles peut varier significativement en fonction de chaque client.

3.3.2 Combinaison

Un moyen simple de combiner nos deux modèles distinctement pour chaque client est par une combinaison convexe des deux sorties :

$$\mathbf{o}_j = f\left(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)}\right), \quad \alpha \in [0,1] \tag{3.7}$$

avec f la fonction sigmoïde, $\mathbf{o}_j^{(I)}$ la sortie de la composante individuelle, et $\mathbf{o}_j^{(P)}$ celle de la composante profonde. $\mathbf{o}^{(I)}$ étant formé de probabilités issues de classification binaire, le vecteur est mis à l'échelle linéairement pour sommer à un. Le paramètre α, restreint à l'intervalle $[0,1]$ par une fonction de *clamp* ($g : x \mapsto \min\{1, \max\{0, x\}\}$) est appris pour chaque individu par descente de gradient sur la log-ressemblance de $\mathbf{o}$. L'entraînement de la combinaison est séparé de celui des deux composantes.

La fonction sigmoïde est appliquée afin d'éviter une explosion des gradients : sans cette fonction, on aurait $\nabla_\alpha - \log\left(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)}\right) = \frac{\mathbf{o}_j^{(I)} - \mathbf{o}_j^{(P)}}{\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)}}$, et donc ce gradient n'est pas borné car le dénominateur peut être très faible voire être nul.

Avec la fonction sigmoïde, on a $f'(x) = f(x)(1 - f(x))$, et donc on obtient :

$$\begin{aligned} \nabla_\alpha - \log(\mathbf{o}_j) &= \nabla_\alpha - \log\left(f(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)})\right) \\ &= \frac{f(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)})\left(1 - f(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)})\right)}{f(\alpha \mathbf{o}_j^{(I)} + (1-\alpha)\mathbf{o}_j^{(P)})} \\ &= 1 - \mathbf{o}_j \quad \in [0,1] \end{aligned} \tag{3.8}$$

Donc dans l'exemple d'un algorithme de descente de gradient stochastique basique avec un exemple ayant pour label y, la mise à jour s'écrit :

$$\alpha \longleftarrow \alpha - \gamma(1 - \mathbf{o}_y) \tag{3.9}$$

Pour les clients n'ayant pas une quantité de données suffisante pour être traitée par la partie individuelle, la valeur de α est fixé à 0, et la prédiction ne s'effectue qu'à partir de la partie profonde. Ainsi, tous les clients peuvent être traités par ce modèle.

3.4 Gestion des choix multiples

Jusqu'ici, notre approche aborde ce problème dans le cadre d'un choix d'un seul siège dans une salle, mais il est très fréquent que plusieurs soient choisis dans une seule réservation, en particulier dans des cinémas, des théâtres ou des salles de concert.

Sans perte de généralité, nous considérons pour illustrer le cas d'une recommandation de deux sièges. De façon quasi systématique, des clients choisissant deux sièges les choisissent côte à côte sur la même ligne : c'est par exemple le cas de plus de 99% des réservations de couple dans la base de données de la salle de concert étudié ici. De ce fait, en ajoutant comme contrainte que les deux sièges doivent être cote à côte, il suffit de prédire le siège le plus à gauche pour en déduire ensuite le deuxième siège réservé, qui sera à sa droite. De cette manière, le problème est ramené à la prédiction d'un seul siège, en l'occurrence celui de gauche dans la paire. Cependant, les données en entrée doivent être modifiées, car ne doivent être considérés disponibles que les sièges ayant leur voisin de droite également disponible. Cette transformation de l'input est illustrée dans la partie inférieure de la figure 3.3 : les sièges qui sont à gauche de sièges indisponibles (ou inexistant car sur le bord) sont considérés comme indisponibles, et parmi les deux labels, seul celui de gauche est conservé.

Cette transformation des données va modifier le calcul des attributs $\mathbf{f}_{i,j}$ pour la composante individuelle : les attributs liés à la position vont utiliser la position du siège de gauche, et

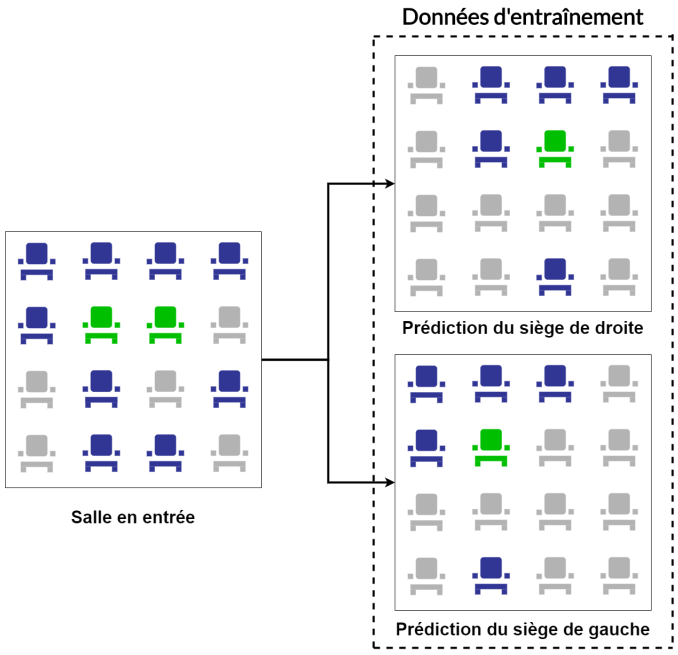

Figure 3.3 Illustration de la transformation d'une salle en entrée pour la prédiction d'une paire de sièges (les sièges disponibles sont en bleu, et un exemple de choix est donné en vert)

les attributs liés à l'espace personnel et la densité vont être calculés comme si les deux sièges avaient fusionné en un seul : par exemple avec le choix sur la Figure 3.3, on observe que le siège à gauche de la paire de sièges reste libre après transformation, et celui de droite reste occupé. Pour ce qui est des sièges au-dessus et en dessous, les variables étant binaires, il serait impossible d'avoir 0.5 bien qu'une place sur les deux soit occupé en dessous par exemple. On observe donc qu'avec cette transformation, les deux sièges du dessus (ou dessous) doivent être disponibles pour que la paire soit considérée disponible.

Cette transformation peut être réalisée de façon symétrique pour le siège de droite, ce qui est représenté sur la partie supérieure de la Figure 3.3. Ainsi, il est possible d'obtenir deux entrées simples à partir d'un choix double, ce qui est donc un moyen de faire de l'augmentation de données pour la composante profonde.

Pour l'évaluation, les deux transformations peuvent être calculées et combinées pour obtenir la prédiction finale : en effet, l'attractivité de deux sièges formant une paire étant estimée séparément, la probabilité qu'ils soient choisis est donc différente, et la probabilité de la paire pourrait être la moyenne de celle des deux sièges. Le calcul effectué est illustré figure 3.4 : la prédiction est effectuée séparément pour les sièges de gauche et de droite, puis avant de faire la moyenne terme à terme, il est nécessaire de translater une des deux prédictions (ici celle de gauche) d'un cran, afin que soit combiné les sièges correspondants à la même paire.

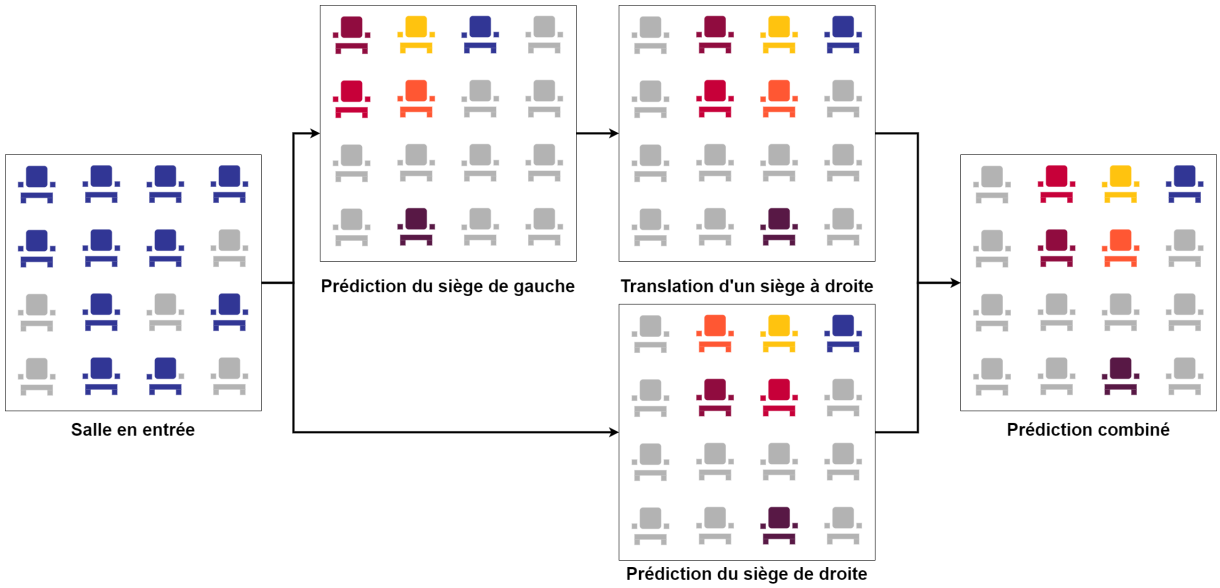

Figure 3.4 Illustration des transformations effectuées pour évaluer la prédiction d'une paire de sièges. La combinaison s'effectue grâce à une moyenne terme à terme des deux probabilités, et la prédiction finale de chaque siège est représenté ici sur le siège choisi le plus à droite.

La même opération peut être effectuée dans le cas de plus de deux sièges, si l'on suppose que tous sont alignés côte à côte : pour chacun des sièges dans la commande, il suffit de calculer le nombre de sièges requis sur la gauche et sur la droite pour déterminer si un siège est disponible ou non. Lors de l'évaluation des modèles, nous nous restreindrons aux cas d'un ou deux sièges.

CHAPITRE 4 RÉSULTATS EXPÉRIMENTAUX

Ce chapitre présente les résultats obtenus sur deux expériences différentes : la première utilise des données générées par Blanchard et al. [17], et la seconde est à partir de données provenant d'une salle de concert nord-américaine. Avant de les détailler successivement, nous présentons les détails expérimentaux communs aux deux expériences, à savoir les métriques utilisées pour mesurer la performance et les modèles comparés. L'ensemble de l'implémentation a été effectué en Python avec les librairies Scikit-learn et Pytorch.

4.1 Détails expérimentaux

Pour juger la performance de notre modèle hybride, nous le comparons à ces deux composantes utilisées indépendamment, une méthode de base, et les résultats de [17] pour leurs données. Chaque modèle est évalué sur un ensemble de test, et l'ensemble d'entraînement est divisé en entraînement/validation avec un ratio de 0.3, pour l'estimation des hyperparamètres.

4.1.1 Métriques

Deux types de métriques sont utilisés ici pour l'évaluation : la première est la précision Top-N, qui mesure la proportion de prédictions correctes parmi les N premiers choix d'un modèle, et la deuxième est lié à la distance moyenne entre les prédictions et le label. Pour chaque salle, elle peut être interprétée comme l'espérance la fonction de perte $L1$ ($L1$-*loss*), et s'écrit :

$$l^{(L1)} = \sum_{j \in P^{(k)}} p_{i,j}(|x_j - x| + |y_j - y|) \tag{4.1}$$

avec (x_j, y_j) les coordonnées du siège j, (x, y) celles du siège choisi (c'est-à-dire le label), et $p_{i,j}$ la probabilité que le client i choisisse le siège j. Pour les modèles individuels et hybrides, ces probabilités sont obtenues par une mise à l'échelle linéaire de celles obtenues en sortie, et pour la partie profonde, cela correspond à la valeur du softmax masqué. La moyenne des $l^{(L1)}$ sur l'ensemble des salles est ensuite calculé pour obtenir la valeur finale.

Ici, contrairement à la précision Top-N, la distribution des probabilités $p_{i,j}$ influe sur la valeur de la $L1$-*loss* : en d'autres termes, des probabilités très uniformes ne changeront pas la précision top-N pour un classement identique, tandis que cela va avoir tendance à accorder plus d'importance à des sièges pouvant être éloignés, et ainsi augmenter la valeur de $l^{(L1)}$.

Cependant, une limite de cette métrique est la non-prise en compte de la configuration de la salle, car la distance au siège choisi est la même peu importe si beaucoup de sièges sont libres ou non.

4.1.2 Modèles

Modèle référence

Le modèle de choix MNL décrit à la sous-section 2.3 est implémenté pour chaque expérience en guise de méthode de base. Ce modèle est entraîné distinctement sur chaque client, c'est-à-dire que chaque client aura sa propre estimation des coefficients $\boldsymbol{\beta}$. L'estimation est effectuée par maximisation de la log-vraisemblance, à partir de l'implémentation de la librairie Python StatsModels.

Modèles individualisés

Quatre algorithmes de classification binaires, introduits à la section 1.1.1, sont comparés pour la partie individualisée. Nous listons ici l'ensemble des hyperparamètres optimisés pour chaque jeu de données. L'ensemble des valeurs ayant permis d'obtenir les résultats présentés ici est résumé en annexe. Comme notre problème est déséquilibré (beaucoup plus de sièges non choisis que de sièges choisis pour chaque salle), pour l'ensemble de ces méthodes, le poids des classes 0 et 1 est changé de telle sorte à être inversement proportionnel à la fréquence de ces deux classes, pour prendre en compte ce déséquilibre.

— **Régression logistique (LR)** : l'optimisation est effectuée en utilisant une méthode de Newton. Malgré des essais pour ajouter une pénalité sur les poids de la régression, les résultats montrent que dans l'ensemble de nos expériences, ne pas mettre de pénalité sur les poids améliore la performance. Il est intéressant de noter ce modèle possède un nombre de paramètres égal au nombre d'attributs plus un (avec l'ordonnée à l'origine), tout comme le modèle MNL. Pour ces deux modèles qui pondèrent chacun des attributs, un affichage des coefficients estimés suffit à apporter une première analyse de l'influence des différents facteurs.

— **Support Vector Machine (SVM)** : différentes familles de noyaux ont été essayées parmi linéaire, polynomial et fonction de base radiale (*rbf*). Le paramètre de régularisation (C) est également optimisé.

— **Gradient Boosted Trees (GBT)** : GBT est optimisé en termes de profondeur maximale (max_depth), nombre d'estimateurs (n_estimators), nombre minimum de données pour séparer un noeud (min_samples_split), et nombre maximal d'attributs

à considérer pour une séparation (max_features).

— **Random Forest (RF)** : RF est optimisé en termes de profondeur maximal (max_depth), nombre minimum de données pour séparer un noeud (min_samples_split), et nombre maximal d'attributs à considérer pour une séparation (max_features).

Modèles profonds

Pour la partie profonde, nous comparons les performances du CNN avec ou sans déconvolution (CNN et CDNN, pour *Convolutional/Deconvolutional Neural Network*). Les deux modèles utilisent l'algorithme Adam [5] pour l'optimisation et la méthode *early-stopping* pour stopper l'apprentissage lorsque l'erreur sur l'ensemble de validation ne diminue plus. Le nombre d'*epochs* choisi pour attendre une amélioration est mis à 20.

— **CNN** : le nombre de blocs formés de couches de convolution/activation/pooling est optimisé pour chaque expérience, tout comme le taux d'apprentissage d'Adam et la taille des batchs. Le nombre initial de *channels* (qui correspond à la troisième dimension des noyaux) vaut 2, puis ce nombre est doublé à chaque bloc jusqu'au dernier. Les paramètres des noyaux sont configurés de sorte à conserver la dimension, c'est-à-dire que le triplet (taille, *padding*, *stride*) vaut $(3, 1, 1)$ ou $(5, 2, 1)$, et les couches de *pooling* divisent par 2 le nombre de pixels entre chaque bloc.

— **CDNN** : le CDNN est formé de 3×2 blocs, avec une couche linéaire entre les parties convolution et déconvolution. Les trois premiers blocs sont similaires à ceux du CNN, sauf pour le nombre de *channels* qui varie en fonction des données. Les couches de *pooling* sont configurées de telle sorte à avoir un neurone par *channel* à la fin de la première partie. Ensuite, les trois autres blocs sont constitués de couches de convolutions (avec un nombre de *channels* symétrique à la première partie), activation et interpolation, avec un facteur d'échelle configuré pour obtenir la taille originale de la salle en sortie.

Les données étant ici binaires et de taille variant entre 12×12 et 31×57 en fonction du problème, elles sont bien moins complexes que celles utilisées couramment en traitement de l'image par réseau de neurones, où les images sont souvent en couleur et avec plus de pixels (comme ImageNet [30] par exemple). Ainsi, avoir une structure très profonde et un nombre conséquent de blocs n'améliore pas les résultats et n'avait donc aucun intérêt ici.

Modèles hybrides

L'ensemble des combinaisons de méthodes individuelles et profondes ont été essayées pour le modèle hybride. La valeur initiale de α est fixé à 0.5 (c'est-à-dire un poids identique pour les deux composantes), puis la descente de gradient stochastique avec *momentum* est utilisé pour l'optimisation de α. Le taux d'apprentissage, la valeur du *momentum*, et l'importance du terme de régularisation (*weight decay*) sont optimisés ici. Pour éviter un potentiel surapprentissage dû au fait que l'ensemble d'entraînement est déjà utilisé par le modèle individuel, l'entraînement de α est effectué comme les autres hyperparamètres sur l'ensemble de validation.

4.2 Données expérimentales de choix de siège

4.2.1 Description des données

À notre connaissance et à celle de Blanchard et al. [17], il n'existe pas de base de données publique de réservations de sièges qui permette une comparaison de performance de systèmes de recommandations pour ce problème. Les auteurs de [17] ont donc construit des jeux de données à partir d'un panel de participants qui devait choisir successivement des positions dans une salle en fonction des disponibilités. Plusieurs expériences avec différentes conditions ont ainsi été effectuées, avec des participants qui ont réalisé entre 24 et 120 choix selon l'expérience. L'ensemble des données générées et les détails sur ces bases de données sont disponibles en ligne. [1]

Pour illustrer les performances de notre approche par rapport à un état de l'art, nous reportons les résultats de deux jeux de données. Le premier (E4-Concert-Singles.FC) est formé de 463 participants qui ont choisi un siège dans 120 configurations d'une salle de concert. La taille de la salle varie parmi 20×10, 10×20, 10×10 et 20×20, avec un taux d'occupation de 25%, 50%, ou 75% (le remplissage est aléatoire). Afin de pouvoir comparer équitablement avec les résultats énoncés dans [17], nous utilisons les mêmes ensembles d'entraînement et test, avec 115 choix constituant l'ensemble d'entraînement et 5 celui de test. Dans le deuxième jeu de données (E2-Movie-Singles.FC/NC :FC), 300 participants ont choisi 12 emplacements dans un cinéma de taille 12×12 avec un taux d'occupation de 75%. Dix choix sont utilisés pour l'entraînement, et deux pour le test.

Il est important de voir ici que plus la salle considérée est grande, plus le modèle individualisé aura d'échantillons. Par exemple, avec les données E2-Movie-Singles, nous obtenons environs

1. `https://seatmaplab.com/public/locationalchoicedatasets`

0.75 × Taille de la salle × Taille de l'historique = 0.75 × 12 × 12 × 10 = 1080 points pour chaque client. C'est pour cela qu'une adaptation des hyperparamètres par jeu de données est importante ici, car le nombre de choix, mais aussi la taille de la salle fait varier le nombre d'échantillons et le déséquilibre des labels (c'est-à-dire la proportion de sièges choisis parmi ceux disponibles).

4.2.2 Résultats et discussion

La précision Top-N avec $N = 1, 3, 5$ et la perte L1 de l'ensemble des modèles sont reportés dans les tableaux 4.1 et 4.2 pour les deux jeux de données. Pour le modèle de Blanchard et al. [17], les auteurs n'ont reporté dans leur article que la précision Top-1.

Tableau 4.1 Résultat de l'ensemble des modèles pour le premier ensemble des données expérimentales de choix de siège.

		E4-Concert-Singles.FC			
Modèle	**Algorithme**	Top-1	Top-3	Top-5	L1-loss
Baseline	MNL	0.4220	0.7175	0.8138	2.755
Blanchard et al. [17]	ML	0.4570	-	-	-
Individuelle	LR	0.3741	0.6773	0.7983	3.971
	SVM	0.3382	0.6393	0.7546	3.738
	GBT	0.4778	0.7490	0.8462	3.074
	RF	0.4285	0.7274	0.8199	5.798
Profond	CNN	0.2944	0.6063	0.7360	4.947
	CDNN	0.2983	0.6157	0.7446	4.868
Modèle Hybride	GBT+CNN	**0.4834**	**0.7529**	**0.8486**	3.568

Les résultats montrent que le modèle hybride améliore les précisions top-N par rapport à toutes les autres méthodes, ce qui confirme l'intérêt de combiner une prédiction basée sur le choix individuel avec une autre plus générale. Cela prouve donc que la composante profonde, malgré des performances plus faibles lorsqu'elle est utilisée seule, apporte de l'information supplémentaire utile pour améliorer la recommandation. Il est également intéressant de constater que la méthode individuelle utilisée seule était déjà en mesure d'améliorer les résultats de [17], en utilisant le même espace d'attributs. Parmi ces méthodes, GBT semble particulièrement efficace ici. Pour le cas des données E2-Movie-Singles (tableau 4.2), la différence entre les différents algorithmes de classification pour la méthode individuelle est moins

Tableau 4.2 Résultat de l'ensemble des modèles pour le second ensemble des données expérimentales de choix de siège.

		E2-Movie-Singles.FC/NC :FC			
Modèle	**Algorithme**	Top-1	Top-3	Top-5	L1-loss
Baseline	MNL	0.3400	0.6250	0.7317	3.521
Blanchard et al. [17]	ML	0.2867	-	-	-
Individuelle	LR	0.3533	0.6250	0.7317	3.481
	SVM	0.2617	0.5333	0.6900	5.015
	GBT	0.3650	0.6167	0.7350	3.473
	RF	0.3583	0.6267	0.7283	6.211
Profond	CNN	0.1069	0.3536	0.5066	6.306
	CDNN	0.1086	0.3536	0.5093	6.316
Modèle hybride	RF+CNN	**0.4533**	**0.6800**	**0.7833**	6.279

claire. Ceci s'explique par un jeu de données avec une petite salle et un nombre de choix faible, ce qui peut provoquer du surapprentissage pour des modèles complexes comme GBT. Ainsi, cette expérience est un exemple où des modèles plus simples comme LR ou MNL pourraient suffire pour avoir une performance correcte tout en restant très interprétable. Pour autant, la combinaison la plus efficace pour ce jeu de données reste avec les forêts aléatoires, et il est intéressant de constater que c'est avec ces données qu'il y a le plus grand écart entre les modèles individuels et l'hybride (amélioration de 36.5% à 45.33% en top-1). Ainsi, on peut penser que le modèle hybride peut compenser le manque d'information due à un historique trop faible. Pour l'autre jeu de données (E2-Concert-Singles), il semblerait que disposer de 115 choix par clients apporte soit suffisant pour le modèle individuel avec GBT pour faire une prédiction convenable, et l'amélioration avec le modèle hybride est donc moins significative. Pour ce qui est des modèles profonds, les deux architectures que sont CNN et CDNN donnent des performances équivalentes ici. On constate en revanche que la *L1-loss* ne s'améliore pas significativement malgré une amélioration sur les autres métriques : ceci est probablement due au fait que la combinaison avec le modèle profond a tendance a "uniformiser" l'ensemble des probabilités du modèle individuel, ce qui empêche l'amélioration sur cette métrique.

4.3 Données de la salle de concert

4.3.1 Description des données

Nous étudions des données anonymisées provenant d'une salle de concert nord-américaine, formées de transactions collectées sur une période de 12 ans, entre 2006 et 2018. La salle n'étant pas carrée, du *padding* est ajoutée autour de chaque ligne pour conserver une largeur constante et pouvoir représenter la salle comme une matrice carrée de taille $h \times w = 31 \times 57$ (voir la figure 4.1 pour une représentation de la salle).

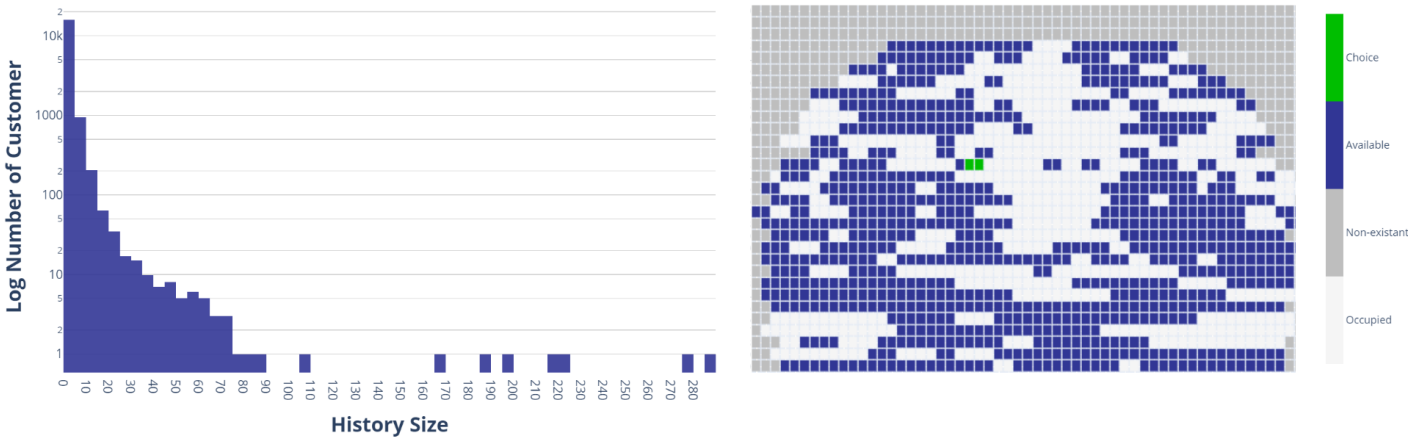

Figure 4.1 À gauche : Histogramme de la taille d'historique pour les données de la salle de concert (échelle logarithmique) À droite : exemple d'une configuration et du choix associé.

Par rapport aux données expérimentales traitées précédemment, des difficultés s'ajoutent ici au traitement de transactions réelles :

— **Le prix** : Dans cette salle (et c'est le cas de beaucoup d'autres), le prix varie en fonction du temps et du siège. Les données collectées au cours du temps donnent l'information sur le prix des sièges réservés, mais il n'y a aucune information concernant celle des sièges non choisis, ce qui empêche de reconstituer la tarification globale pour une performance. Par conséquent, nous ne disposons pas de l'information nécessaire pour considérer le prix dans la prédiction, et le prix restera hors du modèle ici.

— **Occupation de la salle** : il est possible de reconstituer la configuration de la salle à partir des achats de billets au cours du temps, en triant par représentations et en la remplissant au fur et à mesure. Cependant, cette reconstitution sera imprécise, car elle ne prend pas en compte la possibilité de changer de places après la réservation (ces changements ne sont pas collectés), ni les annulations. De plus, il est possible que

la salle de concert décide de ne pas rendre disponibles tout les sièges dès le départ.

— **Abonnements saisonniers** : dans cette salle, les détenteurs d'un abonnement saisonnier conservent le même siège pour toutes les représentations, ce qui rend la recommandation de siège inutile pour ces clients car ils ne le choisissent pas. La prédiction ne s'effectue donc pas sur les abonnements saisonniers.

— **Hétérogénéité du nombre de sièges réservés** : un ou plusieurs sièges peuvent être choisis pour une même représentation. La transformation décrite à la section 3.4 permet de se ramener à une taille de 1 pour tous les choix. Ici, nous nous restreindrons à la sélection d'un ou deux sièges.

Le nombre de réservations initialement dans la base de données était 207738. En résumé, les différentes étapes de traitement pour obtenir les données exploitées finales sont :

1. Restriction à la scène principale (on ne traite pas les balcons et les loges).
   Nombre de réservations conservées : 139623 (67.2%)
2. Suppression des abonnements saisonniers.
   Nombre de réservations conservées : 41423 (29.7%)
3. Restriction à réservations d'un ou deux sièges.
   Nombre de réservations conservées : 30116 (72.7%)
4. Restrictions aux clients ayant au moins effectué cinq réservations.
   Nombre de réservations conservées : 12142 (40.6%)

Afin que l'ensemble des modèles soient comparés sur le même ensemble de test, nous utiliserons le même jeu de données avec une restriction pour les clients ayant un historique suffisamment grand. Ce nombre est mis à 5, et après cette restriction, nous obtenons une base de données de 12142 choix répartis sur 1350 clients. La distribution de la taille d'historique est représentée figure 4.1. Pour les modèles profonds, les modèles d'entraînement et d'évaluation sont obtenus par agrégation de ceux de chaque client étudié.

4.3.2 Résultats et discussion

Cette salle comprenant environ quatre fois plus de sièges que toutes les expériences précédentes, la précision top-N doit être mise à l'échelle en conséquence pour une comparaison équitable, et on pose donc ici $N = 5, 10, 20$. Les résultats reportés tableau 4.3 confirment à nouveau l'utilité d'une architecture hybride, améliorant les performances sur les trois précisions top-N. GBT reste l'algorithme le plus efficace pour la partie individuelle, et contrairement aux expériences précédentes, une différence plus significative existe entre le CNN et le CDNN. D'une manière générale, on peut observer que même en modifiant l'échelle de la

précision top-N, les performances atteintes avec les données générées ne le sont pas ici, ce qui illustre les difficultés énoncées dans la sous-section précédente d'une application avec des données réelles. Il est également important de noter que tous les résultats sont reportés sur les mêmes données, mais qu'un avantage du modèle hybride est d'utiliser le modèle profond dans le cas où des clients ont effectué moins de 5 réservations par le passé.

Tableau 4.3 Résultats pour les données de la salle de concert.

		Salle de concert			
Modèle	**Algorithme**	Top-5	Top-10	Top-20	L1-loss
Baseline	MNL	0.1885	0.2897	0.4160	20.026
Individuelle	LR	0.1782	0.2639	0.3842	18.308
	SVM	0.1227	0.1856	0.2783	23.602
	GBT	0.2366	0.3284	0.4382	19.036
	RF	0.2193	0.3106	0.4287	24.378
Profond	CNN	0.0692	0.1143	0.1886	24.435
	CDNN	0.0907	0.1567	0.2544	22.726
Modèle Hybride	GBT+CDNN	**0.2504**	**0.3356**	**0.4570**	22.164

Cette expérience est celle qui a requis les plus longs temps de calcul : l'entraînement des modèles profond a pris environ deux heures avec une carte graphique Nvidia MX150 et 4Go de RAM, tandis que le modèle hybride accomplit l'ensemble de ses calculs (dont l'entraînement de la partie individuelle) en maximum huit heures. Néanmoins, chaque client étant traité séparément (si l'on assume avoir déjà entraîné les modèles profonds avec les données agrégées), il est possible de paralléliser l'ensemble des calculs pour chaque client, afin d'optimiser les ressources.

Pour illustrer l'évolution de la précision en fonction de l'historique disponible, nous effectuons une deuxième expérience à partir de ces données, mais en ne considérant que les clients ayant au moins 40 choix disponibles pour l'entraînement. Les résultats sont illustrés figure 4.2 : chaque point d'un graphe représente la précision top-N sur l'ensemble de tests si l'on restreint l'ensemble d'entraînement à k choix, avec k variant de 5 à 40. Comme on pouvait l'espérer, on observe que la précision augmente avec la quantité de données disponibles pour le modèle individuel. Cette méthode étant utilisée pour le modèle hybride, ce dernier va également suivre cette augmentation de précision. Cependant, on observe que la différence entre ces deux courbes est plus grande pour de faibles valeurs d'historique (environ moins

de 15). En d'autres termes, le modèle hybride permet de compenser un peu plus le manque d'information pour les clients avec un historique plus restreint.

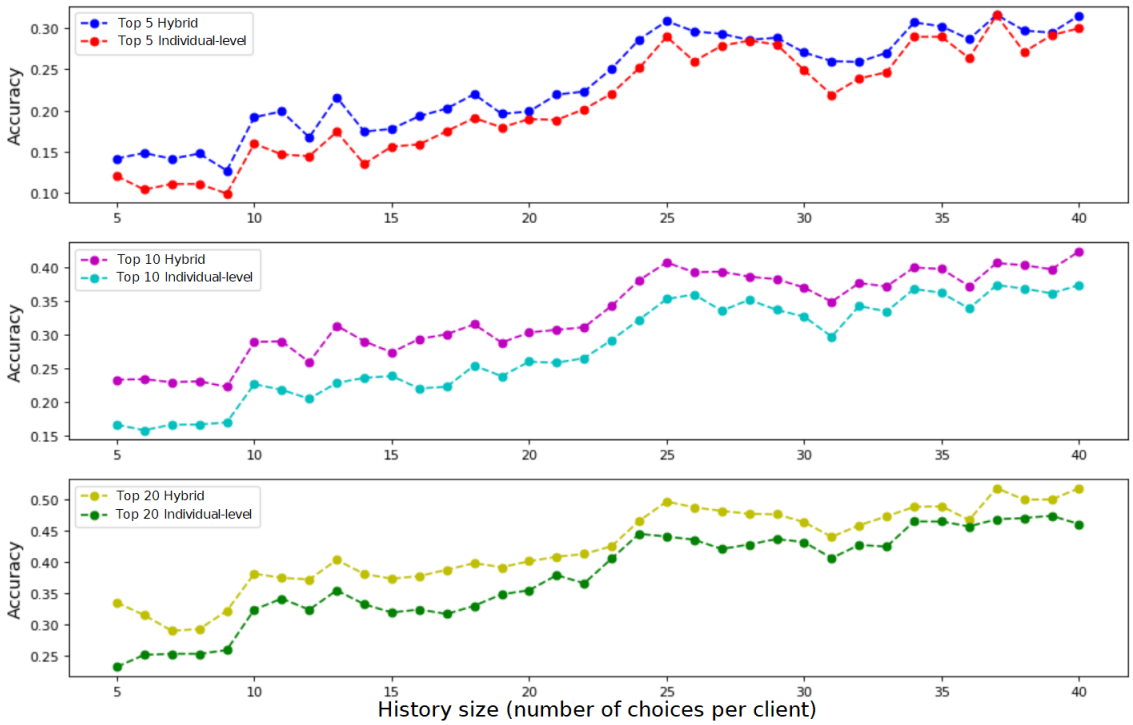

Figure 4.2 Évolution de la précision en fonction du nombre de choix conservé, sur l'ensemble des clients ayant une taille d'historique supérieur à 40 pour les données de la salle de concert.

CHAPITRE 5 CONCLUSION

5.1 Synthèse des travaux

Au problème de recommandation de siège, nous proposons dans ce mémoire un modèle hybride combinant un modèle de choix discret basé sur de l'apprentissage automatique et un réseau de neurones convolutifs. L'intérêt de cette combinaison est double : il permet d'une part d'améliorer les performances globales en ajoutant à un modèle individualisé une prédiction basée sur des attributs généraux, et d'autre part d'être plus flexible sur la quantité de données disponible par client, notamment en résolvant le problème du démarrage à froid. De plus, par une analyse de la partie individuelle qui modélise la prise de décision ainsi que son importance dans la prédiction finale, il est possible conserver un modèle qui reste interprétable si un objectif est l'analyse de comportement lors d'un choix de localisation.

En utilisant des données recueillies pour une étude de choix de placement dans un cinéma ou une salle de concert, nous avons montré que notre modèle améliore la prédiction du choix de chaque client. De plus, avec une autre expérience utilisant des données d'une salle de concert, nous montrons que celui-ci s'adapte à un problème lié à une application réelle, notamment l'hétérogénéité de la taille d'historique de chaque client et celle du nombre de tickets achetés au sein d'une commande.

5.2 Limitations de la solution proposée

Plusieurs limitations peuvent être relevées pour notre approche.

Premièrement, nous pouvons relever la difficulté d'optimiser notre modèle final : la composante profonde possède l'inconvénient usuel d'un algorithme d'apprentissage qui est un nombre élevé d'hyperparamètres, auquel s'ajoutent ceux de notre modèle individuel qui peuvent également être conséquents pour les méthodes les plus complexes, et enfin ceux liés à la combinaison. Ainsi, l'optimisation de tout les hyperparamètres s'avère particulièrement fastidieuse ici.

Ensuite, pour le traitement des données réelles, on peut observer que le pourcentage de données conservées par rapport aux données initialement recueillies est faible, pour plusieurs raisons. D'abord, cela est dû à une partie non exploitable (abonnements saisonniers), mais également à une simplification d'implémentation (conservation de la scène centrale, des réservations de taille 1 ou 2), ou alors pour garder une comparaison équitable avec les méthodes

individuelles (utilisations des clients à plus de 5 choix seulement).

Enfin, une dernière limitation que l'on peut noter est le temps de calcul pour la recommandation. Bien que le code n'ait pas été parallélisé ni exécuté sur une machine très puissante, il est peut-être possible que notre modèle soit trop complexe pour une recommandation en temps réel. La restriction à des algorithmes de classification efficaces peut permettre de pallier ce problème.

5.3 Améliorations futures

Beaucoup de pistes d'améliorations et d'explorations futures sont possibles :

Composante individualisée - Étude effective de l'interprétabilité

En s'inspirant d'autres recherches effectuées sur les modèles de choix de discrets et l'apprentissage automatique [45, 48] et celui plus général de l'interprétabilité en *machine learning*, il serait intéressant d'étudier plus en détail celle de notre composante individuelle en fonction de l'algorithme de classification utilisé. Cela pourrait par exemple conduire à une analyse comportementale qui permettait de tirer des conclusions sur la façon dont est choisis un siège, car ces facteurs demeurent flous à l'heure actuelle (voir la partie 2.1.4 de la revue sur les biais dans le choix)

Composante profonde - Modèles d'attentions

Beaucoup de progrès ont récemment été effectués en apprentissage profond grâce à des modèles d'attentions, notamment pour des données visuelles comme les nôtres [71, 72]. L'idée générale d'un modèle d'attention consiste en l'utilisation de mécanismes permettant de biaiser un réseau de neurones pour qu'il se concentre sur la partie la plus pertinente de l'entrée. Cette stratégie pourrait s'avérer efficace ici, où notre réseau de neurones pourrait se focaliser sur la zone de la salle la plus attractive.

Considération du prix / Tarification

En utilisant d'autres données qui possèdent une tarification connue, il serait intéressant d'étudier la prise en compte du prix dans la prédiction. Pour aller plus loin, utiliser notre modèle comme base pour faire de l'optimisation de recettes en modifiant la tarification en fonction de l'attractivité des sièges (à la manière de Baldin et al. [42]) pourrait être une application intéressante.

Considérations des métadonnées

Un client qui fait a effectué plusieurs réservations par le passé ne les a pas forcément faite pas pour le même type d'évènement : de ce fait, il pourrait être intéresser d'analyser l'influence que peut avoir des métadonnées comme le type de spectacle (par exemple, le genre d'un film dans un cinéma), ou encore la date (jour de la semaine, mois, etc.).

Point de vue spatio-temporel du problème

Dans ce mémoire, le problème est uniquement abordé d'un point de vue "spatial", c'est-à-dire que chaque réservation de siège est étudiée sous le prisme de sa position dans la salle et son interaction spatiale avec les autres sièges. Pour autant, à une réservation est également associée une date d'achat, et plutôt qu'une modélisation d'une configuration de salle par une matrice M, on pourrait imaginer une modélisation d'une salle par une matrice $M(t)$ qui se remplirait au cours du temps t. De façon similaire à notre problème, cette idée a été abordée par Huang et al. [24], mais avec des différences notables par rapport à notre approche (voir 2.2.2).

Essais sur d'autres types de données

Enfin, nos expériences se sont restreintes à des données que l'on pourrait qualifier de culturelles, c'est-à-dire qui ne concerne que des cinémas ou des salles de concert, pour autant il pourrait être intéressant d'appliquer notre problème sur d'autres domaines comme celui du transport aérien ou encore de stade pour des évènements sportifs.

RÉFÉRENCES

[1] Aphex34. (2015) typical cnn architecture. [En ligne]. Disponible : https://commons.wikimedia.org/wiki/File:Typical_cnn.png

[2] H. Noh, S. Hong et B. Han, "Learning deconvolution network for semantic segmentation," dans *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[3] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," dans *ACM International Conference Proceeding Series*, 2016.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, n°. 1, p. 5–32, Oct 2001. [En ligne]. Disponible : https://doi.org/10.1023/A:1010933404324

[5] D. P. Kingma et J. Ba, "Adam : A method for stochastic optimization," 2014.

[6] A. Krizhevsky, I. Sutskever et G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 2017.

[7] J. Long, E. Shelhamer et T. Darrell, "Fully convolutional networks for semantic segmentation," dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[8] S. Ren *et al.*, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[9] V. Dumoulin et F. Visin, "A guide to convolution arithmetic for deep learning," mar 2016. [En ligne]. Disponible : http://arxiv.org/abs/1603.07285

[10] C. C. Aggarwal, *Recommender Systems The Textbook*. Springer, 2016.

[11] X. Su et T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, 2009.

[12] Y. Koren, R. Bell et C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, 2009.

[13] S. Sedhain *et al.*, "AutoRec : Autoencoders meet collaborative filtering," dans *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, 2015.

[14] B. Balu, Oct 2019. [En ligne]. Disponible : https://medium.com/towards-artificial-intelligence/content-based-recommender-system-4db1b3de03e7

[15] M. J. Pazzani et D. Billsus, “Content-based recommendation systems,” dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007.

[16] R. Gil et W. R. Hartmann, “The role and determinants of concession sales in movie theaters : Evidence from the spanish exhibition industry,” *Review of Industrial Organization*, vol. 30, n°. 4, p. 325–347, Jun 2007. [En ligne]. Disponible : https://doi.org/10.1007/s11151-007-9139-7

[17] S. J. Blanchard, T. Dyachenko et K. L. Kettle, “Locational Choices : Studying Consumer Preference for Proximity to Others in Reserved-Seating Venues,” *Journal of Marketing Research*, 2020. [En ligne]. Disponible : http://www.perceptionstudies.com/papers/Blanchard_PS_2020.pdf

[18] E. T. Hall, *The Hidden Dimension*. Doubleday, 1966.

[19] G. D. Harrell, M. D. Hutt et J. C. Anderson, “Path Analysis of Buyer Behavior under Conditions of Crowding,” *Journal of Marketing Research*, 1980.

[20] M. Argyle, *Bodily Communication*. Routledge, 2013.

[21] G. W. Evans et R. E. Wener, “Crowding and personal space invasion on the train : Please don’t make me sit in the middle,” *Journal of Environmental Psychology*, vol. 27, n°. 1, p. 90–94, mar 2007.

[22] J. Schöttl *et al.*, “Investigating the Randomness of Passengers’ Seating Behavior in Suburban Trains,” *Entropy*, vol. 21, n°. 6, p. 600, jun 2019. [En ligne]. Disponible : https://www.mdpi.com/1099-4300/21/6/600

[23] P. Weyers *et al.*, “How to choose a seat in theatres : Always sit on the right side ?” *Laterality*, vol. 11, n°. 2, p. 181–193, mar 2006.

[24] F. Huang et H. Huang, “Event Ticket Price Prediction with Deep Neural Network on Spatial-Temporal Sparse Data,” dec 2019. [En ligne]. Disponible : https://arxiv.org/abs/1912.01139

[25] G. B. Karev, “Cinema Seating in Right, Mixed and Left Handers,” *Cortex*, vol. 36, n°. 5, p. 747–752, jan 2000. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0010945208705501

[26] M. Okubo, “Right movies on the right seat : Laterality and seat choice,” *Applied Cognitive Psychology*, vol. 24, n°. 1, p. 90–99, jan 2010. [En ligne]. Disponible : https://doi.org/10.1002/acp.1556

[27] M. E. R. Nicholls, N. A. Thomas et T. Loetscher, “An Online Means of Testing Asymmetries in Seating Preference Reveals a Bias for Airplanes and Theaters,”

Human Factors, vol. 55, nº. 4, p. 725–731, jan 2013. [En ligne]. Disponible : https://doi.org/10.1177/0018720812471680

[28] V. Harms, M. Reese et L. J. Elias, "Lateral bias in theatre-seat choice," *Laterality*, vol. 19, nº. 1, p. 1–11, jan 2014. [En ligne]. Disponible : https://doi.org/10.1080/1357650X.2012.746349

[29] W. Wittling et R. Roschmann, "Emotion-Related Hemisphere Asymmetry : Subjective Emotional Responses to Laterally Presented Films," *Cortex*, vol. 29, nº. 3, p. 431–448, sep 1993. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0010945213802523

[30] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 2015.

[31] K. Simonyan et A. Zisserman, "Very deep convolutional networks for large-scale image recognition," dans *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[32] K. He *et al.*, "Deep residual learning for image recognition," dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[33] G. Huang *et al.*, "Densely connected convolutional networks," dans *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[34] S. Ioffe et C. Szegedy, "Batch normalization : Accelerating deep network training by reducing internal covariate shift," dans *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[35] A. Odena, V. Dumoulin et C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [En ligne]. Disponible : http://distill.pub/2016/deconv-checkerboard

[36] D. L. McFadden, "Conditional Logit Analysis of Qualitative Choice Behavior," dans *Frontiers in Econometrics*, 1974.

[37] W. H. Greene et D. A. Hensher, "A latent class model for discrete choice analysis : Contrasts with mixed logit," *Transportation Research Part B : Methodological*, 2003.

[38] D. McFadden et K. Train, "Mixed MNL models for discrete response," *Journal of Applied Econometrics*, 2000.

[39] K. E. Train, *Discrete choice methods with simulation*. Cambridge University Press, 2003.

[40] P. Hetrakul et C. Cirillo, "A latent class choice based model system for railway optimal pricing and seat allocation," *Transportation Research Part E : Logistics*

and Transportation Review, vol. 61, p. 68–83, jan 2014. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1366554513001725

[41] X. Wang, H. Wang et X. Zhang, "Stochastic seat allocation models for passenger rail transportation under customer choice," *Transportation Research Part E : Logistics and Transportation Review*, vol. 96, p. 95–112, dec 2016. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1366554516302502

[42] A. Baldin *et al.*, "Revenue and attendance simultaneous optimization in performing arts organizations," *Journal of Cultural Economics*, vol. 42, n°. 4, p. 677–700, nov 2018.

[43] K. G. Willis et J. D. Snowball, "Investigating how the attributes of live theatre productions influence consumption choices using conjoint analysis : The example of the National Arts Festival, South Africa," *Journal of Cultural Economics*, 2009.

[44] J. M. Grisolía et K. G. Willis, "A latent class model of theatre demand," 2012.

[45] X. Zhao *et al.*, "Prediction and behavioral analysis of travel mode choice : A comparison of machine learning and logit models," *Travel Behaviour and Society*, 2020.

[46] Y. Zhang et Y. Xie, "Travel mode choice modeling with support vector machines," *Transportation Research Record*, 2008.

[47] M. Pirra et M. Diana, "A study of tour-based mode choice based on a support vector machine classifier," *Transportation Planning and Technology*, vol. 42, n°. 1, p. 23–36, 2019. [En ligne]. Disponible : https://doi.org/10.1080/03081060.2018.1541280

[48] L. Cheng *et al.*, "Applying a random forest method approach to model travel mode choice behavior," *Travel Behaviour and Society*, 2019.

[49] A. Lhéritier *et al.*, "Airline itinerary choice modeling using machine learning," *Journal of Choice Modelling*, 2019.

[50] Y. Bentz et D. Merunka, "Neural networks and the multinomial logit for brand choice modelling : A hybrid approach," *Journal of Forecasting*, 2000.

[51] H. Hruschka, W. Fettes et M. Probst, "Analyzing purchase data by a neural net extension of the multinomial logit model," dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001.

[52] S. Wang, Q. Wang et J. Zhao, "Multitask learning deep neural networks to combine revealed and stated preference data," 2019.

[53] B. Sifringer, V. Lurkin et A. Alahi, "Let Me Not Lie : Learning MultiNomial Logit," dec 2018. [En ligne]. Disponible : http://arxiv.org/abs/1812.09747

[54] Y. Han *et al.*, “A Neural-embedded Choice Model : TasteNet-MNL Modeling Taste Heterogeneity with Flexibility and Interpretability,” feb 2020. [En ligne]. Disponible : https://arxiv.org/abs/2002.00922

[55] S. Wang *et al.*, “Deep neural networks for choice analysis : A statistical learning theory perspective,” 2018.

[56] A. Mottini et R. Acuna-Agost, “Deep choice model using pointer networks for airline itinerary prediction,” dans *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[57] O. Vinyals, M. Fortunato et N. Jaitly, “Pointer networks,” 2015.

[58] M. Otsuka et T. Osogami, “A deep choice model,” dans *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.

[59] Y. LeCun et C. Cortes, “MNIST handwritten digit database,” 2010. [En ligne]. Disponible : http://yann.lecun.com/exdb/mnist/

[60] S. Zhang *et al.*, “Deep learning based recommender system : A survey and new perspectives,” dans *ACM Computing Surveys*, 2019.

[61] Y. S. Rawat et M. S. Kankanhalli, “ConTagNet : Exploiting user context for image tag recommendation,” dans *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*, 2016.

[62] H. T. Nguyen *et al.*, “Personalized deep learning for tag recommendation,” dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017.

[63] C. Lei *et al.*, “Comparative Deep Learning of Hybrid Representations for Image Recommendations,” dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[64] H. Guo *et al.*, “DeepFM : A factorization-machine based neural network for CTR prediction,” dans *IJCAI International Joint Conference on Artificial Intelligence*, 2017.

[65] J. Lian *et al.*, “xDeepFM : Combining explicit and implicit feature interactions for recommender systems,” dans *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018.

[66] M. F. Dacrema, P. Cremonesi et D. Jannach, “Are we really making much progress ? A worrying analysis of recent neural recommendation approaches,” dans *RecSys 2019 - 13th ACM Conference on Recommender Systems*, 2019.

[67] L. Chen *et al.*, “Human decision making and recommender systems,” *ACM Transactions on Interactive Intelligent Systems*, 2013.

[68] A. Mottini *et al.*, “Understanding customer choices to improve recommendations in the air travel industry,” dans *CEUR Workshop Proceedings*, 2018.

[69] P. Saavedra *et al.*, “Choice-based recommender systems,” dans *CEUR Workshop Proceedings*, vol. 1685. CEUR-WS, 2016, p. 38–46.

[70] B. H. Chaptini, “Use of discrete choice models with recommender systems,” Thèse de doctorat, Massachusetts Institute of Technology, 2005.

[71] V. Mnih *et al.*, “Recurrent Models of Visual Attention,” jun 2014. [En ligne]. Disponible : http://arxiv.org/abs/1406.6247

[72] K. Xu *et al.*, “Show, attend and tell : Neural image caption generation with visual attention,” dans *32nd International Conference on Machine Learning, ICML 2015*, 2015.

ANNEXE : HYPERPARAMÈTRES DES MODÈLES

Cette annexe répertorie l'ensemble des hyperparamètres par expérience pour chacun des modèles testés.

Ces derniers se divisent en deux : ceux du modèle de classification (pour les modèles individuelles, le nom du paramètre de la méthode sur Scikit-learn est repris), et ceux de la forme "X_features", qui indique l'utilisation ou non de la catégorie d'attribut X.

Modèles individuelles

Régression logistique :

Dataset	penalty	solver	max_iter	POS features	PS features	R2 feature	R3 feature
E4-Concert-Singles.FC	None	newton-cg	300	True	True	True	True
E2-Movie-Singles.FC	None	newton-cg	300	True	True	False	False
Concert Hall data	None	newton-cg	300	True	True	True	True

Machines à vecteur de support :

Dataset	C	kernel	max_iter	POS features	PS features	R2 feature	R3 feature
E4-Concert-Singles.FC	100	linear	2000	True	True	True	True
E2-Movie-Singles.FC	50	rbf	2000	True	True	True	True
Concert Hall data	100	linear	2000	True	True	True	True

Gradient Boosted Trees :

Dataset	lr	n_estimator	maxdepth	min_samples_split	min_samples_leaf	max_features
E4-Concert-Singles.FC	0.1	200	2	2	1	2
E2-Movie-Singles.FC	0.1	200	2	2	1	2
Concert Hall data	0.1	200	3	2	1	2

Dataset	POS features	PS features	R2 feature	R3 feature
E4-Concert-Singles.FC	True	True	True	True
E2-Movie-Singles.FC	True	True	False	True
Concert Hall data	True	True	True	True

Forêts aléatoires :

Dataset	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features
E4-Concert-Singles.FC	200	2	2	1	2
E2-Movie-Singles.FC	200	5	3	2	2
Concert Hall data	200	5	2	1	2

Dataset	POS features	PS features	R2 feature	R3 feature
E4-Concert-Singles.FC	True	True	True	True
E2-Movie-Singles.FC	True	True	False	False
Concert Hall data	True	True	True	True

Modèles profonds

CNN :

Dataset	nb_conv_layers	batch_size	lr
E4-Concert-Singles.FC	3	32	1e-4
E2-Movie-Singles.FC	3	32	1e-4
Concert Hall data	4	32	1e-4

CDNN :

Dataset	nb_channels	batch_size	lr
E4-Concert-Singles.FC	3	32	5e-3
E2-Movie-Singles.FC	3	32	5e-3
Concert Hall data	4	32	5e-3

Modèle hybride

Les mêmes hyperparamètres sont gardés présentés dans les tableaux précédents sont conservés pour la combinaison.

Dataset	combination	lr	momentum	weight decay	alpha_init
E4-Concert-Singles.FC	GBT+CNN	1e-3	0.9	0.8	0.5
E2-Movie-Singles.FC	RF+CNN	1e-2	0.9	0.5	0.5
Concert Hall data	GBT+CDNN	3e-2	0.99	0.7	0.5