Prédiction de performance sur des questions dichotomiques : comparaison de modèles pour des tests adaptatifs à grande échelle

Jill-Jênn Vie[1], Fabrice Popineau[1],
Jean-Bastien Grill[2], Éric Bruillard[3], Yolaine Bourda[1]
[1] Laboratoire de recherche en informatique, Bât. 650 Ada Lovelace, 91405 Orsay, France
{jjv, popineau, bourda}@lri.fr
[2] Inria Lille - Nord Europe, 40 avenue Halley, 59650 Villeneuve-d'Ascq, France
grill@clipper.ens.fr
[3] École normale supérieure de Cachan, 61 av. du Président-Wilson, 94235 Cachan, France
eric.bruillard@ens-cachan.fr

Résumé. Les tests adaptatifs sont un moyen d'évaluation qui a récemment gagné en popularité. Ils sélectionnent la prochaine question à poser à un examiné de manière à estimer son niveau efficacement, en fonction de ses réponses aux questions précédentes. Les systèmes de tests adaptatifs se sont d'abord appuyés sur la théorie de la réponse à l'item (TRI) afin de fournir une mesure efficace des traits latents dans des évaluations pouvant être à grande échelle. Plus récemment, dans l'optique de fournir un retour utile à l'examiné, d'autres modèles ont été étudiés dans la théorie du diagnostic cognitif. L'un d'eux est le modèle q-matrice, qui établit un lien entre questions du test et compétences de l'examiné. Dans cet article, nous définissons un protocole basé sur la prédiction de performance pour évaluer deux modèles de tests adaptatifs : q-matrice et le modèle de Rasch issu de la TRI. Les résultats obtenus sur trois jeux de données réelles de différente taille et nature montrent que selon les caractéristiques du test, l'un ou l'autre modèle prédit le mieux la performance de l'examiné.

Mots-clés. Tests adaptatifs, Diagnostic cognitif, Théorie de la réponse à l'item, Q-matrices

1 Introduction

La correction automatique de réponses d'étudiants a gagné en popularité ces dernières années avec le développement d'initiatives en ligne comme le GMAT [20], ou les *massive online open courses* (MOOCs). De tels systèmes doivent être capables de noter des milliers d'étudiants dans des perspectives d'évaluation ou de recrutement, et de fournir des corrections personnalisées dans des buts de formation.

Si on dispose déjà d'une large base de données de réponses à un certain test, il est naturel de se demander quelles sont les questions fournissant le plus d'informations à

propos d'un candidat, c'est-à-dire s'il y a un ordre « idéal » dans lequel poser les questions. Lors d'un oral, l'examinateur choisit habituellement la prochaine question à poser en fonction des précédentes réponses du candidat. Les tests adaptatifs peuvent être considérés comme une version automatisée de ce processus : on pose en permanence les questions les plus utiles jusqu'à ce qu'assez d'informations ont été récoltées. En effet, si le candidat se comporte de façon typique, un sous-ensemble de questions soigneusement choisies peut être suffisant pour deviner quels résultats il va obtenir sur les autres questions. Comme application, on peut imaginer un pré-test adaptatif sur chaque page de présentation d'un MOOC pour donner une idée des concepts développés dans le cours et aider un nouveau venu à se décider à s'inscrire ou non.

Pour répondre à ce besoin, la théorie de réponse à l'item (TRI) est d'une grande aide. Son but initial est de fournir un cadre d'évaluation de performance sur des questions individuelles, appelées *items*, sur les évaluations [12]. Elle a été appliquée avec succès aux tests adaptatifs et de nombreuses méthodes ont été proposées et implémentées [7, 9, 14]. Selon sa performance, un candidat obtient un score, permettant de prédire ses prochaines réponses. La simplicité des modèles TRI le rend assez souple pour l'analyse théorique [1] et a accru d'autant plus sa popularité.

Plus récemment, le *No Child Left Behind Act* de 2001 a demandé des évaluations plus formatrices, en mettant l'accent sur la détection précoce des étudiants présentant des handicaps cérébraux. Au lieu d'un score, les candidats reçoivent à l'issue du test une évaluation détaillée de leurs compétences, indiquant lesquelles sont maîtrisées, lesquelles ne le sont pas [4]. Ces modèles permettent aux enseignants de proposer des exercices appropriés correspondant mieux aux parties mal comprises. La plupart de ces modèles s'appuient sur une q-matrice spécifiant pour chaque question les différentes compétences requises pour la résoudre. Par exemple, un exercice sur les fractions peut exiger de savoir 1) convertir un nombre en fraction, 2) séparer un nombre entier d'une fraction, 3) simplifier avant de soustraire, etc. [5].

1.1 Notre contribution

Dans ce papier, nous proposons un protocole d'évaluation d'algorithmes de tests adaptatifs et l'utilisons pour comparer les performances d'un test adaptatif traditionnel utilisant le modèle de Rasch de la TRI, et d'un diagnostic cognitif plus récent de la littérature en psychométrie utilisant des q-matrices. Plus précisément, on espère répondre à la question suivante : étant donné un certain nombre de questions d'un certain ensemble posées en fonction d'une certaine règle de sélection adaptative, quel modèle prédit le mieux les réponses du candidat aux questions restantes ? À notre connaissance, une telle comparaison n'a jamais été réalisée.

On peut se demander pourquoi l'on compare un modèle d'évaluation avec un modèle de retour personnalisé. Nous montrons ici que même si le modèle de q-matrice a d'abord été conçu à but formateur, il est aussi possible de l'utiliser dans un contexte de

prédiction de performance. En effet, ce modèle évalue les compétences d'un candidat en fonction de ses réponses précédentes. Avec les compétences déterminées et la q-matrice, on peut estimer la probabilité qu'il réponde aux autres questions correctement. Cela nous permet de proposer un protocole de comparaison qui englobe les deux modèles.

Nous avons utilisé dans nos expériences le modèle de la TRI le plus simple et le plus utilisé, le modèle logistique à un paramètre, aussi appelé modèle de Rasch ; le modèle de q-matrice le plus simple ; le critère de sélection d'item maximisant l'information de Fisher et une approche gloutonne à une étape. Nous avons tout de même obtenu des résultats satisfaisants, nous permettant notamment d'affirmer qu'aucun modèle n'est meilleur dans tous les cas : selon le type de test, soit le modèle de Rasch, soit la q-matrice, s'en sort le mieux.

1.2 Plan

Nous exposons d'abord le cadre des tests adaptatifs et deux modèles issus respectivement de la TRI et du diagnostic cognitif. Nous détaillons ensuite la conception de nos modèles et décrivons le protocole proposé pour l'analyse. Finalement, nous listons les caractéristiques de nos trois jeux de données réelles et présentons nos résultats.

2 État de l'art et travaux liés

2.1 Tests adaptatifs

Dans un test adaptatif, on pose des questions personnalisées à un candidat en fonction de sa performance passée. Ainsi, un cadre de test adaptatif repose sur deux routines principales :

- NextItem : un algorithme de sélection de l'item suivant, qui choisit la prochaine question à poser en fonction des réponses précédentes du candidat ;
- TerminationRule : une condition de fin du test, lorsque suffisamment d'informations ont été collectées et que les compétences ont été mesurées.

Le cadre d'un test adaptatif peut être représenté par l'algorithme suivant : tant que le critère de terminaison n'est pas satisfait, l'algorithme choisit une question optimisant un certain critère.

Les critères usuels de sélection de l'item suivant sont de maximiser l'information de Fisher sur les questions possibles, minimiser l'entropie de Shannon de la distribution sur les paramètres, ou bien maximiser la divergence de Kullback-Leibler [26].

2.2 Théorie de la réponse à l'item

Pour nos besoins, la TRI nous donne un modèle souvent utilisé permettant de calculer pour chaque question la probabilité d'y répondre correctement. Pour faire simple, le modèle de Rasch estime le niveau d'un étudiant par un réel unique θ modélisé par une variable aléatoire et caractérise chaque question par un unique réel, sa difficulté d, correspondant au niveau requis pour répondre à la question correctement.

Connaissant le niveau latent θ_i d'un étudiant i donné et la difficulté d_j d'une question j, la probabilité de l'événement « l'étudiant i répond à la question j correctement », notée ci-après par $success_{ij}$, est modélisée par :

$$\Pr\{success_{ij}|\theta_i\} = \frac{1}{1+e^{-\delta(\theta_i - d_j)}},$$

où δ est un paramètre de discrimination. Le but est tout d'abord d'optimiser δ, les paramètres d_j pour chaque question j et θ_i pour chaque étudiant i afin de correspondre à un ensemble d'apprentissage donné. Ensuite, pendant le processus de test adaptatif, une distribution de probabilités sur θ_i est calculée et chaque réponse permet de préciser l'intervalle de confiance autour de θ_i en utilisant la règle de Bayes. La probabilité de $success_{ij}$ connaisant les paramètres peut alors être calculée en intégrant sur la variable latente θ_i :

$$\Pr\{success_{ij}\} = \int \Pr\{success_{ij}|\theta_i\} \Pr\{\theta_i\} \mathrm{d}\theta_i.$$

Ce modèle simple est utilisé dans les tests adaptatifs, où de nombreuses méthodes ont été implémentées pour la sélection de l'item suivant [16].

2.3 Diagnostic cognitif

Le modèle de Rasch que nous venons de décrire utilise un paramètre unique pour représenter les étudiants. Nous présentons maintenant un autre modèle qui essaie d'être plus informatif quant aux compétences de l'étudiant. Chaque étudiant est représenté par un modèle de valeurs binaires (a_1, …, a_K), appelé *vecteur de compétences*, représentant leur maîtrise de K compétences distinctes, donnant ainsi 2^K vecteurs possibles.

Une q-matrice Q [22] représente les différentes compétences mises en jeu dans chaque question. Formellement, Q_{ij} est égal à 1 si la compétence j est considérée, sinon à 0. Si toutes les compétences mises en jeu sont nécessaires pour réussir l'item

correspondant, le modèle est considéré comme faisant partie de la classe *conjonctive*. Si la maîtrise d'une seule compétence est suffisante pour réussir l'item, il sera considéré comme faisant partie de la classe *disjonctive*. On considère ici le modèle NIDA qui est un modèle conjonctif [7] en présence de bruit. Plus précisément, on note s_i (g_i) le paramètre *slip* (*guess*) de l'item i. La probabilité d'une réponse correcte à l'item i est $1 - s_i$ si toutes les compétences requises sont maîtrisées, g_i si au moins l'une de ces compétences n'est pas maîtrisée.

Dans le cas particulier $K = 1$, le modèle de diagnostic cognitif devient proche du modèle de Rasch présenté précédemment, la seule différence résidant dans la fonction de probabilité de succès. Dans le cas de la TRI, cela définit une sigmoïde, tandis que dans le modèle de q-matrice pour $K = 1$, cela définit une fonction affine sur [0, 1].

La détermination de la meilleure q-matrice pour un ensemble de candidats donné est actuellement une question ouverte, les derniers résultats étant des techniques de descente de gradient [2], de factorisation de matrices positives [8] ou l'algorithme EM [13].

Le vecteur de compétences d'un étudiant est une variable cachée modélisée par une variable aléatoire. Le nombre de vecteurs possibles étant fini, il est possible de garder la distribution de probabilités sur ces 2^K vecteurs et de la mettre à jour en utilisant la règle de Bayes. Avec cette distribution et à l'aide de notre q-matrice, on peut en déduire la probabilité d'un étudiant donné de répondre correctement à chaque question du test.

2.4 Travaux liés

Les q-matrices étaient à la base spécifiées par des experts, mais des travaux récents dans l'exploration des données éducatives tentent d'inférer les q-matrices directement à partir des données étudiant [13]. Nous utilisons de telles techniques dans notre simulation. Quelques extensions du modèle de Rasch original existent telles que la théorie multidimensionnelle de la réponse à l'item (TMRI) [21] mais leur complexité est beaucoup plus grande [7]. Le modèle SPARFA pour la réduction de la taille des tests [23] est lui-même une variante de la TMRI avec des coefficients limités à des valeurs positives et donne de bons résultats. Plusieurs modèles de fusion incorporant des paramètres de difficulté et des compétences requises ont été conçus [17] et testés dans des applications en temps réel [24] mais à notre connaissance, aucun travail n'a été réalisé quant à la comparaison des deux modèles.

Dans ce papier, nous utilisons la méthode d'information maximale de Fred Lord [15], en faisant une prédiction à un pas. Un autre domaine de recherche est le test à plusieurs pas (*multistage testing*), dans lequel plusieurs items suivants sont présentés au candidat et non seulement un. De tels regroupements conduisent à des échantillons plus importants et peuvent ne pas être nécessaires dans un test éducatif vu que la réponse à chaque item peut être observée immédiatement [3].

3 Cadre de tests adaptatifs

À présent, nous développons les différentes étapes du processus de test adaptatif utilisé dans la simulation qui nous permettra de comparer les deux modèles présentés. Nos données étudiant sont sous la forme d'une matrice $N_S \times N_Q$ où N_S et N_Q désignent respectivement le nombre d'étudiants et le nombre de questions, et c_{ij} vaut 1 si l'étudiant i a répondu à la question j correctement, 0 sinon.

Nous développons la méthode de validation croisée décrite dans l'algorithme 1. Le jeu de données est partitionné en deux ensembles : *train* et *test*, et un appel à Simulate(*train*, *test*) entraîne notre modèle sur les données *train* et lance une session de test adaptatif pour chaque étudiant de *test*. À chaque étape, une question est choisie et posée à l'étudiant. Les paramètres sont mis à jour en fonction de sa réponse et un indicateur de performance est calculé et enregistré. Voici donc les méthodes principales :

TrainingStep : entraîner le modèle de manière à calibrer les paramètres des questions. Dans le modèle de Rasch, α désigne le paramètre de difficulté d_i pour chaque question i, tandis que dans le modèle de diagnostic cognitif, α désigne les valeurs de la q-matrice Q_{ij} ainsi que les paramètres de *slip* et *guess* s_i et g_i pour chaque question i.
PriorInitialization : initialiser la distribution de probabilité π sur les paramètres de l'étudiant, c'est-à-dire sur les compétences pour le modèle q-matrice et sur la compétence latente θ pour le modèle de Rasch.
NextItem : choisir la meilleure question à poser selon un certain critère étant donné toutes les questions précédentes et les réponses correspondantes.
UpdateParameters : mettre à jour les paramètres étudiant en fonction des réponses précédentes.
TerminationRule : dans notre cas, le critère de terminaison est simple : « Toutes les questions ont été posées. »
PredictPerformance : calculer pour chaque question restante la probabilité que l'étudiant y répondra correctement.
EvaluatePerformance : comparer les vraies réponses avec la performance prédite, de manière à évaluer le modèle.

Algorithm 1 CAT Framework

```
procedure SIMULATE(train, test)
    α ← TRAININGSTEP(train)
    t ← 0
    for all students s in test do
        π ← PRIORINITIALIZATION()
        while TERMINATIONRULE is not satisfied do
            q_{t+1} ← NEXTITEM(q_1, r_1, ..., q_t, r_t, α, π)
            Ask question q_{t+1} to the student s
            Get reply r_{t+1}
            π ← ESTIMATEPARAMETERS(q_1, r_1, ..., q_t, r_t, α)
            p ← PREDICTPERFORMANCE(α, π)
            Σ ← EVALUATEPERFORMANCE(p)
        end while
    end for
end procedure
```

3.1 Évaluation des performances

Une fois que nous avons prédit la performance d'un étudiant, nous devons la comparer à la vérité. Pour cela, nous calculons la log-vraisemblance [11]. Dans tout ce qui suit, nous désignerons cette quantité par « erreur moyenne », un indicateur du pouvoir prédictif.

4 Évaluation

4.1 Jeux de données réelles

Nos algorithmes ont été testés sur trois véritables jeux de données réelles de différentes taille et nature.

SAT. Nous avons utilisé les données SAT également présentes dans [25, 8] et proposées par Titus à l'adresse http://alumni.cs.ucr.edu/~titus/. Il s'agit d'une matrice dichotomique de taille 296 × 40 représentant les résultats de 296 étudiants sur 40 questions de mathématiques, biologie, histoire du monde et français. Dans un test SAT, la plupart des questions ont 5 choix possibles, dont seulement une réponse correcte.

Fraction. Les données de fraction sont sous la forme d'une matrice 536 × 20 représentant les résultats de 536 étudiants sur 20 questions. Une matrice faite à la main pour K = 8 a été spécifiée pour ce jeu de données et étudiée dans [5, 6].

Castor. Il s'agit d'un concours d'informatique pour jeunes de la 6e à la terminale ; c'est la version française du concours Bebras, une initiative originaire de Lituanie aujourd'hui organisée dans 34 pays. Les candidats ont 45 minutes pour résoudre des tâches qui requièrent des compétences algorithmiques mais ne nécessitent pas de savoir programmer. Pour des exemples d'exercices, consulter http://castor-informatique.fr.

176 000 étudiants ont participé à l'édition 2013 du Castor, dont 46 % de filles. Les données que nous avons utilisées pour notre simulation se concentrent sur les réponses de 58 939 élèves de 6e/5e issus de l'édition 2013 composée de 17 tâches à compléter. C'est donc une matrice dichotomique de taille 58939 × 17 dont l'élément (i, j) vaut 1 si i a obtenu un score parfait sur la tâche j, 0 sinon.

4.2 Conception de la simulation

Dans nos expériences, nous analysons l'effet de 2 paramètres : le nombre de questions posées et le nombre de compétences K des q-matrices, pouvant ici varier de 2 à 14.

Nous avons testé nos algorithmes via une méthode de validation croisée par échantillonnages successifs : le modèle était entraîné avec 80 % des étudiants choisis au hasard et testé sur les 20 % restants.

Pour évaluer les deux modèles tout au long du processus de test adaptatif, nous calculons pour chaque étudiant du jeu de test l'erreur moyenne de sa performance prédite sur les questions restantes – c'est-à-dire, la probabilité qu'il y réponde correctement –, vis-à-vis de sa vraie réponse, et nous calculons la valeur moyenne de cette erreur moyenne sur tous les étudiants.

Notre implémentation est écrite en Python et R utilisant les packages rpy2 [10], ltm pour les modèles de traits latents de la TRI [18], catR pour les tests adaptatifs sur le package ltm [16] et CDM [19] pour obtenir la q-matrice spécifiée à la main pour le jeu de données Fraction et déterminer ses paramètres de *slip* et *guess*. Le code source est disponible sous licence MIT sur Bitbucket : http://bitbucket.org/jilljenn/qmatrix/

4.3 Résultats

Nous avons comparé l'implémentation du modèle de Rasch en R et notre implémentation du modèle q-matrice pour différentes valeurs du paramètre K, le nombre de colonnes de la q-matrice. Nous noterons respectivement ces implémentations TRI et Q.

Rapidité de l'algorithme. La durée des phases d'entraînement et de test, sur un Intel Core i5 1,3 GHz est répertoriée dans la Table 1. TRI a la phase d'entraînement la plus rapide et la phase de test la plus lente, ce qui reste raisonnable étant donné que ces valeurs ont été calculées sur 10 000 étudiants. Comme remarqué dans notre estimation de complexité, le temps de simulation pour Q augmente linéairement avec K. Ces résultats montrent que le modèle q-matrice est utilisable pour des tests adaptatifs, après une étape plus coûteuse de précalcul.

	Train phase	Test phase
IRT	4 min 20 s	5 min 20 s
Q $K = 2$	4 min 58 s	4 s
Q $K = 5$	6 min 46 s	3 s
Q $K = 8$	8 min 46 s	4 s
Q $K = 11$	10 min 48 s	5 s
Q $K = 14$	12 min 10 s	5 s

Table 1: Temps de calcul pour chaque algorithme sur le jeu de données Castor.

Évaluation des performances. Les résultats sont présentés dans les Tables 2 à 4 en annexe, les meilleures performances étant présentées en gras. Comme référence, 1.0 est l'erreur obtenue par un algorithme trivial affectant ½ à chaque probabilité, tandis qu'une erreur moyenne de 0 signifie que toutes les réponses ont été prédites correctement.

Sur le jeu de données Castor, la précision des intervalles de confiance nous permet d'affirmer que TRI présente des résultats meilleurs que Q pour n'importe quelle valeur de K à travers tout le test. C'est une illustration du compromis entre une petite valeur de K, donnant une q-matrice peu expressive, et une plus grande valeur de K, conduisant à une convergence tardive. Pour rappel, le but du modèle de la q-matrice est de déduire efficacement une distribution de probabilité sur les vecteurs de compétences possibles. Comme un vecteur plus grand est plus difficile à deviner, il est naturel de penser qu'un modèle avec une valeur de K plus grande requiert plus de questions pour prédire efficacement la performance d'un candidat.

Sur le jeu de données Fraction, la q-matrice écrite à la main obtient l'erreur la plus grande. Au début du test, les algorithmes Q pour $K = 8$ et 11 obtiennent des résultats sensiblement meilleurs que TRI. Comme son nom l'indique, le test Fraction est un test de calcul : il demande des compétences facilement identifiables. De fait, après certaines questions soigneusement choisies, il est naturel de pouvoir prédire la performance d'un candidat sur les questions restantes. Cela peut aussi expliquer que l'erreur moyenne calculée est basse comparée à celle sur les autres jeux de données.

Sur le jeu de données SAT, TRI obtient l'erreur la plus basse parmi tous les algorithmes testés. On observe aussi que la variance augmente tout au long du test, probablement parce que le comportement de l'algorithme doit beaucoup varier si les questions restantes portent sur une discipline différente des premières. Dans les dernières questions du test, l'erreur moyenne de TRI augmente légèrement, ce qui peut indiquer qu'un modèle de Rasch n'est pas assez expressif pour le set multidisciplinaire SAT.

5 Conclusion et perspectives

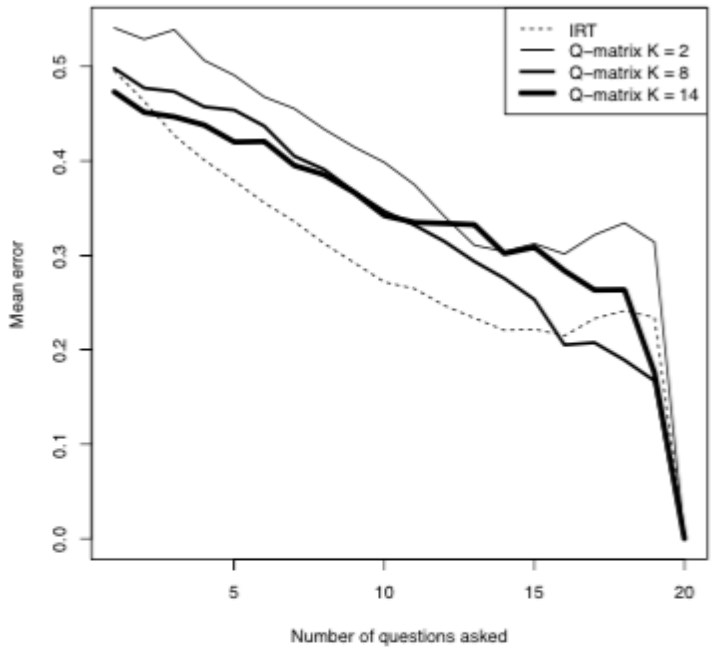
Notre comparaison du modèle de diagnostic cognitif avec la TRI semble indiquer que les q-matrices ont de meilleurs résultats sur certains types d'évaluations ; le test Fraction est un test de calcul, où il y a des redondances d'une question à l'autre afin de vérifier qu'une notion est connue et maîtrisée. Inversement, TRI obtient des résultats plus précis sur le test SAT et sur le concours Castor, ce qui est remarquable étant donné sa simplicité. Le fait que le test SAT soit multidisciplinaire explique partiellement la difficulté des algorithmes considérés pour prédire les réponses, et la nature de Castor en tant que concours peut nécessiter une notion de niveau plutôt que de maîtrise d'une compétence. Ainsi, dans ces deux cas, on préfèrera l'utilisation d'un modèle de Rasch.

Dans ce papier, nous avons défini un protocole pour évaluer des tests adaptatifs, nous permettant de comparer nos algorithmes sur différents types de tests. Notre simulation a mis en valeur le rôle des paramètres *slip* et *guess* en comparaison du paramètre de difficulté du modèle de Rasch, et se conforme à l'hypothèse que des tests de différents types nécessitent différents modèles. Pour confirmer ce comportement et obtenir une caractérisation plus précise de tels tests, nous prévoyons de tester notre implémentation sur de nombreux autres jeux de données.

Bibliographie

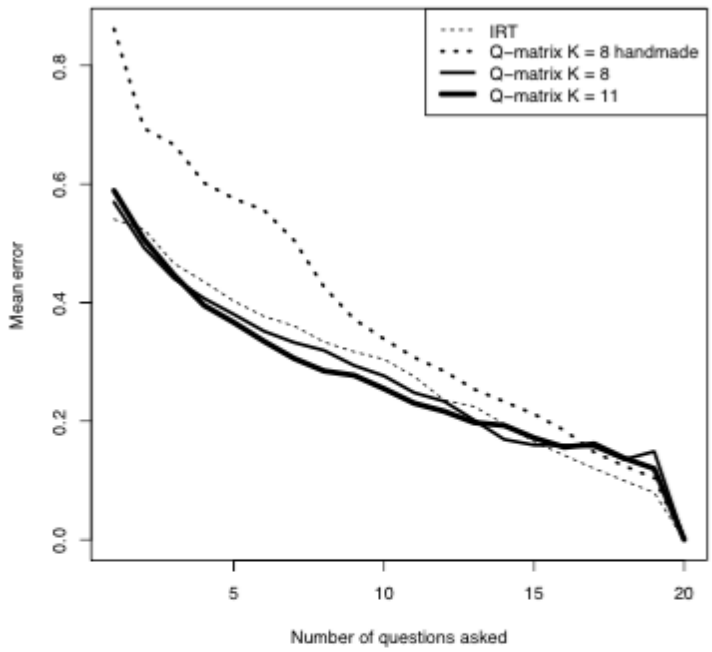
[1] F. B. Baker and S.-H. Kim. Item response theory: Parameter estimation techniques. CRC Press, 2004.
[2] T. Barnes. The q-matrix method: Mining student response data for knowledge. In American Association for Artificial Intelligence 2005 Educational Data Mining Workshop, 2005.
[3] H.-H. Chang. Psychometrics behind computerized adaptive testing. Psychometrika, pages 1–20, 2014.
[4] Y. Cheng. When cognitive diagnosis meets computerized adaptive testing: CD-CAT. Psychometrika, 74(4):619–632, 2009.
[5] J. De La Torre and J. A. Douglas. Higher-order latent trait models for cognitive diagnosis. Psychometrika, 69(3):333–353, 2004.
[6] L. T. DeCarlo. On the analysis of fraction subtraction data: The dina model, classification, latent class sizes, and the q-matrix. Applied Psychological Measurement, 2010.
[7] M. C. Desmarais and R. S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction, 22(1-2):9–38, 2012.
[8] M. C. Desmarais et al. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In 4th International Conference on Educational Data Mining, EDM, pages 41–50, 2011.
[9] T. J. Eggen. Computerized adaptive testing item selection in computerized adaptive learning systems. 2012.
[10] L. Gautier. rpy2: A simple and efficient access to R from Python. 2008.
[11] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477):359–378, 2007.
[12] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. Fundamentals of item response theory. Sage, 1991.
[13] A. Huebner. An overview of recent developments in cognitive diagnostic computer adaptive assessments. Practical Assessment, Research & Evaluation, 15(3):n3, 2010.
[14] Y. Huo, J. de la Torre, E.-Y. Mun, S.-Y. Kim, A. E. Ray, Y. Jiao, and H. R. White. A hierarchical multi-unidimensional irt approach for analyzing sparse, multi-group data for integrative data analysis. Psychometrika, pages 1–22, 2014.
[15] F. M. Lord. Applications of item response theory to practical testing problems. Routledge, 1980.
[16] D. Magis and G. Raˆıche. Random generation of response patterns under computerized adaptive testing with the R package catR. Journal of Statistical Software, 48(8):1–31, 5 2012.
[17] M. McGlohen and H.-H. Chang. Combining computer adaptive testing technology with cognitively diagnostic assessment. Behavior Research Methods, 40(3):808–821, 2008.
[18] D. Rizopoulos. ltm: An R package for latent variable modeling and item response analysis. Journal of Statistical Software, 17(5):1–25, 11 200.6.
[19] A. Robitzsch, T. Kiefer, A. George, and A. U ñlu ¨Cdm: Cognitive diagnosis modeling. R Package version, 3, 2014.
[20] L. M. Rudner. Implementing the graduate management admission test computerized adaptive test. In Elements of adaptive testing, pages 151–165. Springer, 2010.
[21] D. O. Segall. Multidimensional adaptive testing. Psychometrika, 61(2):331–354, 1996.
[22] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement, 20(4):345–354, 1983.
[23] D. Vats, C. Studer, A. S. Lan, L. Carin, and R. G. Baraniuk. Test-size reduction for concept estimation. In Proc. 6th Intl. Conf. on Educational Data Mining, pages 292–295, 2013.
[24] C. Wang. Mutual information item selection method in cognitive diagnostic computerized adaptive testing with short test length. Educational and Psychological Measurement, 73(6):1017–1035, 2013.
[25] T. Winters, C. Shelton, T. Payne, and G. Mei. Topic extraction from item-level grades. In American Association for Artificial Intelligence 2005 Workshop on Educational Datamining, Pittsburgh, PA, volume 1, page 3, 2005.
[26] X. Xu, H. Chang, and J. Douglas. A simulation study to compare CAT strategies for cognitive diagnosis. In annual meeting of the American Educational Research Association, Chicago, 2003.

Annexes



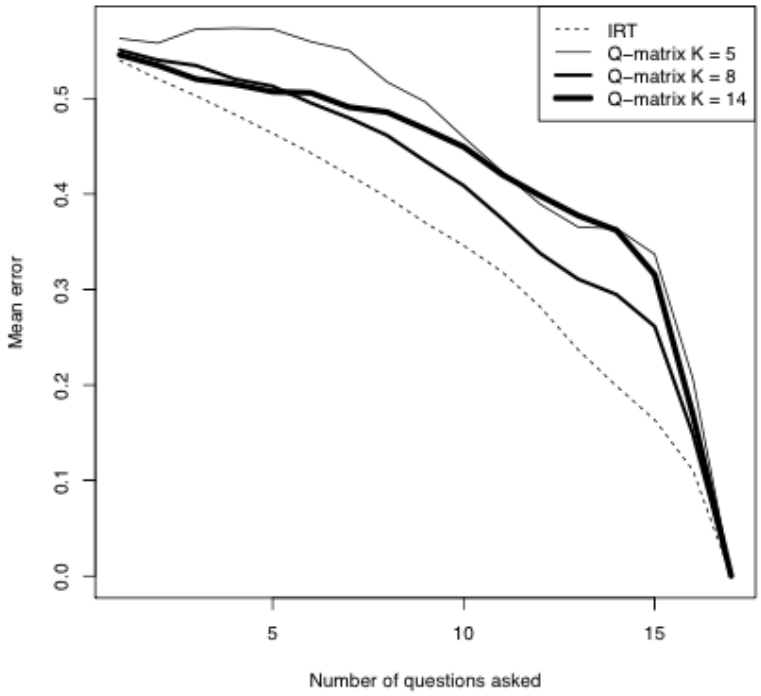
	After 4 q.	After 10 q.	After 16 q.
Q $K = 2$	0.522 ± 0.007	0.417 ± 0.010	0.315 ± 0.018
Q $K = 5$	0.469 ± 0.007	0.365 ± 0.012	0.306 ± 0.019
Q $K = 8$	0.463 ± 0.007	0.367 ± 0.013	$\mathbf{0.242 \pm 0.018}$
Q $K = 11$	0.456 ± 0.008	0.364 ± 0.013	0.331 ± 0.023
Q $K = 14$	0.441 ± 0.007	0.350 ± 0.012	0.296 ± 0.021
IRT	$\mathbf{0.409 \pm 0.008}$	$\mathbf{0.285 \pm 0.012}$	$\mathbf{0.248 \pm 0.022}$

Table 1. Erreur moyenne pour le jeu de données SAT.



	After 4 q.	After 10 q.	After 16 q.
Q $K = 2$	0.464 ± 0.012	0.326 ± 0.013	0.196 ± 0.017
Q $K = 5$	0.440 ± 0.011	0.289 ± 0.014	$\mathbf{0.146 \pm 0.013}$
Q $K = 8$	0.407 ± 0.011	0.276 ± 0.015	0.159 ± 0.015
Q $K = 11$	$\mathbf{0.395 \pm 0.009}$	$\mathbf{0.255 \pm 0.013}$	0.156 ± 0.015
Q $K = 14$	0.422 ± 0.009	0.274 ± 0.014	0.180 ± 0.018
IRT	0.435 ± 0.012	0.304 ± 0.013	$\mathbf{0.142 \pm 0.012}$
Q* $K = 8$	0.596 ± 0.008	0.346 ± 0.007	0.182 ± 0.007

Table 2. Erreur moyenne pour le jeu de données Fraction.



	After 4 q.	After 10 q.	After 16 q.
Q $K = 2$	0.555 ± 0.004	0.456 ± 0.005	0.167 ± 0.012
Q $K = 5$	0.574 ± 0.004	0.460 ± 0.006	0.206 ± 0.016
Q $K = 8$	0.520 ± 0.004	0.409 ± 0.006	0.148 ± 0.013
Q $K = 11$	0.519 ± 0.004	0.462 ± 0.007	0.218 ± 0.014
Q $K = 14$	0.515 ± 0.003	0.449 ± 0.006	0.169 ± 0.014
IRT	$\mathbf{0.484 \pm 0.003}$	$\mathbf{0.346 \pm 0.005}$	$\mathbf{0.111 \pm 0.010}$

Table 3. Erreur moyenne pour le jeu de données Castor.