université PARIS-SACLAY

CentraleSupélec

NNT : 2016SACLC090

Thèse de doctorat de l'Université Paris-Saclay préparée à CentraleSupélec

École doctorale n° 580
ED Sciences et technologies de l'information et de la communication
Spécialité de doctorat : Informatique

par

M. Jill-Jênn Vie

Modèles de tests adaptatifs pour le diagnostic de connaissances dans un cadre d'apprentissage à grande échelle

Thèse présentée et soutenue à Cachan, le 5 décembre 2016.

Composition du Jury :

Mme	Amel Bouzeghoub	Professeur Télécom SudParis	(Présidente du jury)
Mme	Nathalie Guin	Maître de conférences HDR LIRIS	(Rapporteur)
M.	Sébastien George	Professeur des universités Université du Maine	(Rapporteur)
Mme	Vanda Luengo	Professeur des universités UPMC	(Examinatrice)
Mme	Monique Grandbastien	Professeur émérite LORIA	(Examinatrice)
Mme	Yolaine Bourda	Professeur CentraleSupélec	(Directrice de thèse)
M.	Éric Bruillard	Professeur des universités ENS Paris-Saclay	(Codirecteur de thèse)
M.	Fabrice Popineau	Professeur CentraleSupélec	(Coencadrant de thèse)

Remerciements

Mes remerciements vont avant tout à mes parents Lucien Vie et Catherine Hui Bon Hoa, qui m'ont supporté depuis plus de vingt-cinq ans, soit à peu près huit fois plus que mes encadrants.

Merci à Fabrice Popineau [1] d'avoir encadré cette thèse, Yolaine Bourda [2] de l'avoir dirigée, Éric Bruillard [3] de l'avoir codirigée, Sébastien George et Nathalie Guin d'avoir accepté d'être rapporteurs de cette thèse, Amel Bouzeghoub, Monique Grandbastien et Vanda Luengo d'avoir accepté de faire partie du jury, Hiba Hajri d'avoir partagé mon bureau avec un sourire permanent, et enfin Tomoko Kōzu (神津智子) de m'avoir soutenu moi-même pendant ma thèse, tout en ayant soutenu le contraire. Merci aux rock stars du labo : Jean Condé, Matthieu Cisel, Mehdi Khaneboubi, Tiphaine Liu et hors du labo : Bruno Martin, Diego Riofrío, Franck Silvestre, Maria Denami, Maÿlis Limouzineau. Je dois également remercier Gilles Dowek pour m'avoir aidé à trouver cet encadrement de thèse, Serge Abiteboul pour m'avoir encouragé à de nombreuses reprises malgré son emploi du temps de ministre et surtout Françoise Tort, qui en plus d'être une personne d'une extrême gentillesse, est tout de même l'une des instigatrices du concours Castor [4], a travaillé sans relâche sur l'élaboration des sujets, et a même rédigé les nouveaux programmes d'informatique des classes de seconde, première et terminale ; et tout ça, personne ne le sait.

Je remercie également tous ceux qui m'ont fait penser à autre chose que mon sujet de thèse, mes lointains [5] amis : Amelle Vandevelde, Clélia de Mulatier, Coline

1. Superviseur de la traduction de la bible de 1 200 pages de Russell & Norvig : *Artificial Intelligence: A Modern Approach*, aficionado de TeX et MetaPost à l'association GUTenberg au point d'y inviter Knuth et Zapf ; et tout ça, personne ne le sait.

2. Dont j'ai pu confirmer la réputation de *maman* à CentraleSupélec, qui a toujours fait preuve d'une étonnante efficacité de relecture de mes épreuves et d'une rigoureuse organisation dont j'ai — semble-t-il — beaucoup à apprendre…

3. Qui a pu éclairer mon travail à l'aide de recherches issues des sciences de l'éducation.

4. Il s'agit d'un concours d'initiation à l'algorithmique pour les enfants de la 6e à la terminale, composé d'activités ludiques. http://castor-informatique.fr

5. Géographiquement seulement.

Wiatrowski, Jean-Bastien Grill, Jean-Pierre Boudine, Larissa & Tobit Caudwell, Laurent Bonnans, Michel Blockelet, Rose Glaeser, Svetlana Pinet [6], Thomas Refis, Vincent Maioli, mes très lointains[5] amis Bruno & Laure Le Floch, Fayadhoi Ibrahima, Florence Brücken, Juanjuan Liu, Juliana Gutierrez-Mazariegos, Julien Guertault, Sarah & Mat Danskin, Tancrède Lepoint, Xian Zhang, les membres des associations Prologin, Mangaki et Paris ACM SIGGRAPH ainsi que le club algo de l'ENS Paris-Saclay : Christoph Dürr, Clémence Réda, Clément Beauseigneur, Garance Gourdel, Guillaume Aubian, Jérémie Perrin, Lê Thành Dũng Nguyễn [7], Lucas Gréaux, Noël Nadal, Olivier Marty, Rémi Dupré, Shendan Jin, Thomas Espitau et Victor Lanvin. Profitons-en pour remercier au passage le département informatique dont j'ai détourné les fonds autant que se peut pour envoyer les candidats au concours SWERC à Porto pendant 3 ans : merci Catherine Forestier, Imane Mimouni, Paul Gastin et Serge Haddad.

Que serais-je sans les passions que je partage avec Antoine Pietri, Basile Clement, Clément Dervieux, Cyril Amar, David Lambert, Julien Reichert, La Jeune Fille au chat, Pierre Hénon et Sam Vie, qu'il s'agisse de cinéma, de mangas, d'opéra ou de chats ? Faire de la musique permet également de quitter l'écran à défaut du clavier, merci aux talentueuses Camille Laïly et Solène Pichereau pour ces nombreux concerts que j'ai pris tant de plaisir à accompagner. Il me faut mentionner également les multiples discussions fructueuses depuis le début de la thèse avec Antoine Amarilli [8], Elsa Caboche, Étienne Simon, Julien Simoni, Morpheen, Roger Mansuy et Ryan Lahfa. Vous n'imaginez pas à quel point vous m'avez influencé, et motivé à persévérer.

Concevoir un système de recommandation de mangas était au départ un simple projet annexe pour tester mes algorithmes sur de vraies données, mais cela a indirectement conduit à l'émission *La Faute à l'algo* que Michel Blockelet et moi avons réalisée avec l'équipe de Nolife ; à encadrer Alexis Rivière en stage à Mangaki, ce qui m'a renforcé dans ma volonté de former des jeunes ; à encadrer François Voisin avec Basile Clement, ce qui nous a motivés à étudier les processus à point déterminantal et a donné lieu au chapitre 5 de cette thèse ainsi qu'à une présentation de Mangaki à Deezer France et MFG Labs, puis devant des investisseurs japonais à Tokyo.

Il me faut remercier encore Fabrice Popineau, avec qui nous avons passé de longues heures à rendre cette thèse archivable au format PDF/A-1b. J'espère que les sources sur GitHub inspireront d'autres personnes. Merci à la communauté TeX.SE [9] d'avoir réponse à tout, même aux bugs les plus invraisemblables.

6. On fait la course depuis le primaire ; elle a soutenu sa thèse la première, mais j'imagine que ça se jouera au premier qui aura son article dans *Nature*.
7. Dont les diacritiques ont fait planter plus d'un système.
8. Ainsi que ses relectures inestimables d'une précision étonnante.
9. C'est grâce à ces plateformes innovantes que des communautés peuvent s'entraider.

Enfin, je remercie ces artistes, professeurs ou entrepreneurs, dont les échanges parfois rares mais toujours précieux m'ont poussé à toujours faire plus [10] : Emmanuel Lévy, Fabrice Popineau, Frederic Vander Elst, Greg Wilson, Jean-Christophe Poulain, Jonathan Dhiver, Lewis Trondheim, Marc Alperovitch, Marie-Amélie Jéhanno, Matti Schneider, Pierre Berger ; ainsi que tous les agrégatifs de la préparation 2014 à l'ENS Paris-Saclay [11], pour cette ambiance de travail positive où l'on est à tour de rôle enseignant et jury, l'année préférée de ma scolarité. Merci à Benjamin Dadoun, Claire Brécheteau, Cristina Sirangelo, Didier Lesesvre, Lilian Besson, Loïc Devilliers, Ludovic Sacchelli, Rémi Cheval, Sylvain Schmitz, et bien sûr la prof de tous les profs, Claudine « Pikachu » Picaronny.

10. Il existe un terme japonais pour ça, d'ailleurs : 頑張る (*gambaru*), qui signifie « Je vais faire de mon mieux. » Il est d'ailleurs difficile de passer de l'agrégation de mathématiques, où la maîtrise et l'exactitude de ce que l'on raconte est facilement évaluable car l'environnement est clairement défini, à la thèse, où l'on doit défendre ses arguments dans une jungle de modèles et constructions intellectuelles et où les articles sont éternellement perfectibles.

11. École qui, rappelons-le, n'existait même pas à l'époque.

Acronymes

CC composantes de connaissances. 33

CD-CAT *Cognitive Diagnostic Computerized Adaptive Tests*. 35

DINA *Deterministic Input, Noisy And*. 35

ECPE *Examination for the Certificate of Proficiency in English*. 56

GenMA *General Multidimensional Adaptive*. 20

GMAT *Graduate Management Admission Test*. 27

InitialD *Initial Determinant*. 20

MIRT *Multidimensional Item Response Theory*. 31

MOOC *Massive Open Online Courses*. 15

PPD processus à point déterminantal. 97

SPARFA *Sparse Factor Analysis*. 31

TIMSS *Trends in International Mathematics and Science Study*. 57

Nomenclature

$\boldsymbol{\theta}_\mathbf{i}$	caractéristiques de l'apprenant i dans MIRT, GenMA, page 31
$\mathbf{d}_\mathbf{j}$	paramètres de discrimination de la question j dans MIRT, GenMA, page 31
δ_j	paramètre de facilité de la question j dans MIRT, GenMA, page 31
Φ	fonction logistique, page 28
$\propto$	proportionnel à, page 36
θ_i	valeur de niveau de l'apprenant i pour le modèle de Rasch, page 28
D	données des apprenants, page 27
d_j	valeur de difficulté de la question j pour le modèle de Rasch, page 28
g_i	paramètre de chance pour le modèle DINA, page 35
K	nombre de composantes de connaissances, page 35
k	nombre moyen de composantes de connaissances par question pour SPARFA, GenMA, page 48
q_{jk}	entrée (j,k) de la q-matrice, page 82
s_i	paramètre d'inattention pour le modèle DINA, page 35
V	caractéristiques des questions dans MIRT, GenMA, page 85

Table des matières

Chapitre 1

Introduction

1.1 Évaluation adaptative à grande échelle

L'individualisation de l'enseignement et de l'évaluation est un enjeu fort de notre siècle. Donner la même feuille d'exercices à tous les élèves d'une classe conduit à des situations où certains élèves finissent très vite et demandent d'autres exercices, tandis que d'autres peinent à résoudre un exercice. Il serait plus profitable pour les élèves d'avoir des feuilles d'exercices personnalisées. Mais le travail consistant à piocher des exercices dans une banque de façon à maintenir un certain équilibre pour une unique feuille universelle est déjà difficile pour un professeur, alors concevoir des feuilles d'exercices différentes pour chaque étudiant peut sembler complexe à gérer. Heureusement, les progrès en traitement de l'information rendent cela possible. On voit ainsi aujourd'hui des professeurs d'université distribuer des énoncés différents à tous leurs étudiants lors d'un examen (Zeileis, Umlauf, Leisch, et al., 2012), ce qui pose toutefois des questions d'impartialité de l'évaluation. Personnaliser l'évaluation permet également de poser moins de questions à chaque apprenant (H.-H. Chang, 2014), ce qui est d'autant plus utile que les apprenants passent aujourd'hui trop de temps à être testés (Zernike, 2015).

Avec l'arrivée des *Massive Open Online Courses* (MOOC), ce besoin en évaluation adaptative s'est accentué. Des cours de nombreuses universités à travers le monde peuvent être suivis par des centaines de milliers d'étudiants. Mais la pluralité des profils de ces apprenants, notamment leurs âges et leurs parcours, fait qu'il devient crucial d'identifier les connaissances que les apprenants ont accumulées dans le passé, afin de personnaliser leurs expériences d'apprentissage et d'aider le professeur à mieux connaître sa classe pour améliorer son cours. Or, répondre à de nombreuses questions d'un test de positionnement au début d'un cours risque de paraître fastidieux pour les apprenants (Desmarais et R. S. J. D.

Baker, 2012). Il est alors encouragé de ne poser des questions que lorsque c'est nécessaire, par exemple ne pas poser des questions trop difficiles tant que l'apprenant n'a pas répondu à des questions faciles, et ne pas poser des questions requérant des compétences que l'apprenant semble déjà maîtriser, au vu de ses réponses précédentes (H.-H. Chang, 2014).

Cette réduction du nombre de questions d'un test a été étudiée depuis longtemps en psychométrie. La théorie de la réponse à l'item suppose qu'un faible nombre de variables peut expliquer les réponses d'un étudiant à plusieurs questions, et cherche à déterminer les questions les plus informatives pour dévoiler les facteurs latents de l'étudiant (Hambleton et Swaminathan, 1985). Cette théorie a ainsi permis de développer des modèles de tests *adaptatifs*, qui posent une question à un apprenant, évaluent sa réponse et choisissent en fonction de celle-ci la question suivante à lui poser. Alors que la théorie de la réponse à l'item remonte aux années 50, ces tests sont aujourd'hui utilisés en pratique par le GMAT, une certification administrée à des centaines de milliers d'étudiants chaque année. Toutefois, les enjeux de ces tests sont davantage liés à l'évaluation qu'à la formation : leur objectif est de mesurer les apprenants afin de leur remettre ou non un certificat, plutôt que de leur faire un retour sur leurs lacunes. Un tel retour leur serait plus utile pour s'améliorer, et également renforcerait leur engagement. Ainsi, les organismes de certification se placent davantage du côté des institutions, qui décident d'une barre d'admissibilité et d'un quota d'entrants, tandis qu'une plateforme de MOOC se place davantage du côté de ses utilisateurs, les apprenants.

1.2 Diagnostic de connaissances

On distingue deux types de tests adaptatifs. Des tests adaptatifs *sommatifs* mesurent l'apprenant et lui renvoient un simple score, tandis que des tests adaptatifs *formatifs* font un diagnostic des connaissances de l'apprenant afin qu'il puisse s'améliorer, par exemple sous la forme de points à retravailler.

Les tests adaptatifs qui se contentent de mesurer l'apprenant opèrent de façon purement statistique, agnostique du domaine. En revanche, les tests formatifs ont besoin d'une représentation du domaine : en effet, pour expliquer à l'apprenant ce qu'il semble ne pas avoir compris, il faut avoir fait le lien entre les questions et les composantes de connaissances impliquées dans leur résolution.

Dans cette thèse, nous avons répertorié des modèles de tests adaptatifs issus de différents pans de la littérature. Nous les avons comparés de façon qualitative et quantitative. Nous avons ainsi proposé et implémenté un protocole expérimental pour comparer les principaux modèles de tests adaptatifs sur plusieurs jeux de données réelles. Cela nous a amenés à proposer un modèle hybride de diagnostic

de connaissances adaptatif. Enfin, nous avons élaboré une stratégie pour poser plusieurs questions au tout début du test afin de réaliser une meilleure estimation initiale des connaissances de l'apprenant.

Nous avons souhaité adopter un point de vue venant de l'apprentissage automatique, plus précisément du filtrage collaboratif, pour attaquer le problème du choix des questions à poser pour réaliser un diagnostic. En filtrage collaboratif, on se demande comment s'aider d'une communauté active pour avoir une idée des préférences d'un utilisateur en fonction des préférences des autres utilisateurs. En évaluation adaptative, on se demande comment s'aider d'un historique de passage d'un test pour avoir une idée de la performance d'un apprenant en fonction de la performance des autres apprenants. Il n'est pas question ici de faire une analogie directe entre l'apprentissage et la consommation de culture, mais plutôt de s'inspirer des techniques étudiées dans cet autre domaine : il est indéniable que les plateformes de consommation de biens culturels sont davantage préparées que les MOOC à recevoir des milliers d'utilisateurs, traiter les grandes quantités de données qu'ils récoltent et adapter leur contenu en conséquence. Ainsi, les algorithmes qu'on y retrouve ne reposent pas seulement sur une solide théorie statistique mais également sur un souci de mise en pratique efficace en grande dimension.

1.3 Problèmes

Dans cette thèse, nous nous sommes intéressés aux problèmes suivants liés aux modèles de tests adaptatifs.

Réduction du nombre de questions d'un test

Si un intervenant ne peut poser que k questions d'une banque de n questions (où $k < n$) à un apprenant, lesquelles choisir ? Pour ce problème, nous avons considéré différents modèles de tests adaptatifs ainsi que différentes stratégies du choix des premières questions. Laquelle de ces approches fournit les meilleurs résultats en fonction de k ?

Méthodologie de comparaison de modèles

Comment comparer des modèles de tests adaptatifs différents sur un même jeu de données ? Quels critères considérer pour la comparaison ? Quels sont les avantages et limitations des modèles de tests adaptatifs ?

Également, comment comparer différentes stratégies de choix d'un ensemble de questions ?

Méthodologie de choix de modèles

Dans une situation donnée, en fonction des données dont un enseignant dispose et de ses objectifs, quel modèle de test adaptatif a-t-il intérêt à choisir ?

Élaboration d'un test adaptatif dans un MOOC

Dans le cas pratique d'une utilisation d'un test adaptatif dans un MOOC, comment pourrait-on procéder ? (Vie et al., 2015b)

1.4 Contributions

1.4.1 Hypothèses

Dans le cadre de cette thèse, nous cherchons à évaluer les connaissances des apprenants, et non d'autres dimensions telles que leur persévérance, leur organisation ou leur capacité à faire preuve de précaution lorsqu'ils répondent à des questions. En réduisant le nombre de questions, nous réduisons le temps que les apprenants passent à être évalués, ce qui les empêche de se lasser de répondre à trop de questions et laisse plus de temps pour d'autres activités d'apprentissage.

Nous avons considéré que les réponses de l'apprenant pouvaient être correctes ou incorrectes, c'est-à-dire qu'ils produisent des *motifs de réponse dichotomiques*. Cela comprend les questions à choix multiples, les questions à réponse ouverte courte à condition d'avoir une fonction capable de traiter la réponse de l'apprenant afin de savoir si elle est correcte ou non, ou même des tâches plus complexes que l'apprenant peut résoudre ou ne pas résoudre. Ce cadre nous a permis d'analyser des données issues de différents environnements éducatifs : des tests standardisés, des plateformes de jeux sérieux ou des MOOC.

Afin de pouvoir mener nos expériences sur des données de test existantes, nous avons fait la supposition que le niveau de l'apprenant n'évolue pas pendant qu'il passe le test. Les modèles considérés fournissent donc une « photographie » de la connaissance d'un apprenant à un instant donné. Aussi nous supposons que l'apprenant répondra de la même façon indépendamment de l'ordre dans lequel nous posons les questions. Celles-ci doivent donc être localement indépendantes. Nous ne supposons aucun profil de l'apprenant autre que ses réponses aux questions posées, ce qui nous permet de proposer des tests anonymes. L'apprenant n'a donc pas à craindre que ses erreurs puissent être enregistrées et associées à son identité indéfiniment (Executive Office of the President et Podesta, 2014).

Enfin, les modèles que nous considérons ne posent jamais deux fois la même question au sein d'un test. Même si dans certains scénarios, présenter le même item plusieurs fois est utile, par exemple lors de l'apprentissage du vocabulaire

d'une langue (Altiner, 2011), nous préférons poser des questions différentes pour réduire les risques que l'apprenant devine la bonne réponse.

1.4.2 Système de comparaison de tests adaptatifs

Nous avons identifié plusieurs modèles de tests adaptatifs et les avons comparés selon plusieurs angles qualitatifs :

- capacité à modéliser plusieurs dimensions de l'apprenant, c'est-à-dire à mesurer l'apprenant selon plusieurs composantes de connaissances ;
- capacité à faire un diagnostic de l'apprenant utile pour qu'il puisse s'améliorer ;
- nécessité de disposer de données d'entraînement ou non pour faire fonctionner le test ;
- complexité en temps pour l'entraînement des modèles.

Nous les avons également comparés selon plusieurs angles quantitatifs :

- capacité à requérir peu de questions de l'utilisateur pour converger vers un diagnostic ;
- capacité à aboutir à un diagnostic vraisemblable.

Ce cadre nous a permis d'évaluer des modèles de tests adaptatifs sur plusieurs jeux de données réelles :

- un test multidisciplinaire SAT ;
- un examen d'anglais ECPE ;
- un test de soustraction de fractions ;
- un test standardisé de mathématiques ;
- une édition du concours d'initiation à l'informatique Castor ;
- les données d'un MOOC de Coursera.

Le protocole expérimental que nous avons conçu est générique : il s'appuie sur les composants que l'on retrouve dans tous les modèles de tests adaptatifs, et peut ainsi être réutilisé pour de nouveaux modèles, et de nouveaux jeux de données.

Cette première analyse (Vie et al., 2015a) nous a permis de faire un état de l'art intercommunautaire des modèles de tests adaptatifs récents que nous avons publié dans un livre sur l'analytique de l'apprentissage (Vie et al., 2016a), et de concevoir une méthodologie pour étudier les modèles de tests adaptatifs. Nous avons ainsi pu mettre en évidence que selon le type de test, le meilleur modèle n'est pas le même, et proposer les deux autres contributions suivantes.

1.4.3 GenMA, un modèle hybride adaptatif de diagnostic de connaissances

La comparaison que nous avons effectuée a permis de mettre en exergue les limitations des différents modèles : par exemple, un modèle sommatif basé sur la théorie de la réponse à l'item est généralement plus prédictif qu'un modèle formatif basé sur un modèle de diagnostic cognitif.

Afin de pallier les limitations des deux types de modèles, nous avons proposé un nouveau modèle adaptatif de diagnostic de connaissances appelé *General Multidimensional Adaptive* (GenMA) qui mesure à la fois le niveau de l'apprenant et son degré de maîtrise selon plusieurs composantes de connaissances pour lui faire un diagnostic utile pour s'améliorer. GenMA est donc un modèle hybride car il s'appuie sur une représentation des composantes de connaissances, et sur la théorie de la réponse à l'item.

GenMA est plus rapide à calibrer qu'un modèle de théorie de la réponse à l'item de même dimension. De plus, il est meilleur que le modèle existant de diagnostic cognitif sur tous les jeux de données étudiés. Nous l'avons présenté à la conférence EC-TEL 2016 (Vie et al., 2016b).

1.4.4 InitialD, tirer les k premières questions pour démarrer

Pour l'utilisation du modèle GenMA, nous avons comparé différentes stratégies du choix des k premières questions à poser à un nouvel apprenant. Adapter le processus d'évaluation dès la première question peut conduire à des estimations imprécises du niveau de l'apprenant, car celui-ci peut faire des fautes d'inattention ou deviner la bonne réponse. Une variante des tests adaptatifs nommée tests à étapes multiples consiste à poser plusieurs questions avant d'estimer le niveau de l'apprenant pour minimiser le taux d'erreur du premier diagnostic.

Nous avons proposé une nouvelle stratégie appelée *Initial Determinant* (InitialD) pour choisir les k premières questions, qui repose sur une mesure de diversité inspirée par les systèmes de recommandation. Ainsi, InitialD cherche à sélectionner k questions diversifiées, afin de minimiser la redondance de ce qui est mesuré.

Nous montrons ainsi que l'adaptation a ses limites, puisque parfois poser un petit groupe de k questions est plus informatif pour le modèle de test adaptatif que poser k questions une par une, de façon adaptative. De façon théorique, la meilleure stratégie adaptative réalise un diagnostic plus fin que la meilleure stratégie non adaptative, mais nous avons mis en évidence que certaines stratégies adaptatives habituellement utilisées dans les tests adaptatifs se comportent moins bien que les stratégies non adaptatives que nous avons proposées.

1.5 Publications

Poster à EDM 2015

Jill-Jênn Vie, Fabrice Popineau, Éric Bruillard et Yolaine Bourda (2015a). "Predicting Performance over Dichotomous Questions: Comparing Models for Large-Scale Adaptive Testing". In: *8th International Conference on Educational Data Mining (EDM 2015)*.

Workshop à EIAH 2015

Jill-Jênn Vie, Fabrice Popineau, Jean-Bastien Grill, Éric Bruillard et Yolaine Bourda (2015b). « Prédiction de performance sur des questions dichotomiques : comparaison de modèles pour des tests adaptatifs à grande échelle ». In : *Atelier Évaluation des Apprentissages et Environnements Informatiques, EIAH 2015*.

Conférence à EC-TEL 2016

Jill-Jênn Vie, Fabrice Popineau, Yolaine Bourda et Éric Bruillard (2016b). "Adaptive Testing Using a General Diagnostic Model". In: *European Conference on Technology Enhanced Learning*. Springer, p. 331–339.

Chapitre de journal Springer 2016

Jill-Jênn Vie, Fabrice Popineau, Éric Bruillard et Yolaine Bourda (2016a). "A review of recent advances in adaptive assessment". In: *Learning analytics: Fundaments, applications, and trends: A view of the current state of the art*. Springer, à paraître.

Revue STICEF 2016

Jill-Jênn Vie, Fabrice Popineau, Éric Bruillard et Yolaine Bourda (2016). « Utilisation de tests adaptatifs dans les MOOC dans un cadre de crowdsourcing ». In : STICEF, soumis.

1.6 Plan

Vous arrivez à la fin du chapitre 1, où nous avons introduit le concept de test adaptatif pour le diagnostic de connaissances, les problèmes auxquels nous nous sommes attaqués et les contributions que nous avons proposées, ainsi que le plan de la thèse.

Dans le chapitre 2, nous décrivons les différents modèles de tests adaptatifs que nous avons rencontrés dans des communautés scientifiques différentes (théorie de la réponse à l'item, modèles basés sur des composantes de connaissances, apprentissage automatique), ainsi que leurs limitations.

Dans le chapitre 3, nous proposons une méthode pour comparer de façon qualitative et quantitative des modèles de tests adaptatifs différents sur un même jeu de données. Nous l'appliquons à la comparaison de deux modèles, un issu de la théorie de la réponse à l'item, un autre basé sur des composantes de connaissances, sur cinq jeux de données.

Dans le chapitre 4, nous présentons un nouveau modèle de test adaptatif (GenMA), qui peut être vu à la fois comme issu de la théorie de la réponse à l'item multidimensionnelle et comme un modèle basé sur des composantes de connaissance. En utilisant le système de comparaison décrit au chapitre 3, nous montrons qu'il a une capacité plus prédictive que les autres modèles, sur tous les jeux de données testés.

Dans le chapitre 5, nous présentons une nouvelle stratégie de choix des k premières questions (InitialD) pour le modèle GenMA, basée sur une métrique qui quantifie à quel point un ensemble de questions peut être informatif. Nous montrons qu'elle permet de converger plus vite vers un diagnostic vraisemblable de l'apprenant.

Enfin, dans le chapitre 6, nous décrivons les perspectives de notre travail, et nos futures pistes de recherche.

Chapitre 2

État de l'art

2.1 Introduction

De plus en plus d'évaluations de connaissances sont faites par ordinateur, que ce soient pour les célèbres tests standardisés (SAT, TOEFL, GMAT) ou celles que l'on trouve dans les MOOC. L'automatisation de l'évaluation a simplifié le stockage et l'analyse des données des apprenants, ce qui permet de proposer des évaluations plus précises et plus courtes pour des apprenants futurs. L'analytique de l'apprentissage [1] consiste à collecter des données d'apprenants, déterminer des motifs permettant d'améliorer l'apprentissage au sens large, et de continuellement mettre à jour les modèles en fonction des nouvelles données récoltées (Chatti et al., 2012). D'autres travaux ont porté sur l'adaptation de l'enseignement à plusieurs niveaux : celui de la conception d'un cours en fonction des occurrences précédentes, celui de la tâche présentée à l'apprenant, et celui de l'étape de résolution suggérée à l'apprenant (Aleven et al., 2016). En évaluation, un test est dit *adaptatif*, s'il choisit la question suivante à poser en fonction des réponses que donne l'apprenant au cours du test. Réduire la durée de l'évaluation est d'autant plus utile qu'aujourd'hui les apprenants sont surévalués : par exemple, les écoliers américains entre l'école primaire et le lycée [2] passent 20 à 25 heures par an à subir 8 évaluations obligatoires (Zernike, 2015), ce qui pour un élève de 4e représente 2,34 % du temps d'instruction, est générateur de stress, et laisse moins de temps pour les cours. C'est pourquoi le gouvernement américain a demandé moins de tests ou des tests plus utiles et réfléchis.

Les premiers modèles utilisés pour les tests adaptatifs (Hambleton et Swaminathan, 1985) sont *sommatifs* : ils affectent un score, une valeur de niveau aux

1. En anglais, *learning analytics*.
2. Entre l'équivalent américain du CM1 français et l'équivalent américain de la classe de première française.

examinés, ce qui permet de les classer, et par exemple de déterminer ceux qui obtiendront un certificat. Plus récemment, on s'est demandé comment construire des tests adaptatifs *formatifs*, qui font un retour plus utile à l'apprenant pour qu'il puisse s'améliorer, par exemple sous la forme de points maîtrisés et de points à retravailler. Ainsi, en combinant des modèles de diagnostic cognitif, qui déterminent un profil discret correspondant à ce que maîtrise l'apprenant et ce qu'il ne maîtrise pas, avec des modèles de tests adaptatifs, on a pu proposer des tests adaptatifs de diagnostic cognitif (Ferguson, 2012 ; Huebner, 2010), qui en peu de questions indiquent à l'apprenant à la fin du test les points à retravailler. De tels retours permettent aux professeurs de juger le niveau de leur classe selon plusieurs dimensions, et aux apprenants d'avoir une indication de leurs points faibles et de leurs points forts.

Dans cette thèse, nous nous sommes concentrés sur l'utilisation de données de tests existants d'apprenants ayant répondu de façon correcte ou incorrecte à des questions, pour proposer des versions adaptatives de ces tests comportant moins de questions. De plus, nous souhaitons que ces tests soient formatifs, et qu'ils puissent faire un retour à l'apprenant. Ce retour peut être agrégé à différents niveaux (celui de l'étudiant, d'une classe, d'une école, d'un district, d'un état ou d'un pays), sur des tableaux de bord (*dashboards*), de façon à prendre des décisions informées (Shute, Leighton, et al., 2016).

Dans ce chapitre, nous commençons par décrire plus précisément le domaine de l'analytique de l'apprentissage et ses méthodes, puis nous présentons les différents modèles de tests adaptatifs issus de différentes communautés, ainsi que leurs limitations.

2.2 Analytique de l'apprentissage pour l'évaluation

En technologies de l'éducation, il existe deux domaines très proches qui sont celui de la fouille de données éducatives [3] et l'analytique de l'apprentissage. La première consiste à se demander comment extraire de l'information à partir de données éducatives, en utilisant les modèles mathématiques adéquats. La deuxième se veut plus holistique et s'intéresse aux effets que les systèmes éducatifs ont sur l'apprentissage, et comment représenter les informations récoltées sur les apprenants de façon à ce qu'elles puissent être utilisées par des apprenants, des professeurs ou des administrateurs et législateurs.

Plus généralement, l'analytique de l'apprentissage consiste à se demander comment utiliser les données récoltées sur les apprenants pour améliorer l'ap-

3. En anglais, *educational data mining*.

prentissage, au sens large.

En ce qui concerne l'évaluation, Chatti et al. (2012) décrivent différents objectifs de l'analytique de l'apprentissage : le besoin d'un retour intelligent dans les évaluations et le problème de déterminer l'activité suivante à présenter à l'apprenant. Les méthodes utilisées sont regroupées en plusieurs classes : statistiques, visualisation d'information, fouille de données (dont les méthodes d'apprentissage automatique), et analyse de réseaux sociaux.

Comme le disent Desmarais et R. S. J. D. Baker (2012), « Le ratio entre la quantité de faits observés et la largeur de l'évaluation est particulièrement critique pour des systèmes qui couvrent un large nombre de compétences, dans la mesure où il serait inacceptable de poser des questions pendant plusieurs heures avant de faire une évaluation utilisable. » Ils décrivent donc l'importance de réduire la longueur des tests lorsqu'on cherche à évaluer beaucoup de compétences.

Dans les systèmes éducatifs, il y a une différence entre l'*adaptativité*, la capacité à modifier les contenus des cours en fonction de différents paramètres et d'un ensemble de règles préétablies, et l'*adaptabilité*, qui consiste à permettre aux apprenants de personnaliser les contenus de cours par eux-mêmes. Chatti et al. (2012) précisent que « des travaux récents en apprentissage adaptatif personnalisé ont critiqué le fait que les approches traditionnelles soient dans une hiérarchie descendante et ignorent le rôle crucial des apprenants dans le processus d'apprentissage. » Il devrait y avoir un meilleur équilibre entre donner à l'apprenant ce qu'il a besoin d'apprendre (adaptativité) et lui donner ce qu'il souhaite apprendre (adaptabilité), de la façon qu'il souhaite l'apprendre (s'il préfère plus d'exemples, ou plus d'exercices). Dans tous les cas, construire un profil des connaissances de l'apprenant est une tâche cruciale.

Comme cas d'utilisation, considérons un nouvel arrivant sur un MOOC. Celui-ci ayant acquis des connaissances de différents domaines, certains prérequis du cours peuvent ne pas être maîtrisés tandis que d'autres leçons pourraient être sautées. Ainsi, il serait utile de pouvoir évaluer ses besoins et préférences de façon adaptative, pour filtrer le contenu du cours en conséquence et minimiser la surcharge d'information. Lynch et Howlin (2014) décrivent un tel algorithme qui identifie l'état des connaissances d'un apprenant en posant quelques questions au début d'un cours.

En analytique de l'apprentissage, parmi les méthodes employées pour construire des modèles prédictifs, on trouve l'apprentissage automatique [4]. Une application populaire consiste à prédire si un apprenant sur un MOOC va obtenir son certificat à partir de différentes variables liées aux traces de l'apprenant : le nombre d'heures passées à consulter les cours, à regarder les vidéos, le nombre de messages postés sur le forum, entre autres. Cela permet de détecter les apprenants en difficulté

4. En anglais, *machine learning*.

à un instant donné du cours, pour les inviter à se rendre sur le forum, ou leur indiquer des ressources utiles pour les motiver à continuer. La majorité de ces modèles prédictifs s'attaquent à prédire une certaine variable objectif à partir d'un nombre fixé de variables, mais à notre connaissance, peu de modèles interrogent l'apprenant sur ses besoins et préférences. Nous estimons qu'il reste encore beaucoup de recherche à faire vers des modèles d'analytique de l'apprentissage plus interactifs, et les travaux de cette thèse vont dans ce sens.

Deux éléments issus des systèmes de recommandation peuvent être transposés au cadre éducatif de l'analytique de l'apprentissage. Le premier est la technique du filtrage collaboratif (cf. section 2.3.3 page 39), qui permet de concevoir un système de recommandation de ressources pédagogiques (Chatti et al., 2012 ; Manouselis et al., 2011 ; Verbert et al., 2011). Le second est le problème du démarrage à froid de l'utilisateur, dans la mesure où lorsqu'un nouvel utilisateur utilise un système de recommandation, le système n'a que peu d'information sur lui et doit donc lui poser des questions de façon à éliciter ses préférences.

Le temps de réponse lors d'une évaluation a été étudié en psychologie cognitive, car le temps qu'un apprenant met pour répondre à une question peut indiquer quelques aspects sur le processus cognitif (H.-H. Chang, 2014) et joue un rôle dans la performance (Papamitsiou, Terzis, et Economides, 2014). Cela requiert des modèles statistiques spécifiques que nous ne considérons pas ici.

2.3 Modèles de tests adaptatifs

Dans notre cas, nous cherchons à filtrer et à ordonner les questions à poser à un apprenant. Plutôt que de poser les mêmes questions à tout le monde, les tests adaptatifs (Linden et Glas, 2010) choisissent la question suivante à poser à un certain apprenant en fonction des réponses qu'il a données depuis le début du test. Cela permet une adaptation à chaque étape de la séquence de questions. Leur conception repose sur deux critères : un critère de *terminaison* et un critère de *choix de la question suivante*. Tant que le critère de terminaison n'est pas satisfait (par exemple, poser un nombre de questions fixé à l'avance), les questions sont posées selon le critère de choix de la question suivante (par exemple, poser la question la plus informative pour déterminer les connaissances de l'apprenant). Lan, Waters, et al. (2014) ont prouvé que de tels tests adaptatifs pouvaient permettre, sur certains jeux de données de tests en mathématiques, d'obtenir une mesure aussi précise que des tests non adaptatifs, tout en requérant moins de questions.

Raccourcir la taille des tests est utile à la fois pour le système, qui doit équilibrer la charge du serveur, et pour les apprenants, qui risqueraient de se lasser de devoir fournir trop de réponses (Lynch et Howlin, 2014 ; Chen, Choi, et Darwiche, 2015). Ainsi, les tests adaptatifs deviennent de plus en plus utiles dans l'ère actuelle

des MOOC, où la motivation des apprenants joue un rôle important sur leur apprentissage (Lynch et Howlin, 2014). Lorsqu'on implémente ces tests pour une utilisation réelle, des contraintes supplémentaires s'appliquent : pour qu'un apprenant n'ait pas à patienter longuement entre deux questions du test, le calcul du critère du choix de la question suivante doit se faire dans un temps raisonnable, ainsi la complexité en temps de ce calcul est importante. De même, lorsqu'on évalue des connaissances, un certain degré d'incertitude est à prendre en compte : un apprenant risque de faire des fautes d'inattention ou de deviner une bonne réponse alors qu'il n'a pas compris la question. C'est pourquoi une simple dichotomie sur le niveau de l'apprenant, c'est-à-dire poser des questions plus difficiles lorsqu'un apprenant réussit une question ou poser des questions plus faciles lorsqu'il échoue, n'est pas suffisant. Il faut considérer des méthodes plus robustes, tels que des modèles probabilistes pour l'évaluation des compétences.

Les tests adaptatifs ont été étudiés au cours des dernières années et ont été développés en pratique. Par exemple, 238 536 tests de ce type ont été administrés via le *Graduate Management Admission Test* (GMAT), développé par le Graduate Management Admission Council (GMAC) entre 2012 et 2013. Étant donné un modèle de l'apprenant (Peña-Ayala, 2014), l'objectif est de fournir une mesure précise des caractéristiques d'un nouvel apprenant tout en minimisant le nombre de questions posées. Ce problème s'appelle la *réduction de longueur d'un test* (Lan, Waters, et al., 2014) et est également lié à la prédiction de performance future (Bergner, Droschler, et al., 2012 ; Thai-Nghe et al., 2011). En apprentissage automatique, ce problème est connu sous le nom d'apprentissage actif : choisir les éléments à étiqueter de façon adaptative afin de maximiser l'information récoltée à chaque pas.

Dans ce qui suit, nous ne permettons pas à l'apprenant de revenir en arrière pour corriger ses réponses, mais certaines variantes de modèles de tests adaptatifs le permettent (Han, 2013 ; Wang, Fellouris, et H.-H. Chang, 2015).

En fonction du but de l'évaluation, plusieurs modèles peuvent être utilisés, selon si l'on souhaite estimer un niveau général de connaissances, faire un diagnostic détaillé, ou identifier les connaissances maîtrisées par l'apprenant (Mislevy et al., 2012). Dans ce qui suit, nous proposons une répartition de ces modèles dans les trois catégories suivantes : théorie de la réponse à l'item pour des tests sommatifs, modèles de diagnostic cognitif pour des tests formatifs basés sur des composantes de connaissances, et enfin apprentissage automatique.

Dans ce qui suit, on suppose que D désigne la matrice binaire $m \times n$ des succès (1) ou échecs (0) des m apprenants sur les n questions d'un test. Ainsi « $D_{ij} = 1$ » désigne l'événement « L'apprenant i a répondu correctement à la question j ».

2.3.1 Théorie de la réponse à l'item

La *théorie de la réponse à l'item* consiste à supposer que les réponses d'un apprenant que l'on observe lors d'un test peuvent être expliquées par un certain nombre de valeurs cachées, qu'il convient d'identifier.

Modèle de Rasch

Le modèle le plus simple de tests adaptatifs est le *modèle de Rasch*, aussi connu sous le nom de modèle logistique à un paramètre. Il modélise un apprenant par une valeur unique de niveau, et les questions ou tâches à résoudre par une valeur de difficulté. La propension d'un apprenant à résoudre une tâche ne dépend que de la différence entre la difficulté de la tâche et le niveau de l'apprenant. Ainsi, si un apprenant i a un niveau θ_i et souhaite résoudre une question j de difficulté d_j :

$$Pr(D_{ij} = 1) = \Phi(\theta_i - d_j) \tag{2.1}$$

où $\Phi : x \mapsto 1/(1 + e^{-x})$ est la fonction logistique. Ainsi, plus l'apprenant a un haut niveau, plus grande est sa chance de répondre correctement à chacune des questions et plus une question a une difficulté basse, plus grande est la chance de n'importe quel apprenant d'y répondre correctement.

Spécifier toutes les valeurs de difficulté à la main serait coûteux pour un expert, et fournirait des valeurs subjectives qui risquent de ne pas correspondre aux données observées. Ce modèle est suffisamment simple pour qu'il soit possible de calibrer automatiquement et de façon efficace les paramètres de niveau et difficulté, à partir d'un historique de réponses. En particulier, aucune connaissance du domaine n'est prise en compte.

Ainsi, lorsqu'un apprenant passe un test, les variables observées sont ses résultats (vrai ou faux) sur les questions qui lui sont posées, et la variable que l'on souhaite estimer est son niveau, en fonction des valeurs de difficulté des questions qui lui ont été posées ainsi que de ses résultats. L'estimation est habituellement faite en déterminant le maximum de vraisemblance, facile à calculer en utilisant la méthode de Newton pour trouver les zéros de la dérivée de la fonction de vraisemblance. Ainsi, le processus adaptatif devient : étant donné une estimation du niveau de l'apprenant, quelle question poser afin d'obtenir un résultat informatif pour affiner cette estimation ? Il est en effet possible de quantifier l'information que chaque question j donne sur le paramètre de niveau. Il s'agit de l'information de Fisher, définie par la variance du gradient de la log-vraisemblance en fonction du paramètre de niveau :

$$I_j(\theta) = E_{X_j}\left[\left(\frac{\partial}{\partial\theta}\log f(X_j, \theta, d_j)\right)^2 \middle| \theta\right] \tag{2.2}$$

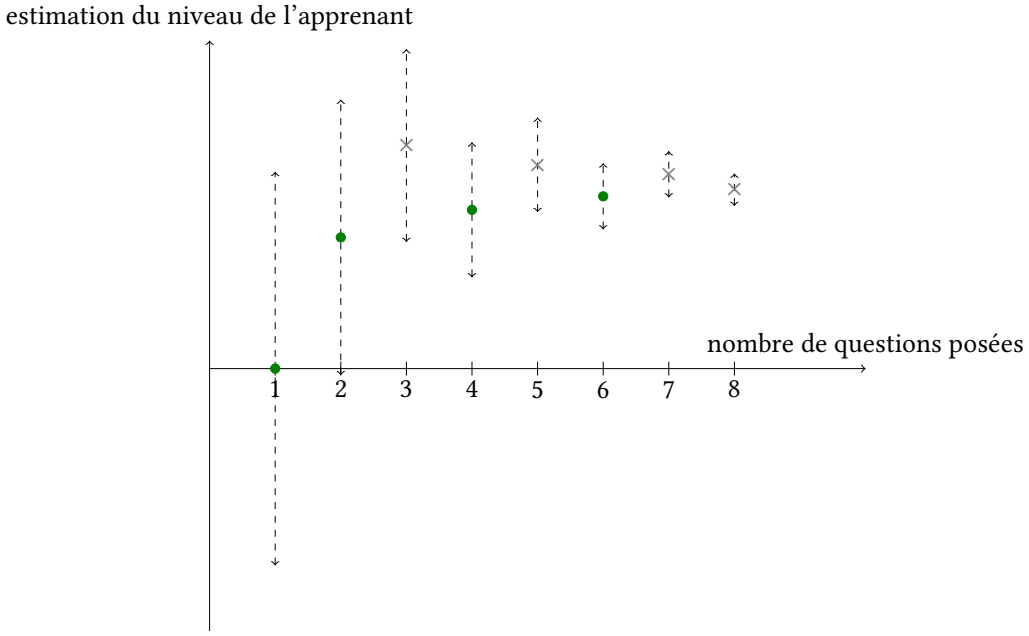
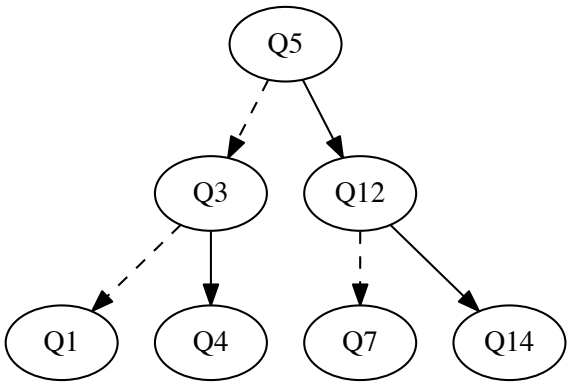


Figure 2.1 – Évolution de l'estimation du niveau via un test adaptatif basé sur le modèle de Rasch. Les croix désignent des mauvaises réponses, les points des bonnes réponses.

- θ est le niveau de l'apprenant qui passe le test en cours ;
- d_j est la difficulté de la question j ;
- X_j est la variable correspondant au succès/échec de l'apprenant sur la question j : elle vaut 1 si i a répondu correctement à j et 0 sinon ;
- et $f(X_j, \theta, d_j)$ est la fonction de probabilité que X_j vaille 1, qui dépend de θ comme indiqué plus haut : $f(X_j, \theta, d_j) = \Phi(\theta - d_j)$.

Ainsi, un test adaptatif peut être conçu de la façon suivante : étant donné l'estimation actuelle du niveau de l'apprenant, choisir la question qui va apporter le plus d'information sur son niveau, mettre à jour l'estimation en fonction du résultat (succès ou échec), et ainsi de suite. À la fin du test, on peut visualiser le processus comme dans les figures 2.1 et 2.2 : l'intervalle de confiance sur le niveau de l'apprenant est réduit après chaque résultat, et les questions sont choisies de façon adaptative.

Le modèle de Rasch est unidimensionnel, donc il ne permet pas d'effectuer un diagnostic cognitif. Il reste pourtant populaire pour sa simplicité, sa généricité (Desmarais et R. S. J. D. Baker, 2012 ; Bergner, Droschler, et al., 2012) et sa robustesse (Bartholomew et al., 2008). Verhelst (2012) a montré qu'avec la simple donnée supplémentaire d'une répartition des questions en catégories, il est possible de renvoyer à l'examiné un profil utile à la fin du test, spécifiant quels sous-scores



Apprenant 1	**Apprenant 2**
On pose la q. 5 à l'apprenant.	On pose la q. 5 à l'apprenant.
Correct !	Incorrect.
On pose la q. 12 à l'apprenant.	On pose la q. 3 à l'apprenant.
Incorrect !	Correct !
On pose la q. 7 à l'apprenant.	On pose la q. 4 à l'apprenant.
Incorrect.	Incorrect.
Le niveau de l'apprenant est 6.	Le niveau de l'apprenant est 3.

Figure 2.2 – Deux exemples de déroulement de test adaptatif pour des apprenants ayant des motifs de réponse différents. Ici on considère que les questions sont de difficulté croissante.

de catégorie sont plus bas ou plus haut que la moyenne.

Théorie de la réponse à l'item multidimensionnelle

Il est naturel d'étendre le modèle de Rasch à des compétences multidimensionnelles. En théorie de la réponse à l'item multidimensionnelle, aussi appelée *Multidimensional Item Response Theory* (MIRT) (Reckase, 2009), les apprenants et les questions ne sont plus modélisés par de simples scalaires mais par des vecteurs de dimension d. La probabilité qu'un apprenant réponde correctement à une question dépend seulement du produit scalaire du vecteur de l'apprenant et du vecteur de la question, plus un paramètre de facilité. Ainsi, un apprenant a plus de chances de répondre à des questions qui sont corrélées à son vecteur de compétences, et poser une question apporte de l'information dans la direction de son vecteur.

Ainsi, si l'apprenant $i \in \{1, \dots, m\}$ est modélisé par le vecteur $\mathbf{\theta_i} \in \mathbb{R}^d$ et la question $j \in \{1, \dots, n\}$ par le vecteur $\mathbf{d_j} \in \mathbb{R}^d$ et le paramètre de facilité $\delta_j \in \mathbb{R}$:

$$Pr(D_{ij} = 1) = \Phi(\mathbf{\theta_i} \cdot \mathbf{d_j} + \delta_j). \tag{2.3}$$

Notez qu'on retrouve le modèle de Rasch lorsque $d = 1$ et $d_{j1} = 1$, avec un paramètre de facilité δ_j à la place d'un paramètre de difficulté d_j.

Lorsqu'on considère un modèle de type MIRT, l'apprenant et les questions ont des caractéristiques selon plusieurs dimensions. L'information de Fisher qu'apporte une question n'est plus un scalaire mais une matrice, dont on cherche habituellement à maximiser soit le déterminant (règle D), soit la trace (règle T). Choisir la question avec la règle D apporte la plus grande réduction de volume dans la variance de l'estimation du niveau, tandis que choisir la question avec la règle T augmente l'information moyenne de chaque dimension du niveau, en ignorant la covariance entre composantes.

Ce modèle plus riche a beaucoup plus de paramètres : d paramètres doivent être estimés pour chacun des m apprenants et $d + 1$ paramètres pour chacune des n questions, soit $d(n+m)+n$ paramètres au total. Ayant de nombreux paramètres, ce modèle est plus difficile à calibrer que le modèle de Rasch (Desmarais et R. S. J. D. Baker, 2012 ; Lan, Waters, et al., 2014).

SPARFA

Lan, Waters, et al. (2014) ont défini un nouveau modèle de tests adaptatifs appelé *Sparse Factor Analysis* (SPARFA). Leur probabilité que l'apprenant réponde correctement à une certaine question repose sur un produit scalaire, ce qui est semblable au modèle MIRT, avec des contraintes supplémentaires.

Si l'apprenant $i \in \{1, \dots, m\}$ est modélisé par le vecteur $\boldsymbol{\theta}_\mathbf{i} \in \mathbb{R}^d$ et la question $j \in \{1, \dots, n\}$ par le vecteur $\mathbf{d}_\mathbf{j} \in \mathbb{R}^d$ et le paramètre de facilité $\delta_j \in \mathbb{R}$:

$$Pr(D_{ij} = 1) = \Phi(\boldsymbol{\theta}_\mathbf{i} \cdot \mathbf{d}_\mathbf{j} + \delta_j). \tag{2.4}$$

Si l'on note V la matrice ayant pour lignes les vecteurs $\mathbf{d}_\mathbf{j}$, SPARFA ajoute comme contrainte que V doit être une matrice uniquement constituée d'entrées positives. De plus, V doit être creuse, c'est-à-dire que la majorité de ses entrées est nulle.

En ajoutant la contrainte que V est creuse, Lan, Waters, et al. (2014) font la supposition que chaque question fait appel à peu de caractéristiques de l'apprenant : en effet, le calcul de la probabilité que l'apprenant i réponde correctement à la question j dépend seulement de $\boldsymbol{\theta}_\mathbf{i} \cdot \mathbf{d}_\mathbf{j} + \delta_j$. Ainsi, pour chaque k tel que d_{jk} vaut 0, ce qui arrive souvent puisque V est creuse, le niveau de l'apprenant θ_{ik} ne sera pas pris en compte dans le calcul de ses chances de succès pour répondre à la question j.

En ajoutant la contrainte que les entrées de V sont positives, Lan, Waters, et al. (2014) supposent que le fait que l'apprenant ait un grand niveau dans une dimension ne peut pas diminuer ses chances de répondre correctement à une question.

Nous aurions voulu intégrer le modèle SPARFA dans notre comparaison de modèles au chapitre suivant, mais leur code n'est pas en accès libre. De plus, le test ainsi considéré est sommatif selon plusieurs dimensions, mais pas formatif, car les caractéristiques extraites par SPARFA ne sont pas facilement interprétables. Lan, Waters, et al. (2014) essaient d'interpréter a posteriori les colonnes de la matrice V, en utilisant des tags spécifiés par des experts sur les questions, mais ce n'est pas toujours possible.

Tests à étapes multiples

Jusqu'à présent, nous n'avons considéré que des questions posées une par une. Mais les premières étapes d'un test adaptatif conduisent à des estimations du niveau de l'apprenant peu représentatives de la réalité, car il y a peu de réponses observées sur lesquelles s'appuyer pour effectuer un diagnostic. C'est pourquoi d'autres recherches en psychométrie portent sur des tests à étapes multiples (Yan, A. A. v. Davier, et Lewis, 2014), qui adaptent le processus d'évaluation seulement après qu'un groupe de questions a été posé. Ainsi, l'adaptation se fait au niveau des groupes et non des questions : après avoir posé un premier ensemble de k_1 questions à un apprenant, un autre ensemble de k_2 questions est sélectionné en fonction de sa performance sur le premier ensemble, et ainsi de suite, voir la figure 2.3. Cela permet également à l'apprenant de vérifier ses réponses avant de valider, ce qui déclenche le processus suivant de questions.

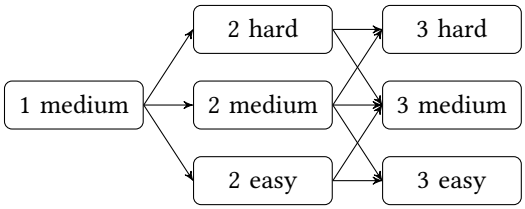


Figure 2.3 – Un exemple de test à étapes multiples. Les questions sont posées par groupe.

Il y a ainsi un compromis entre adapter le processus de façon séquentielle, après chaque question, et ne le faire que lorsque suffisamment d'information a été récoltée sur l'apprenant. Wang, Lin, et al. (2016) suggèrent de poser un groupe de questions au début du test, lorsque peu d'information sur l'apprenant est disponible, puis progressivement réduire le nombre de questions de chaque groupe afin d'augmenter les opportunités d'adapter le processus. Aussi, poser des groupes de questions permet d'équilibrer les ensembles de questions en termes de connaissances évaluées, tandis que poser les questions une par une peut conduire à un test où les connaissances évaluées peuvent beaucoup changer d'une question à l'autre.

2.3.2 Modèles de diagnostic cognitif basés sur les composantes de connaissances

Les *modèles de diagnostic cognitif* font l'hypothèse que la résolution des questions ou tâches d'apprentissage peut être expliquée par la maîtrise ou non-maîtrise de certaines composantes de connaissances (CC), ce qui permet de transférer de l'information d'une question à l'autre. Par exemple, pour calculer $1/7 + 8/9$ correctement, un apprenant est censé maîtriser l'addition, et la mise au même dénominateur. En revanche, pour calculer $1/7 + 8/7$, il suffit de savoir additionner deux fractions de même dénominateur. Ces modèles cognitifs requièrent la spécification des CC impliqués dans la résolution de chacune des questions du test, sous la forme d'une matrice binaire appelée *q-matrice*, qui fait le lien entre les questions et les CC : c'est ce qu'on appelle un modèle de transfert. Un exemple de q-matrice est donné à la table 2.1 pour un test de 20 questions de soustraction de fractions comportant 8 composantes de connaissances. Le jeu de données de test correspondant est étudié dans (DeCarlo, 2010) et à la section 3.4.5 page 56 de cette thèse.

	Comp. de connaissances							
	1	2	3	4	5	6	7	8
Q1	0	0	0	1	0	1	1	0
Q2	0	0	0	1	0	0	1	0
Q3	0	0	0	1	0	0	1	0
Q4	0	1	1	0	1	0	1	0
Q5	0	1	0	1	0	0	1	1
Q6	0	0	0	0	0	0	1	0
Q7	1	1	0	0	0	0	1	0
Q8	0	0	0	0	0	0	1	0
Q9	0	1	0	0	0	0	0	0
Q10	0	1	0	0	1	0	1	1
Q11	0	1	0	0	1	0	1	0
Q12	0	0	0	0	0	0	1	1
Q13	0	1	0	1	1	0	1	0
Q14	0	1	0	0	0	0	1	0
Q15	1	0	0	0	0	0	1	0
Q16	0	1	0	0	0	0	1	0
Q17	0	1	0	0	1	0	1	0
Q18	0	1	0	0	1	1	1	0
Q19	1	1	1	0	1	0	1	0
Q20	0	1	1	0	1	0	1	0

Description des huit composantes de connaissances :

1. convertir un nombre entier en fraction
2. séparer un nombre entier d'une fraction
3. simplifier avant de soustraire
4. mettre au même dénominateur
5. soustraire une fraction d'un entier
6. poser la retenue lors de la soustraction des numérateurs
7. soustraire les numérateurs
8. réduire les fractions sous forme irréductible

Table 2.1 – Exemple de q-matrice pour un test de 20 questions de soustraction de fractions.

Modèle DINA

Le modèle *Deterministic Input, Noisy And* (DINA), qui signifie « entrée déterministe avec un *et* bruité », suppose que l'apprenant résoudra une certaine question i avec probabilité $1 - s_i$ s'il maîtrise toutes les CC impliquées dans sa résolution, sinon avec probabilité g_i. Le paramètre g_i est le paramètre de chance de la question i, c'est-à-dire la probabilité de deviner la bonne réponse alors que l'on ne maîtrise pas les CC nécessaires, tandis que s_i est le paramètre d'inattention, c'est-à-dire la probabilité de se tromper alors qu'on maîtrise les CC associées. Il existe d'autres variantes de modèles cognitifs tels que le modèle DINO (*Deterministic Input, Noisy Or*, c'est-à-dire entrée déterministe, avec un « ou » avec bruit) où ne maîtriser qu'une seule des CC impliquées dans une question i suffit à la résoudre avec probabilité $1 - s_i$, et si en revanche aucune CC impliquée n'est maîtrisée, la probabilité d'y répondre correctement est g_i.

S'il y a K composantes de connaissances mises en œuvre dans le test, l'état latent d'un apprenant est représenté par un vecteur de K bits $(c_1, \dots, c_K)$, un par CC : c_k vaut 1 si l'apprenant maîtrise la k-ième CC, 0 sinon. Chaque réponse que l'apprenant donne sur une question nous donne de l'information sur les états possibles qui pourraient correspondre à l'apprenant. Xu, H. Chang, et Douglas (2003) ont utilisé des stratégies de tests adaptatifs pour inférer l'état latent de l'apprenant en utilisant peu de questions, c'est ainsi qu'ont été développés les modèles de tests adaptatifs pour le diagnostic cognitif, en anglais *Cognitive Diagnostic Computerized Adaptive Tests* (CD-CAT) (Cheng, 2009). À partir d'une estimation a priori de l'état mental de l'apprenant, on peut inférer son comportement sur les questions restantes du test, et choisir des questions informatives en fonction. À chaque étape, le système maintient une distribution de probabilité sur les 2^K états mentaux possibles et l'affine après chaque réponse de l'apprenant en utilisant une approche bayésienne.

Pour converger rapidement vers l'état latent le plus vraisemblable, la meilleure question à poser est celle qui réduit le plus l'incertitude (Doignon et Falmagne, 2012 ; Huebner, 2010), c'est-à-dire l'entropie de la distribution sur les états latents possibles :

$$H(\pi) = -\sum_{c \in \{0,1\}^K} \pi(c) \log \pi(c). \tag{2.5}$$

Nous présentons un exemple de test adaptatif basé sur le modèle DINA, à partir de la q-matrice spécifiée dans la figure 2.4. Si la tâche 1 est présentée à l'apprenant, et qu'il la résout correctement, cela signifie que les CC **form** et **mail** ont de bonnes chances d'être maîtrisées. Il est donc peu informatif de présenter la tâche 2. Si la tâche 4 est présentée, et que l'apprenant échoue, la CC **url** risque de ne pas être maîtrisée, ainsi il n'est pas nécessaire de poser la tâche 3. À la fin

		Composantes de connaissances			
		form	**mail**	**copier**	**url**
T1	Envoyer un mail	**form**	**mail**		
T2	Remplir un formulaire	**form**			
T3	Partager une URL			**copier**	**url**
T4	Entrer une URL				**url**

Figure 2.4 – Exemple de q-matrice pour un test adaptatif basé sur le modèle DINA.

du test, le système peut dire à l'apprenant « Vous semblez maîtriser les CC **form** et **mail** mais pas **url**.

Comme le dit H.-H. Chang (2014), « Une étude conduite à Zhengzhou indique que CD-CAT encourage la pensée critique, en rendant les étudiants plus autonomes en résolution de problèmes, et offre de la remédiation personnalisée facile à suivre, ce qui rend l'apprentissage plus intéressant. » En effet, une fois que l'état mental de l'apprenant a été identifié, on peut l'orienter vers des ressources utiles pour combler ses lacunes.

Comme l'espace des états latents possibles est discret, on peut maintenir une distribution de probabilité $(\pi_i)_{i\geq 0}$ sur les vecteurs de compétences possibles, mise à jour après chaque réponse de l'apprenant. Connaissant la réponse de l'apprenant à la i-ème question, la mise à jour de π_{i-1} est faite par la règle de Bayes. Soit x un état latent, s_i et g_i les paramètres d'inattention et de chance associés à la i-ème question et soit a_i une variable qui vaut 1 si la réponse de l'apprenant est correcte, 0 sinon. Si les CC associées à x sont suffisantes pour répondre à la question correctement,

$$\pi_i(x) \propto \pi_{i-1}(x) \cdot [a_i \cdot (1 - s_i) + (1 - a_i) \cdot s_i] \tag{2.6}$$

sinon

$$\pi_i(x) \propto \pi_{i-1}(x) \cdot [a_i \cdot g_i + (1 - a_i) \cdot (1 - g_i)]. \tag{2.7}$$

En effet : si x a bien les compétences requises, il peut soit donner la bonne réponse en ne faisant pas d'erreur d'inattention (résultat $a_i = 1$ avec probabilité $1 - s_i$), soit faire une erreur d'inattention (résultat $a_i = 0$ avec probabilité s_i).

La complexité du choix de la question suivante est $O(2^K K|Q|)$, ce qui est impraticable pour de larges valeurs de K. C'est pourquoi en pratique $K \leq 10$ (Su et al., 2013).

La q-matrice peut être coûteuse à construire. Ainsi, calculer une q-matrice automatiquement est un sujet de recherche à part entière. Barnes (2005) utilise une

technique d'escalade de colline [5] (qui consiste à modifier un bit de la q-matrice, regarder si le taux d'erreur du modèle diminue, et itérer le processus) tandis que Winters et al. (2005) et Desmarais et al. (2011) ont essayé des méthodes de factorisation de matrice pour recouvrer des q-matrices à partir de données d'apprenants. Ils ont découvert que pour des domaines bien distincts comme le français et les mathématiques, ces techniques permettent de séparer les questions qui portent sur ces domaines. Une critique est que même si l'on obtient via ces méthodes automatiques des matrices qui correspondent bien aux données, les colonnes risquent de ne plus être interprétables. Koedinger, McLaughlin, et Stamper (2012) ont réussi à combiner des q-matrices de différents experts par externalisation ouverte (*crowdsourcing*) de façon à obtenir des q-matrices plus riches, toujours interprétables, et qui correspondent davantage aux données.

Un avantage du modèle DINA est qu'il n'a pas besoin de données de test pour être déjà adaptatif. La q-matrice suffit à administrer des tests, où l'on suppose alors que les apprenants ont autant de chance de maîtriser une composante que de ne pas la maîtriser. À l'aide d'un historique des réponses des apprenants, on peut avoir un a priori sur les composantes qu'un nouvel apprenant maîtrisera ou non, et améliorer l'adaptation.

Modèle de hiérarchie sur les attributs

Il est toutefois possible de réduire la complexité en supposant des relations de prérequis entre composantes de connaissances (CC) : si la maîtrise d'une CC implique celle d'une autre CC, le nombre d'états possibles décroît et donc la complexité en temps fait de même. Cette approche est appelée modèle de hiérarchie sur les attributs (en anglais, *Attribute Hierarchy Model*) (Leighton, Gierl, et Hunka, 2004) et permet d'obtenir des représentations de connaissances qui correspondent mieux aux données (Rupp et al., 2012).

Nous allons à présent présenter la théorie des espaces de connaissances, découverte indépendamment des modèles de hiérarchie sur les attributs, mais qui y ressemble beaucoup.

Théorie des espaces de connaissances basés sur les compétences

Doignon et Falmagne (2012) ont développé la théorie des espaces de connaissances, qui repose sur une représentation abstraite des connaissances similaire aux composantes de connaissances (CC) qui apparaissent dans les q-matrices considérées par le modèle DINA. Ainsi, *l'état des connaissances* d'un apprenant peut être modélisé par l'ensemble des CC qu'il maîtrise. Supposons qu'il existe un certain nombre de CC à apprendre, pour lesquelles on connaît des relations

5. En anglais, *hill-climbing technique*.

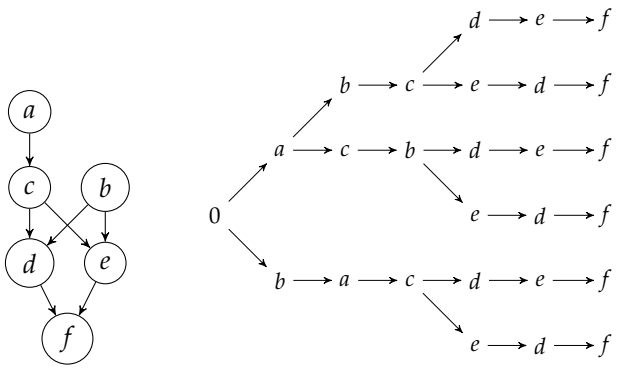


Figure 2.5 – À gauche, un graphe de dépendance. À droite les parcours d'apprentissage possibles pour apprendre toutes les CC.

de prérequis, c'est-à-dire quelles CC doivent être maîtrisées avant d'apprendre une certaine CC (voir figure 2.5). À partir de ce graphe, on peut calculer les états de connaissances dans lesquels l'apprenant peut se trouver. Par exemple, dans la figure 2.5, $\{a, c\}$ est un état des connaissances possible tandis que $\{c\}$ ne l'est pas, car a doit être maîtrisé avant c. Donc pour cet exemple, il y a 10 états de connaissances possibles pour l'apprenant : $\emptyset$, $\{a\}$, $\{b\}$, $\{a, b\}$, $\{a, c\}$, $\{a, b, c\}$, $\{a, b, c, d\}$, $\{a, b, c, e\}$, $\{a, b, c, d, e\}$ et $\{a, b, c, d, e, f\}$. Un test adaptatif peut donc déterminer l'état des connaissances de l'apprenant d'une façon similaire au modèle de hiérarchie sur les attributs décrit plus haut dans cette section. Une fois que l'état des connaissances de l'apprenant a été identifié, le modèle peut lui suggérer les prochaines CC à apprendre pour progresser, à travers ce que l'on appelle un parcours d'apprentissage (voir figure 2.5). Par exemple, si l'apprenant a pour état de connaissances $\{a\}$, il peut choisir d'apprendre b ou c.

Falmagne et al. (2006) proposent un test adaptatif pour deviner de façon efficace l'état des connaissances de l'apprenant en minimisant l'entropie de la distribution sur les états des connaissances possibles de l'apprenant, mais leur méthode n'est pas robuste aux erreurs d'inattention. Ce modèle a été implémenté dans le système ALEKS, qui appartient désormais à McGraw-Hill Education et est utilisé par des millions de personnes aujourd'hui (Kickmeier-Rust et Albert, 2015 ; Desmarais et R. S. J. D. Baker, 2012).

Lynch et Howlin (2014) ont implémenté un test adaptatif analogue à la construction de Falmagne et al. (2006) au début d'un MOOC de façon à deviner ce que l'apprenant maîtrise déjà et l'orienter automatiquement vers des ressources utiles du cours. Pour résister aux erreurs d'inattention et aux apprenants qui devinent les bonnes réponses sans avoir les CC nécessaires, ils combinent des modèles de la théorie des espaces de connaissances et de la théorie de la réponse à l'item, sans donner les détails de leurs constructions.

Certains modèles plus fins pour le diagnostic de connaissances considèrent des représentations de connaissances plus riches, telles que des réseaux bayésiens (Shute, 2011 ; Rupp et al., 2012) ou des ontologies du domaine couvert par le test (Mandin et Guin, 2014 ; Kickmeier-Rust et Albert, 2015). Toutefois, de telles représentations sont coûteuses à construire, car il faut spécifier le poids d'une certaine relation entre la représentation de connaissances et chaque question du test.

2.3.3 Lien avec l'apprentissage automatique

Tests adaptatifs et filtrage collaboratif

Une application de l'apprentissage automatique est l'élaboration de systèmes de recommandation, capables de recommander des ressources à des utilisateurs en fonction d'autres ressources qu'ils ont appréciées. En technologies de l'éducation, de tels systèmes sont appliqués à la recommandation de ressources pédagogiques (Chatti et al., 2012 ; Manouselis et al., 2011 ; Verbert et al., 2011).

Le but est de prédire le comportement d'un utilisateur face à une ressource inédite, à partir de ses préférences sur une fraction des ressources qu'il a consultées. Deux techniques sont principalement utilisées.

1. Des recommandations *basées sur le contenu*, qui analysent le contenu des ressources de façon à calculer une mesure de similarité entre ressources, pour recommander des ressources similaires à celles appréciées par l'utilisateur. Ici, la communauté d'utilisateurs n'a pas d'impact sur les recommandations.
2. Le *filtrage collaboratif*, où la mesure de similarité entre ressources dépend seulement des préférences des utilisateurs : des produits étant préférés par les mêmes personnes sont supposés proches. À partir des données communiquées par les autres internautes (*collaboratif*), il est possible de faire le tri de façon automatique (*filtrage*) pour un nouvel utilisateur, par exemple en identifiant des internautes ayant aimé des produits similaires et en lui suggérant des ressources qui les ont satisfaits.

Dans notre cas, nous devons prédire la performance d'un apprenant sur une question inédite, en fonction de son comportement sur d'autres questions et du comportement que d'autres apprenants ont eu dans le passé sur le même test. Les techniques de filtrage collaboratif ont été appliquées à deux problèmes issus de la fouille de données éducatives : la recommandation de ressources éducatives à des apprenants (Manouselis et al., 2011 ; Verbert et al., 2011) et la prédiction de performance d'un apprenant sur un test (Toscher et Jahrer, 2010 ; Thai-Nghe et al., 2011 ; Bergner, Droschler, et al., 2012).

En filtrage collaboratif, on fait l'hypothèse que l'on dispose d'utilisateurs ayant noté certains objets : m_{ui} désigne la note que l'utilisateur u affecte à l'objet i.

	Zootopie	12 Monkeys	Oldboy	Paprika
Sacha	?	5	2	?
Ondine	4	1	?	5
Pierre	3	3	1	4
Joëlle	5	?	2	?

Table 2.2 – Un exemple de problème de complétion de matrice.

La matrice observée $M = (m_{ui})$ est creuse, c'est-à-dire qu'une faible partie de ses entrées est renseignée. Le problème consiste à déterminer les entrées manquantes de M (voir table 2.2). Afin d'accomplir cette tâche, on suppose en général que M a un faible rang, c'est-à-dire que les notes des utilisateurs sont dans un espace de faible dimension, ou encore qu'on peut les exprimer par un faible nombre de composantes.

L'historique d'un test peut également être représenté par une matrice $M = (m_{ui})$ où l'élément m_{ui} représente 1 si l'apprenant u a répondu correctement à la question i, 0 sinon. Administrer un test adaptatif à un nouvel apprenant revient à ajouter une ligne dans la matrice et choisir les composantes à révéler (les questions à poser) de façon à inférer les composantes restantes (les questions qui n'ont pas été posées).

Un autre élément qui apparaît dans les systèmes de recommandation peut être utile à notre analyse, celui du *démarrage à froid de l'utilisateur* : étant donné un nouvel utilisateur, comment lui recommander des ressources pertinentes ? La seule référence que nous ayons trouvée au problème du démarrage à froid dans un contexte éducatif vient de Thai-Nghe et al. (2011) : « Dans des environnements éducatifs, le problème du démarrage à froid n'est pas aussi dérangeant que dans des environnements commerciaux, où de nouveaux utilisateurs ou produits apparaissent chaque jour ou même chaque heure. Donc, les modèles n'ont pas besoin d'être réentraînés continuellement. » Toutefois, l'arrivée des MOOC en 2011 a accentué le besoin de mettre à jour fréquemment les modèles de recommandation de ressources.

Parmi les techniques les plus populaires pour s'attaquer au problème du démarrage à froid de l'utilisateur, une méthode qui nous intéresse particulièrement est un test adaptatif qui présente certaines ressources à l'apprenant et lui demande de les noter. Golbandi, Koren, et Lempel (2011) construisent un arbre de décision qui pose des questions à un nouvel utilisateur et choisit en fonction de ses réponses la meilleure question à lui poser de façon à identifier rapidement un groupe d'utilisateurs qui lui sont proches. Les meilleures questions sont celles qui séparent la population en trois parties de taille similaire, selon si l'utilisateur a apprécié la ressource, n'a pas apprécié la ressource, ou ne connaît pas la res-

source. La différence principale avec notre cadre éducatif est que les apprenants risquent de moins coopérer avec un système d'évaluation qu'avec un système de recommandations commercial, car leur objectif n'est pas d'obtenir des bonnes recommandations mais un bon score. Ainsi, leurs réponses risquent de ne pas refléter les compétences qu'ils maîtrisent vraiment. C'est pourquoi les modèles que l'on considère pour les tests adaptatifs doivent prendre en compte le fait que l'apprenant puisse faire des fautes d'inattention, ou deviner la bonne réponse.

Stratégies adaptatives pour le compromis exploration-exploitation

Dans certaines applications de tests adaptatifs, on souhaite maximiser une fonction objectif pendant qu'on pose les questions. Par exemple, supposons qu'un site commercial cherche à maximiser le nombre de clics sur ses publicités. Il y a un compromis entre explorer l'espace des publicités en présentant à l'utilisateur des publicités plus risquées, et exploiter la connaissance de l'utilisateur en lui présentant des publicités sur des domaines susceptibles de lui plaire.

Dans un contexte éducatif, on peut se demander quelle serait la tâche qui permettrait de maximiser la progression de l'apprenant, tout en cherchant à identifier ce qu'il maîtrise ou non. C'est la technique que Clement et al. (2015) adoptent pour les systèmes de tuteurs intelligents : ils personnalisent les séquences d'activités d'apprentissage de façon à identifier les CC de l'apprenant tout en maximisant son progrès, défini comme la performance sur les dernières activités.

Pour cela, ils utilisent des modèles de bandits. Le problème du bandit manchot consiste à se demander, si l'on dispose de k machines à sous d'espérances de gain inconnues, quelles machines essayer séquentiellement pour maximiser le gain. Il y a donc un compromis entre exploiter les machines dont on sait qu'elles ont une bonne espérance avec un intervalle de confiance serré, et explorer les machines plus risquées, parce qu'elles ont un intervalle de confiance plus large.

Clement et al. (2015) utilisent deux modèles de bandits, l'un se basant sur la zone proximale de développement de Vygotski [6] (Vygotsky, 1980) sous la forme d'un graphe de prérequis, l'autre sur une q-matrice. Ils ont comparé ces deux approches sur 400 apprenants de 7 et 8 ans, et ont découvert que le graphe de dépendance se comportait mieux que le modèle qui utilise une q-matrice spécifiée par un expert. Leur technique est adaptée à des populations d'apprenants de niveaux variés, notamment ceux ayant des difficultés.

6. En français, Vygotski, en anglais, Vygotsky.

2.4 Comparaison de modèles de tests adaptatifs

Comme le disent Desmarais et R. S. J. D. Baker (2012) : « Les modèles de tests adaptatifs doivent être validés sur des données réelles afin de garantir que le modèle évalue bien ce qu'on croit qu'il évalue. Une validation usuelle réside dans la capacité de l'évaluation à prédire la performance future au sein du système d'apprentissage. »

Habituellement, on compare pour un même modèle plusieurs stratégies de choix de la question suivante. Cheng (2009) compare ainsi plusieurs critères de sélection de la question suivante pour le modèle DINA utilisé dans un cadre de tests adaptatifs. Pour ses expériences, elle considère des données simulées.

Lallé (2013) utilise une technique de validation croisée pour comparer des modèles de diagnostic de connaissances, mais pas dans le cadre de tests adaptatifs. Plus rarement, certaines recherches comparent des modèles de tests adaptatifs différents sur de mêmes données de test : Lan, Waters, et al. (2014) comparent SPARFA et le modèle de Rasch, Bergner, Droschler, et al. (2012) comparent des algorithmes de filtrage collaboratif au modèle de Rasch. Toutefois, nous n'avons pas observé de comparaison de modèles sommatifs avec des modèles formatifs.

2.5 Conclusion

Dans ce chapitre, nous avons présenté différents modèles de tests adaptatifs identifiés dans divers pans de la littérature, que nous avons classés selon trois catégories : théorie de la réponse à l'item, modèles de diagnostic basés sur les composantes de connaissances et apprentissage automatique. Nous avons mentionné différents travaux consistant à les comparer sur un même jeu de données.

L'avantage principal des modèles de théorie de la réponse à l'item est qu'ils peuvent être calibrés automatiquement à partir d'un historique des réponses d'un test, ce qui est utile dans des évaluations faites par ordinateur où l'on a facilement accès à un large historique d'utilisation, en particulier dans les MOOC. Ils conviennent donc tout à fait au cadre de l'analytique de l'apprentissage. En contrepartie, de tels tests sommatifs sont moins utiles à l'apprentissage que les modèles basés sur les composantes de connaissances qui permettent d'indiquer à l'apprenant et son professeur les points à améliorer. Ceux-ci ne dépendent plus d'un historique d'utilisation mais d'une représentation de connaissances minimale, faisant le lien entre chaque question et les CC qu'elle requiert pour être résolue correctement. Enfin, il nous a semblé important d'introduire les modèles d'apprentissage automatique pour leur cadre générique qui consiste à tenter d'optimiser une fonction objectif bien définie à partir de données existantes, et qui ont été utilisés dans de multiples domaines, associés à de larges bases de données.

Dans le chapitre suivant, nous allons faire une comparaison qualitative de ces modèles et proposer un système de comparaison quantitative de modèles de tests adaptatifs.

Chapitre 3

Système de comparaison de modèles de tests adaptatifs

3.1 Introduction

Des modèles de tests adaptatifs ont été proposés dans différentes communautés, que ce soit en théorie de la réponse à l'item, dans les modèles de diagnostic cognitif et en apprentissage automatique, c'est pourquoi il nous a semblé important de les comparer sur un même jeu de données. En ce qui nous concerne, comme nous cherchons à réduire le nombre de questions posées, nous devons vérifier que le diagnostic obtenu en peu de questions reflète bien le comportement de l'apprenant sur les questions restantes du test.

Dans ce chapitre, nous commençons par présenter les différents composants d'un test adaptatif que nous pouvons paramétrer, puis nous exposons notre première contribution, un système de comparaison de modèles de tests adaptatifs. Notre méthodologie de comparaison est faite selon deux axes, un axe qualitatif, purement descriptif, et un autre quantitatif, inspiré de la méthode de validation croisée que l'on rencontre en statistiques et en apprentissage automatique. Cela consiste à séparer un jeu de données de réponses d'apprenants existant en deux parties, l'une utilisée pour calibrer les caractéristiques des modèles, l'autre pour simuler des tests adaptatifs. Ainsi, une partie des données des apprenants est cachée pour évaluer les prédictions que le modèle de test adaptatif fait de l'apprenant simulé, après chacune de ses réponses, et le procédé est ainsi entièrement automatisable.

Nous avons implémenté ce système et nous l'avons appliqué à la comparaison du modèle de Rasch issu de la théorie de la réponse à l'item et du modèle DINA de diagnostic cognitif. Nous décrivons les jeux de données réelles utilisés dans notre implémentation (des données de tests standardisés mais aussi de concours et de

MOOC), nous précisons les choix que nous avons faits dans la spécification des différents composants des modèles de Rasch et DINA, et enfin nous présentons nos résultats.

Pour terminer ce chapitre, nous proposons une méthodologie de choix de modèles pour différents usages dans un MOOC, et l'illustrons par une analyse des données d'un MOOC de Coursera. Étant donné que la plupart des MOOC partagent la même structure, cette méthode peut être appliquée à des données provenant d'autres plateformes de MOOC tels que edX (notamment la plateforme française France Université Numérique).

3.2 Composants modulables d'un test adaptatif

3.2.1 Modèle de réponse de l'apprenant

Tous les modèles de tests adaptatifs reposent sur des caractéristiques des questions et des apprenants, spécifiés par un expert ou déterminés automatiquement au moyen d'algorithmes. Ils reposent sur une expression de la probabilité qu'un certain apprenant réponde à une certaine question, en fonction de leurs caractéristiques.

3.2.2 Calibrage des caractéristiques

Les caractéristiques des questions et des apprenants peuvent être spécifiées à la main par un enseignant, ou bien calculées automatiquement à partir de ce qu'on appelle des *données d'entraînement*. Si le test est administré pour la première fois, il n'y a pas de données d'entraînement, sinon on dispose d'un historique de réponses d'une population I d'apprenants face à des questions d'un ensemble Q, sous la forme d'une matrice $|I| \times |Q|$ dont l'élément m_{ij} vaut 1 si l'apprenant i a répondu correctement à la question j, 0 sinon.

En général, les valeurs calculées automatiquement conduisent à une erreur du modèle plus faible, car les algorithmes de calibrage sont justement conçus pour minimiser le taux d'erreur autant que possible sur les données d'entraînement, contrairement à un humain dont les valeurs affectées peuvent être subjectives et ne pas correspondre à la réalité.

Il est également possible de spécifier une partie des caractéristiques et de calculer automatiquement les autres. Lorsqu'il y a plusieurs caractéristiques à optimiser, il est possible d'en optimiser une première en fixant toutes les autres, puis optimiser la deuxième en fixant toutes les autres, et ainsi de suite, puis itérer plusieurs fois cette optimisation séquentielle de toutes les caractéristiques jusqu'à obtenir un taux d'erreur suffisamment faible.

3.2.3 Initialisation des paramètres d'un nouvel apprenant

Au début d'un test adaptatif, le système n'a aucune information sur l'apprenant, car il n'a fourni aucune réponse et nous ne considérons aucune donnée sur l'apprenant nous permettant de l'identifier, notamment des données démographiques comme son âge ou son pays.

Le système peut supposer que l'apprenant est de niveau nul, ou de niveau moyen au sein de la population.

3.2.4 Choix de la question suivante

À un certain moment du test, le système doit, à partir des questions déjà posées et de leurs résultats, choisir la question suivante. Ainsi, la fonction qui choisit la question suivante prend en paramètre une liste de couples $\{(i_k, r_k)\}_k$ où r_k vaut 1 si l'apprenant a répondu correctement à la question i_k, 0 sinon.

3.2.5 Retour fait à la fin du test

L'apprenant obtient à la fin du test ses caractéristiques qui ont été calculées pendant le processus. Cela peut comprendre la liste des questions qu'il a résolues correctement ou non, munies éventuellement de leurs caractéristiques. Par exemple, le modèle de Rasch renvoie une valeur réelle de niveau tandis que le modèle DINA indique la probabilité que le candidat maîtrise chacune des CC.

Pour visualiser cette information, diverses méthodes peuvent être employées. Pour le modèle de Rasch, on peut indiquer à l'apprenant où il se trouve au sein de la population (par exemple, dans les 10 % meilleurs). Pour le modèle DINA, on peut tracer des jauges de maîtrise des différentes compétences, à partir de sa probabilité de les maîtriser.

3.3 Évaluation qualitative

Plusieurs aspects font qu'on peut préférer un modèle de test adaptatif plutôt qu'un autre. Par exemple, la mise en œuvre d'un modèle de test peut requérir la construction d'une représentation des connaissances, ce qui peut être coûteux si l'on a plusieurs milliers de questions à apparier avec une centaine de composantes de connaissances.

Dimension Est-ce que le modèle considère une ou plusieurs dimensions de l'apprenant ?

Nombre de paramètres Combien de paramètres au niveau des questions doivent être estimés lors du calibrage des caractéristiques du modèle ?

	Dimension	Calibrage	De zéro	Nombre de paramètres
Rasch	1	Auto	Non	n
MIRT	$K \leq 4$	Auto	Non	$(K+1)n$
SPARFA	$K \leq 16$	Auto	Non	$(k+1)n$
DINA	$K \leq 15$	Manuel	Oui	$2n$
AHM	$K \leq 90$	Manuel	Oui	$2n$
KST	$K \leq 90$	Manuel	Oui	0
Bandits	$K \leq 7$	Manuel	Oui	0

Table 3.1 – Comparaison qualitative des modèles présentés

Calibrage Le calibrage des caractéristiques du modèle est-il entièrement fait de façon manuelle, entièrement fait de façon automatique, ou partiellement manuel ?

Interprétabilité Dans les évaluations formatives, il est important de pouvoir nommer les composantes de connaissances dont l'apprenant a dû faire preuve, de façon satisfaisante ou insatisfaisante. Disposer d'une q-matrice spécifiée par un humain permet d'accroître l'interprétabilité du système, car il est alors possible d'identifier les lacunes de l'apprenant soulignées par le test.

De zéro Est-ce que le modèle a besoin de données existantes d'apprenants ayant déjà passé le test pour fonctionner ou est-ce que le test peut être adaptatif dès sa première administration ?

Complexité Quelle est la complexité de chacun des composants modulables ?

Nous avons ainsi comparé les modèles présentés au chapitre précédent : Rasch, MIRT (théorie de la réponse à l'item multidimensionnelle), SPARFA (*Sparse Factor Analysis*), DINA, AHM (modèle de hiérarchie sur les attributs), KST (théorie des espaces de connaissances) et enfin Bandits (le modèle de bandits dans les systèmes de tuteurs intelligents). Les résultats sont répertoriés dans la table 3.1. En l'occurrence, tous les modèles automatiques ne sont pas interprétables sans intervention d'un expert a posteriori. n désigne le nombre de questions, K la dimension du modèle et dans le cas de SPARFA, k désigne le nombre moyen de composantes de connaissances par question.

Les modèles issus de la théorie de la réponse à l'item (Rasch, MIRT et SPARFA) ont une calibration de leurs paramètres qui est automatique. C'est pourquoi ils nécessitent des données existantes d'apprenants sur les questions d'un test pour être calibrés. Pour MIRT, nous supposons $K \leq 4$ car nous ne sommes pas parvenus à faire converger un modèle MIRT de dimension 5 sur nos jeux

de données comportant 20 questions et moins de 1000 apprenants après 2000 itérations. Pour SPARFA, nous n'avons pas accès à l'implémentation mais dans (Lan, Studer, et al., 2014), les auteurs en calibrent une instance de dimensions $K = 16$. Comme $k < K$, leur méthode peut estimer des modèles de plus grande dimension que MIRT.

Les modèles basés sur les composantes de connaissances nécessitent de préciser une q-matrice voire un graphe de prérequis sur les CC (KST, AHM). En contrepartie, le test peut être administré sans donnée préalable. La seule chose qui diffère entre les modèles KST et AHM, c'est que KST ne considère pas de paramètres d'inattention et de chance. DINA et KST ont deux paramètres d'inattention et de chance à fixer à la main ou estimer automatiquement par question, ce qui fait $2n$ paramètres en tout.

Enfin, le modèle de bandit requiert une q-matrice, et de façon optionnelle un graphe de prérequis sur les activités à présenter à l'apprenant. Dans leur expérience, Clement et al. (2015) considèrent une q-matrice de $K = 7$. Ils n'ont aucun paramètre à estimer pour administrer un test, donc le test peut être administré de zéro.

3.4 Méthodologie de comparaison quantitative de modèles

Nous allons employer un formalisme qui vient de l'apprentissage automatique pour définir notre problème.

3.4.1 Apprentissage automatique à partir d'exemples

Lorsqu'on cherche à modéliser un phénomène naturel, on peut utiliser un modèle statistique, dont on estime les paramètres en fonction des occurrences observées. Par exemple, si on suppose qu'une pièce suit une loi de Bernoulli et tombe sur Face avec probabilité p et Pile avec probabilité $1 - p$, on peut estimer p à partir de l'historique des occurrences des lancers de la pièce. Plus l'historique est grand, meilleure sera l'estimation de p. À partir de ce modèle, il est possible de faire des prédictions sur les futurs lancers de la pièce.

L'apprentissage automatique consiste à construire, à partir d'exemples, des modèles capables de prédire des caractéristiques sur des données inédites, par exemple : reconnaître des chiffres sur des codes postaux, ou des chats sur des images. Plus il y a d'exemples pour entraîner le modèle, meilleures sont ses prédictions.

En ce qui nous concerne, nous souhaitons utiliser l'historique des réponses d'apprenants sur un test pour permettre de concevoir automatiquement un test

adaptatif composé des mêmes questions. Dans le cadre d'un MOOC, par exemple, il sera possible de réutiliser les données des apprenants d'une session pour proposer des tests plus efficaces pour la session suivante.

On distingue deux types d'apprentissage automatique. L'*apprentissage supervisé* consiste à disposer d'échantillons étiquetés, c'est-à-dire appariés avec une variable d'intérêt, et à devoir prédire les étiquettes d'échantillons inédits. L'*apprentissage non supervisé* consiste à disposer d'échantillons non étiquetés, et donc à déterminer des motifs récurrents au sein des échantillons ou à en extraire des caractéristiques utiles pour expliquer les données.

En apprentissage supervisé, on appelle *classifieur* un modèle qui prédit une variable discrète, et *régresseur* un modèle qui prédit une variable continue. Ainsi, à partir des *caractéristiques* d'un échantillon $\mathbf{x} = (x_1, \dots, x_d)$, par exemple les couleurs des pixels d'une image, un classifieur peut prédire une variable y dite *étiquette*, par exemple le chiffre 1 si l'image est un chat, 0 sinon. Une fois ce modèle entraîné sur des échantillons de caractéristiques $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(e)}$ étiquetées par les variables $y^{(1)}, \dots, y^{(e)}$ (c'est la *phase d'entraînement*), on peut s'en servir pour prédire les étiquettes d'échantillons inédits $\mathbf{x'}^{(1)}, \dots, \mathbf{x'}^{(t)}$ (c'est la *phase de test*).

Ainsi on distingue les données d'entraînement $X_{train} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(e)})$, sous la forme d'une matrice de taille $e \times d$ où e est le nombre d'échantillons et d la dimension des caractéristiques, et leurs étiquettes $\mathbf{y}_{train} = (y^{(1)}, \dots, y^{(e)})$ des données de test $X_{test} = (\mathbf{x'}^{(1)}, \dots, \mathbf{x'}^{(t)})$.

En ce qui nous concerne, nous disposons des résultats de plusieurs apprenants sur les questions d'un test, et nous cherchons à prédire les résultats d'un nouvel apprenant alors qu'il passe ce même test, sous la forme de succès ou échecs à chacune des questions. Notre problème commence par une phase d'apprentissage non supervisé, car à partir du simple historique des résultats au test, il faut extraire des caractéristiques sur les apprenants et les questions qui expliquent ces résultats. Puis, on se ramène à une phase d'apprentissage supervisé pour un nouvel apprenant car il s'agit d'un problème de classification binaire : on cherche à prédire à partir des réponses que donne l'apprenant ses résultats sur le reste des questions du test. Une particularité est que l'apprentissage du modèle est ici interactif, dans la mesure où c'est le modèle qui choisit les questions à poser (c'est-à-dire, les éléments à faire étiqueter) à l'apprenant afin d'améliorer son apprentissage des caractéristiques de l'apprenant. Cette approche s'appelle apprentissage actif (*active learning*), et dans ce cadre elle comporte du bruit, car l'apprenant peut faire des fautes d'inattention ou deviner la bonne réponse.

3.4.2 Extraction automatique de q-matrice

Il peut arriver que pour un test donné, on ne dispose pas de q-matrice. Nous avons mentionné à la section 2.3.2 page 36 diverses méthodes utilisées pour en extraire automatiquement à partir de données d'apprenants.

Nous avons testé des approches plus génériques. Zou, Hastie, et Tibshirani (2006) présentent un algorithme pour l'analyse de composantes principales creuses, qui détermine deux matrices W et H tels que :

$$M \simeq WH \text{ et } H \text{ est creuse.} \tag{3.1}$$

Lee, Huang, et Hu (2010) proposent une analyse de composantes principales creuses avec une fonction de lien logistique, ce qui est plus approprié pour notre problème pour lequel la matrice que nous cherchons à approximer est binaire.

$$M \simeq \Phi(WH) \text{ et } H \text{ est creuse.} \tag{3.2}$$

Dans ces deux cas, H est composée majoritairement de 0. Pour en extraire une q-matrice, nous fixons à 1 les entrées non nulles.

3.4.3 Validation bicroisée

Pour valider un modèle d'apprentissage supervisé, une méthode courante consiste à estimer ses paramètres à partir d'une fraction des données et leurs étiquettes, calculer les prédictions faites sur les données restantes et les comparer avec les vraies étiquettes. Cette méthode s'appelle *validation croisée*. Ainsi, le jeu de données X est divisé en deux parties X_{train} et X_{test}, le modèle est entraîné sur X_{train} et ses étiquettes $\mathbf{y}_{train}$ et fait une prédiction sur les données X_{test} appelée $\mathbf{y}_{pred}$, qui est ensuite comparée aux vraies valeurs $\mathbf{y}_{test}$ pour validation. S'il s'agit d'un problème de régression, on peut utiliser par exemple la fonction de coût RMSE (*root mean squared error*) :

$$\mathrm{RMSE}(\mathbf{y}_{test}, \mathbf{y}_{pred}) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y_i^* - y_i)^2} \tag{3.3}$$

où $\mathbf{y}_{pred} = (y_1, \dots, y_n)$ et $\mathbf{y}_{test} = (y_1^*, \dots, y_n^*)$.

S'il s'agit d'un problème de classification binaire, on utilise habituellement la fonction de coût *log-loss* (aussi appelée coût logistique ou perte d'entropie mutuelle) :

$$\mathrm{logloss}(\mathbf{y}_{test}, \mathbf{y}_{pred}) = \frac{1}{n}\sum_{k=1}^{n}\log(1 - |y_k^* - y_k|). \tag{3.4}$$

		Questions							
		1	2	3	4	5	6	7	8
Entraînement	Alice	0	1	1	1	0	0	0	1
	Bob	1	0	1	1	0	0	0	1
	Charles	1	0	1	0	0	0	0	0
	Daisy	1	0	0	1	1	1	1	1
	Everett	1	0	0	0	1	0	0	1
	Filipe	0	1	0	1	1	1	1	1
	Gwen	0	0	0	1	0	0	1	1
Test	Henry	0	0	0	0	1	0	0	1
	Ian	1	1	1	1	0	1	1	0
	Jill	0	1	1	1	0	0	1	0
	Ken	1	1	1	0	1	1	0	1

Figure 3.1 – Jeu de données séparé pour la validation bicroisée.

Toutes les valeurs prédites étant comprises entre 0 et 1, cette fonction pénalise beaucoup plus une grosse différence entre valeur prédite (comprise entre 0 ou 1) et valeur réelle (égale à 0 ou 1) que la RMSE.

Afin d'obtenir une validation plus robuste, il faut s'assurer que la proportion de 0 et de 1 soit la même dans les étiquettes d'entraînement et dans les étiquettes d'évaluation. Pour une validation encore meilleure, on peut recourir à une validation croisée à k paquets : le jeu de données X est divisé en k paquets, et k validations croisées sont faites en utilisant $k-1$ paquets parmi les k pour entraîner le modèle et le paquet restant pour l'évaluer.

Dans notre cadre, nous avons deux types de populations : les apprenants de l'historique, pour lesquels nous avons observé les résultats à toutes les questions, et les apprenants pour lesquels on souhaite évaluer le modèle de test adaptatif. Nous faisons donc une validation bicroisée, car nous séparons les apprenants en deux groupes d'entraînement et de test, et également les questions en deux groupes de test et de validation. À la figure 3.1, un exemple de découpage est présenté. Chaque ligne correspond à un apprenant et pour chaque apprenant de test, seules les questions 1, 2, 4, 6, 7 sont posées, les questions 3, 5, 8 étant conservées pour validation (en grisé). Pour chaque apprenant du groupe d'entraînement, nous connaissons toutes ses réponses et pouvons entraîner nos modèles à partir de ces données. Pour chaque apprenant du groupe de test, nous simulons un test adaptatif qui choisit les questions à poser, hors celles de validation. Nous vérifions alors, à chaque étape du test adaptatif pour l'apprenant, les prédictions du modèle de son comportement sur l'ensemble des questions de validation.

Notre comparaison de modèles a deux aspects : qualitatifs en termes d'interprétabilité ou d'explicabilité et quantitatifs en termes de vitesse de convergence de la phase d'entraînement et performance des prédictions.

3.4.4 Évaluation quantitative

Rapidité de convergence vers un diagnostic Combien de questions sont nécessaires pour faire converger le diagnostic ?

Pouvoir prédictif Est-ce que le diagnostic fait après un nombre réduit de questions permet effectivement d'expliquer les résultats de l'apprenant sur le reste du test ?

Nous cherchons à comparer le pouvoir prédictif de différents modèles de tests adaptatifs qui modélisent la probabilité qu'un certain apprenant résolve une certaine question d'un test. Ces modèles sont comparés sur un jeu de données réel D de taille $|I| \times |Q|$ où D_{iq} vaut 1 si l'apprenant i a répondu correctement à la question q, 0 sinon. Pour faire une validation bicroisée, nous séparons les apprenants de l'ensemble I en U paquets et les questions de l'ensemble Q en V paquets. Ainsi, si on numérote les paquets d'apprenants I_i pour $i = 1, \dots, U$ et les paquets de questions Q_j pour $j = 1, \dots, V$, l'expérience (i, j) consiste à, pour chaque modèle T :

1. entraîner le modèle T sur tous les paquets d'apprenants sauf le i-ème (l'ensemble d'apprenants d'entraînement $I_{train} = I \setminus I_i$) ;
2. simuler des tests adaptatifs sur les apprenants du i-ème paquet (l'ensemble d'apprenants de test $I_{test} = I_i$) en utilisant les questions de tous les paquets sauf le j-ème, et après chaque réponse de l'apprenant, en évaluant le taux d'erreur du modèle T sur le j-ème paquet de questions (l'ensemble de questions de validation $Q_{val} = Q_j$). On fait donc un appel à SIMULATE(modèle M, I_{train}, I_{test}), voir l'algorithme 1.

Par exemple, sur la figure 3.2, après que la question la plus informative a été choisie puis posée, les caractéristiques de l'apprenant sont mises à jour, une prédiction est faite sur l'ensemble de questions de validation et cette prédiction est évaluée étant donné le vrai motif de réponse de l'apprenant.

Les variables qui apparaissent dans l'algorithme sont les suivantes :

— D représente le jeu de données, ainsi $D[I_{train}]$ correspond aux motifs de réponse des apprenants de l'ensemble d'entraînement et $D[s][Q_{val}]$ correspond aux motifs de réponse de l'apprenant s sur l'ensemble de questions de validation Q_{val} ;
— α représente les caractéristiques des apprenants de l'ensemble d'entraînement ;
— κ représente les caractéristiques des questions de l'ensemble d'entraînement ;
— π_t représente les caractéristiques d'un apprenant qui passe le test, à l'instant t ;
— q_t est la question posée à l'apprenant au temps t ;

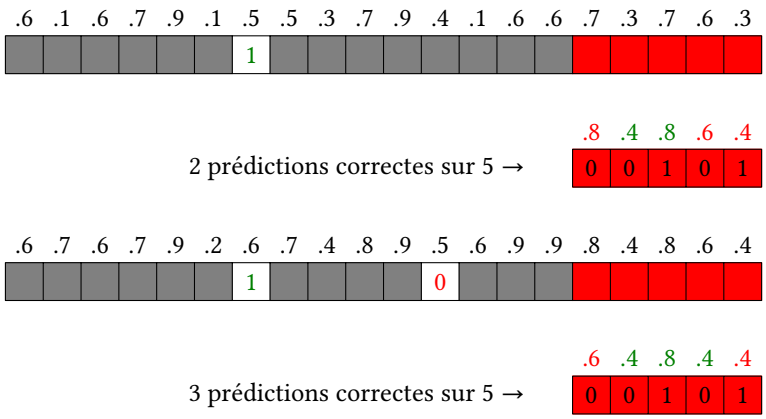


Figure 3.2 – Exemple de phase de test.

- r_t le succès ou échec de l'apprenant sur la question q_t ;
- p désigne la probabilité de succès de l'apprenant en cours sur chaque question de l'ensemble de validation Q_{val} ;
- enfin, σ_t désigne la performance du modèle M à l'instant t, c'est-à-dire après avoir posé t questions.

Algorithme 1 Simulation d'un modèle de tests adaptatifs

procédure SIMULATE(modèle M, I_{train}, I_{test})
  $\alpha, \kappa \leftarrow$ TRAININGSTEP($M, D[I_{train}]$)
  pour tout apprenant s de l'ensemble I_{test} **faire**
    $\pi_0 \leftarrow$ PRIORINITIALIZATION(α)
    pour t de 0 à $|Q \setminus Q_{val}| - 1$ **faire**
      $q_{t+1} \leftarrow$ NEXTITEM($\{(q_k, r_k)\}_{k=1,\dots,t}, \kappa, \pi_t$)
      Poser la question q_{t+1} à l'apprenant s
      Récupérer la valeur de succès ou échec r_{t+1} de sa réponse
      $\pi_{t+1} \leftarrow$ ESTIMATEPARAMETERS($\{(q_k, r_k)\}_{k=1,\dots,t+1}, \kappa$)
      $p \leftarrow$ PREDICTPERFORMANCE(κ, π_t, Q_{val})
      $\sigma_{t+1} \leftarrow$ EVALUATEPERFORMANCE($p, D[s][Q_{val}]$)

Pour chaque modèle testé, nous avons implémenté les routines suivantes :

- **TrainingStep**(M, $D[I_{train}]$) : calibrer le modèle sur les motifs de réponse des apprenants $D[I_{train}]$ et renvoyer les caractéristiques α des apprenants et κ des questions ;
- **PriorInitialization**(α) : initialiser les caractéristiques π_0 d'un nouvel apprenant au début de son test, éventuellement en fonction des caractéristiques des apprenants de l'ensemble d'entraînement ;

- **NextItem**($\{(q_k, r_k)\}_k, \kappa, \pi_t$) : choisir la question à poser telle que la probabilité que l'apprenant y réponde correctement est la plus proche de 0,5 (H.-H. Chang, 2014), en fonction des réponses précédentes de l'apprenant et de l'estimation en cours de son niveau ;
- **EstimateParameters**($\{(q_k, r_k)\}_k, \kappa$) : mettre à jour les caractéristiques de l'apprenant en fonction de ses réponses aux questions posées ;
- **PredictPerformance**(κ, π_t, Q_{val}) : calculer pour chacune des questions du test la probabilité que l'apprenant en cours de test y réponde correctement et renvoyer le vecteur de probabilités obtenu ;
- **EvaluatePerformance**($p, D[s][Q_{val}]$) : comparer la performance prédite à la vraie performance de l'apprenant sur l'ensemble de questions de validation D[s][Q_{val}], de façon à évaluer le modèle. La fonction d'erreur peut être la *log loss* ou le nombre de prédictions incorrectes.

De façon à visualiser les questions posées par un modèle de tests adaptatifs, on peut construire l'arbre binaire de décision correspondant. La racine est la première question posée, puis chaque réponse fausse renvoie vers le nœud gauche, chaque réponse vraie renvoie vers le nœud droit (Ueno et Songmuang, 2010 ; Yan, A. A. v. Davier, et Lewis, 2014). En chaque nœud on peut calculer le taux d'erreur en cours du modèle sur l'ensemble des questions de validation, et le meilleur modèle est celui dont le taux d'erreur moyen est minimal.

Pour calculer le taux d'erreur, nous avons choisi la *log loss*, courante pour les problèmes de classification binaire :

$$e(p, a) = \frac{1}{|Q_{val}|} \sum_{k \in Q_{val}} a_k \log p_k + (1 - a_k) \log(1 - p_k) \tag{3.5}$$

où p est la performance prédite sur les $|Q_{val}|$ questions et a est le vrai motif de réponse de l'apprenant en cours. Notez que si $p = a$, on a bien $e(p, a) = 0$. À titre d'exemple, si pour un des apprenants, après 4 questions, la performance prédite sur l'ensemble de questions de validation est $[0{,}617; 0{,}123; 0{,}418; 0{,}127; 0{,}120]$ tandis que son véritable motif de réponse est $[1, 0, 1, 0, 0]$ (c'est-à-dire, [*correct*, *incorrect*, *correct*, *incorrect*, *incorrect*]), la *log loss* obtenue est 0,350.

Ainsi, EvaluatePerformance calcule la *log loss* et le nombre de prédictions incorrectes entre la performance prédite p et a qui vaut $D[s][Q_{val}]$ pour l'apprenant s.

Lors de chaque expérience (i, j), on enregistre pour chaque apprenant t valeurs d'erreurs où t est le nombre de questions posées, soit $|Q \setminus Q_{val}|$. Ainsi, on peut déterminer le taux d'erreur moyen que chaque modèle a obtenu après avoir posé un certain nombre de questions. Ces valeurs sont stockées dans une matrice de taille $U \times V$ dont chaque case correspond à l'expérience (i, j) correspondant à un ensemble d'apprenants d'entraînement $I_{test} = I_i$ et un ensemble de questions de

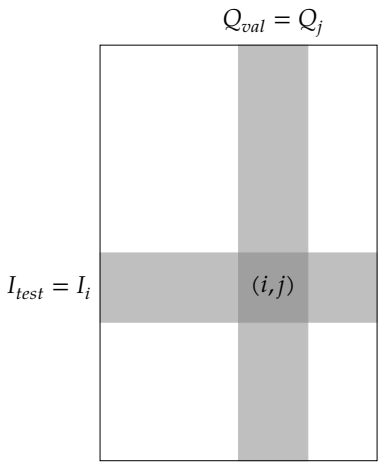


Figure 3.3 – Validation bicroisée selon 6 paquets d'apprenants et 4 paquets de questions.

validation $Q_{val} = Q_j$ (voir figure 3.3). En calculant le taux d'erreur moyen selon chaque colonne, on peut visualiser comment les modèles se comportent pour chaque ensemble de question de validation. On calcule la moyenne de toutes les cases pour tracer les courbes correspondant à chaque modèle.

3.4.5 Jeux de données

Pour nos expériences, nous avons utilisé les jeux de données réelles suivants.

SAT

Le SAT est un test standardisé aux États-Unis. Il est multidisciplinaire, car les questions portent sur 4 catégories : mathématiques, biologie, histoire et français. Dans ce jeu de données, 296 apprenants ont répondu à 40 questions. Ce jeu a été étudié par Winters et al. (2005) et Desmarais et al. (2011) pour déterminer une q-matrice automatiquement via une factorisation de matrices positives.

ECPE

Il s'agit d'une matrice 2922×28 représentant les résultats de 2922 apprenants sur 28 questions d'anglais de l'examen *Examination for the Certificate of Proficiency in English* (ECPE). Ce test standardisé cherche à mesurer trois attributs, c'est pourquoi la q-matrice correspondante a 3 CC : règles morphosyntaxiques, règles cohésives, règles lexicales.

Fraction

Ce jeu de données regroupe les résultats de 536 collégiens sur 20 questions de soustraction de fractions. Les items et la q-matrice correspondante sont décrits dans DeCarlo (2010). La q-matrice est également décrite à la figure 2.1 page 34.

TIMSS

Le *Trends in International Mathematics and Science Study* (TIMSS) effectue un test standardisé de mathématiques. Les données sont librement disponibles sur leur site pour les chercheurs. En l'occurrence, ce jeu de données provient de l'édition 2003 du TIMSS. C'est une matrice binaire de taille 757×23 qui regroupe les résultats de 757 apprenants de 4e sur 23 questions de mathématiques. La q-matrice a été définie par des experts du TIMSS et comporte 13 des 15 composantes de connaissances décrites dans Su et al. (2013).

Castor

Le Castor est un concours d'informatique où les candidats, collégiens ou lycéens, doivent résoudre des problèmes d'algorithmique déguisés au moyen d'interfaces. Le jeu de données provient de l'édition 2013, où 58 939 élèves de 6e et 5e ont dû résoudre 17 problèmes. La matrice est encore dichotomique, c'est-à-dire que son entrée (i, j) vaut 1 si l'apprenant i a eu le score parfait sur la question j, 0 sinon.

3.4.6 Spécification des modèles

Le code est en Python, un langage lisible pour concevoir des scripts en peu de lignes de code, et fait appel à des fonctions en R au moyen du package `RPy2`. Des détails concernant l'implémentation sont donnés dans l'annexe A.

Modèle de Rasch

Chaque apprenant a une unique caractéristique correspondant à son niveau, tandis que chaque question a une unique caractéristique correspondant à sa difficulté.

TrainingStep La phase d'apprentissage consiste à déterminer l'estimateur du maximum de vraisemblance pour les paramètres des apprenants et des questions. Comme le modèle est simple, l'expression de la dérivée de la vraisemblance est simple et on peut déterminer les paramètres qui l'annulent par la méthode de Newton.

PriorInitialization Lorsqu'un nouvel apprenant passe le test, on initialise son niveau à 0.

NextItem Comme pour chaque modèle, la question choisie est celle pour laquelle la probabilité que l'apprenant y réponde correctement est la plus proche de 0,5.

UpdateParameters Après chaque réponse de l'apprenant, l'estimation ses caractéristiques est faite par le maximum de vraisemblance. Si trop peu de réponses ont été fournies, l'estimation est faite selon une approche bayésienne (R. Philip Chalmers, 2012).

PredictPerformance Pour rappel, la formule est donnée par l'expression :

$$Pr(D_{ij} = 1) = \Phi(\theta_i - d_j). \tag{3.6}$$

Modèle DINA

Chaque apprenant a une caractéristique qui est son état latent, défini à la section 2.3.2 page 35, et chaque question a pour caractéristiques la liste des CC qu'elle requiert dans la q-matrice, ainsi qu'un paramètre d'inattention et un paramètre de chance.

Pendant la phase de calibrage, nous calculons, à partir d'une q-matrice et d'une population, les paramètres d'inattention et de chance expliquant le mieux les données.

TrainingStep Si l'on dispose d'une q-matrice, la phase d'apprentissage consiste à déterminer les états latents des apprenants d'entraînement, ainsi que les paramètres d'inattention et de chance minimisant le taux d'erreur du modèle, pour chaque question.

Pour déterminer les états latents des apprenants, on simule le fait de leur poser toutes les questions en utilisant le modèle DINA. Si on ne dispose pas de q-matrice, nous l'extrayons automatiquement en utilisant la méthode décrite à la section 3.4.2 page 51. Nous estimons ensuite les paramètres d'inattention et de chance de chaque question via optimisation convexe.

PriorInitialization Lorsqu'un nouvel apprenant passe le test, on suppose qu'il a autant de chance d'être dans chacun des 2^K états latents possibles. On va maintenir cette distribution de probabilité sur les 2^K états latents possibles, initialisée à cette distribution uniforme : pour tout $c \in \{0,1\}^K$, $\pi(c) = 1/2^K$.

NextItem Comme pour chaque modèle, la question choisie est celle pour laquelle la probabilité que l'apprenant y réponde correctement est la plus proche de 0,5.

UpdateParameters Après chaque réponse de l'apprenant, une mise à jour de la distribution de probabilité est faite, de façon bayésienne. Voir la section 2.3.2 page 36 pour les formules utilisées.

PredictPerformance Pour rappel, la formule est donnée par l'expression :

$$Pr(D_{ij} = 1) = \begin{cases} 1 - s_j & \text{si l'apprenant } i \text{ maîtrise toutes les CC requises} \\ & \text{pour répondre correctement à la question } j \\ g_j & \text{sinon.} \end{cases} \tag{3.7}$$

3.5 Résultats

3.5.1 Évaluation qualitative

Rasch Le modèle de Rasch est unidimensionnel, il n'a pas besoin de q-matrice pour fonctionner et il fait un retour à l'apprenant sous la forme d'une valeur de niveau. Cela permet à l'apprenant de se situer au sein des autres apprenants mais pas de comprendre les points du cours qu'il doit approfondir. Les paramètres estimés peuvent être interprétés comme des valeurs de niveau pour les apprenants et de difficulté pour les questions, mais peuvent correspondre à des erreurs d'énoncé. Enfin, afin de calibrer les paramètres de difficulté des questions et de niveau des apprenants, le modèle de Rasch a besoin de données d'entraînement.

DINA Le modèle DINA est multidimensionnel, requiert une q-matrice, fait un retour à l'apprenant sous la forme d'une probabilité de maîtriser chacune des composantes de connaissances. Grâce à la q-matrice, on peut interpréter les différentes dimensions. Si la q-matrice est mal définie, des valeurs aberrantes apparaîtront pour les paramètres d'inattention et de chance. Le modèle DINA peut fonctionner sans historique, en supposant une distribution uniforme a priori sur les états latents possibles.

Dans ce qui suit, m désigne le nombre d'apprenants, n le nombre de questions et K le nombre de CC.

Complexité du modèle de Rasch Il est difficile de calculer la complexité de la phase de calibrage, car il s'agit d'une méthode numérique. Toutefois, de tous les modèles testés, il a été le plus rapide à estimer. Le choix de la question suivante coûte $O(n)$ car connaissant une estimation du niveau de l'apprenant, calculer la probabilité de chaque question s'effectue en temps constant. Si l'on note A le temps passé à déterminer les caractéristiques de l'apprenant, la complexité de la phase de test est $O(mn(n + A))$.

Complexité du modèle DINA Le choix de la question suivante coûte $O(K2^K n)$ opérations. L'estimation des caractéristiques de l'apprenant s'effectue en $O(K2^K)$. Les phases d'entraînement et de test de DINA ont une complexité $O(mn^2K2^K)$. C'est pourquoi K est généralement choisi inférieur à 10 (Su et al., 2013).

3.5.2 Évaluation quantitative

Les résultats sont donnés dans les figures 3.4 à 3.7. Les valeurs du taux d'erreur (*log loss*) sont répertoriées dans les tables 3.2 à 3.6. Entre parenthèses, la précision des prédictions sur l'ensemble de validation.

SAT

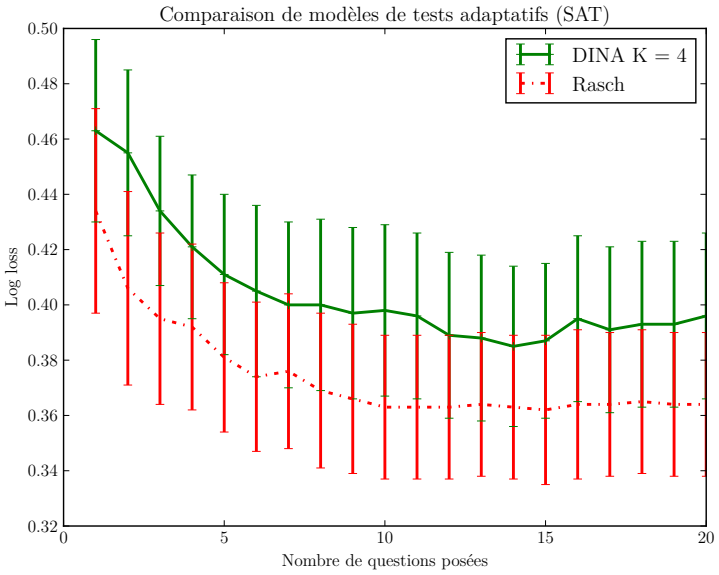


Figure 3.4 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données SAT.

	Après 10 questions	Après 15 questions
DINA	0,398 $\pm$ 0,031 (82 %)	0,387 $\pm$ 0,028 (82 %)
Rasch	0,363 $\pm$ 0,026 (83 %)	0,362 $\pm$ 0,027 (84 %)

Table 3.2 – Valeurs de *log loss* obtenues pour le jeu de données SAT.

Dans la figure 3.4, le modèle de Rasch réalise un diagnostic légèrement meilleur

que le modèle DINA avec une q-matrice calculée automatiquement. Comme Desmarais et al. (2011), notre extraction de q-matrice a réussi à identifier que les questions 1 à 10 partageaient une CC (mathématiques) ainsi que les questions 31 à 40 (français) mais a eu plus de mal à identifier les questions de biologie et d'histoire.

Le modèle de Rasch converge en 10 questions mais plafonne à 82 % de précision tandis que le modèle DINA continue à augmenter légèrement sa précision (voir table 3.2).

Nous faisons l'hypothèse que comme ce jeu de données est multidisciplinaire et que la plupart des questions portent sur une unique CC, poser une question de mathématiques ne va pas apporter beaucoup d'information sur les CC en français ; c'est pourquoi le modèle de Rasch peut en quelques questions avoir une bonne information sur l'ensemble du test, tandis que le modèle DINA récolte de l'information seulement sur la maîtrise ou non maîtrise de la CC sur laquelle porte chaque question posée.

ECPE

Dans la figure 3.5, les modèles se valent. DINA est en moyenne très légèrement meilleur.

Après 10 questions, les deux modèles plafonnent à 74 % de précision (voir table 3.3).

Nous faisons l'hypothèse que comme le jeu de données a beaucoup de motifs de réponse différents, les questions sont indépendantes donc les modèles ont du mal à prédire le comportement des apprenants sur les questions restantes du test.

Fraction

Dans la figure 3.6, le meilleur modèle en moyenne est le modèle DINA dont la q-matrice a été spécifiée par un expert. Après avoir posé 4 questions de façon adaptative, le modèle DINA est capable de prédire en moyenne 86 % de l'ensemble de question de validation correctement, soit en moyenne plus de 8 questions sur 10 (voir table 3.4).

Nous faisons l'hypothèse que comme il s'agit d'un jeu de données de soustraction de fractions, l'information que l'apprenant maîtrise ou non le fait de mettre au même dénominateur est suffisant pour prédire son comportement sur des questions qui ne lui ont pas été posées. Il n'y a pas besoin de considérer des valeurs de niveau. De plus, comme les questions font souvent appel à plusieurs CC, peu de questions sont nécessaires pour converger.

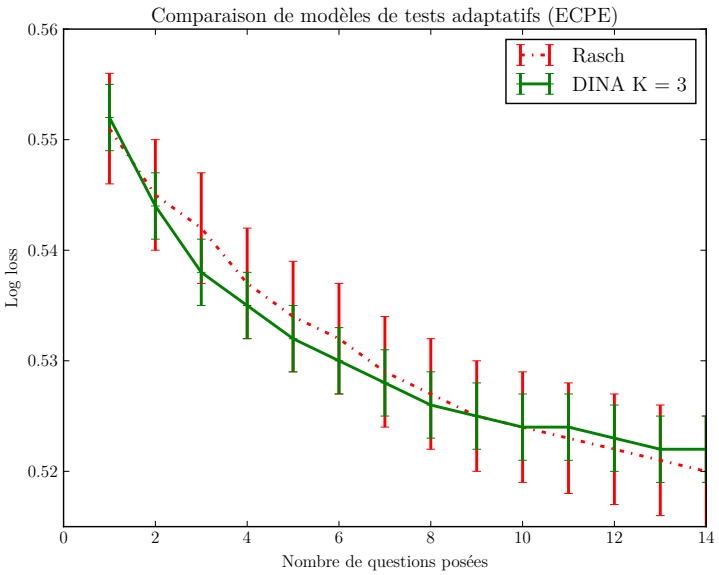


Figure 3.5 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données ECPE.

	Après 5 questions	Après 10 questions
Rasch	0,534 $\pm$ 0,005 (73 %)	0,524 $\pm$ 0,005 (74 %)
DINA	0,532 $\pm$ 0,003 (73 %)	0,524 $\pm$ 0,003 (74 %)

Table 3.3 – Valeurs de *log loss* obtenues pour le jeu de données ECPE.

TIMSS

Dans la figure 3.7, Rasch est meilleur que DINA.

Après 8 questions, les prédictions plafonnent à 71 % et les intervalles de confiance de la *log loss* des modèles sont, comme dans le jeu de données ECPE, très serrés (voir table 3.5).

Nous faisons l'hypothèse que ces jeux de données se ressemblent : il y a beaucoup de motifs de réponse possibles, donc les questions semblent indépendantes.

Castor

Dans la figure 3.8, Rasch est bien meilleur que DINA avec une q-matrice de taille $K = 3$ calculée automatiquement.

Nous faisons l'hypothèse que la q-matrice a été mal spécifiée, ce qui a conduit à des erreurs de diagnostic.

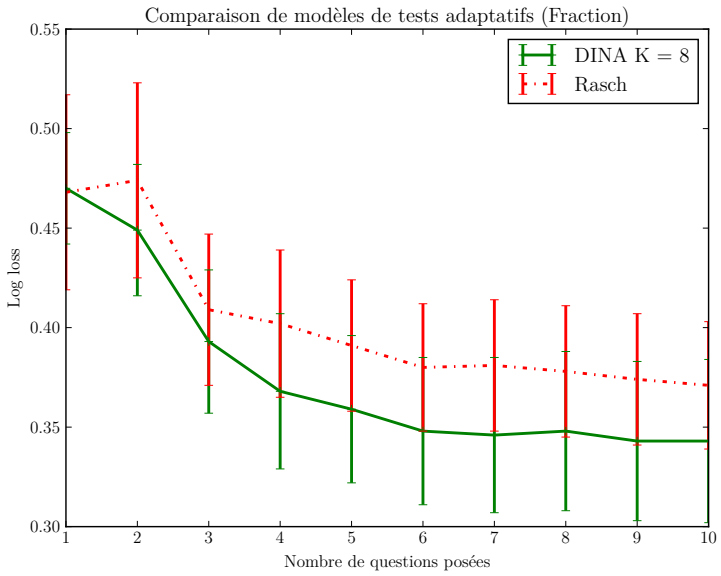


Figure 3.6 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données Fraction.

	Après 4 questions	Après 7 questions
Rasch	0,402 $\pm$ 0,037 (84 %)	0,381 $\pm$ 0,033 (85 %)
DINA	0,368 $\pm$ 0,039 (86 %)	0,346 $\pm$ 0,039 (86 %)

Table 3.4 – Valeurs de *log loss* obtenues pour le jeu de données Fraction.

3.5.3 Discussion

Selon le jeu de données, le meilleur modèle n'est pas le même. Par exemple, pour des tâches procédurales telles que le test Fraction, le modèle DINA a une haute précision en prédiction de performance. Pour tous les jeux de données, le modèle de Rasch a de bonnes performances tout en étant très simple. L'avantage du modèle DINA est qu'il est formatif : la q-matrice spécifiée par un expert permet de faire un retour à l'apprenant à l'issue de test pour lui indiquer ce qu'il semble ne pas avoir maîtrisé.

Il est utile de remarquer que pour le modèle DINA avec $K = 1$, l'apprenant peut être modélisé par une probabilité d'avoir l'unique CC ou non. Si la question ne requiert aucune CC, il a une probabilité constante $1 - s_i$ d'y répondre. Sinon, sa probabilité est $(1 - p)g_i + p(1 - s_i) = g_i + p(1 - s_i - g_i)$ soit une valeur qui croît entre g_i et $1 - s_i$ de façon linéaire avec p. Cela donne une interprétation géométrique du modèle de Rasch comparé au modèle DINA, et indique peut-être

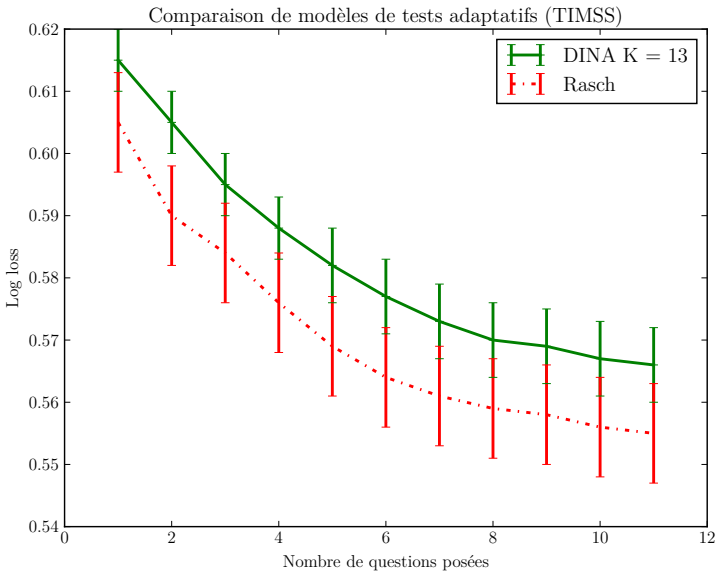


Figure 3.7 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données TIMSS.

	After 4 questions	After 8 questions
DINA	0,588 $\pm$ 0,005 (68 %)	0,570 $\pm$ 0,006 (70 %)
Rasch	0,576 $\pm$ 0,008 (70 %)	0,559 $\pm$ 0,008 (71 %)

Table 3.5 – Valeurs de *log loss* obtenues pour le jeu de données TIMSS.

une limitation du modèle : le meilleur élève possible a une probabilité de répondre à chaque question i plafonnée par $1 - s_i$, tandis que Rasch n'est pas limité. Ainsi, le modèle de DINA est plus prudent que le modèle Rasch, ce qui peut expliquer pourquoi la *log loss* du modèle DINA est souvent plus faible que celle de Rasch.

Le calcul automatique d'une q-matrice est un problème difficile : s'il y a $|Q|$ questions et K composantes de connaissances, il y a $|Q|K$ bits donc $2^{|Q|K}$ q-matrices possibles. Pour chacune, le calcul des paramètres d'inattention et de chance est un problème d'optimisation convexe.

Notre calcul a conduit à des résultats peu satisfaisants, sachant que la méthode de calibration du modèle de Rasch est efficace tandis que si l'on souhaite calculer une distribution a priori sur les apprenants du modèle DINA, la complexité est grande dans la mesure où il faut simuler l'administration de chaque question à chaque apprenant de l'ensemble d'entraînement, or chaque fois qu'un apprenant répond à une question, il faut mettre à jour la distribution de probabilité sur les états possibles ce qui a une complexité $O(2^K K)$, ce qui donne une complexité

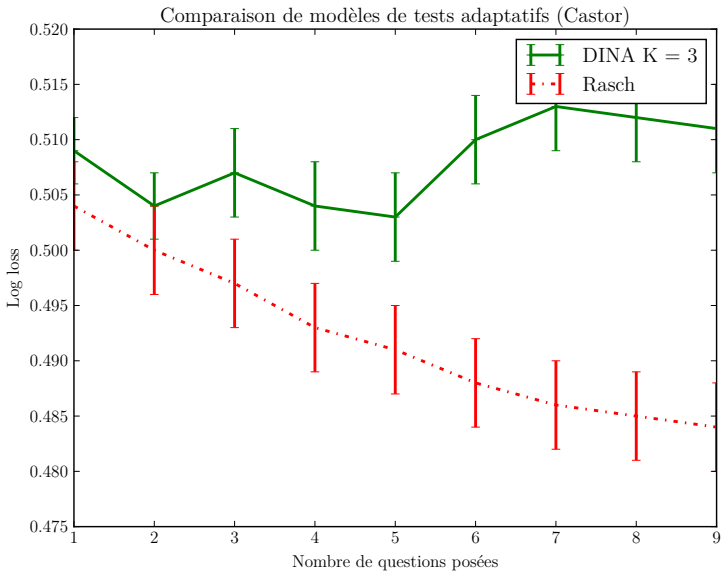


Figure 3.8 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données Castor.

	Après 4 questions	Après 8 questions
DINA	0,504 $\pm$ 0,004 (78 %)	0,512 $\pm$ 0,004 (77 %)
Rasch	0,493 $\pm$ 0,004 (78 %)	0,485 $\pm$ 0,004 (79 %)

Table 3.6 – Valeurs de *log loss* obtenues pour le jeu de données Castor.

totale de $NM2^K K$ où N est le nombre d'apprenants de l'ensemble d'entraînement et M le nombre de questions.

Le modèle DINA en lui-même mélange des paramètres discrets (les bits de la q-matrice) et des paramètres continus (les paramètres d'inattention et de chance), ce qui fait qu'il ne s'agit ni d'un problème d'optimisation linéaire en nombres entiers, ni d'un problème d'optimisation convexe. La méthode naïve d'escalade de colline fait tomber dans des minima locaux qui ne donnent pas un modèle qui correspond aux données de façon satisfaisante. Ainsi on préférera le modèle de Rasch ou ses analogues multidimensionnels de la théorie de la réponse à l'item.

3.6 Applications aux MOOC

Forts de la description des modèles de tests adaptatifs au chapitre précédent, et de leur comparaison qualitative dans ce chapitre, nous proposons à présent

une méthodologie de choix de modèles en fonction du type de test que l'on souhaite administrer dans un MOOC. Dans une seconde section, nous illustrons la réduction du nombre de questions obtenue par un test adaptatif simulé sur un MOOC de Coursera.

3.6.1 Méthodologie de choix de modèles

Test adaptatif au début d'un MOOC

Au début d'un cours, il faut identifier les connaissances de l'apprenant avec le moins de questions possible. C'est un problème de démarrage à froid de l'apprenant, où il faut identifier si celui-ci a bien les prérequis du cours. Si un graphe de prérequis entre composantes de connaissances (CC) est disponible, nous suggérons d'utiliser le modèle de Falmagne et al. (2006), décrit à la section 2.3.2 page 37, ou son analogue composé de paramètres d'inattention et de chance, le modèle de hiérarchie sur les attributs. Si une q-matrice est disponible, nous suggérons d'utiliser le modèle DINA, décrit à la section 2.3.2 page 35. Sinon, le modèle de Rasch permet au moins de classer les apprenants. Si aucun historique sur le test n'est disponible, par exemple parce qu'il s'agit de la première édition du cours, les seuls modèles envisageables parmi ceux présentés sont celui de Falmagne et al. (2006) qui nécessite un graphe de prérequis, ou le modèle DINA qui nécessite une q-matrice.

Une autre application consiste à faire un test adaptatif à partir du graphe de prérequis sur les CC développées dans le cours. Ainsi, il sera possible d'indiquer à l'apprenant s'il peut se passer de suivre certaines parties du cours.

Test adaptatif au milieu d'un MOOC

Les apprenants aiment pouvoir savoir sur quoi ils vont être testés, sous la forme d'une autoévaluation qui « ne compte pas ». Cet entraînement de passage de tests a un effet bénéfique sur leur apprentissage (Dunlosky et al., 2013). Il y a toutefois plusieurs scénarios à considérer. Si les apprenants ont accès au cours alors qu'ils passent ce test à faible enjeu, le modèle de test adaptatif doit prendre en compte le fait que le niveau de l'apprenant puisse changer alors qu'il passe le test, par exemple parce qu'il consulte son cours avant de répondre à chaque question. Dans ce cas, les modèles qui tentent de faire progresser le plus les élèves, tel que celui proposé par Clement et al. (2015) décrit à la section 2.3.3 page 41, sont appropriés. Ils requièrent soit un graphe de prérequis, soit une q-matrice. Si les apprenants n'ont pas accès au cours pendant le test, le modèle DINA convient, à condition qu'une q-matrice soit spécifiée.

Test adaptatif à la fin d'un MOOC

Un test d'évaluation à la fin d'un cours peut se baser sur les modèles de tests adaptatifs usuels, de façon à mesurer les apprenants efficacement et leur attribuer une note. Pour ce dernier examen, nous supposons que le retour peut se limiter à un score, ainsi le modèle de Rasch est le plus simple à mettre en place.

3.6.2 Simulation d'un test adaptatif

Nous avons simulé un modèle de test adaptatif sur de véritables données de MOOC issues d'un cours d'analyse fonctionnelle donné par John Cagnol, professeur à CentraleSupélec, sur la plateforme Coursera en 2014.

Le cours a accueilli 25354 inscrits et était composé de 8 leçons, à la fin de chacune un quiz, plus un examen final.

Extraction de données

Nous avons tenté d'extraire un maximum de données de test, à l'exception des QCM qui se trouvaient au sein de chaque vidéo car elles avaient trop peu de réponses possibles. Ainsi, pour chaque test il nous fallait récupérer les succès et échecs des apprenants sur la plateforme : un ensemble de motifs de réponse binaires (vrai ou faux), sous la forme $(r_1, \dots, r_n)$ où n est le nombre de questions posées dans un test.

Cependant, sur un MOOC, les apprenants ne participent pas à tous les quiz. Ainsi, il faut se demander comment considérer les entrées manquantes. De plus, parfois les apprenants tentent plusieurs fois de répondre à un quiz. Ainsi, il faut choisir quel essai considérer, le premier ou celui de score maximum (Bergner, Colvin, et Pritchard, 2015). Dans notre cas, nous avons considéré à chaque fois le premier essai, le dernier ayant de grandes chances d'être un succès.

À partir de la base de données SQL de ce MOOC, nous avons ainsi pu extraire les tests suivants :

- Quiz 1 topologie : 5770 essais de 3672 étudiants sur 6 questions.
- Quiz 2 espaces métriques et normés : 3296×7
- Quiz 3 espaces de Banach et fonctions linéaires continues : 2467×7 (dont une réponse ouverte)
- Quiz 4 espaces de Hilbert : 1807×6
- Quiz 5 lemme de Lax-Milgram : 1624×7
- Quiz 6 espaces L_p : 1504×6
- Quiz 7 distributions et espaces de Sobolev : 1358×9
- Quiz 8 application à la simulation d'une membrane : 1268×7
- Exam : 599×10

Représentation du domaine

Nous souhaitions nous placer dans le cas où un nouvel apprenant apparaît sur un MOOC et souhaite tester ses connaissances afin de déterminer s'il peut se passer de certaines parties du cours. Pour ce faire, il nous a fallu construire :

- une représentation des connaissances mises en œuvre dans le cours, sous la forme d'un graphe de prérequis $G = (V, E)$ où V est l'ensemble des composantes de connaissances et une arête $u \to v$ désigne la relation de prérequis : « u doit être maîtrisé pour maîtriser v » ;
- un lien entre chaque question et les composantes de connaissances (CC) qu'elle requiert. Pour simplifier, nous avons considéré que chaque question requérait une CC principale, et le graphe de prérequis permet d'indiquer quels sont les CC qu'il faut avoir maîtrisé pour maîtriser cette CC principale.

Cela nous a permis de construire un modèle de hiérarchie sur les attributs, défini à la section 2.3.2 page 37 et similaire au modèle de théorie des espaces de connaissances. Ainsi, à partir de ce modèle de test adaptatif, pour chaque apprenant qui passe le test, les informations dont nous disposons sur lui sont :

- le résultat (vrai ou faux) à chaque question que le système lui a posée ;
- une distribution de probabilité π sur les états latents possibles dans lesquels peut se trouver l'apprenant, c'est-à-dire : quels CC il semble maîtriser et quels CC il semble ne pas maîtriser (voir section 2.3.2 page 35).

Spécification des paramètres

Les réponses des candidats à un QCM ne reflètent pas nécessairement leur maîtrise d'un sujet, la réponse peut être facile à deviner ou inversement, un apprenant peut faire une erreur d'inattention. Ainsi, nous avons considéré un paramètre ε correspondant à la probabilité de deviner la bonne réponse tandis que toutes les CC requises ne sont pas maîtrisées, ainsi qu'à la probabilité de se tromper bien qu'on ait toutes les CC requises. C'est-à-dire que tous les paramètres d'inattention et de chance sont fixés à une valeur unique ε.

Déroulement du test adaptatif

Au fur et à mesure que l'apprenant répond à des questions, le système peut mettre à jour l'estimation qu'il se fait de son état latent. Chaque réponse à une question posée à l'apprenant permet de mettre à jour une information a priori sur ses états latents possibles, de façon bayésienne.

Certaines questions sont plus informatives que d'autres. Par exemple, poser une question reliée à une composante qui n'a pas d'arc sortant est peu avantageux car la probabilité que l'étudiant la maîtrise est faible, or une réponse fausse n'apportera pas beaucoup d'information. Il est possible de quantifier plus

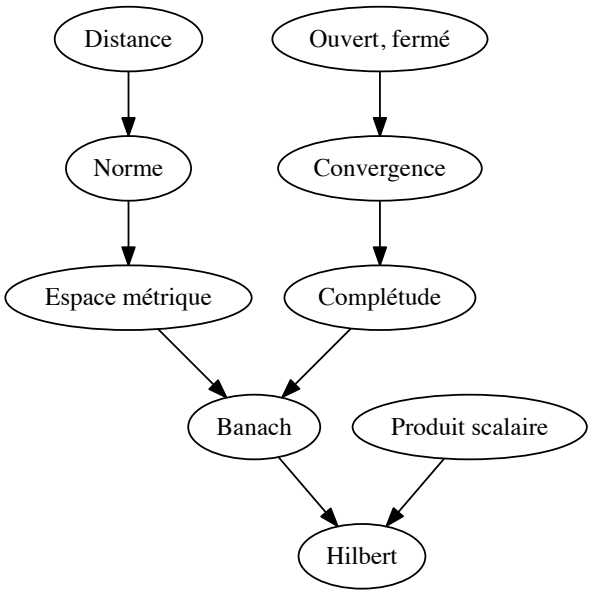


Figure 3.9 – Un exemple de graphe de prérequis.

formellement l'information que chaque question peut apporter. En théorie de l'information, une manière de représenter l'incertitude est l'entropie. Pour une variable X pouvant prendre des valeurs avec des probabilités $(p_i)_{1\leq i\leq n}$:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i. \tag{3.8}$$

Par exemple, une pièce parfaitement équilibrée peut prendre la valeur Pile avec probabilité 50 % et Face avec la même probabilité, ainsi son entropie est de 1, tandis qu'une pièce pouvant prendre la valeur Pile avec probabilité 90 % a une entropie de 0,470. La pièce équilibrée est donc celle d'incertitude maximale. Dans notre cas, en choisissant la question faisant le plus abaisser l'entropie, on vise à converger rapidement vers l'état latent de l'apprenant.

Exemple de déroulement

Afin d'illustrer cette approche, voici un exemple de déroulement de test adaptatif. Si l'on considère le graphe de prérequis de la figure 3.9 et que l'apprenant maîtrise les notions de produit scalaire, d'espace métrique et celle de complétude mais pas d'espace de Banach, un test minimisant l'entropie à chaque étape et

s'arrêtant lorsqu'un état a atteint une probabilité de 95 % se déroulera comme suit :

— **Q1.** Est-ce que l'apprenant maîtrise « Produit scalaire » ?
— Oui.
— **Q2.** Est-ce que l'apprenant maîtrise « Norme » ?
— Oui.
— **Q3.** Est-ce que l'apprenant maîtrise « Complétude » ?
— Oui.
— **Q4.** Est-ce que l'apprenant maîtrise « Banach » ?
— Non.
— **Q5.** Est-ce que l'apprenant maîtrise « Espace métrique » ?
— Oui.
— Alors, l'apprenant maîtrise Produit scalaire, Distance, Norme, Ouvert/fermé, Complétude, Produit scalaire, mais pas Banach, Hilbert.

Ainsi, 5 questions ont été posées au lieu de 9 afin de déterminer l'état mental de l'apprenant et lui faire un retour.

Protocole expérimental

Afin de simplifier l'étude tout en conservant un grand nombre de réponses, nous avons considéré le graphe de prérequis à la figure 3.9 et avons choisi un sous-ensemble de 9 questions tirées des quiz 1 à 4 du MOOC. Cela nous a permis de construire une matrice de motifs de réponse binaires de 3713 étudiants sur ces 9 questions portant sur les 9 composantes de connaissances Banach, Complétude, Convergence, Distance, Espace métrique, Hilbert, Norme, Ouvert et fermé, Produit scalaire.

Chaque question a été choisie pour couvrir une composante de connaissances (et toutes celles qui sont nécessaires à sa maîtrise), ainsi chaque question correspond à un nœud du graphe de prérequis. Le nombre de motifs de réponse de chaque type est donné dans la table 3.7 et sa non-uniformité laisse entendre qu'il existe des corrélations entre les réponses aux questions (sinon, le nombre d'occurrences serait le même d'un motif de réponse à un autre).

Validation

Deux métriques nous permettent de valider notre modèle de test adaptatif. La première est le nombre moyen de questions avant arrêt du test (appelé *temps de convergence moyen*), c'est-à-dire avant que le critère de terminaison soit validé. La deuxième est le nombre de prédictions incorrectes (appelé *erreur de prédiction*), car il faut vérifier que le test ne converge pas vers un diagnostic qui ne correspond pas à la réalité.

Motif de réponse	Nombre d'occurrences
000000010	1129
000000000	460
010110110	271
110111111	263
010110010	122
111111111	116
110111011	77
110110110	70
110110010	42
010010010	41

Table 3.7 – Les 10 motifs de réponse les plus fréquents pour le jeu de données extrait du MOOC d'analyse fonctionnelle.

Valeur de ε	Temps de convergence	Erreur de prédiction
0	$5{,}009 \pm 0{,}003$	$1{,}075 \pm 0{,}040$
0,01	$5{,}430 \pm 0{,}016$	$1{,}086 \pm 0{,}041$
0,02	$6{,}879 \pm 0{,}019$	$1{,}086 \pm 0{,}041$
0,03	$7{,}671 \pm 0{,}027$	$0{,}956 \pm 0{,}037$
0,04	$7{,}807 \pm 0{,}023$	$1{,}086 \pm 0{,}041$
0,05	$8{,}671 \pm 0{,}027$	$0{,}956 \pm 0{,}037$

Table 3.8 – Métriques principales pour la validation du modèle de test adaptatif sur les données du MOOC d'analyse fonctionnelle

Pour chaque apprenant de notre jeu de données, nous simulons un test adaptatif à l'aide du modèle de hiérarchie sur les attributs, qui choisit la question qui réduit le plus son incertitude (entropie). Dès qu'un état mental dépasse la probabilité 95 %, le test s'arrête. Cela permet de déterminer le nombre de questions moyen avant arrêt, ainsi que le nombre de prédictions incorrectes. Les résultats sont donnés dans la table 3.8.

Discussion

La valeur de robustesse $\varepsilon = 0$ correspond à un test où l'on suppose que si l'apprenant répond correctement à une question, alors il maîtrise la composante de connaissances correspondante. Un tel test converge en 5 questions en moyenne, et prédit correctement 8 des 9 réponses du motif de réponse. Ainsi, en ne posant

que 55 % des questions du test en fonction des réponses précédentes, il obtient un succès de 89 %.

Une plus grande valeur de robustesse ϵ n'améliore pas tellement les prédictions, ce qui peut être expliqué par le faible nombre d'états possibles (35). Le graphe des prérequis à la figure 3.9 est rudimentaire, ce qui ne lui permet pas d'exprimer les connaissances d'un tel domaine des mathématiques. Toutefois, cet exemple minimal de test adaptatif réduit le nombre de questions posées de façon satisfaisante, tout en garantissant la fiabilité du test. Ce modèle est à préférer au modèle de Rasch car il permet d'indiquer à l'apprenant les composantes de connaissances qu'il semble maîtriser. Cela requiert toutefois un graphe de prérequis qui peut être coûteux à construire selon la discipline dans laquelle l'apprenant est évalué.

3.7 Conclusion

Dans ce chapitre, nous avons détaillé les différents composants modulables dans la conception d'un système de test adaptatif, nous permettant de comparer différents modèles de tests adaptatifs sur un même jeu de données. La méthode de validation que nous proposons, la validation bicroisée, est souvent utilisée en apprentissage automatique, notamment pour valider des techniques de filtrage collaboratif.

Nous avons implémenté ce système et nous l'avons appliqué à la comparaison d'un modèle sommatif, le modèle de Rasch, et d'un modèle formatif, le modèle DINA, sur des données réelles. Nous avons mis en évidence que selon le type de test, le meilleur modèle n'est pas le même. Comme Rupp et al. (2012), nous ne cherchons pas à déterminer un meilleur modèle pour tous les usages, nous cherchons à identifier quel modèle convient le mieux à quel usage et nous avons proposé une méthodologie pour comparer leur capacité à efficacement réduire la taille des tests.

Dans la littérature, nous avons observé que la plupart des modèles qui se basent sur des q-matrices sont évalués sur des données simulées (Desmarais et al., 2011 ; Cheng, 2009). Ici, nous ne considérons que des données réelles d'apprenants, et notre système de comparaison peut être testé sur n'importe quel jeu de données de test comportant des succès ou échecs d'apprenants sur des questions. Il peut également être généralisé à des tests à étapes multiples comme nous le verrons à la section 5.1 page 93. Le fait de considérer seulement les réussites ou les échecs d'apprenants face à des questions ou tâches permet d'appliquer un modèle de test adaptatif à des données issues d'interfaces plus complexes telles que des jeux sérieux.

Nous avons terminé ce chapitre en proposant une méthodologie de choix d'un

modèle de test adaptatif en fonction du type de test qui peut apparaître dans un MOOC. Nous l'avons illustrée par une simulation d'un test adaptatif basé sur le modèle de hiérarchie sur les attributs, appliqué à des données réelles d'un MOOC de Coursera. Pour cette simulation, nous avons construit une représentation du domaine couvert par un test d'analyse fonctionnelle, et avons mis en évidence que le nombre de questions du test pouvait être réduit grâce à ce modèle de tests adaptatifs.

Le système de comparaison développé dans ce chapitre va nous être utile pour valider le nouveau modèle de tests adaptatifs que nous proposons, décrit dans le chapitre suivant.

Chapitre 4

GenMA : un modèle hybride de diagnostic de connaissances

4.1 Introduction

Dans le chapitre précédent, nous avons mis en évidence que le modèle de Rasch considère une valeur de difficulté des questions et un niveau de l'apprenant global, tandis que le modèle DINA fait un retour à l'apprenant en termes de maîtrise de plusieurs composantes de connaissances (CC). Notre comparaison quantitative a de plus prouvé que le modèle de Rasch restait compétitif avec le modèle DINA, car il déterminait en peu de questions des valeurs de niveau capables de bien prédire le comportement de l'apprenant sur les questions restantes du test.

Il devient naturel de se demander comment se comporterait un modèle qui prendrait en compte à la fois une notion de difficulté des questions d'un test, et une notion de différentes composantes de connaissances impliquées dans la résolution de ces questions. Ce serait un moyen d'avoir un modèle qui exprimerait le fait qu'une question du test puisse faire davantage appel à une composante de connaissances qu'à une autre. Ainsi à la fin du test, l'apprenant obtiendrait un diagnostic composé d'un degré de maîtrise selon chaque composante de connaissances. Un tel diagnostic serait plus riche que celui du modèle DINA, qui renvoie à l'apprenant la probabilité que celui-ci maîtrise ou non chaque composante de connaissances.

Comme nous l'avons vu lors de la comparaison qualitative menée à la section 3.3 page 47, le modèle de théorie de la réponse à l'item multidimensionnelle (MIRT) est effectivement une généralisation du modèle de Rasch à plusieurs dimensions. Toutefois, ce modèle étant entièrement automatique, les dimensions extraites sont difficilement interprétables. De plus, pour de grandes dimensions, ce modèle est difficile à calibrer.

Dans ce chapitre, nous proposons un nouveau modèle hybride appelé GenMA (pour *General Multidimensional Adaptive*). Il repose sur un modèle de diagnostic cognitif qui à notre connaissance n'a pas été utilisé pour administrer des tests adaptatifs (Yan, A. A. v. Davier, et Lewis, 2014). Nous avons implémenté ce modèle et, en utilisant le système de comparaison décrit au chapitre précédent, nous avons pu mettre en évidence que le modèle GenMA réduit davantage le nombre de questions requises pour converger vers un diagnostic que les autres modèles de tests adaptatifs formatifs. De plus, le diagnostic fourni par GenMA est vraisemblable : il permet de prédire le comportement de l'apprenant sur les questions restantes du test.

À partir d'une q-matrice spécifiée par un expert ou calculée automatiquement, qui indique quelles CC sont requises pour répondre à chaque question, la phase de calibrage de GenMA détermine automatiquement des paramètres de discrimination entre questions et composantes de connaissances. Cette première phase extrait ainsi des caractéristiques des questions qui seront utiles pour réaliser un test adaptatif efficace.

Plutôt qu'une approche totalement automatique et difficilement interprétable comme le modèle MIRT, ou une approche principalement manuelle comme le modèle DINA, notre modèle GenMA est semi-automatique, dans la mesure où la q-matrice spécifiée guide l'estimation des caractéristiques des questions et donc des apprenants.

À chaque étape du test, en fonction des réponses de l'apprenant, un diagnostic est calculé sous la forme d'un vecteur représentant le niveau de l'apprenant, dont chaque dimension correspond à une composante de connaissances. Si la q-matrice a été spécifiée par un expert, les CC sont interprétables, ainsi à la fin du test il est possible de nommer et quantifier les points forts et les lacunes de l'apprenant.

Ce chapitre présente d'abord les différentes techniques d'extraction de caractéristiques et de réduction de dimension qui sont à la base de la phase de calibrage du modèle GenMA. Il présente ensuite notre modèle de tests adaptatifs GenMA, ainsi que les résultats de notre système de comparaison sur le modèle GenMA, le modèle de Rasch et le modèle DINA.

4.2 Extraction de caractéristiques cachées

La théorie de la réponse à l'item suppose que les réponses des apprenants peuvent être expliquées par un faible nombre de facteurs. Ainsi, si l'on parvient à partir de peu de réponses de l'apprenant à identifier ces facteurs, on peut généraliser le comportement de l'apprenant à toutes les questions du test. Par exemple, il est inutile de poser davantage de questions portant sur le même sujet si l'apprenant au cours du test a prouvé à plusieurs reprises qu'il ne maîtrisait pas

ce sujet. Pour réduire la taille d'un test, on cherche donc, entre autres, à éliminer la redondance dans les questions qu'on pose.

Il existe de nombreuses façons de réduire le nombre de dimensions d'un jeu de données afin d'interpréter plus facilement les motifs qui y apparaissent. C'est pourquoi nous avons jugé important de les mentionner afin de comprendre ce qui différencie ces méthodes, lesquelles modélisent la variance des données d'une façon particulière, et lesquelles sont approchées.

Toutes les méthodes que nous décrivons dans cette section consistent à tenter d'extraire automatiquement des caractéristiques des questions à partir de données d'apprenants y ayant répondu de façon correcte ou incorrecte. Ainsi, nous nous concentrons ici sur la phase d'apprentissage des modèles de tests adaptatifs décrite à la section 3.2.2 page 46. Les caractéristiques des questions extraites seront sous la forme de vecteurs à d dimensions [1], où d est inférieur au nombre de questions du test, tels que pour un apprenant fixé, des questions de caractéristiques proches induisent des motifs de réponse de l'apprenant proches. De façon similaire, pour une question fixée, des utilisateurs de caractéristiques proches auront des motifs de réponse proches. Celles-ci nous permettront d'exécuter des tests adaptatifs pour de nouveaux apprenants.

Il est intéressant de tenter d'interpréter les dimensions des caractéristiques des questions a posteriori : ainsi, on pourra par exemple identifier que telle composante évalue la capacité de la question à mesurer la capacité de l'apprenant à savoir additionner, tandis que telle autre évalue la capacité de l'apprenant à savoir multiplier.

Dans cette section, on note D la matrice des réponses succès / échecs des apprenants aux questions d'un test. Cette matrice est de taille $m \times n$, où m est le nombre d'apprenants et n le nombre de questions du test. Nous commençons par présenter le problème général de la factorisation de matrice, puis l'analyse en composantes principales, l'analyse de facteurs et enfin la théorie de la réponse à l'item multidimensionnelle.

4.2.1 Factorisation de matrice pour la réduction de dimension

Formellement, le problème de la factorisation de matrice consiste à résoudre l'équation :

$$D \simeq UV^T \tag{4.1}$$

— d est un entier compris entre 1 et n ;

1. Cette écriture est appelée *représentation distribuée* en apprentissage automatique.

— U est une matrice de taille $m \times d$, qui ici correspondrait aux caractéristiques de l'apprenant ;
— V est une matrice de taille $n \times d$, qui ici correspondrait aux caractéristiques des questions ;
— V^T indique la matrice transposée de V.

Résoudre cette équation permet d'exprimer les lignes de D, c'est-à-dire les réponses des apprenants aux n questions, comme une combinaison linéaire des d lignes de V^T, qui sont en nombre plus faible. On réduit ainsi la dimension du problème : au lieu de devoir déterminer les réponses de l'apprenant aux n questions, on se ramène à un problème intermédiaire : il faut déterminer les caractéristiques de l'apprenant sur seulement d dimensions. Ainsi, moins de réponses de l'apprenant seront nécessaires pour obtenir une estimation suffisante.

Dans le problème générique de la factorisation de matrice, les lignes de U sont appelés *poids* ou *facteurs* tandis que les lignes de V^T sont appelées composantes. Dans notre problème, on peut interpréter la i-ème ligne de U comme la maîtrise de l'apprenant selon plusieurs dimensions d, et la j-ème ligne de V comme l'importance de chaque dimension dans la résolution de la question j.

Les caractéristiques des questions ainsi extraites sont les lignes de V.

4.2.2 Analyse en composantes principales

L'analyse en composantes principales est une méthode descriptive qui consiste à déterminer par une factorisation de matrice les composantes qui expliquent le plus la variance des données. Pour procéder à cette analyse, on fait habituellement une décomposition en valeurs singulières, qui décompose la matrice D de la façon suivante :

$$D = U\Sigma W^T \tag{4.2}$$

— r est un entier compris entre 1 et n correspondant au rang de la matrice D des motifs de réponse des apprenants ;
— U est une matrice unitaire de taille $m \times r$, c'est-à-dire que $U^T U = I$ et donc que les lignes de U sont deux à deux orthogonales ;
— Σ est une matrice diagonale de taille $r \times r$ dont les éléments diagonaux $\sigma_1, \dots, \sigma_r$ sont classés par ordre décroissant : $\sigma_1 \geq \cdots \geq \sigma_r$;
— W est une matrice unitaire de taille $n \times r$;
— W^T indique la matrice transposée de W.

Cette décomposition est habituellement obtenue en diagonalisant la matrice $D^T D$ puis en se ramenant à D.

La première ligne de W^T est alors la composante expliquant le plus la variance des données, donc la direction séparant le mieux les données et les lignes de W

sont orthogonales deux à deux. La décomposition en valeurs singulières est donc un processus qui consiste à réordonner les composantes de façon à obtenir une représentation « éclatée » du jeu de données.

Une méthode de réduction de dimension consiste ensuite à calculer l'approximation de rang d de la matrice D pour une valeur de d choisie entre 1 et n, donnée par :

$$D_d = U_d \Sigma_d W_d^T \simeq D \tag{4.3}$$

- U_d est la matrice U tronquée aux d premières colonnes ;
- Σ_d est la matrice diagonale Σ dont on n'a conservé que les d éléments les plus grands, c'est-à-dire les d premiers : $\sigma_1 \geq \cdots \geq \sigma_d$;
- W_d est la matrice W tronquée aux d premières colonnes.

L'approximation D_d obtenue est bien une matrice de rang d, proche de la matrice D initiale des données des apprenants.

Ainsi, l'analyse en composantes principales est un processus déterministe qui conduira toujours au même résultat, et les caractéristiques des questions sont les colonnes de $\Sigma_d W_d^T$.

Une limitation de cette méthode est que si les données que l'on souhaite décomposer comportent des erreurs de mesure, aussi appelées *bruit*, la décomposition en valeurs singulières va prendre le bruit pour de la variance qui explique les données. Or, comme nous l'avons vu, les apprenants peuvent répondre faux à des questions par inattention, ou deviner la bonne réponse alors qu'ils ne maîtrisent pas les composantes nécessaires. C'est pourquoi on considère généralement des modèles probabilistes, qui modélisent le bruit comme une gaussienne.

4.2.3 Analyse de facteurs

Faire une analyse de facteurs consiste à supposer que les données sont issues d'un modèle vérifiant :

$$D = UV^T + E \tag{4.4}$$

- d est un entier compris entre 1 et $n - 1$;
- U est une matrice de taille $m \times d$, qui ici correspondrait aux caractéristiques de l'apprenant ;
- V est une matrice de taille $n \times d$, qui ici correspondrait aux caractéristiques des questions ;
- V^T indique la matrice transposée de V ;
- E est l'erreur de mesure dont la covariance est une matrice diagonale.

Cela consiste à supposer que les réponses de l'apprenant suivent une loi de probabilité telle que l'erreur de mesure a une variance différente selon chaque

dimension. Avec cette hypothèse, l'estimation des composantes est séparée de celle du bruit, donc le modèle est robuste aux erreurs des apprenants.

Les composantes ne sont alors plus orthogonales comme dans une analyse en composantes principales et sont ainsi plus facilement interprétables. Les caractéristiques des questions extraites par cette méthode sont les lignes de V.

4.2.4 Théorie de la réponse à l'item multidimensionnelle

Dans notre cas, la matrice D des données des apprenants ne comporte que des 1 et des 0, correspondant respectivement aux succès et échecs des apprenants sur des questions du test.

Ainsi, si l'on modélise l'apprenant d'après la théorie de la réponse à l'item multidimensionnelle (MIRT) de dimension d, si l'apprenant a un grand facteur selon une dimension impliquée dans la résolution d'une question, il aura simplement plus de chances de répondre correctement à la question. Pour rappel, la probabilité que l'apprenant i réponde correctement à la question j est donnée par l'expression :

$$Pr(D_{ij} = 1) = \Phi(\boldsymbol{\theta}_\mathbf{i} \cdot \mathbf{d}_\mathbf{j} + \delta_j) \tag{4.5}$$

- Φ est la fonction logistique définie sur les réels : $\Phi(x) = 1/(1 + e^{-x})$, qui tend vers 0 en $-\infty$ et vers 1 en $+\infty$;
- $\boldsymbol{\theta}_\mathbf{i}$ est le vecteur des caractéristiques de l'apprenant i ;
- $\mathbf{d}_\mathbf{j}$ est le vecteur des caractéristiques de la question j ;
- δ_j est un paramètre de facilité de la question j.

Écrit sous forme matricielle, le modèle MIRT devient, à la fonction logistique près, une analyse de facteurs :

$$D \simeq \Phi(\Theta V^T) \tag{4.6}$$

- d est un entier compris entre 1 et $n - 1$;
- Θ est de taille $m \times (d + 1)$, sa ligne i est $(\theta_{i1}, \dots, \theta_{id}, 1)$;
- V est de taille $n \times (d + 1)$, sa ligne j est $(d_{j1}, \dots, d_{jd}, \delta_j)$;
- V^T indique la matrice transposée de V.

L'estimation ne peut jamais être exacte, car Φ tend vers 0 et 1 en $\pm\infty$. Toutefois, Φ tend vite vers ses limites, par exemple $\Phi(4)$ vaut déjà environ 0,982.

Cette estimation revient à faire une phase d'apprentissage non supervisé, car les caractéristiques des questions sont directement extraites des données des apprenants D, elles ne sont pas spécifiées par un expert. Il y a $d(m+n)$ paramètres à estimer, donc la calibration des paramètres peine à converger lorsqu'on doit traiter des données de beaucoup d'apprenants.

4.3 Description du modèle GenMA

Nous proposons un nouveau modèle de test adaptatif basé sur un modèle de diagnostic cognitif. GenMA est hybride dans la mesure où il tire son inspiration d'un modèle de la théorie de la réponse à l'item, et requiert la spécification d'une q-matrice : on suppose que l'on connaît les composantes de connaissances (CC) mises en œuvre pour chaque question, sous la forme d'une q-matrice, dont l'élément q_{jk} vaut 1 si la CC k est impliquée dans la résolution de la question j, 0 sinon.

Comme pour chaque modèle de test adaptatif, il nous faut spécifier les points suivants.

Modèle de réponse Un modèle de la probabilité de répondre correctement à chaque question, basé sur une représentation du domaine et des caractéristiques de l'apprenant et des questions à identifier.

TrainingStep Une phase d'apprentissage de ces caractéristiques à partir des données d'entraînement, c'est-à-dire des données d'apprenants ayant répondu aux questions du test par le passé.

PriorInitialization Une méthode qui définit les caractéristiques initiales d'un nouvel apprenant.

NextItem Un critère de sélection de la question suivante.

EstimateParameters Une méthode d'estimation des caractéristiques de l'apprenant, c'est-à-dire son diagnostic, à partir des données observées : ses réponses aux questions posées.

Retour Un retour fait à la fin du test.

4.3.1 Modèle de réponse de l'apprenant sur une question

Caractéristiques de l'apprenant

L'apprenant i est modélisé par un vecteur $\boldsymbol{\theta}_\mathbf{i} = (\theta_{i1}, \ldots, \theta_{iK})$ de dimension K, où θ_{ik} représente son niveau selon chaque composante de connaissances k. Le niveau de l'apprenant selon une composante peut être positif, auquel cas ce paramètre participe à ce que l'apprenant réponde correctement aux questions qui requièrent cette composante. Il peut être négatif, auquel cas ce paramètre correspond à une lacune de l'apprenant, qui le pénalisera lorsqu'il répondra aux questions qui requièrent cette composante.

Caractéristiques des questions

Au lieu d'associer à chaque question j un vecteur de bits comme dans le modèle DINA, le modèle GenMA lui associe un vecteur de valeurs réelles $\mathbf{d}_\mathbf{j} =$

$(d_{j1}, \ldots, d_{jK})$, correspondant à des paramètres de discrimination selon chaque composante de connaissances, et un paramètre de facilité δ_j.

Modèle de diagnostic général

M. Davier (2005) a proposé un modèle de diagnostic cognitif dont la loi de probabilité est un cas particulier de MIRT (théorie de la réponse à l'item multidimensionnelle) et s'appuie sur un degré de maîtrise de chaque CC et d'une notion de difficulté des questions. Il s'agit du *modèle général de diagnostic* (*general diagnostic model for partial credit data*). La probabilité que l'apprenant i réponde correctement à la question j, c'est-à-dire que $D_{ij} = 1$, est donnée par :

$$Pr(D_{ij} = 1) = \Phi\left(\sum_{k=1}^{K} \theta_{ik} q_{jk} d_{jk} + \delta_j\right) \tag{4.7}$$

- K est le nombre de CC évaluées par le test ;
- θ_{ik} son niveau dans la CC k ;
- q_{jk} l'élément (j,k) de la q-matrice qui vaut 1 si la CC k est impliquée dans la résolution de la question j, 0 sinon ;
- d_{jk} est la discrimination de la question j selon la CC k ;
- δ_j est la facilité de la question j.

Lien avec MIRT Si la q-matrice ne contient que des 1, alors tous les q_{jk} valent 1, donc la probabilité que l'apprenant i réponde correctement à la question j devient :

$$Pr(D_{ij} = 1) = \Phi\left(\sum_{k=1}^{K} \theta_{ik} q_{jk} d_{jk} + \delta_j\right) = \Phi\left(\boldsymbol{\theta}_\mathbf{i} \cdot \mathbf{d}_\mathbf{j} + \delta_j\right) \tag{4.8}$$

qui est la loi de probabilité qui régit le modèle MIRT, voir section 2.3.1 page 31.

Lien avec le modèle de Rasch Si $K = 1$, que les paramètres de discrimination d_{j1} et de q-matrice q_{j1} sont tous fixés à 1, et que l'on remplace le paramètre de facilité δ_j par un paramètre de difficulté opposé $d_j = -\delta_j$, la probabilité que l'apprenant i réponde correctement à la question j devient :

$$Pr(D_{ij} = 1) = \Phi\left(\sum_{k=1}^{K} \theta_{ik} q_{jk} d_{jk} + \delta_j\right) = \Phi\left(\theta_{i1} - d_j\right) \tag{4.9}$$

qui est la loi de probabilité qui régit le modèle de Rasch, voir section 2.1 page 29.

La phase de calibrage d'un tel modèle conduit à une extraction des caractéristiques de chaque question j : $\mathbf{d_j} = (d_{j1}, \ldots, d_{jk})$ et δ_j similaire à MIRT de

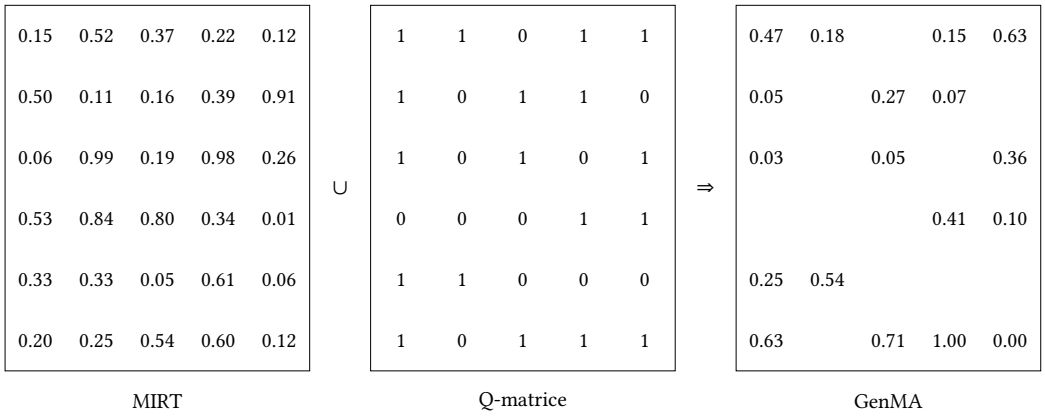


Figure 4.1 – Le modèle hybride GenMA, qui combine MIRT et une q-matrice.

dimension $d = K$ mais la q-matrice force une contrainte supplémentaire : pour chaque entrée nulle de la q-matrice q_{jk}, la composante correspondante dans la caractéristique de la question j est nulle : si $q_{jk} = 0$, alors $d_{jk} = 0$. Ainsi, il y a moins de paramètres à estimer (voir figure 4.1).

Contrairement au modèle DINA, où il faut maîtriser toutes les composantes de connaissances mises en œuvre dans une question afin d'y répondre correctement [2], le modèle de diagnostic général suppose que plus on maîtrise de composantes de connaissances mises en œuvre dans une question, plus grandes seront nos chances d'y répondre correctement. Et les paramètres de discrimination de chaque question permettent de favoriser certaines composantes plutôt que d'autres, dans le calcul de la probabilité de succès.

À titre d'exemple, supposons que le paramètre de discrimination d'une question j soit très grand pour une CC k mise en œuvre dans sa résolution ($q_{jk} = 1$ et d_{jk} est très grand). Si l'apprenant a un petit niveau $\theta_{ik} = \varepsilon > 0$ selon la CC k, alors comme $\theta_{ik} q_{jk} d_{jk}$ est grand, l'apprenant a de bonnes chances de répondre correctement à la question j. Si en revanche $\theta_{ik} = -\varepsilon < 0$, c'est-à-dire que l'apprenant a une lacune selon la composante k, alors $\theta_{ik} q_{jk} d_{jk}$ est faible et l'apprenant a peu de chances de répondre correctement à la question j. Ainsi, le paramètre de discrimination permet d'indiquer qu'une CC est plus ou moins importante dans le calcul de la probabilité qu'une certaine question soit résolue correctement. Et comme indiqué à la section 4.3.1 page précédente, si $q_{jk} = 0$ alors $d_{jk} = 0$, c'est-à-dire que le niveau de l'apprenant pour des composantes de connaissances non impliquées dans la résolution d'une question n'interviennent pas dans le calcul de ses chances d'y répondre correctement.

2. Pour rappel, le A de DINA signifie « And ».

4.3.2 Calibrage des caractéristiques

Les paramètres de facilité δ_j et de discrimination d_{jk} pour chaque question j et composante de connaissances k sont calibrés à partir des données des apprenants D, en utilisant l'algorithme de Metropolis-Hastings Robbins-Monro (R. Philip Chalmers, 2012 ; Cai, 2010).

Dans un test, chaque question fait habituellement appel à peu de CC, c'est-à-dire que la q-matrice est creuse : elle contient majoritairement des 0. C'est pourquoi le modèle GenMA est plus rapide à calibrer que le modèle MIRT général : il y a moins de paramètres à estimer car la q-matrice spécifie les seules caractéristiques intéressantes à estimer, voir à nouveau la figure 4.1. Si la q-matrice contient s entrées fixées à 1, alors au lieu de devoir estimer nd caractéristiques pour les n questions en dimension d, il suffit d'en estimer $s + n : s$ paramètres de discrimination en tout, et 1 paramètre de facilité pour chacune des n questions.

4.3.3 Initialisation des paramètres d'un nouvel apprenant

Au début du test, on suppose que l'apprenant est de niveau nul : $\boldsymbol{\theta} = (0, \dots, 0)$. Ainsi, pour chaque question j, la probabilité que l'apprenant y réponde correctement ne dépend que de son paramètre de facilité δ_j.

4.3.4 Choix de la question suivante

Pour le choix de la question suivante dans le modèle GenMA, nous choisissons de maximiser le déterminant de l'information de Fisher à chaque étape. Il s'agit de la règle D spécifiée à la section 2.3.1 page 31.

Cela correspond à choisir la question qui va le plus réduire la variance sur le paramètre à estimer, c'est-à-dire les caractéristiques $\boldsymbol{\theta}$ de l'apprenant qui passe le test.

4.3.5 Estimation des caractéristiques d'un nouvel apprenant

Après que l'apprenant i a répondu à une question, en fonction de sa réponse on calcule les paramètres $(\theta_{i1}, \dots, \theta_{iK})$ les plus vraisemblables, c'est-à-dire son niveau selon chaque composante de connaissances.

Pour cela, on calcule l'estimateur du maximum de vraisemblance, c'est-à-dire les paramètres $\boldsymbol{\theta}_\mathbf{i}$ qui maximisent la probabilité d'observer ces résultats sachant $\boldsymbol{\theta}_\mathbf{i}$ et l'expression de la probabilité que l'apprenant i a répondu correctement à la question j, comme un modèle de type MIRT habituel, c'est-à-dire en utilisant une régression logistique.

On suppose que le calibrage des questions a été effectué sur des données d'entraînement, et qu'on dispose des caractéristiques des questions dans une matrice V de taille $m \times d$ dont la ligne V_j correspond aux caractéristiques de la question j. À un certain moment du test, on a posé les questions $(q_1, \dots, q_t)$ de caractéristiques $V_{q_1}, \dots, V_{q_t}$ pour lesquelles on a observé les réponses $(r_1, \dots, r_t) \in \{0,1\}^t$ et on se demande quelle va être la performance de l'apprenant sur une certaine question j de caractéristiques V_j.

On cherche donc à estimer les caractéristiques de l'apprenant $\hat{\boldsymbol{\theta}}$ tels que pour chaque $k = 1, \dots, t$, $\Phi(\hat{\boldsymbol{\theta}} \cdot V_{q_k}) = r_k$. Ainsi, on pourra calculer la probabilité que l'apprenant réponde correctement à la question j de caractéristiques V_j, donnée par l'expression $\Phi(\hat{\boldsymbol{\theta}} \cdot V_j)$.

Il s'agit d'un problème d'apprentissage automatique, appelé *classification binaire*. Le modèle MIRT permet de résoudre ce problème en faisant une régression logistique.

Régression logistique Le modèle de régression logistique est utilisé pour la prédiction de variables dichotomiques (vrai ou faux), telles que les réponses des apprenants dans notre cas. Lorsqu'on a n éléments de dimension d $(\mathbf{x_1}, \dots, \mathbf{x_n})$ pour lesquels on observe des résultats vrai/faux $\mathbf{y} = (y_1, \dots, y_n) \in \{0,1\}^n$, la régression logistique consiste à estimer un paramètre $\boldsymbol{\theta} \in \mathbb{R}^d$ tel que $\Phi(\boldsymbol{\theta}^T X) = \mathbf{y}$ où X est la matrice ayant pour lignes les vecteurs $(\mathbf{x_1}, \dots, \mathbf{x_n})$. Ce modèle est apprécié pour sa propriété de généralisation à partir de peu de données.

Notre problème est directement encodable ainsi :
— il y a $n = t$ échantillons ;
— les échantillons $\mathbf{x_i}$ sont les caractéristiques $V_{q_1}, \dots, V_{q_t}$ des questions ;
— les étiquettes des échantillons sont les réponses correspondantes de l'apprenant $r_1, \dots, r_t$.

4.3.6 Retour à la fin du test

À la fin du test, l'apprenant reçoit un diagnostic sous la forme de valeurs de niveau selon chaque composante de connaissances (CC). Il s'agit du vecteur $\boldsymbol{\theta_i} = (\theta_{i1}, \dots, \theta_{iK})$. Si la valeur θ_{ik} selon la CC k est négative, il s'agit d'une lacune. Sinon, il s'agit d'un degré de maîtrise.

Contrairement à un modèle de type MIRT habituel, complètement automatique et où il faudrait interpréter les composantes a posteriori, le modèle GenMA a été calibré de façon que chaque dimension corresponde à une colonne de la q-matrice spécifiée par un expert, donc à une CC bien définie. Ainsi, les composantes du vecteur de niveau sont directement interprétables. GenMA est donc un modèle

	Dimension	Calibrage	De zéro	Nombre de paramètres
Rasch	1	Auto	Non	n
MIRT	$K \leq 4$	Auto	Non	$(K+1)n$
DINA	$K \leq 15$	Manuel	Oui	$2n$
GenMA	$K \leq 15$	Semi-auto	Non	$(k+1)n$

Table 4.1 – Comparaison qualitative des modèles de tests adaptatifs

formatif : le diagnostic qu'il fait à l'apprenant est plus utile pour la progression de l'apprenant qu'un simple score, car il indique ses éventuels points forts et lacunes.

4.4 Validation

4.4.1 Qualitative

Nous avons suivi l'analyse qualitative menée à la section 3.3 page 47. Les résultats sont répertoriés dans la table 4.1.

GenMA est multidimensionnel donc mesure des valeurs selon plusieurs CC, contrairement à Rasch qui ne mesure qu'une unique valeur correspondant au niveau de l'apprenant sur tout le test.

Comme indiqué à la section 4.3.6 page précédente, le modèle GenMA est interprétable car semi-automatique. Au lieu de déterminer automatiquement toutes les caractéristiques, GenMA permet d'orienter le calibrage en laissant un expert spécifier les paramètres de discrimination à estimer au moyen d'une q-matrice, qui fait le lien entre les questions et les CC. GenMA est à la fois basé sur la théorie de la réponse à l'item et sur une représentation des composantes de connaissances mises en œuvre dans le test, c'est donc un modèle hybride.

Cette spécification permet en outre d'accélérer la convergence, car il y a moins de paramètres à estimer que dans un modèle général de type MIRT. Si la matrice a en moyenne k entrées non nulles par ligne, GenMA estime pour ses questions kn paramètres de discrimination et n paramètres de facilité.

4.4.2 Modèles comparés

Pour quantifier la réduction du nombre de questions posées par le modèle GenMA, et la validité du diagnostic qu'il fournit, nous avons suivi le protocole décrit au chapitre précédent avec les modèles suivants.

Rasch Modèle unidimensionnel qui ne requiert pas de q-matrice.

MIRT Modèle de dimension 2 automatique.

DINA Le modèle DINA avec une q-matrice spécifiée par un expert.

GenMA Notre modèle GenMA avec la même q-matrice.

4.4.3 Jeux de données

Les jeux de données réelles suivants sont décrits plus en détail à la section 3.4.5 page 56.

Fraction $m = 536$ collégiens, $n = 20$ questions, q-matrice $K = 8$ CC.

ECPE $m = 2922$ apprenants, $n = 28$ questions, q-matrice $K = 3$ CC.

TIMSS $m = 757$ collégiens, $n = 23$ questions, q-matrice $K = 13$ CC.

4.4.4 Implémentation

Pour l'implémentation, nous utilisons le package `mirt`, voir la section A page 123 en annexe, qui permet de fixer les entrées non nulles à estimer (au moyen d'une q-matrice). Le package `mirtCAT` nous permet de poser les questions. Les résultats sont donnés dans les figures 4.2 à 4.4. Les valeurs obtenues de la *log loss* sont répertoriées dans les tables 4.2 à 4.5.

La validation croisée est faite sur 5 paquets d'apprenants et 2 paquets de questions.

4.4.5 Résultats et discussion

Sur chacun des jeux de données testés, GenMA a un plus grand pouvoir prédictif que l'autre modèle formatif DINA. La réponse à une question donnée par l'apprenant apporte plus d'information sur son niveau car chaque question a, contrairement au modèle DINA, des paramètres de discrimination selon chaque composante de connaissances.

Fraction

Dans le jeu de données Fraction, 4 questions sur 10 sont suffisantes pour prédire correctement 80 % en moyenne des réponses sur les 10 questions de l'ensemble de validation (voir table 4.2).

Les modèles Rasch, MIRT et DINA convergent en 4 ou 5 questions tandis que GenMA continue à apprendre car c'est un modèle de plus grande dimension que les autres. DINA est de dimension 8 comme GenMA mais c'est un modèle discret.

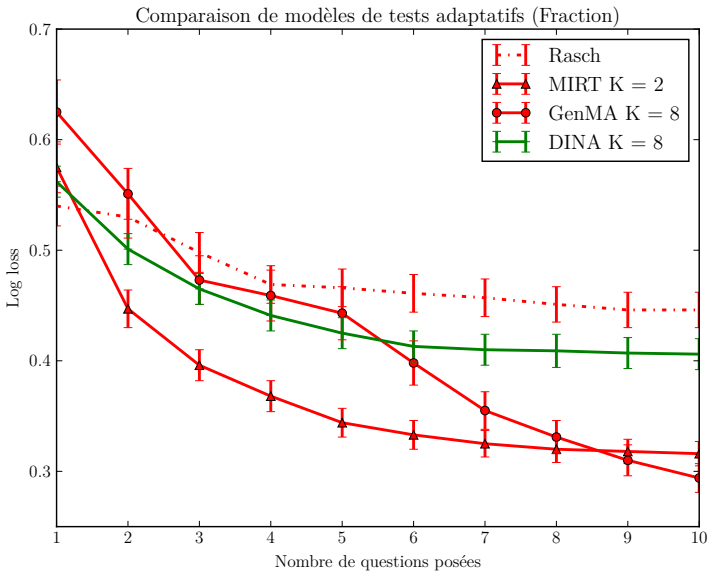


Figure 4.2 – Évolution de la *log loss* moyenne de prédiction en fonction du nombre de questions posées, pour le jeu de données Fraction.

	Après 4 questions	Après 7 questions	Après 10 questions
Rasch	0,469 $\pm$ 0,017 (79 %)	0,457 $\pm$ 0,017 (79 %)	0,446 $\pm$ 0,016 (79 %)
DINA	0,441 $\pm$ 0,014 (80 %)	0,410 $\pm$ 0,014 (82 %)	0,406 $\pm$ 0,014 (82 %)
MIRT	0,368 $\pm$ 0,014 (83 %)	0,325 $\pm$ 0,012 (86 %)	0,316 $\pm$ 0,011 (86 %)
GenMA	0,459 $\pm$ 0,023 (79 %)	0,355 $\pm$ 0,017 (85 %)	0,294 $\pm$ 0,013 (88 %)

Table 4.2 – Valeurs de *log loss* obtenues pour le jeu de données Fraction. Entre parenthèses, le taux de questions prédites correctement.

ECPE

Dans la figure 4.3, DINA et Rasch ont une performance similaire, ce qui est surprenant étant donné que Rasch ne requiert aucune connaissance du domaine. Nous supposons que cela apparaît car il n'y a que 3 CC décrites dans la q-matrice, donc le nombre d'états possibles pour un apprenant est $2^3 = 8$ pour 2^{28} motifs de réponse possibles. Ainsi, les paramètres d'inattention et de chance sont très hauts (voir la table 4.4), ce qui explique pourquoi l'information gagnée à chaque question est basse. Par exemple, la question qui requiert les CC 2 et 3 a un grand taux de succès de 88 %, ce qui rend cette question plus facile à résoudre que d'autres questions qui ne requièrent que la CC 2 ou 3, donc le seul moyen pour le modèle DINA d'exprimer ce comportement est d'accroître le paramètre de chance.

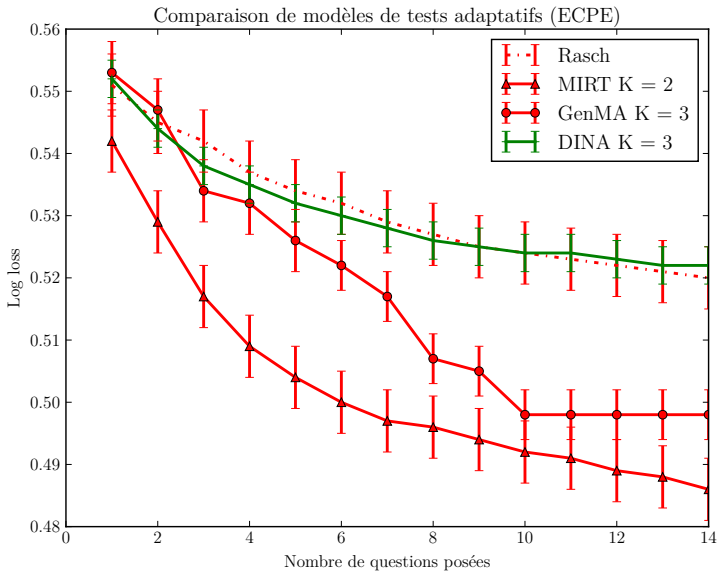


Figure 4.3 – Évolution de la *log loss* en fonction du nombre de questions posées, pour le jeu de données ECPE.

	Après 4 questions	Après 8 questions	Après 12 questions
DINA	0,535 $\pm$ 0,003 (73 %)	0,526 $\pm$ 0,003 (74 %)	0,523 $\pm$ 0,003 (74 %)
MIRT	0,509 $\pm$ 0,005 (76 %)	0,496 $\pm$ 0,005 (76 %)	0,489 $\pm$ 0,005 (77 %)
GenMA	0,532 $\pm$ 0,005 (73 %)	0,507 $\pm$ 0,004 (75 %)	0,498 $\pm$ 0,004 (76 %)
Rasch	0,537 $\pm$ 0,005 (73 %)	0,527 $\pm$ 0,005 (74 %)	0,522 $\pm$ 0,005 (74 %)

Table 4.3 – Valeurs de *log loss* obtenues pour le jeu de données Fraction. Entre parenthèses, le taux de questions prédites correctement.

À l'inverse, GenMA est un modèle plus expressif.

MIRT à 2 dimensions a un taux d'erreur plus faible que GenMA, ce qui laisse entendre qu'un modèle prédictif n'est pas nécessairement explicatif. Toutefois afin de faire un retour à l'utilisateur, notre modèle fait un diagnostic correspondant davantage à la réalité qu'un modèle DINA basé sur les q-matrices.

Nous faisons l'hypothèse que la q-matrice a été mal spécifiée.

TIMSS

Dans la figure 4.4, Rasch a une erreur plus faible que DINA. Dès 4 questions, GenMA a une erreur beaucoup plus faible, comparable à celle obtenue par MIRT.

Les modèles Rasch, DINA et MIRT convergent en 4 questions, tandis que

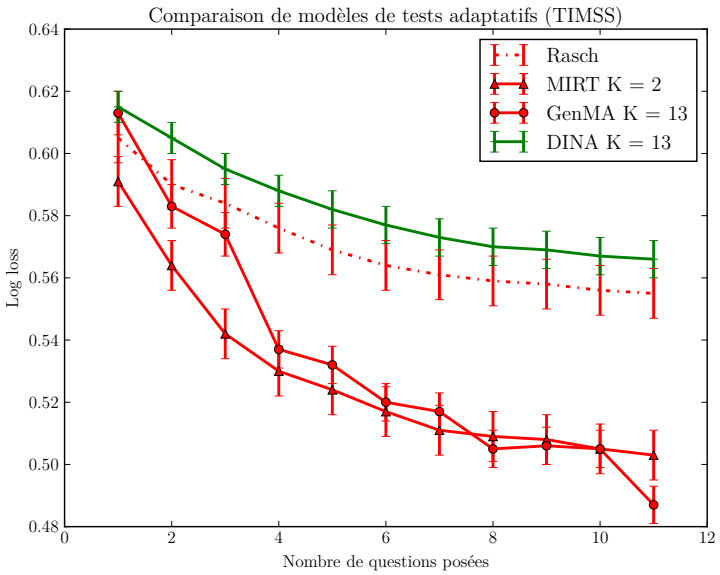


Figure 4.4 – Évolution de la *log loss* en fonction du nombre de questions posées, pour le jeu de données TIMSS.

	Après 4 questions	Après 8 questions	Après 11 questions
Rasch	0,576 $\pm$ 0,008 (70 %)	0,559 $\pm$ 0,008 (71 %)	0,555 $\pm$ 0,008 (71 %)
DINA	0,588 $\pm$ 0,005 (68 %)	0,570 $\pm$ 0,006 (70 %)	0,566 $\pm$ 0,006 (70 %)
GenMA	0,537 $\pm$ 0,006 (72 %)	0,505 $\pm$ 0,006 (75 %)	0,487 $\pm$ 0,006 (77 %)
MIRT	0,530 $\pm$ 0,008 (73 %)	0,509 $\pm$ 0,008 (75 %)	0,503 $\pm$ 0,008 (75 %)

Table 4.5 – Valeurs de *log loss* obtenues pour le jeu de données TIMSS. Entre parenthèses, le taux de questions prédites correctement.

GenMA continue à affiner son diagnostic : à la 11^{e} question, GenMA est le modèle le plus précis, car il prédit correctement 77 % des réponses sur l'ensemble de validation (voir table 4.5).

MIRT a une bonne propriété de généralisation à partir de peu de questions mais le diagnostic en deux dimensions qu'il crée n'est pas formatif, car ses dimensions ne sont pas interprétables sans intervention d'un expert. En revanche, les 13 dimensions du modèle GenMA correspondent chacune à une composante de connaissances de la matrice.

4.5 Conclusion

Dans ce chapitre, nous avons proposé un modèle hybride de tests adaptatifs, que nous avons validé en utilisant plusieurs jeux de données réelles, au moyen de notre système de comparaison défini à la section 3.4.4 page 53.

Comme indiqué dans la comparaison qualitative, le modèle MIRT ne peut pas facilement converger lorsqu'on le lance sur un grand nombre de dimensions. GenMA en revanche estime un nombre plus faible de paramètres, seulement ceux qui font le lien entre les questions et les composantes de connaissances (CC). C'est pourquoi il est possible de calibrer un modèle de plus grande dimension et faire un diagnostic plus riche à l'apprenant pour s'améliorer. GenMA est ainsi un modèle semi-automatique : un expert peut spécifier une q-matrice pour orienter la calibration des paramètres.

À présent que nous avons identifié un bon modèle pour faire du diagnostic adaptatif de connaissances, nous allons comparer différentes stratégies pour choisir les premières questions à poser dans un test.

q-matrice					taux de succès
entrées			chance	inattention	
1	1	0	0,705	0,085	80 %
0	1	0	0,724	0,101	83 %
1	0	1	0,438	0,266	57 %
0	0	1	0,480	0,162	70 %
0	0	1	0,764	0,040	88 %
0	0	1	0,717	0,066	85 %
1	0	1	0,544	0,085	72 %
0	1	0	0,802	0,040	89 %
0	0	1	0,534	0,199	70 %
1	0	0	0,483	0,163	65 %
1	0	1	0,556	0,099	72 %
1	0	1	0,195	0,305	43 %
1	0	0	0,633	0,122	75 %
1	0	0	0,517	0,212	65 %
0	0	1	0,749	0,040	88 %
1	0	1	0,549	0,126	70 %
0	1	1	0,816	0,058	88 %
0	0	1	0,729	0,086	84 %
0	0	1	0,473	0,150	71 %
1	0	1	0,239	0,295	46 %
1	0	1	0,621	0,097	75 %
0	0	1	0,322	0,188	63 %
0	1	0	0,637	0,075	81 %
0	1	0	0,313	0,322	53 %
1	0	0	0,512	0,272	61 %
0	0	1	0,555	0,211	70 %
1	0	0	0,265	0,369	44 %
0	0	1	0,659	0,086	81 %

Table 4.4 – Paramètres d'inattention et de chance pour la q-matrice du jeu de données ECPE.

Chapitre 5

InitialD : une heuristique pour le démarrage à froid

5.1 Introduction

Notre comparaison de modèles au chapitre précédent a mis en évidence que GenMA était le modèle de test adaptatif formatif réduisant le mieux les questions à poser. À présent, nous allons comparer différentes stratégies de choix des questions à poser à un nouvel apprenant.

Jusqu'à présent nous avons considéré des modèles de tests adaptatifs qui présentent des questions à l'apprenant une par une. Une variante consiste à ne commencer à adapter le processus de test que lorsque plusieurs réponses de l'apprenant ont été récoltées. Cela consiste à réduire l'adaptativité du processus pour obtenir en contrepartie plus de précision dans l'estimation des caractéristiques de l'apprenant. R. Philip Chalmers (2016) évoque aussi la possibilité de poser un groupe de questions au début du test avant de démarrer le processus adaptatif, et c'est sur ce problème que nous nous concentrons : le choix des toutes premières k questions.

Ce problème est de façon combinatoire plus difficile à résoudre que celui du choix de la question suivante : si maximiser une fonction objectif pour un seul élément reste faisable, en revanche itérer sur tous les sous-ensembles de questions à k éléments pour calculer une fonction objectif devient impraticable. On recourt alors généralement à des heuristiques, des algorithmes d'approximation, qui ont une borne de complexité prouvée.

Dans ce chapitre, nous nous intéressons au choix des k premières questions qui vont nous aider à estimer le niveau de l'apprenant. Un autre type d'application est la génération automatique de fiches d'exercices sur un MOOC, à partir d'exercices piochés dans les banques des QCM qui se trouvent déjà dans le MOOC. À noter que

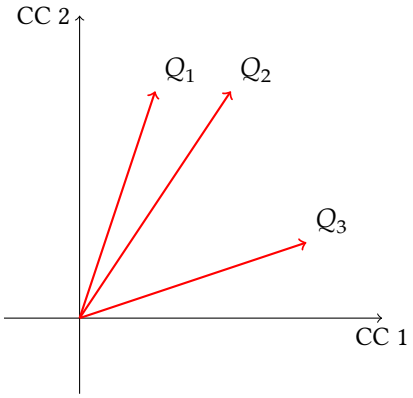


Figure 5.1 – Caractéristiques de trois questions sur deux composantes de connaissances.

l'algorithme que nous présentons n'est pas déterministe : relancer la génération plusieurs fois donnera des planches d'exercices différentes, ce qui est intéressant pour diversifier les questions présentées aux apprenants.

Nous avons vu au chapitre précédent que les modèles issus de la théorie de réponse à l'item à plusieurs dimensions consistaient à affecter à chaque question un vecteur de caractéristiques indiquant la direction dans laquelle le vecteur mesure le niveau de l'apprenant. Ainsi, des questions ayant des vecteurs proches ont des motifs de réponse proches : comme elles mesurent les mêmes composantes, il y a de fortes chances qu'un apprenant qui parvient à en résoudre une parvienne à résoudre les autres.

Comme on cherche à réduire le plus possible le nombre de questions, on a intérêt à choisir des questions ayant des vecteurs peu corrélés deux à deux. Par exemple, si l'on suppose que l'on a trois questions dont les caractéristiques en dimension 2 sont représentées à la figure 5.1, et qu'on souhaite n'en choisir que deux, il vaut mieux choisir la 1re et la 3e, afin d'avoir une mesure de l'apprenant la moins redondante possible. Nous présentons dans ce chapitre une méthode pour tirer des questions éloignées les unes des autres, basée sur une loi de probabilité appelée processus à point déterminantal.

Nous avons ainsi une mesure de ce qui constitue un « bon » ensemble de k questions à poser : si le volume de l'espace clos engendré par les vecteurs caractéristiques des questions est grand, l'information apportée par ces questions sur le niveau de l'apprenant sera discriminante sur plusieurs dimensions. Calculer le volume de l'espace engendré par chacun des choix de k vecteurs parmi n a une complexité $O(\binom{n}{k}k^2d + k^3)$, donc c'est impraticable. Mais la formulation du problème que nous proposons dans ce chapitre permet de tirer un bon ensemble, pas nécessairement le meilleur, avec une complexité $O(nk^3)$, permettant l'application

de notre approche à de grandes banques de questions.

Dans ce chapitre, nous rappelons ce qui caractérise un bon ensemble de questions à poser au début du test. Puis nous donnons la définition d'un processus à point déterminantal et expliquons comment notre problème peut s'y ramener. Enfin, nous décrivons notre stratégie de choix des k toutes premières questions appelée InitialD (pour *Initial Determinant*) et la validons via un protocole expérimental qui généralise notre méthode décrite à la section 3.4.4 page 53. Sur tous les jeux de données étudiés, InitialD est meilleur que les autres approches connues.

5.1.1 Caractérisation de la qualité d'un ensemble de questions

À la section 2.3 page 33, nous avons mentionné les *tests à étapes multiples* qui consistent à poser un groupe de questions à l'apprenant, obtenir ses réponses en bloc, pour ensuite choisir le groupe suivant de questions à poser, plutôt que d'adapter le processus question après question. Cela permet d'avoir plus d'informations sur l'apprenant avant de réaliser la première estimation de son niveau qui permettra de choisir le groupe de questions suivant. De plus, cela permet à l'apprenant d'avoir plus de recul sur les exercices qui lui sont posés et de se relire avant de valider, plutôt que d'obtenir des questions portant sur des composantes de connaissances (CC) diverses question après question.

Ainsi, le problème devient : comment choisir les k premières questions à présenter à un nouveau venu ? Elles doivent mesurer des CC diversifiées afin d'estimer au mieux le niveau de l'apprenant.

5.1.2 Visualisation géométrique d'un test adaptatif

Pour mieux comprendre notre approche, voici une interprétation géométrique de ce qu'il se passe lorsqu'un test adaptatif multidimensionnel est administré.

Pour rappel, la phase d'apprentissage du modèle GenMA de dimension K consiste à déterminer les caractéristiques $\mathbf{d_j} = (d_{j1}, \dots, d_{jK})$ et δ_j de chaque question j et les caractéristiques $\boldsymbol{\theta}_\mathbf{i} = (\theta_{i1}, \dots, \theta_{iK})$ de chaque apprenant i. La probabilité qu'un apprenant i réponde correctement à une question j est ensuite donnée par l'expression $\Phi(\boldsymbol{\theta}_\mathbf{i} \cdot \mathbf{d_j})$. Pour visualiser, on peut représenter les questions par des points à coordonnées $(d_{j1}, \dots, d_{jK})$ pour chaque j et l'apprenant i par le vecteur $\boldsymbol{\theta}_\mathbf{i}$. Les questions qui ont le plus de chances d'être résolues par l'apprenant correspondent aux points qui se trouvent le plus dans la direction de $\boldsymbol{\theta}_\mathbf{i}$.

Ainsi, poser un jeu de k questions revient à choisir k points de l'espace à présenter à l'apprenant, ce qui permettra après étiquetage par succès/échec en

fonction de ses réponses de déterminer une première estimation de son vecteur de niveau $\boldsymbol{\theta}$.

Pour estimer les caractéristiques de l'apprenant, on souhaite choisir l'estimateur du maximum de vraisemblance. Mais si les réponses que l'apprenant a faites jusque-là sont toutes correctes ou toutes incorrectes, l'estimateur tend vers $\pm\infty$ et il faut choisir un autre estimateur. Ce problème avait déjà été mis en évidence par Lan, Waters, et al. (2014) et par Magis (2015).

5.1.3 Stratégies de choix de k questions

Aléatoire Une première méthode naïve consiste à choisir k questions au hasard.

Incertitude maximale Une des méthodes en apprentissage automatique consiste à choisir les questions les plus incertaines, c'est-à-dire celles pour lesquelles la probabilité que l'apprenant réponde correctement est la plus proche de 0,5. Toutefois, cela risque d'apporter de l'information redondante (Hoi et al., 2006).

Déterminant maximal Une façon de choisir des questions peu corrélées les unes des autres consiste à choisir un ensemble de questions dont le parallélotope [1] forme un grand volume. Le volume d'un parallélotope formé par des vecteurs $V = (\mathbf{v_1}, \dots, \mathbf{v_n})$ où $\mathbf{v_1}, \dots, \mathbf{v_n}$ sont les lignes de la matrice V, est donné par $Vol(\{\mathbf{v_i}\}_{i=1,\dots,n}) = \sqrt{\det VV^T}$.

5.2 Processus à point déterminantal

Nous allons présenter une loi de probabilité, tirée de la théorie des matrices aléatoires, qui a récemment été appliquée en apprentissage automatique (Kulesza et Taskar, 2012). Cette loi permet, étant donné des objets munis de caractéristiques, d'échantillonner efficacement des éléments « diversifiés » pour une certaine mesure de distance. Cela a par exemple des applications en recommandation pour sélectionner des produits diversifiés, dans les moteurs de recherche afin que les résultats en tête de la recherche portent sur des thèmes différents (par exemple, pour une requête « jaguar », l'animal et la voiture) ou encore en génération automatique de résumé, à partir d'un corpus de textes, par exemple des articles de presse dont on souhaiterait sélectionner les thèmes principaux.

Tout d'abord, il nous faut définir la notion de noyau, qui est une généralisation du produit scalaire. Soit $d \geq 1$ un entier, une fonction symétrique $K : \mathbb{R}^d \times \mathbb{R}^d \to$

1. Une généralisation du parallélogramme en dimension n quelconque.

$\mathbb{R}$ est un *noyau* si pour tout n entier, pour tous $\mathbf{x_1}, \dots, \mathbf{x_n} \in \mathbb{R}^d$ et pour tous $(c_1, \dots, c_n) \in \mathbb{R}^d$, $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x_i}, \mathbf{x_j}) \geq 0$.

Pour implémenter cet échantillonnage, il faut :
- un ensemble de n éléments à échantillonner, identifiés par les indices $X = \{1, \dots, n\}$
- pour chaque élément $i \in X$, un vecteur $\mathbf{x_i}$ de dimension d correspondant aux caractéristiques de l'élément i ;
- un noyau K permettant de décrire une valeur de similarité pour chaque paire d'éléments. Ce noyau permet de définir une matrice symétrique L telle que $L_{ij} = K(\mathbf{x_i}, \mathbf{x_j})$.

Pour nos usages, nous avons utilisé le simple noyau linéaire $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}$, mais il est possible d'utiliser le noyau gaussien :

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2}\right). \tag{5.1}$$

Formellement, un processus stochastique $Y \subset \{1, \dots, n\}$ est un processus à point déterminantal (PPD) s'il vérifie pour tout ensemble $A \subset \{1, \dots, n\}$:

$$Pr(A \subset Y) \propto \det L_A \tag{5.2}$$

où L_A est la sous-matrice carrée de L indexée par les éléments de A en ligne et colonne.

Dans notre cas, cette loi est intéressante car des éléments seront tirés avec une probabilité proportionnelle au carré du volume du parallélotope qu'ils forment. En effet, chaque élément L_{ij} de la matrice L vaut $L_{ij} = K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}$ donc si on note B la matrice ayant pour lignes $\mathbf{x_1}, \dots, \mathbf{x_n}$, on a $L = BB^T$. Si à présent on note B_A la matrice ayant pour lignes les vecteurs $\mathbf{x_i}$ pour i appartenant à A, $L_A = B_A B_A^T$ et donc $Pr(A \subset Y) \propto \det L_A = \det B_A B_A^T = Vol(\{\mathbf{x_i}\}_{i \in A})^2$.

Or, plus le volume d'un ensemble de vecteurs est grand, moins ces vecteurs sont corrélés. Ainsi, des éléments diversifiés auront plus de chances d'être tirés par un PPD. On peut encore le voir de la façon suivante : des vecteurs de questions similaires apportent une information similaire. Afin d'avoir le plus d'information possible au début du test il vaut mieux choisir des vecteurs écartés deux à deux.

Il existe des algorithmes efficaces pour échantillonner selon une PPD (Kulesza et Taskar, 2012), y compris lorsqu'on fixe à l'avance le nombre d'éléments qu'on souhaite sélectionner (k-PPD) : la complexité de tirage est $O(nk^3)$ où n est le nombre de questions, à condition d'avoir calculé la diagonalisation de la matrice L au préalable, ce qui peut se faire avec une complexité $O(n^3)$ par exemple avec la méthode de Gauss-Jordan. En revanche, le problème de déterminer le mode de cette

distribution (c'est-à-dire l'ensemble X de plus grande probabilité a posteriori) est un problème NP-difficile, c'est pourquoi des algorithmes d'approximation ont été développés. Ce n'est que récemment que les PPD sont appliqués à l'apprentissage statistique, mais surtout à des méthodes de diversification et de résumé.

Un autre avantage de cette méthode est que le choix de k questions est probabiliste, ainsi on ne pose pas nécessairement les mêmes k premières questions à tous les apprenants, ce qui présente certains avantages en termes de sécurité et de diversification de la banque de questions.

5.3 Description de la stratégie InitialD

Notre contribution consiste à appliquer la méthode de tirage d'éléments diversifiés selon un PPD au choix de questions diversifiées au début d'un test, de façon automatique.

Étant donné des données d'apprenants D correspondant à des succès et échecs de m apprenants sur n questions, et une q-matrice de taille $n \times K$, on calibre un modèle GenMA. On extrait donc des caractéristiques en dimension K pour chacune des n questions du test : chaque question j a pour caractéristiques le vecteur $\mathbf{d_j} = (d_{j1}, \dots, d_{jK})$.

La stratégie InitialD consiste à considérer les questions $X = \{1, \dots, n\}$ et pour chaque question j les caractéristiques $\mathbf{d_j} = (d_{j1}, \dots, d_{jK})$. Le noyau choisi est le noyau linéaire : $K(\mathbf{d_i}, \mathbf{d_j}) = \mathbf{d_i} \cdot \mathbf{d_j}$, et nous cherchons à tirer k questions parmi les n selon un PPD. Nous faisons l'hypothèse que les questions ainsi choisies seront peu redondantes, donc constitueront un bon résumé des questions du test pour l'apprenant.

L'algorithme de tirage est tiré de (Kulesza et Taskar, 2012) et est implémenté en Python. Sa complexité est $O(nk^3)$ où k est le nombre de questions sélectionnées et n est le nombre de questions du test, après une coûteuse étape de diagonalisation de complexité $O(n^3)$. Ainsi, cette complexité convient à une grande base de questions comme peut l'être celle sur un MOOC, car l'étape de tirage est linéaire en le nombre de questions de la banque.

5.4 Validation

À partir d'un jeu de données réelles des réponses des apprenants, nous allons comparer quatre stratégies pour choisir les k premières questions. Le modèle de test adaptatif considéré est GenMA.

5.4.1 Stratégies comparées

Nous avons comparé quatre stratégies. Les trois premières ne sont pas adaptatives, la quatrième l'est. Chacune des 3 premières correspond donc à une implémentation de la fonction FirstBundle.

Random Les questions sont choisies au hasard.

Uncertainty On suppose que l'apprenant est de niveau initial $(0, \dots, 0)$ et on choisit k questions de probabilité estimée proche de 0,5, c'est-à-dire d'incertitude maximale.

InitialD L'algorithme présenté à la section précédente qui choisit les k questions selon un processus à point déterminantal.

CAT Enfin, nous ajoutons à ces trois stratégies la sélection adaptative habituelle, question par question, afin de comparer nos trois stratégies non adaptatives aux métriques obtenues avec la stratégie adaptative.

5.4.2 Jeux de données réelles

Pour les jeux de données Fraction et TIMSS, grâce aux q-matrices et au modèle GenMA nous obtenons une représentation distribuée des questions de dimension 8, que nous utilisons pour calculer la matrice de similarité et échantillonner les questions.

5.4.3 Protocole expérimental

Notre protocole est similaire à celui développé pour la comparaison de modèles de tests adaptatifs à la section 3.4.4 page 53, à l'exception d'une méthode FirstBundle qui prend en argument la stratégie S choisie, le nombre de questions à poser k, les caractéristiques des questions $(\mathbf{d_j})_{j=1,\dots,n}$ et $(\delta_j)_{j=1,\dots,n}$, les caractéristiques initiales de l'apprenant $\boldsymbol{\theta} = (0, \dots, 0) \in \mathbb{R}^K$ et renvoie un ensemble Y de k questions à poser à l'apprenant. Contrairement au chapitre précédent, ici nous ne comparons plus des modèles différents mais des stratégies différentes pour le même modèle GenMA.

Nous séparons les apprenants en deux ensembles d'entraînement et de test (80 % et 20 %) et calibrons le modèle GenMA avec les apprenants d'entraînement. Puis, pour chaque apprenant de test, nous choisissons k premières questions à poser, récoltons ses réponses et estimons son vecteur de niveau (voir algorithme 2).

Nous mesurons alors deux métriques, pour différentes valeurs du nombre de questions k.

Qualité du diagnostic

Quelle est la performance des prédictions qui découlent de ce premier groupe de questions, en termes de *log loss* et de nombre de prédictions incorrectes ?

Distance au diagnostic final

Quelle est la différence entre le paramètre estimé à partir de k questions et le paramètre estimé lorsqu'on a posé toutes les questions ? Cette valeur est calculée par Lan, Waters, et al. (2014) pour comparer les méthodes de sélection de questions.

Algorithme 2 Simulation de choix des k premières questions

procédure SIMULATEPRETEST(stratégie S, I_{train}, I_{test})
  $(\mathbf{d_j})_j, (\delta_j)_j \leftarrow$ TRAININGSTEP($D[I_{train}]$)
  pour tout apprenant s de l'ensemble I_{test} **faire**
    pour k de 1 à n **faire**
      $\theta \leftarrow$ PRIORINITIALIZATION()
      $Y \leftarrow$ FIRSTBUNDLE($S, k, (\mathbf{d_j})_j, (\delta_j)_j, \theta$)
      Poser les questions Y à l'apprenant s
      Récupérer les valeurs de succès ou échec correspondantes $(r_i)_{i \in Y}$ de ses réponses
      $\theta \leftarrow$ ESTIMATEPARAMETERS($\{(i, r_i)\}_{i \in Y}, \theta$)
      $p \leftarrow$ PREDICTPERFORMANCE($\theta, (\mathbf{d_j})_j$)
      $\sigma_k \leftarrow$ EVALUATEPERFORMANCE($p, D[s], \theta$)

5.4.4 Résultats

Les résultats sont donnés dans les figures 5.2 à 5.5.

TIMSS

Dans la figure 5.2, InitialD est bien meilleur que Random, bien meilleur que CAT, bien meilleur que Uncertainty (voir 5.1). Dans les premières questions, CAT a une erreur comparable à celle de Uncertainty, car les deux modèles choisissent la question pour laquelle la probabilité que l'apprenant y réponde correctement est la plus proche de 0,5. Mais InitialD explore davantage en choisissant un groupe de questions diversifiées.

Dès la première question, InitialD a une meilleure performance. C'est parce que choisir la question de plus grand « volume » correspond à choisir la question

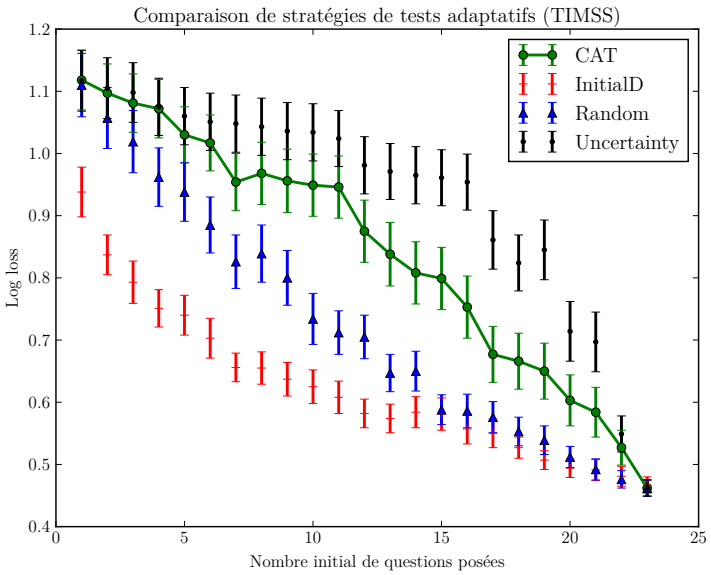


Figure 5.2 – *Log loss* du modèle GenMA après qu'un groupe de questions a été posé selon certaines stratégies pour le jeu de données TIMSS.

	Après 3 questions	Après 12 questions	Après 20 questions
CAT	1,081 ± 0,047 (62 %)	0,875 ± 0,050 (66 %)	0,603 ± 0,041 (75 %)
Uncertainty	1,098 ± 0,048 (58 %)	0,981 ± 0,046 (68 %)	0,714 ± 0,048 (72 %)
InitialD	0,793 ± 0,034 (61 %)	0,582 ± 0,023 (70 %)	0,494 ± 0,015 (74 %)
Random	1,019 ± 0,050 (58 %)	0,705 ± 0,035 (68 %)	0,512 ± 0,017 (74 %)

Table 5.1 – Valeurs de *log loss* pour le jeu de données TIMSS.

dont le vecteur caractéristique a la plus grande norme, ou encore : la question la plus discriminante.

Dans la figure 5.3, on voit que InitialD converge plus vite vers le vrai paramètre que les autres stratégies (voir 5.2).

Fraction

Dans la figure 5.4, InitialD est meilleur que les autres stratégies. Uncertainty est la stratégie de plus grande variance, tandis que Random a une erreur comparable à CAT (voir 5.3).

Dans la figure 5.5, le modèle qui converge le plus vite vers le vrai paramètre est InitialD pour la première moitié des questions, et CAT pour la deuxième moitié des questions, ce qui semble être un compromis entre choisir un groupe de questions avant de faire la première estimation, et adapter pour converger plus

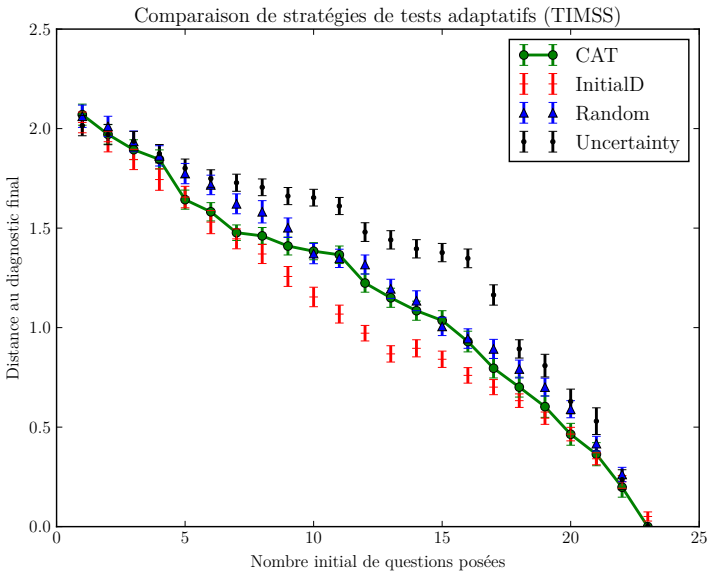


Figure 5.3 – Distances au diagnostic final après qu'un groupe de questions a été posé selon certaines stratégies pour le jeu de données TIMSS.

	Après 3 questions	Après 12 questions	Après 20 questions
CAT	1,894 ± 0,050	1,224 ± 0,046	0,464 ± 0,055
Uncertainty	1,937 ± 0,049	1,480 ± 0,047	0,629 ± 0,062
InitialD	1,845 ± 0,051	0,972 ± 0,039	0,465 ± 0,034
Random	1,936 ± 0,052	1,317 ± 0,048	0,590 ± 0,043

Table 5.2 – Distances au diagnostic final pour le jeu de données TIMSS.

vite vers le vrai paramètre à estimer (voir 5.4).

5.4.5 Discussion et applications

Si le nombre de questions à poser k, le nombre de questions disponibles n et le nombre de dimensions d sont des petites valeurs, il est possible de simuler tous les choix possibles de k questions parmi n. Toutefois, en pratique, les banques de questions sur des plateformes de MOOC seront telles que la complexité de InitialD, $O(nk^3)$ après un précalcul de $O(n^3)$, sera un avantage.

La méthode proposée dans ce chapitre ne cherche pas à déterminer le meilleur ensemble de questions à poser, mais un bon ensemble de questions tiré au hasard. Ajouter de l'aléa dans cette technique présente plusieurs avantages : les premières questions posées à chaque candidat ne sont pas les mêmes. Si cela constitue une surcharge supplémentaire lorsqu'on doit corriger manuellement les exercices des

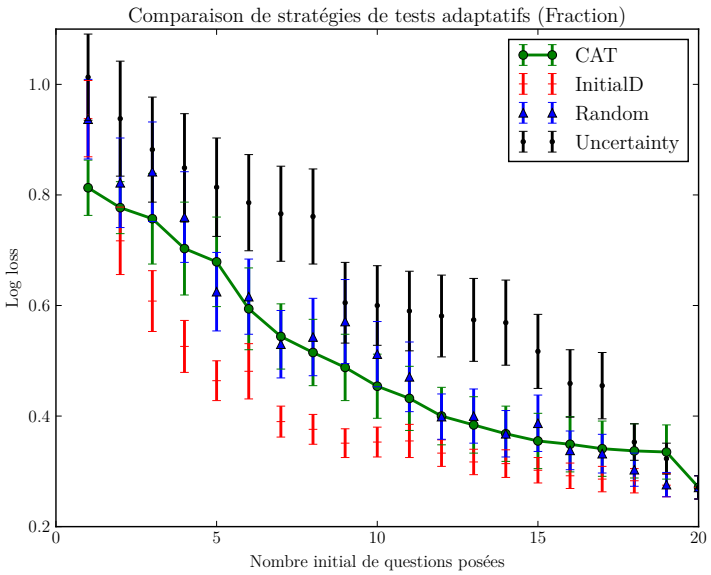


Figure 5.4 – *Log loss* du modèle GenMA après qu'un groupe de questions a été posé selon certaines stratégies pour le jeu de données Fraction.

	Après 3 questions	Après 8 questions	Après 15 questions
CAT	0,757 ± 0,082 (67 %)	0,515 ± 0,060 (82 %)	0,355 ± 0,050 (88 %)
Uncertainty	0,882 ± 0,095 (72 %)	0,761 ± 0,086 (76 %)	0,517 ± 0,067 (86 %)
InitialD	0,608 ± 0,055 (74 %)	0,376 ± 0,027 (82 %)	0,302 ± 0,023 (86 %)
Random	0,842 ± 0,090 (70 %)	0,543 ± 0,070 (80 %)	0,387 ± 0,051 (86 %)

Table 5.3 – Valeurs de *log loss* pour le jeu de données Fraction.

apprenants, en revanche lorsqu'ils sont administrés automatiquement sur une plateforme, cela permet d'éviter que les apprenants ne s'échangent les réponses, ou de trop utiliser les mêmes exercices de sa banque.

La stratégie InitialD peut être améliorée en ne tirant pas un seul ensemble de k questions mais plusieurs, et en conservant le meilleur des échantillons. Tirer ℓ fois k questions a une complexité $O(\ell n k^3)$, déterminer le meilleur ensemble a une complexité $O(\ell k^3)$. Faire plusieurs tirages augmente la probabilité de déterminer ainsi le meilleur ensemble de questions.

Génération automatique de fiches d'exercices

Le test préalable peut être également appliqué à la génération d'une fiche d'exercices « diversifiée » étant donné un historique de réponses.

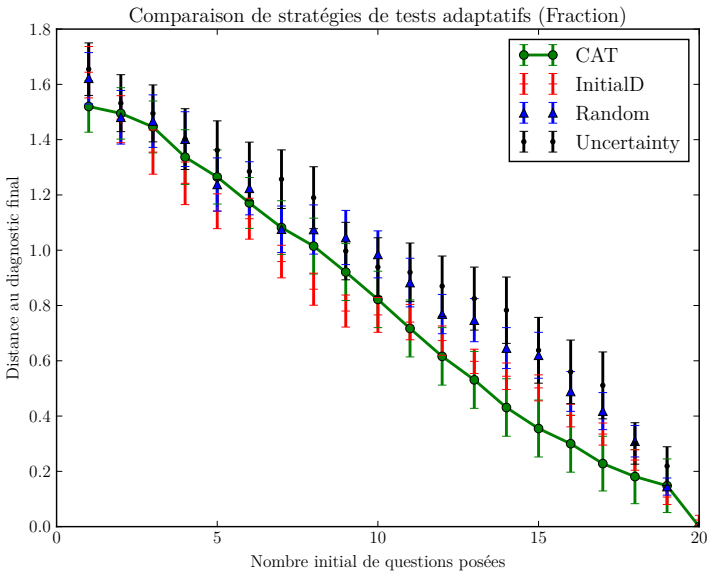


Figure 5.5 – Distances au diagnostic final après qu'un groupe de questions a été posé selon certaines stratégies pour le jeu de données Fraction.

	Après 3 questions	Après 8 questions	Après 15 questions
CAT	1,446 ± 0,094	1,015 ± 0,101	0,355 ± 0,103
Uncertainty	1,495 ± 0,103	1,190 ± 0,112	0,638 ± 0,119
InitialD	1,355 ± 0,080	0,859 ± 0,058	0,502 ± 0,047
Random	1,467 ± 0,095	1,075 ± 0,089	0,620 ± 0,083

Table 5.4 – Distances au diagnostic final pour le jeu de données Fraction.

Démarrage à froid de question

Cette méthode pourrait être appliquée au problème de démarrage à froid de la question : lorsqu'une nouvelle question est ajoutée à un test existant, on ne dispose d'aucune information concernant son niveau. Une méthode consiste à, de façon similaire, la poser à des apprenants qui ont des niveaux diversifiés pour estimer ses caractéristiques. C'est l'approche qu'adoptent Anava et al. (2015) dans un contexte de filtrage collaboratif. On peut imaginer sur un MOOC repérer le nombre de personnes actuellement connectées et tirer un sous-ensemble d'apprenants à qui poser la question.

5.5 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle manière de choisir les k premières questions à administrer à un nouvel apprenant, via InitialD, un algorithme probabiliste efficace inspiré de la littérature en apprentissage automatique. Grâce à la complexité de $O(nk^3)$ pour choisir k questions à poser parmi n (après un précalcul de complexité $O(n^3)$ qui peut être parallélisé car il s'agit d'une diagonalisation de matrice), notre méthode peut être appliquée à des grandes banques de questions, telles que celles que peut contenir un cours en ligne.

Nous avons également mis en évidence qu'un processus non adaptatif peut être utile pour les toutes premières questions, tandis qu'une évaluation adaptative peut donner de meilleurs résultats plus loin dans le test.

Chapitre 6

Conclusion et perspectives

6.1 Conclusion

6.1.1 Travaux effectués

Dans cette thèse, nous avons décidé de poser le regard de l'apprentissage automatique sur l'évaluation adaptative des apprenants. Cela nous a permis d'avoir un point de vue générique, nous permettant de comparer différents modèles sur les mêmes données de tests et de fournir des stratégies originales. Cela nous permettra également d'importer de nombreux autres modèles tels que des machines à vecteurs de support [1], ou des machines à hausse de gradient [2]. Les modèles d'apprentissage automatique ont parfois une mauvaise réputation due à la difficulté d'interpréter leurs déductions, mais ce n'est pas le cas de modèles linéaires ou log-linéaires tels que la régression logistique, qui est pourtant déjà un exemple de modèle d'apprentissage automatique.

Nous avons mis en évidence à la section 3.6.1 page 66 que selon le type de test : au début d'un MOOC, au milieu d'un MOOC ou à la fin d'un MOOC, le modèle le plus approprié n'était pas le même. Pour choisir un modèle, il faut se poser les questions suivantes :

— De quelle représentation du domaine dispose-t-on ? Un graphe de prérequis sur les composantes de connaissances (CC) développées dans le cours ? Ou bien une q-matrice, qui fait le lien entre questions et CC requises ?
— S'agit-il de la première administration du test ou dispose-t-on déjà de données d'apprenants y ayant répondu ?
— Est-ce que les connaissances de l'apprenant évoluent alors qu'il passe le test ? Par exemple, a-t-il accès à des notes de cours ?

1. En anglais, *support vector machine*.
2. En anglais, *gradient boosting machine*.

— Souhaite-t-on poser peu de questions pour identifier les connaissances de l'apprenant ou le faire progresser le plus possible alors qu'il passe le test ?

Les modèles de tests adaptatifs que nous avons étudiés proviennent de divers domaines de la littérature. Les angles sous lesquels nous avons choisi de les comparer qualitativement (voir section 3.3 page 47) nous ont permis de repérer que certains modèles nommés différemment étaient presque identiques (par exemple, la théorie des espaces de connaissances et le modèle de hiérarchie sur les attributs, voir section 2.3.2 page 37).

Comparer le modèle de diagnostic général aux modèles de théorie de la réponse à l'item (voir section 4.3.1 page 82), et étudier comment le modèle DINA, au départ un simple modèle de diagnostic cognitif, a été utilisé dans des tests adaptatifs, nous a permis de proposer le modèle de tests adaptatifs GenMA, tirant parti des avantages des autres modèles. GenMA est formatif, fournit un diagnostic plus vraisemblable que le modèle DINA car il incorpore des notions de discrimination selon chaque composante de connaissances, et est plus rapide à calibrer que les modèles habituels de théorie de réponse à l'item multidimensionnelle (MIRT).

Voir la phase de calibrage de GenMA comme un problème d'apprentissage non supervisé à partir des réponses des apprenants, nous a permis de redéfinir le problème de choisir les k premières questions d'une façon géométrique. Nous avons ainsi pu proposer la stratégie InitialD, basée sur un algorithme d'échantillonnage de processus à point déterminantal, récemment utilisé en apprentissage automatique pour sa faible complexité (Kulesza et Taskar, 2012). InitialD peut tirer un ensemble de k questions diversifiées parmi n avec une complexité $O(nk^3)$, qui permettent au modèle de diagnostic GenMA de converger vers un diagnostic plus vraisemblable que des stratégies usuelles.

6.1.2 Limitations

Dans les tests adaptatifs, nous n'observons pas toutes les réponses mais seulement celles aux questions que nous avons posées. Or, dans toutes les approches développées dans cette thèse requérant des données, nous avons supposé que nous disposions des réponses de tous les apprenants à toutes les questions. Certains modèles se comportent différemment si tous les apprenants n'ont pas répondu à toutes les questions. En filtrage collaboratif où les utilisateurs notent en moyenne 1 % de tous les produits, on est cependant habitué à traiter ce problème des matrices creuses. Ainsi, on pourrait appliquer ces techniques à notre problème.

Dans cette thèse, nous nous sommes concentrés sur l'évaluation d'un seul apprenant. Certaines méthodes en analytique de l'apprentissage s'intéressent à la manière dont un groupe de personnes résout un problème donné (Goggins et al., 2015), à partir des traces d'utilisation de la plateforme et de l'utilisation d'une plateforme de discussion.

6.2 Perspectives

6.2.1 Extraction de q-matrice automatique

Pour nos expériences, nous avons dû extraire des q-matrices automatiquement. La méthode que nous avons testée a fourni de bons résultats, mais ce domaine reste à explorer : en effet, les q-matrices que nous avons obtenues n'ont aucune garantie d'être les meilleures q-matrices possibles et sont parfois difficilement interprétables.

6.2.2 Tester différentes initialisations des modèles de tests adaptatifs

Nous nous sommes rendus compte que considérer un a priori dans le modèle DINA (par exemple, « 86 % des apprenants ont pour état latent `0010` ») fournissait de moins bons résultats qu'un a priori uniforme. C'est une piste de recherche à explorer.

De même, dans GenMA nous considérons qu'au démarrage du test, l'apprenant est de niveau 0. Il serait intéressant de supposer qu'il est de niveau moyen au sein de la population.

6.2.3 Différents noyaux pour InitialD

Dans notre expérience menée à la section 5.2 page 96, nous avons considéré un noyau linéaire. Il serait intéressant de tester d'autres noyaux : noyau gaussien, ou d'autres noyaux. Peut-être permettraient-ils de déterminer un meilleur ensemble de k questions.

6.2.4 Largeur optimale du prétest non adaptatif

Nous pourrions déterminer pour un jeu de données quelle valeur de k permet d'obtenir le meilleur diagnostic initial avant de procéder à un test adaptatif (voir figure 5.5 page 104).

6.2.5 Généralisation de la théorie de la réponse à l'item multidimensionnelle

La théorie de la réponse à l'item est basée sur un produit scalaire entre les caractéristiques de l'apprenant $\boldsymbol{\theta}_i$ et les caractéristiques des questions $\mathbf{d}_j$. Or, on pourrait généraliser cela en appliquant la méthode du noyau, comme dans les machines à vecteur de support.

Faire cela aurait sans doute un meilleur pouvoir prédictif, mais l'on perdrait l'interprétation du diagnostic, car les caractéristiques estimées de l'apprenant ne correspondraient plus à des degrés de maîtrise ou des lacunes. Ainsi, on pourrait repérer des apprenants susceptibles d'avoir des lacunes, mais on ne pourrait pas leur expliquer pourquoi. Cela peut être utile dans certaines applications où un enseignant doit détecter les apprenants qui ont besoin d'aide.

6.2.6 Prendre en compte la progression de l'apprenant pendant le test

La théorie de la réponse à l'item suppose l'indépendance locale entre les questions : les réponses aux questions sont indépendantes, conditionnellement à son niveau. Cela suppose, entre autres, que chaque apprenant répond de la même manière aux questions peu importe l'ordre dans lequel on les pose, et que le niveau de l'apprenant reste le même tout au long du test. Cette hypothèse est raisonnable dans le cadre d'un test de positionnement au début d'un cours ou d'un test rapide de diagnostic de connaissances en mi-parcours. Toutefois, dans les systèmes de tuteurs intelligents on observe plutôt des modèles temporels, où le niveau de l'apprenant peut évoluer alors qu'il accomplit des tâches : bandits à plusieurs bras (Clement et al., 2015), *Bayesian Knowledge Tracing* (Koedinger, McLaughlin, et Stamper, 2012), modèles de Markov cachés. Par exemple, Clement et al. (2015) tentent de maximiser le progrès de l'apprenant alors que des questions lui sont posées, avec des méthodes de bandit qui résolvent un compromis entre exploration des connaissances de l'apprenant et exploitation de ces données pour faire progresser l'apprenant. Ces modèles proviennent de l'apprentissage par renforcement, issu de l'apprentissage automatique, et nous pourrions nous en inspirer pour proposer des tests adaptatifs dans des systèmes de tuteurs intelligents.

6.2.7 Incorporer des informations supplémentaires sur les questions et les apprenants

Plus le système a de données sur les apprenants, meilleures sont les prédictions. Ainsi il serait intéressant d'intégrer d'autres informations de l'apprenant dans son profil : à l'aide de ces données supplémentaires, les caractéristiques des apprenants seraient enrichies, et les modèles potentiellement plus prédictifs. Toutefois, le système risque de favoriser une classe d'apprenants plutôt qu'une autre, ce qui pose des problèmes d'impact disparate, c'est-à-dire de discrimination involontaire (Feldman et al., 2015). Si par exemple, le système enregistre qu'en Norvège, le score en algèbre est plus faible que dans les autres pays et administre un test à un nouvel apprenant norvégien, il risque de ne pas poser de questions en algèbre et d'avoir

un a priori négatif sur l'apprenant pour cette composante de connaissances.

Pour s'attaquer au problème de démarrage à froid de l'apprenant, on peut incorporer des liens entre les questions comme ce que font Van den Oord, Dieleman, et Schrauwen (2013) sur des musiques. Par exemple, on pourrait faire de la fouille de données sur les mots de l'énoncé, ou bien accoler des mots-clés aux questions, qui seraient un premier moyen d'aller vers une q-matrice. Toutes ces données sont autant de caractéristiques qui permettront d'enrichir le système, c'est-à-dire de déterminer des vecteurs de plus grande dimension.

6.2.8 Considérer une représentation plus riche du domaine

Dans cette thèse, nous avons considéré des q-matrices comme représentation du domaine évalué par le test. Certaines approches utilisent des représentations plus riches telles que des ontologies, sur lesquelles il est possible de faire des inférences (Mandin et Guin, 2014). Ce genre de structure est difficile à construire, et surtout à combiner. Nous apprécions l'approche q-matrice, qui permet à plusieurs experts de partager et de combiner leur représentation du domaine (Koedinger, McLaughlin, et Stamper, 2012). Il y a donc un compromis entre l'information apportée en tant que représentation du domaine évalué, et la facilité de mise en œuvre du test.

De telles représentations plus riches permettraient d'administrer un test sans besoin de données précédentes. C'est une piste de recherche à explorer.

6.2.9 Incorporer des générateurs automatiques d'exercices

Il est parfois utile de répéter les questions, par exemple lorsqu'on apprend du vocabulaire au moyen de cartes de support visuel, et de systèmes par répétitions espacées (Altiner, 2011). Notre approche correspond davantage à l'apprentissage d'une compétence plutôt que d'un item particulier. Aussi, ne pas poser la même question plusieurs fois mais plutôt des variantes, par exemple en mathématiques en changeant l'opération à effectuer, permet de réduire le risque que l'apprenant devine la bonne réponse. Il serait intéressant de coupler cela avec des générateurs automatiques d'exercices (Cablé, Guin, et Lefevre, 2013) : à partir d'un système pouvant générer un exercice à partir des CC que l'on souhaite évaluer, la plateforme pourrait diversifier les questions qu'elle pose et les apprenants pourraient demander de nouveaux exercices jusqu'à ce qu'ils maîtrisent les CC. Cela conviendrait au modèle DINA, mais pas aux modèles GenMA et Rasch qui ont besoin d'un historique de passage sur un item pour fonctionner.

6.2.10 Incorporer des systèmes de recommandation de ressources

Les tests adaptatifs permettent une meilleure personnalisation en organisant les ressources d'apprentissage. Le séquençage du programme (*curriculum sequencing*) consiste à définir des parcours d'apprentissage dans un espace d'objectifs d'apprentissage (Desmarais et R. S. J. D. Baker, 2012). Cela consiste à faire passer des évaluations de compétences pour adapter le contenu d'apprentissage à partir d'un minimum de faits observés. Par exemple, à l'issue d'un diagnostic de connaissances, le système a une idée plus précise des composantes de connaissances que maîtrise l'apprenant et peut ainsi filtrer le contenu du cours qui lui sera utile pour s'améliorer.

Si l'on disposait d'une base de données faisant le lien entre des ressources éducatives et les composantes de connaissances sur lesquelles elles portent, il serait possible d'orienter un apprenant vers des ressources utiles, à partir des lacunes identifiées par notre système après un passage de test adaptatif. Si de plus, les ressources étaient évaluées par d'autres apprenants sous la forme de retours du type : « Cette ressource m'a été utile / ne m'a pas été utile pour comprendre », on se ramènerait à la conception d'un système de recommandation.

Les sites d'e-commerce font la distinction entre le retour explicite donné par un apprenant, par exemple « J'ai apprécié ce produit » et le retour implicite, tel que « Cette personne est restée longtemps sur cette page », ce qui peut indiquer qu'il est intéressé par son contenu. Ainsi, ces données d'utilisation sont traitées par les plateformes d'e-commerce à l'insu des utilisateurs pour mieux connaître leurs clients. Dans des plateformes d'apprentissage en ligne, il y a rarement de retour explicite (Verbert et al., 2011), donc ces techniques de retour implicite sont des méthodes d'intérêt pour améliorer la pertinence des recommandations de ressources d'apprentissage, en utilisant par exemple le temps passé sur une page, les recherches d'un apprenant, la liste des ressources téléchargées et les commentaires postés. Pour l'évaluation des apprenants, on pourrait imaginer qu'après avoir résolu correctement une question, l'apprenant reçoit une question supplémentaire : « Pour résoudre cette question, vous semblez avoir consulté les ressources suivantes : [...] Lesquelles vous ont été utiles pour résoudre cette question ? » Ces données peuvent aider à recommander des ressources utiles à de futurs apprenants ayant des difficultés pour résoudre ces questions.

6.2.11 Considérer des interfaces plus riches pour l'évaluation

Dans cette thèse, nous avons considéré que les données d'évaluations des apprenants étaient sous la forme de succès ou échecs d'apprenants sur des ques-

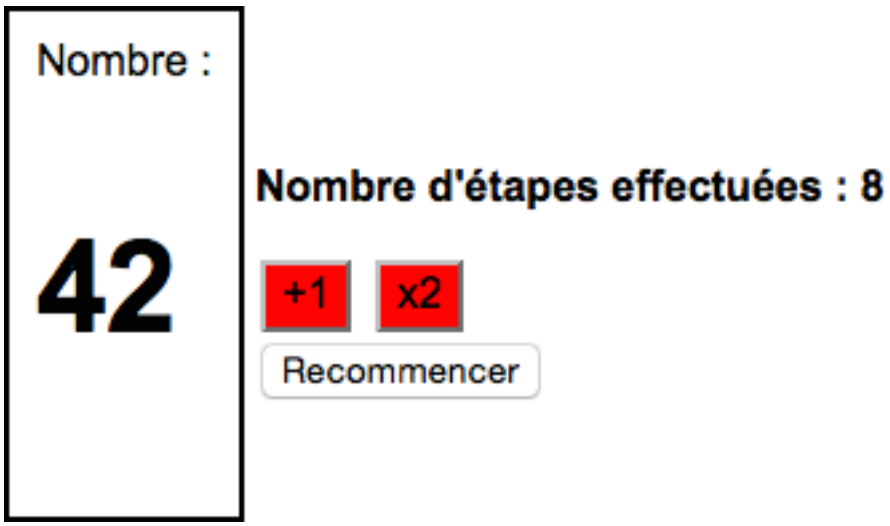


Figure 6.1 – Exemple d'interface d'évaluation tirée du concours Castor.

tions. Cela comprend les tests comportant des questions à choix multiples, ou bien des questions à réponse ouverte courte. Mais nous supposons que nos modèles peuvent être réutilisés tels quels pour l'analyse de données d'évaluations d'apprenants ayant résolu correctement ou non des tâches, telles que des niveaux de jeux sérieux, ou des interfaces plus complexes comme celles que l'on peut trouver dans le concours d'informatique Castor. Un exemple est donné à la figure 6.1 : le problème à résoudre consiste à construire le nombre 51 en partant de zéro à l'aide d'une calculette qui ne contient que les boutons $+1$ et $\times 2$. Les examinés doivent aboutir au résultat en appuyant sur un nombre minimum de boutons pour obtenir tous les points. Tant qu'ils n'ont pas déterminé le nombre optimal, l'interface les invite à recommencer, sachant que chaque essai infructueux ne fait pas perdre de point. Cette activité très ludique est bénéfique pour l'apprentissage, car la tâche à résoudre est bien définie et l'examiné peut soit jouer avec l'interface, soit tenter de résoudre le problème sur papier. Lorsque l'examiné a résolu l'exercice, un texte lui explique qu'il vient de faire de l'informatique au moyen d'un paragraphe (en l'occurrence, pour cet exercice, il s'agit du système binaire).

Ces tâches, paramétrisables (dans cet exemple, on peut demander d'abord à l'apprenant de construire le nombre 5), peuvent être liées à des composantes de connaissances, et à des notions de difficulté. On peut considérer que l'apprenant réussit la tâche s'il a obtenu le score maximal, 0 sinon, et ainsi se ramener aux hypothèses que nous avons considérées pendant cette thèse.

Il est également possible d'enregistrer tous les essais infructueux de l'apprenant pour détecter s'il a eu du mal à trouver la bonne réponse, et choisir l'activité suivante à lui présenter en conséquence. Nous aimerions explorer cette piste de

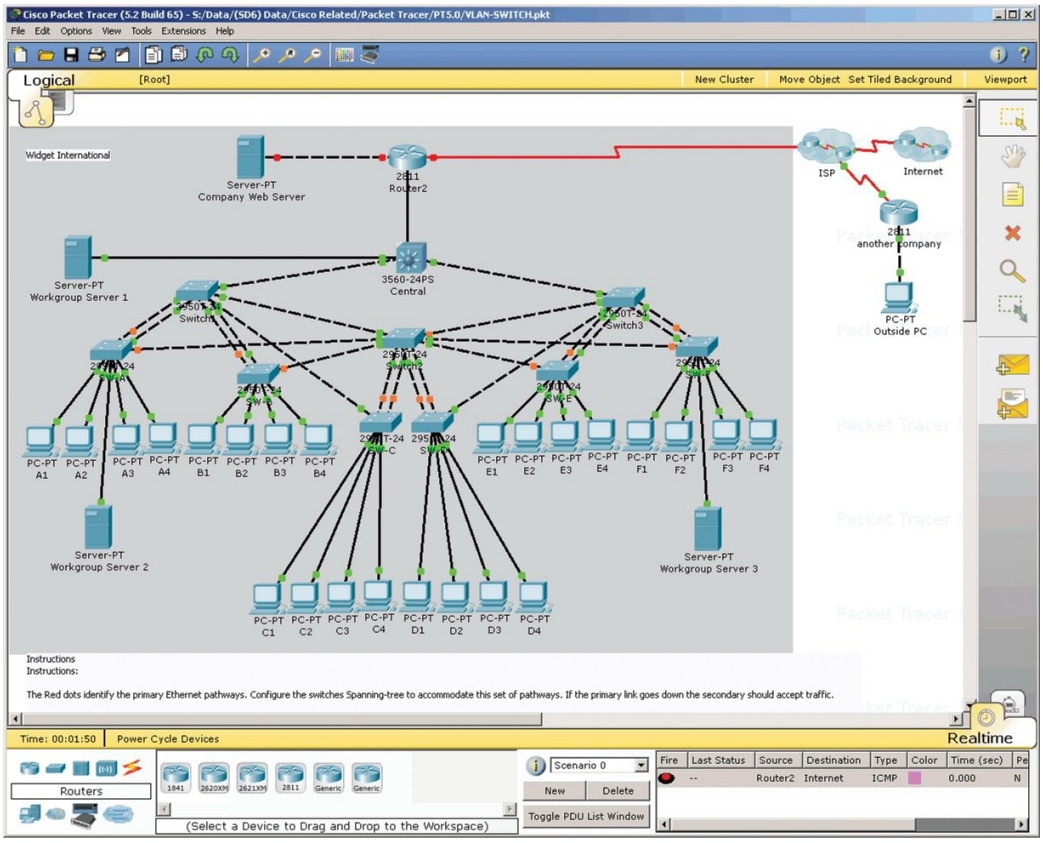


Figure 6.2 – Capture d'écran de Packet Tracer, développé par Cisco.

recherche.

6.2.12 Évaluation furtive dans les jeux sérieux

Shute (2011) décrit la notion d'évaluation furtive. Le principe est de collecter les données d'une plateforme éducative alors que des apprenants sont en train de l'utiliser, sous la forme d'actions et de leurs résultats, et de faire des inférences sur le niveau de l'apprenant selon les différentes compétences associées à ces actions. On peut citer par exemple le Packet Tracer de Cisco étudié par Rupp et al. (2012) (voir figure 6.2) permettant de comprendre par la pratique comment fonctionne le routage des paquets dans un réseau ou *Newton's Playground* (Shute, Ventura, et Kim, 2013), où les apprenants jouent avec une interface leur faisant découvrir des notions de physique. L'approche est basée sur deux éléments principaux : la conception d'une évaluation centrée sur les faits, et le retour fait à l'apprenant pour soutenir l'apprentissage.

Shute (2011) insiste sur la capacité de l'évaluation furtive via les jeux vidéo à

entretenir le flux (*flow*), c'est-à-dire l'état mental atteint par une personne lorsqu'elle est complètement absorbée par une activité, telle qu'une résolution de problème, et se trouve dans un état maximal de concentration, de plein engagement et de satisfaction dans son accomplissement. Ils motivent leur recherche par le fait qu'aujourd'hui, les problèmes auxquels nous sommes confrontés nécessitent de réfléchir de façon créative, critique, collaborative et systémique. Ainsi, comme dit Shute (2011), « apprendre et réussir dans un monde complexe et dynamique ne peut être facilement mesuré par un test de connaissances composé de questions à choix multiple. »Ils proposent plutôt de repenser la notion d'évaluation, en identifiant les *compétences clés du 21e siècle* et les façons d'évaluer leur acquisition par les apprenants.

Évaluation centrée sur les faits et réseaux bayésiens

La conception d'une évaluation centrée sur les faits a été formalisée par Mislevy et al. (2012). Les lignes principales sont les suivantes.

Modèle de compétences Quelles connaissances et compétences sont censées être évaluées ?

Modèle des faits Quels comportements ou performance devraient révéler ces constructions ?

Modèle des tâches Quelles tâches devraient éliciter ces comportements qui constituent les faits observés ?

Afin de modéliser les liens entre compétences et faits, Shute (2011) utilise des réseaux bayésiens. Nous pensons toutefois que cette approche est coûteuse à construire et doit être reproduite pour chaque nouvelle interface d'évaluation. De plus, elle impose de fixer a priori un modèle d'évaluation auquel un apprenant pourrait ne pas adhérer. Toutefois, cela permet de proposer un diagnostic alors même que l'on ne dispose d'aucun historique d'évaluation : le diagnostic n'a besoin que du modèle de compétences pour fonctionner. Rupp et al. (2012) comparent, pour le Packet Tracer, une méthode de diagnostic utilisant des réseaux bayésiens avec une approche utilisant le modèle de hiérarchie sur les attributs décrit à la section 2.3.2 page 37 qui requiert une q-matrice, et montre que les approches sont complémentaires.

Nous aimerions déterminer si notre approche peut s'appliquer dans ce cadre où il y a plusieurs moyens d'aboutir à la bonne réponse, et que deux chemins de résolution requièrent des composantes de connaissances différentes.

6.3 Le futur de l'évaluation

Une application prometteuse de notre recherche est la conception d'autoévaluation formative adaptative : avant les évaluations à fort enjeu en fin de cours, les apprenants aiment s'entraîner à passer des tests, afin d'avoir une idée de ce qui sera attendu d'eux. Cela a d'ailleurs un effet bénéfique sur l'apprentissage (Dunlosky et al., 2013). Concevoir des exercices ou spécifier des valeurs de difficulté ou de discrimination est coûteux pour l'enseignant, mais nous avons vu dans cette thèse différentes méthodes pour automatiser ces processus : s'aider de l'historique de passage d'un test pour calibrer automatiquement des paramètres de discrimination, générer des fiches d'exercices diversifiées.

Les tests que nous proposons ne supposent aucune connaissance sur l'apprenant, ainsi les résultats peuvent être enregistrés de façon anonyme, ce qui permet à l'apprenant de faire table rase et de ne pas craindre que ses erreurs le poursuivent tout au long de la vie, ce qui est une crainte de nombreux parents d'élèves aujourd'hui (Executive Office of the President et Podesta, 2014). Nos modèles de tests permettent à l'apprenant d'avoir une photographie de ses connaissances à un certain moment, et un diagnostic afin de l'aider à se positionner sur une carte de compétences et comprendre ce qu'il aurait intérêt à apprendre ensuite.

Dans la communauté de l'analytique de l'apprentissage, certains argumentent qu'à partir du moment où une même plateforme enregistrera tout ce qu'on fait, des tests explicites disparaîtront au profit d'une *évaluation intégrée*, un contrôle continu automatique utilisant les données à disposition pour prédire la performance de l'étudiant et proposer une éducation sur mesure (Shute, Leighton, et al., 2016 ; Redecker et Johannessen, 2013). En effet, si l'apprenant est continuellement observé par la plateforme et si un tuteur numérique peut répondre à ses questions et recommander des activités (Korn, 2016), ils peuvent être les seuls acteurs de leur progrès et il n'y a plus besoin d'évaluer leurs connaissances à la fin du cours.

Nous pensons toutefois que même dans un futur où l'évaluation intégrée est possible, il y aura toujours besoin de tests adaptatifs, pour un usage ponctuel tel qu'un test standardisé (GMAT, GRE) ou pour les nouveaux apprenants au début d'un cours, dans la mesure où les plateformes en ligne seront amenées à avoir des apprenants de tous âges et profils, et qu'il faudra avoir en peu de temps une idée des connaissances qu'ils ont emmagasinées dans leur expérience (R. S. Baker et Inventado, 2014 ; Lynch et Howlin, 2014). Une comparaison peut être faite avec les tests de positionnement en langues dans les grandes écoles, qui consistent à répartir les apprenants en différents groupes de niveaux.

Dans le futur, une plateforme pourra d'abord demander à l'apprenant les composantes de connaissances (CC) qu'il pense maîtriser. À partir de ces informations, la plateforme pourra lancer un test adaptatif, à l'issue duquel elle pourra lui faire un retour de type : « Vous avez dit que vous maîtrisiez cette CC pourtant j'ai posé

cette question et vous n'y avez pas répondu correctement. » Ainsi, la plateforme pourra argumenter sur ce désaccord. L'apprenant pourra à son tour rectifier le diagnostic en prouvant à la plateforme qu'il maîtrise effectivement cette CC, et ainsi de suite. Nous pensons qu'il devrait y avoir plus de recherche en analytique de l'apprentissage pour de tels systèmes, plus interactifs.

Après un test, la plateforme peut fournir à l'apprenant un diagnostic composé de points à retravailler. Mais l'apprenant peut décider de vouloir aller plus loin dans ce qu'il maîtrise. La plateforme pourra alors lui proposer un parcours pédagogique en prenant en compte ce que l'apprenant doit renforcer et ce qu'il souhaite apprendre. Il faut trouver le bon équilibre entre permettre à l'apprenant de naviguer dans le cours sans le décourager devant l'immensité de ce qu'il peut apprendre. Afin que l'apprenant puisse être maître de son apprentissage, la plateforme doit lui redonner du contrôle en étant plus transparente sur les déductions qu'elle fait à partir de ses réponses aux tests.

Aujourd'hui, l'analytique de l'apprentissage ne se concentre pas que sur une adaptation automatique mais aussi des moyens de renforcer la motivation des apprenants. Cela pose des questions ouvertes quant au partage des informations que la plateforme emmagasine sur l'apprenant. Une autre question est de savoir à quel point la plateforme devrait communiquer à l'apprenant les informations qu'elle déduit de son comportement. Un avantage serait la confiance que l'apprenant a dans le système, mais un risque serait que les apprenants modifient leur comportement en conséquence de façon à truquer et contourner le système.

Table des figures

Liste des tableaux

Annexe A

Implémentation des modèles

Tout le code pour effectuer la comparaison quantitative de modèles est disponible sur GitHub à l'adresse `http://github.com/jilljenn/qna`.

A.1 Modèles de tests adaptatifs

Certains modèles sont implémentés en R, d'autres en Python. Nous avons donc utilisé le package RPy2 (Gautier, 2008) qui permet d'appeler des fonctions R avec du code Python, afin de pouvoir comparer tous les modèles avec du code générique.

Rasch `irt.py` Le modèle de Rasch est implémenté au moyen des packages `ltm` (Rizopoulos, 2006) (pour *latent trait models*) et `catR` (Magis et Raîche, 2012) (pour *computerized adaptive testing in R*).

DINA `qmatrix.py` Nous avons implémenté le modèle DINA en Python. Au départ, nous utilisions le package `CDM` (Robitzsch et al., 2014) (pour *cognitive diagnosis modeling*) pour estimer les paramètres d'inattention et de chance, mais nous avons finalement implémenté notre propre code de calibration, après avoir reconnu qu'il s'agissait d'un problème d'optimisation convexe. Afin d'accélérer la procédure d'entraînement parfois coûteuse, nous utilisons `pypy` : il s'agit d'un interpréteur Python qui compile le code à la volée en code machine, afin de fournir une exécution plus rapide de code Python. Pour l'utiliser, il suffit de taper `pypy fichier.py` au lieu de `python fichier.py`.

MIRT et GenMA `mirt.py` Les modèles MIRT et GenMA sont implémentés au moyen des packages `mirt` (R. Philip Chalmers, 2012) et `mirtCAT` (R. P. Chalmers, 2015).

A.2 Comparaison quantitative

Modèles

Le code de validation bicroisée est implémenté en Python. Un fichier de configuration `conf.py` permet de spécifier le jeu de données sur lequel effectuer l'expérience, ainsi que des paramètres tels que le nombre de paquets d'apprenants et le nombre de paquets de questions pour lancer la validation bicroisée définie à la section 3.4.3 page 51.

Constitution des paquets Un fichier `subset.py` construit les paquets d'apprenants et de questions pour la validation bicroisée.

Exécution des modèles Pour faire tourner un modèle, il faut appeler `python cat.py MODEL` où `MODEL` désigne `irt` pour Rasch, `qmspe` pour DINA avec une q-matrice spécifiée par un expert, `qm` pour DINA avec une extraction automatique de q-matrice, `mirt D` pour MIRT de dimension `D`, et enfin `mirtq` pour GenMA avec une q-matrice spécifiée par un expert.

Évaluation des performances À la fin, `combine.py` permet de combiner toutes les expériences effectuées, `stats.py` calcule les taux d'erreurs et `plot.py FOLDER TYPE` trace la courbe de type `TYPE` pour l'expérience `FOLDER`, où `TYPE` désigne `mean` si l'on souhaite tracer la *log loss*, `count` si l'on souhaite calculer le nombre de prédictions incorrectes ou `delta`, la distance au diagnostic final comme définie à la section 5.4.3 page 100.

Stratégies

Toutes les stratégies de choix de k questions notamment InitialD sont implémentées dans le fichier `coldstart.py`.

Bibliographie

Aleven, Vincent, Bruce M. McLaren, Jonathan Sewall, Martin van Velsen, Octav Popescu, Sandra Demi, Michael Ringenberg, and Kenneth R. Koedinger (2016). "Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers". In: *International Journal of Artificial Intelligence in Education* 26.1, pp. 224–269. ISSN: 1560-4306. DOI: 10.1007/s40593-015-0088-2. URL: http://dx.doi.org/10.1007/s40593-015-0088-2 (cit. on p. 23).

Altiner, Cennet (2011). "Integrating a computer-based flashcard program into academic vocabulary learning". MA thesis. Iowa State University (cit. on pp. 19, 111).

Anava, Oren, Shahar Golan, Nadav Golbandi, Zohar Karnin, Ronny Lempel, Oleg Rokhlenko, and Oren Somekh (2015). "Budget-Constrained Item Cold-Start Handling in Collaborative Filtering Recommenders via Optimal Design". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 45–54 (cit. on p. 104).

Baker, Ryan Shaun and Paul Salvador Inventado (2014). "Educational data mining and learning analytics". In: *Learning Analytics*. Springer, pp. 61–75 (cit. on p. 116).

Barnes, Tiffany (2005). "The q-matrix method: Mining student response data for knowledge". In: *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop* (cit. on p. 36).

Bartholomew, David J., Fiona Steele, Jane Galbraith, and Irini Moustaki (2008). *Analysis of multivariate social science data*. CRC press (cit. on p. 29).

Bergner, Yoav, Kimberly Colvin, and David E. Pritchard (2015). "Estimation of ability from homework items when there are missing and/or multiple attempts". In: *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. ACM, pp. 118–125 (cit. on p. 67).

Bergner, Yoav, Stefan Droschler, Gerd Kortemeyer, Saif Rayyan, Daniel Seaton, and David E. Pritchard (2012). "Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory." In: *International Educational Data Mining Society* (cit. on pp. 27, 29, 39, 42).

Cablé, Baptiste, Nathalie Guin, et Marie Lefevre (2013). « Un outil auteur pour une génération semi-automatique d'exercices d'auto-évaluation ». In : *6e Conférence sur les Environnements Informatiques pour l'Apprentissage Humain*, p. 155 (cf. p. 111).

Cai, Li (2010). "High-dimensional exploratory item factor analysis by a Metropolis–Hastings Robbins–Monro algorithm". In: *Psychometrika* 75.1, pp. 33–57 (cit. on p. 84).

Chalmers, R. P. (2015). *mirtCAT: computerized adaptive testing with multidimensional item response theory. R package version 0.6. 1* (cit. on p. 123).

Chalmers, R. Philip (2012). "mirt: A multidimensional item response theory package for the R environment". In: *Journal of Statistical Software* 48.6, pp. 1–29 (cit. on pp. 58, 84, 123).

— (2016). "Generating Adaptive and Non-Adaptive Test Interfaces for Multidimensional Item Response Theory Applications". In: *Journal of Statistical Software* 71.1, pp. 1–38 (cit. on p. 93).

Chang, Hua-Hua (2014). "Psychometrics Behind Computerized Adaptive Testing". In: *Psychometrika*, pp. 1–20 (cit. on pp. 15, 16, 26, 36, 55).

Chatti, Mohamed Amine, Anna Lea Dyckhoff, Ulrik Schroeder, and Hendrik Thüs (2012). "A reference model for learning analytics". In: *International Journal of Technology Enhanced Learning* 4.5-6, pp. 318–331 (cit. on pp. 23, 25, 26, 39).

Chen, Suming, Arthur Choi, and Adnan Darwiche (2015). "Computer Adaptive Testing Using the Same-Decision Probability". In: *12th Annual Bayesian Modeling Applications Workshop (BMAW)* (cit. on p. 26).

Cheng, Ying (2009). "When cognitive diagnosis meets computerized adaptive testing: CD-CAT". In: *Psychometrika* 74.4, pp. 619–632 (cit. on pp. 35, 42, 72).

Clement, Benjamin, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2015). "Multi-Armed Bandits for Intelligent Tutoring Systems". In: *JEDM-Journal of Educational Data Mining* 7.2, pp. 20–48 (cit. on pp. 41, 49, 66, 110).

Davier, Matthias (2005). "A general diagnostic model applied to language testing data". In: *ETS Research Report Series* 2005.2, pp. i–35 (cit. on p. 82).

DeCarlo, Lawrence T. (2010). "On the analysis of fraction subtraction data: The DINA model, classification, latent class sizes, and the Q-matrix". In: *Applied Psychological Measurement* (cit. on pp. 33, 57).

Desmarais, Michel C. et al. (2011). "Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization". In: *4th International Conference on Educational Data Mining, EDM*, pp. 41–50 (cit. on pp. 37, 56, 61, 72).

Desmarais, Michel C. and Ryan S. J. D. Baker (2012). "A review of recent advances in learner and skill modeling in intelligent learning environments". In: *User Modeling and User-Adapted Interaction* 22.1-2, pp. 9–38 (cit. on pp. 15, 25, 29, 31, 38, 42, 112).

Doignon, Jean-Paul and Jean-Claude Falmagne (2012). *Knowledge spaces*. Springer Science & Business Media (cit. on pp. 35, 37).

Dunlosky, John, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham (2013). "Improving students' learning with effective learning techniques promising directions from cognitive and educational psychology". In: *Psychological Science in the Public Interest* 14.1, pp. 4–58 (cit. on pp. 66, 116).

Executive Office of the President and John Podesta (2014). *Big data: seizing opportunities, preserving values*. Tech. rep. The White House (cit. on pp. 18, 116).

Falmagne, Jean-Claude, Eric Cosyn, Jean-Paul Doignon, and Nicolas Thiéry (2006). "The assessment of knowledge, in theory and in practice". In: *Formal concept analysis*. Springer, pp. 61–79 (cit. on pp. 38, 66).

Feldman, Michael, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian (2015). "Certifying and removing disparate impact". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 259–268 (cit. on p. 110).

Ferguson, Rebecca (2012). "Learning analytics: drivers, developments and challenges". In: *International Journal of Technology Enhanced Learning* 4.5-6, pp. 304–317 (cit. on p. 24).

Gautier, L. (2008). "rpy2: A simple and efficient access to R from Python". In: URL: http://rpy.sourceforge.net/rpy2.html (cit. on p. 123).

Goggins, Sean P., Wanli Xing, Xin Chen, Bodong Chen, and Bob Wadholm (2015). "Learning Analytics at" Small" Scale: Exploring a Complexity-Grounded Model for Assessment Automation." In: *J. UCS* 21.1, pp. 66–92 (cit. on p. 108).

Golbandi, Nadav, Yehuda Koren, and Ronny Lempel (2011). "Adaptive bootstrapping of recommender systems using decision trees". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 595–604 (cit. on p. 40).

Hambleton, Ronald K. and Hariharan Swaminathan (1985). *Item response theory: Principles and applications*. Vol. 7. Springer Science & Business Media (cit. on pp. 16, 23).

Han, Kyung T. (2013). "Item Pocket Method to Allow Response Review and Change in Computerized Adaptive Testing." In: *Applied Psychological Measurement* 37.4, pp. 259–275 (cit. on p. 27).

Hoi, Steven C. H., Rong Jin, Jianke Zhu, and Michael R. Lyu (2006). "Batch mode active learning and its application to medical image classification". In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 417–424 (cit. on p. 96).

Huebner, Alan (2010). "An Overview of Recent Developments in Cognitive Diagnostic Computer Adaptive Assessments." In: *Practical Assessment, Research & Evaluation* 15.3, p. 7 (cit. on pp. 24, 35).

Kickmeier-Rust, Michael D. and Dietrich Albert (2015). "Competence-Based Knowledge Space Theory". In: *Measuring and Visualizing Learning in the Information-Rich Classroom*, p. 109 (cit. on pp. 38, 39).

Koedinger, Kenneth R., Elizabeth A. McLaughlin, and John C. Stamper (2012). "Automated Student Model Improvement." In: *International Educational Data Mining Society* (cit. on pp. 37, 110, 111).

Korn, M. (2016). "Imagine Discovering that your Teaching Assistant is Really a Robot. The Wall Street Journal". In: *The Wall Street Journal*. Accédé le 4 octobre 2016. URL: http://www.wsj.com/articles/if-your-teacher-sounds-like-a-robot-you-might-be-on-to-something-1462546621 (cit. on p. 116).

Kulesza, Alex and Ben Taskar (2012). "Determinantal point processes for machine learning". In: *arXiv preprint arXiv:1207.6083* (cit. on pp. 96–98, 108).

Lallé, Sébastien (2013). « Assistance à la construction et à la comparaison de techniques de diagnostic des connaissances ». Thèses. Université de Grenoble. URL : https://tel.archives-ouvertes.fr/tel-01135183 (cf. p. 42).

Lan, Andrew S., Christoph Studer, Andrew E. Waters, and Richard G. Baraniuk (2014). "Tag-aware ordinal sparse factor analysis for learning and content analytics". In: *arXiv preprint arXiv:1412.5967* (cit. on p. 49).

Lan, Andrew S., Andrew E. Waters, Christoph Studer, and Richard G. Baraniuk (2014). "Sparse factor analysis for learning and content analytics". In: *The Journal of Machine Learning Research* 15.1, pp. 1959–2008 (cit. on pp. 26, 27, 31, 32, 42, 96, 100).

Lee, Seokho, Jianhua Z. Huang, and Jianhua Hu (2010). "Sparse logistic principal components analysis for binary data". In: *The annals of applied statistics* 4.3, pp. 1579–1601 (cit. on p. 51).

Leighton, Jacqueline P., Mark J. Gierl, and Stephen M. Hunka (2004). "The Attribute Hierarchy Method for Cognitive Assessment: A Variation on Tatsuoka's Rule-Space Approach". In: *Journal of Educational Measurement* 41.3, pp. 205–237 (cit. on p. 37).

Linden, Wim J. van der and Cees A. W. Glas (2010). *Elements of adaptive testing*. Springer (cit. on p. 26).

Lynch, Danny and Colm P. Howlin (2014). *Real world usage of an adaptive testing algorithm to uncover latent knowledge* (cit. on pp. 25–27, 38, 116).

Magis, David (2015). "Empirical comparison of scoring rules at early stages of CAT". In: (cit. on p. 96).

Magis, David and Gilles Raîche (2012). "Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package catR". In:

Journal of Statistical Software 48.8, pp. 1–31. ISSN: 1548-7660. URL: http://www.jstatsoft.org/v48/i08 (cit. on p. 123).

Mandin, Sonia and Nathalie Guin (2014). "Basing learner modelling on an ontology of knowledge and skills". In: *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*. IEEE, pp. 321–323 (cit. on pp. 39, 111).

Manouselis, Nikos, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper (2011). "Recommender systems in technology enhanced learning". In: *Recommender systems handbook*. Springer, pp. 387–415 (cit. on pp. 26, 39).

Mislevy, Robert J., John T. Behrens, Kristen E. Dicerbo, and Roy Levy (2012). "Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining". In: *JEDM-Journal of Educational Data Mining* 4.1, pp. 11–48 (cit. on pp. 27, 115).

Papamitsiou, Zacharoula K., Vasileios Terzis, and Anastasios A. Economides (2014). "Temporal learning analytics for computer based testing". In: *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*. ACM, pp. 31–35 (cit. on p. 26).

Peña-Ayala, Alejandro (2014). "Educational data mining: A survey and a data mining-based analysis of recent works". In: *Expert systems with applications* 41.4, pp. 1432–1462 (cit. on p. 27).

Reckase, Mark (2009). *Multidimensional item response theory*. Vol. 150. Springer (cit. on p. 31).

Redecker, Christine and Øystein Johannessen (2013). "Changing assessment—Towards a new assessment paradigm using ICT". In: *European Journal of Education* 48.1, pp. 79–96 (cit. on p. 116).

Rizopoulos, Dimitris (2006). "ltm: An R Package for Latent Variable Modeling and Item Response Analysis". In: *Journal of Statistical Software* 17.5, pp. 1–25. ISSN: 1548-7660. URL: http://www.jstatsoft.org/v17/i05 (cit. on p. 123).

Robitzsch, A., T. Kiefer, A. C. George, and A. Ünlü (2014). "CDM: Cognitive diagnosis modeling". In: *R Package version* 3 (cit. on p. 123).

Rupp, A., Roy Levy, Kristen E. Dicerbo, Shauna J. Sweet, Aaron V. Crawford, Tiago Calico, Martin Benson, Derek Fay, Katie L. Kunze, Robert J. Mislevy, et al. (2012). "Putting ECD into practice: The interplay of theory and data in evidence models within a digital learning environment". In: *JEDM-Journal of Educational Data Mining* 4.1, pp. 49–110 (cit. on pp. 37, 39, 72, 114, 115).

Shute, Valerie J. (2011). "Stealth assessment in computer-based games to support learning". In: *Computer games and instruction* 55.2, pp. 503–524 (cit. on pp. 39, 114, 115).

Shute, Valerie J., Jacqueline P. Leighton, Eunice E. Jang, and Man-Wai Chu (2016). "Advances in the science of assessment". In: *Educational Assessment* 21.1, pp. 34–59 (cit. on pp. 24, 116).

Shute, Valerie J., Matthew Ventura, and Yoon Jeon Kim (2013). "Assessment and learning of qualitative physics in newton's playground". In: *The Journal of Educational Research* 106.6, pp. 423–430 (cit. on p. 114).

Su, Yu-Law, K. M. Choi, W. C. Lee, T. Choi, and M. McAninch (2013). "Hierarchical cognitive diagnostic analysis for TIMSS] 2003 mathematics". In: *Centre for Advanced Studies in Measurement and Assessment* 35, pp. 1–71 (cit. on pp. 36, 57, 60).

Thai-Nghe, Nguyen, Lucas Drumond, Tomáš Horváth, Lars Schmidt-Thieme, et al. (2011). "Multi-relational factorization models for predicting student performance". In: *Proc. of the KDD Workshop on Knowledge Discovery in Educational Data*. Citeseer (cit. on pp. 27, 39, 40).

Toscher, A. and Michael Jahrer (2010). "Collaborative filtering applied to educational data mining". In: *KDD cup* (cit. on p. 39).

Ueno, Maomi and Pokpong Songmuang (2010). "Computerized adaptive testing based on decision tree". In: *2010 10th IEEE International Conference on Advanced Learning Technologies*. IEEE, pp. 191–193 (cit. on p. 55).

Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen (2013). "Deep content-based music recommendation". In: *Advances in Neural Information Processing Systems*, pp. 2643–2651 (cit. on p. 111).

Verbert, Katrien, Hendrik Drachsler, Nikos Manouselis, Martin Wolpers, Riina Vuorikari, and Erik Duval (2011). "Dataset-driven research for improving recommender systems for learning". In: *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. ACM, pp. 44–53 (cit. on pp. 26, 39, 112).

Verhelst, Norman D. (2012). "Profile analysis: a closer look at the PISA 2000 reading data". In: *Scandinavian Journal of Educational Research* 56.3, pp. 315–332 (cit. on p. 29).

Vie, Jill-Jênn, Fabrice Popineau, Yolaine Bourda, and Éric Bruillard (2016a). "A review of recent advances in adaptive assessment". In: *Learning analytics: Fundaments, applications, and trends: A view of the current state of the art*. Springer, in press (cit. on p. 19).

— (2016b). "Adaptive Testing Using a General Diagnostic Model". In: *European Conference on Technology Enhanced Learning*. Springer, pp. 331–339 (cit. on p. 20).

Vie, Jill-Jênn, Fabrice Popineau, Jean-Bastien Grill, Éric Bruillard, and Yolaine Bourda (2015a). "Predicting Performance over Dichotomous Questions: Comparing Models for Large-Scale Adaptive Testing". In: *8th International Conference on Educational Data Mining (EDM 2015)* (cit. on p. 19).

— (2015b). « Prédiction de performance sur des questions dichotomiques : comparaison de modèles pour des tests adaptatifs à grande échelle ». In : *Atelier*

Évaluation des Apprentissages et Environnements Informatiques, EIAH 2015 (cf. p. 18).

Vygotsky, Lev Semenovich (1980). *Mind in society: The development of higher psychological processes*. Harvard university press (cit. on p. 41).

Wang, Shiyu, Georgios Fellouris, and Hua-Hua Chang (2015). "Sequential Design for Computerized Adaptive Testing that Allows for Response Revision". In: *arXiv preprint arXiv:1501.01366* (cit. on p. 27).

Wang, Shiyu, Haiyan Lin, Hua-Hua Chang, and Jeff Douglas (2016). "Hybrid Computerized Adaptive Testing: From Group Sequential Design to Fully Sequential Design". In: *Journal of Educational Measurement* 53.1, pp. 45–62 (cit. on p. 33).

Winters, Titus, Christian Shelton, Tom Payne, and Guobiao Mei (2005). "Topic extraction from item-level grades". In: *American Association for Artificial Intelligence 2005 Workshop on Educational Datamining, Pittsburgh, PA*. Vol. 1, p. 3 (cit. on pp. 37, 56).

Xu, Xueli, H. Chang, and Jeff Douglas (2003). "A simulation study to compare CAT strategies for cognitive diagnosis". In: *annual meeting of the American Educational Research Association, Chicago* (cit. on p. 35).

Yan, Duanli, Alina A. von Davier, and Charles Lewis (2014). *Computerized Multistage Testing*. CRC Press (cit. on pp. 32, 55, 76).

Zeileis, Achim, Nikolaus Umlauf, Friedrich Leisch, et al. (2012). *Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond*. Department of Economics (Inst. für Wirtschaftstheorie und Wirtschaftsgeschichte) (cit. on p. 15).

Zernike, K. (2015). "Obama Administration Calls for Limits on Testing in Schools". In: *The New York Times*. Accédé le 2 avril 2016. URL: http://www.nytimes.com/2015/10/25/us/obama-administration-calls-for-limits-on-testing-in-schools.html (cit. on pp. 15, 23).

Zou, Hui, Trevor Hastie, and Robert Tibshirani (2006). "Sparse principal component analysis". In: *Journal of computational and graphical statistics* 15.2, pp. 265–286 (cit. on p. 51).

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : **Modèles de tests adaptatifs pour le diagnostic de connaissances dans un cadre d'apprentissage à grande échelle**

Mots-clés : tests adaptatifs, diagnostic de connaissances, cours en ligne ouverts et massifs (MOOC), théorie de la réponse à l'item, q-matrice, analytique de l'apprentissage

Résumé : Cette thèse porte sur les tests adaptatifs dans les environnements d'apprentissage. Elle s'inscrit dans les contextes de fouille de données éducatives et d'analytique de l'apprentissage, où l'on s'intéresse à utiliser les données laissées par les apprenants dans des environnements éducatifs pour optimiser l'apprentissage au sens large.

L'évaluation par ordinateur permet de stocker les réponses des apprenants facilement, afin de les analyser et d'améliorer les évaluations futures. Dans cette thèse, nous nous intéressons à un certain type de test par ordinateur, les tests adaptatifs. Ceux-ci permettent de poser une question à un apprenant, de traiter sa réponse à la volée, et de choisir la question suivante à lui poser en fonction de ses réponses précédentes. Ce processus réduit le nombre de questions à poser à un apprenant tout en conservant une mesure précise de son niveau. Les tests adaptatifs sont aujourd'hui implémentés pour des tests standardisés tels que le GMAT ou le GRE, administrés à des centaines de milliers d'étudiants. Toutefois, les modèles de tests adaptatifs traditionnels se contentent de noter les apprenants, ce qui est utile pour l'institution qui évalue, mais pas pour leur apprentissage. C'est pourquoi des modèles plus formatifs ont été proposés, permettant de faire un retour plus riche à l'apprenant à l'issue du test pour qu'il puisse comprendre ses lacunes et y remédier. On parle alors de diagnostic adaptatif.

Dans cette thèse, nous avons répertorié des modèles de tests adaptatifs issus de différents pans de la littérature. Nous les avons comparés de façon qualitative et quantitative. Nous avons ainsi proposé un protocole expérimental, que nous avons implémenté pour comparer les principaux modèles de tests adaptatifs sur plusieurs jeux de données réelles. Cela nous a amenés à proposer un modèle hybride de diagnostic de connaissances adaptatif, meilleur que les modèles de tests formatifs existants sur tous les jeux de données testés. Enfin, nous avons élaboré une stratégie pour poser plusieurs questions au tout début du test afin de réaliser une meilleure première estimation des connaissances de l'apprenant. Ce système peut être appliqué à la génération automatique de feuilles d'exercices, par exemple sur un cours en ligne ouvert et massif (MOOC).

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title. **Adaptive Testing using Cognitive Diagnosis for Large-Scale Learning**

Keywords. adaptive testing, cognitive diagnosis, massive open online courses (MOOCs), item response theory, q-matrix, learning analytics

Abstract. This thesis studies adaptive tests within learning environments. It falls within educational data mining and learning analytics, where student educational data is processed so as to optimize their learning.
Computerized assessments allow us to store and analyze student data easily, in order to provide better tests for future learners. In this thesis, we focus on computerized adaptive testing. Such adaptive tests which can ask a question to the learner, analyze their answer on the fly, and choose the next question to ask accordingly. This process reduces the number of questions to ask to a learner while keeping an accurate measurement of their level. Adaptive tests are today massively used in practice, for example in the GMAT and GRE standardized tests, that are administered to hundreds of thousands of students. Traditionally, models used for adaptive assessment have been mostly summative : they measure or rank effectively examinees, but do not provide any other feedback. Recent advances have focused on formative assessments, that provide more useful feedback for both the learner and the teacher ; hence, they are more useful for improving student learning.
In this thesis, we have reviewed adaptive testing models from various research communities. We have compared them qualitatively and quantitatively. Thus, we have proposed an experimental protocol that we have implemented in order to compare the most popular adaptive testing models, on real data. This led us to provide a hybrid model for adaptive cognitive diagnosis, better than existing models for formative assessment on all tried datasets. Finally, we have developed a strategy for asking several questions at the beginning of a test in order to measure the learner more accurately. This system can be applied to the automatic generation of worksheets, for example on a massive online open course (MOOC).

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France