

Comparaison de l'impact de techniques de diagnostic des connaissances sur l'apprentissage d'une stratégie d'aide

Sébastien Lallé, Jack Mostow, Vanda Luengo, Nathalie Guin

▶ **To cite this version:**

Sébastien Lallé, Jack Mostow, Vanda Luengo, Nathalie Guin. Comparaison de l'impact de techniques de diagnostic des connaissances sur l'apprentissage d'une stratégie d'aide. Journée EIAH&IA 2013, May 2013, Toulouse, France. 10 p. hal-00825330

HAL Id: hal-00825330
https://hal.science/hal-00825330

Submitted on 23 May 2013

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

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

Comparaison de l'impact de techniques de diagnostic des connaissances sur l'apprentissage d'une stratégie d'aide

Sébastien Lallé[1, 2, 3], Jack Mostow[3], Vanda Luengo[1], Nathalie Guin[2]

[1] LIG METAH, Université Joseph Fourier, Grenoble, France
[2] LIRIS, Université Lyon 1, CNRS, Lyon, France
[3] Carnegie Mellon University, Pittsburgh PA, United States of America
{sebastien.lalle, vanda.luengo}@imag.fr, mostow@cs.cmu.edu,
Nathalie.Guin@liris.univ-lyon1.fr

Résumé. Nous décrivons une méthode pour comparer des techniques de diagnostic des connaissances en fonction de leur impact sur les prises de décision de l'EIAH. À partir de traces d'apprenants, nous utilisons une méthode de classification automatique pour prédire le résultat (succès ou échec) d'une aide donnée par l'EIAH, en prenant en considération des informations sur l'apprenant, la tâche et l'état des connaissances de l'apprenant. Il en résulte une stratégie d'aide apprise automatiquement et permettant de sélectionner dans une situation donnée l'aide ayant la probabilité la plus élevée de succès. Le moyen usuel mais coûteux d'évaluer une telle stratégie d'aide est de l'implémenter et de la tester sur de nouveaux apprenants, ce qui nécessite du temps et des moyens. Nous utilisons plutôt des données déjà collectées, où les aides sont données aléatoirement aux apprenants, pour simuler l'impact d'une stratégie d'aide. Nous comparons ensuite différentes stratégies d'aide basées sur différentes techniques de diagnostic. Nous avons testé notre méthode en utilisant les données du Reading Tutor (projet LISTEN), destiné à l'apprentissage de la lecture. Nous rapportons la précision de la prédiction de quatre techniques de diagnostic en validation croisée, puis comparons l'impact des stratégies d'aide basées sur ces techniques. Notre méthode fournit une métrique pour comparer différentes techniques de diagnostic fondées sur leur utilité pour les prises de décision de l'EIAH.

Mots clés : Technique de diagnostic, aide, classification automatique, prédiction

1 Introduction

Un challenge dans le domaine des EIAH (Environnements Informatiques pour l'Apprentissage Humain) est d'évaluer l'impact des techniques de diagnostic des connaissances sur les prises de décision de l'EIAH, en particulier le choix d'un type d'aide. Aider les apprenants dans l'EIAH peut avoir un impact fort sur l'apprentissage [1] : plus un EIAH adapte l'aide à un apprenant et à une situation, plus l'apprenant a de chances d'apprendre.

Cet article est une traduction de l'article *Comparing Student Models in Different Formalisms by Predicting their Impact on Help Success* publié dans les actes de la *16th International Conference on Artificial Intelligence in Education*.

Cet article propose d'utiliser des traces d'EIAH et des techniques de diagnostic des connaissances pour décider quelle aide donner dans une situation donnée, puis montre comment comparer plusieurs techniques de diagnostic via le succès des aides. Les sections 2 et 3 reviennent sur l'état de l'art sur l'apprentissage de stratégies d'aide et les techniques de diagnostic. La section 4 présente les traces. La section 5 décrit l'algorithme d'apprentissage d'une stratégie d'aide. La section 6 rapporte les résultats avant la conclusion.

2 Travaux précédents et positionnement

Plusieurs articles montrent des résultats positifs sur l'apprentissage de stratégies d'aide. Andes [7] utilise un réseau bayésien pour adapter les indices fournis aux apprenants, aux problèmes et aux contextes, mais demande la conception manuelle de modèles d'adaptation, ce qui peut être coûteux.

ADVISOR [4] et des travaux ultérieurs [2, 6, 7] utilisent de l'apprentissage par renforcement pour adapter l'EIAH afin d'optimiser des métriques des performances des apprenants, comme le temps nécessaire à la résolution des exercices. ADVISOR et Barnes et Stamper [2, 7] se basent seulement sur le résultat d'une seule technique de diagnostic, alors que nous comparons plusieurs techniques.

Le Reading Tutor (projet LISTEN) [19] choisit aléatoirement une aide parmi plusieurs types d'aide, dont Heiner et al. [13] comparent le succès en mesurant le nombre de mots lus correctement après avoir reçu une aide. Nous utilisons le même principe et d'autres informations (dont des techniques de diagnostic) pour apprendre une stratégie d'aide, ce que Heiner et al. ne faisait pas.

Table 1. Synthèse de l'état de l'art sur la génération et/ou l'évaluation de stratégies d'aide.

Travaux	Éléments considérés pour l'aide	Méthodologie d'évaluation
Gertner *et al.* [11]	But du problème, problème courant, contexte, connaissances maîtrisées	Expérimentations (pré et post tests)
Beck *et al.* [4]	Problème courant, technique de diagnostic	Simulation de la probabilité de succès, expérimentations
Heiner *et al.* [13]	Niveau de l'apprenant, difficulté du mot lu	Estimation de l'amélioration du taux de bonnes réponses via des traces
Barnes, *et al.* [3, 23]	Problème courant, technique de diagnostic	Expérimentations (nombres de problèmes résolus, d'erreurs, d'indices donnés)
Chi *et al.* [6]	Données sur l'apprenant, le domaine et le système	Expérimentations (pré et post tests)
Cet article	Données sur l'apprenant et le domaine, techniques de diagnostic	Estimation de l'amélioration du taux de bonnes réponses via des traces

La table 1 résume ces travaux en fonction des éléments utilisés par les stratégies d'aide, et la façon dont elles ont été évaluées (expérimentations ou simulations).

D'autres travaux ont exploré plusieurs moyens de comparer des techniques de diagnostic [17]. Plusieurs articles [5, 12, 21, 26] comparent des techniques basées sur le modèle Knowledge Tracing via la précision de la prédiction des futures réponses (correctes ou non) des apprenants. Dans ces travaux, une technique de diagnostic est un outil prédictif. D'autres travaux font de même pour des techniques basées sur le modèle Constraint-based [14] et l'Item Response Theory [9]. L'intérêt de comparer des techniques de diagnostic est que leurs résultats dépendent du domaine considéré, des traces, et des algorithmes utilisés. Par exemple, Pavlik *et al.* montrent que la technique Performance Factor Models (PFM) bat la technique Bayesian Knowledge Tracing [21], mais Gong *et al.* trouvent l'inverse, sans que les raisons soient clairement identifiées [12]. De plus, nous ne connaissons pas de comparaisons quantitatives de techniques de diagnostic basées sur des modèles différents.

3 Techniques de diagnostic des connaissances

Nous décrivons trois modèles de techniques de diagnostic des connaissances, c.-à-d. des techniques estimant les connaissances du domaine maîtrisées ou non par un apprenant, utilisées dans cet article.

Le *Knowledge Tracing* [8] se base sur un modèle cognitif qui associe des connaissances à des actions successives observables. Ces observations servent à estimer la probabilité qu'un apprenant maîtrise une connaissance, via quatre paramètres : la probabilité de connaître la connaissance dès le début, la probabilité de l'apprendre lors d'une action, la probabilité de répondre juste sans maîtriser la connaissance nécessaire, et la probabilité de répondre faux tout en maîtrisant la connaissance. Basé sur ce modèle, le *Bayesian Knowledge Tracing* utilise l'inférence bayésienne, tandis que le *Performance Factor Model* utilise un modèle linéaire.

Le C*onstraint-based modeling* [20] représente le domaine en termes de contraintes dont la violation révèle une connaissance non sue appelant une rétroaction corrective. Une connaissance dans un modèle Constraint-based est une contrainte (Cr, Cs), où Cr identifie les situations où la contrainte est applicable, et Cs spécifie la réponse juste dans ces situations. On peut inférer à partir de traces la probabilité pour un apprenant de satisfaire une contrainte quand elle est applicable.

Enfin, le *Control-based* [16] (derivant de cKc [2]) représente les connaissances du domaine avec : un ensemble de problèmes, d'opérateurs pour résoudre le problème, des indicateurs sur la représentation du problème (comme *graphique* ou *preuve* en géométrie), et des connaissances de l'apprenant pour décider si une réponse est correcte ou non. Le Control-based est implémenté avec un réseau bayésien dynamique, pour inférer la probabilité de maîtriser les connaissances, sachant les problèmes, opérateurs, indicateurs, et le résultat observé d'un opérateur (correct ou non).

4 Traces expérimentales

Nous utilisons les traces du Reading Tutor (projet LISTEN) [19], qui donne des exercices de lecture en anglais et écoute les enfants lire à voix haute. Cet EIAH utilise la

reconnaissance automatique de la parole (RAP) pour décider si chaque mot est correctement lu ou non, et pour mesurer l'hésitation entre les mots. Nous disons qu'un mot est *lu couramment* si l'enfant le lit correctement selon la RAP et sans hésitation. Le Reading Tutor peut donner divers types d'aide sur un mot, comme le prononcer, l'épeler, prononcer un mot rimant avec… [13] Certains types d'aide ne sont pas possibles pour tous les mots (comme épeler un mot d'une lettre). Le Reading Tutor choisit aléatoirement entre tous les types possibles pour un mot donné.

Chaque aide ainsi choisie génère un *essai*. Nous disons qu'un essai aide l'apprenant à apprendre si le même mot est lu couramment à la prochaine lecture (au moins le lendemain) par l'apprenant. Donc si un apprenant a une aide sur un mot *W* le jour *i*, nous considérons la prochaine lecture de *W* le jour *j* avec $j>i$, de façon à éviter les biais liés à la récence de l'aide (figure 1). Nous avons 30 838 essais dans nos traces, collectés par le Reading Tutor durant l'année scolaire 2002-2003, avec 96 apprenants et 1078 mots différents. Pour simplifier l'analyse, nous ignorons les cas où un apprenant reçoit plusieurs types d'aide différents pour lire un même mot le jour *i*.

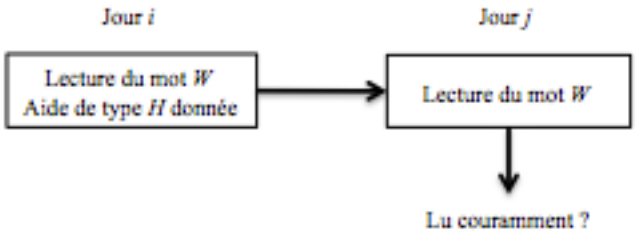


Fig. 1. L'aide *H* donnée sur le mot *W* le jour *i* est un succès si *W* est lu couramment le jour *j*.

5 Apprentissage d'une stratégie d'aide

Nous apprenons automatiquement une stratégie d'aide comme suit : nous enrichissons les traces d'apprenants avec les informations données par une technique de diagnostic, sélectionnons un ensemble d'éléments affectant le succès d'une aide (comme le niveau de l'apprenant ou la longueur d'un mot) via un modèle linéaire, et apprenons une stratégie d'aide via un algorithme de classification automatique.

5.1 Enrichissement des traces avec les techniques de diagnostic

Une technique de diagnostic infère ou met à jour un modèle d'apprenant via les interactions observables d'un apprenant avec l'EIAH [25]. Nous considérons dans notre étude quatre techniques : Performance Factor Modeling [21], Bayesian Knowledge Tracing [8], une technique basée sur le Constraint-based [18], et une technique basée sur le Control-based issu de cKc [16] (table 2). Ces techniques sont génériques, donc utilisables dans divers domaines. Plusieurs outils permettent l'implémentation de ces techniques, comme l'inférence bayésienne ou des régressions linéaires. Inférer ou mettre à jour un modèle d'apprenant signifie mettre à jour la probabilité qu'un appre-

nant maîtrise une connaissance, en utilisant les performances observées de l'apprenant (comme les réponses correctes et incorrectes). Ces estimations permettent ensuite de prédire les futures performances de l'apprenant mettant en jeu cette connaissance.

Table 2. Aperçu des quatre techniques de diagnostic utilisées dans cet article.

Technique de diagnostic	Mise à jour	Résultat (prédiction)	Ref
Performance Factor Model	Régression linéaire	Probabilité de répondre correctement	[21]
Bayesian Knowledge Tracing	Modèle de Markov caché	Probabilité de répondre correctement	[8]
Constraint-based	Contraintes	Probabilité d'enfreindre une contrainte	[18]
Control-based	Réseau bayésien dynamique	Probabilité d'utiliser une connaissance de façon correcte	[16]

Deux difficultés se posent ici. Premièrement, les techniques Constraint-based sont généralement mises à jour à la fin des exercices. Pour une mise à jour mot par mot dans notre domaine, nous associons aux contraintes une loi de puissance (*power law*). Deuxièmement, les performances liées à une connaissance ne sont pas directement observables. Dans notre modèle de la lecture, une connaissance est l'association d'un graphème vers un phonème. Par exemple, le mot *chemist* associe : *ch*→/K/, *e*→/EH/, *m*→/M/, *i*→/IH/, *s*→/S/, and *t*→/T/. Cependant, la RAP ne peut reconnaître que des mots. Nous utilisons donc une approche où une étape observée (un mot lu) est associée à plusieurs connaissances (graphème vers phonème). La mise à jour des connaissances se fait indépendamment et la prédiction par multiplication des prédictions pour chaque connaissance nécessaire à la lecture d'un mot [26].

Pour les paramètres des techniques, nous utilisons l'algorithme EM (expectation maximization) pour le Bayesian Knowledge Tracing et Control-based, et les packages *stats* et *igraph* de R pour Performance Factors Models et Constraint-based.

5.2 Sélection des éléments affectant l'aide

Nous considérons qu'un type d'aide *H* sur un mot *W* le jour *i* est un succès si *W* est lu couramment à la prochaine lecture le jour j ($j>i$). Pour trouver dans les traces les éléments qui prédisent le mieux un succès, nous utilisons une régression linéaire pas à pas (*stepwise*) avec le succès comme variables à expliquer et les éléments des traces comme prédicteurs. La mesure d'optimisation utilisée pour la régression pas à pas est AIC :

$$AIC = 2 \times k - 2 \times \ln(L)$$

Ici, *k* est le nombre de paramètres du modèle et *L* la vraisemblance (*data likelihood*). Un test ANOVA à un facteur mesure si un élément est significatif ($p<0.01$). Les éléments initiaux à tester sont : le niveau de lecture de l'apprenant, son taux de mots lus

couramment, le niveau de difficulté de l'exercice, la longueur du mot, la fréquence du mot en anglais, la position du mot dans la phrase, le nombre de fois où l'apprenant a lu le mot auparavant, et la classe du mot (définie comme l'ensemble des aides possibles pour ce mot).

5.3 Apprentissage d'un classificateur pour prédire le succès d'une aide

En utilisant le résultat des techniques de diagnostic, les éléments affectant l'aide, et le type d'aide, le problème est de prédire si une aide sera un succès. Pour cela, nous apprenons automatiquement trois classificateurs, deux basés sur des règles (Part [10] et JRip [7]) et un basé sur des arbres décisionnels, utilisant Weka[1]. Un exemple de règle de classification apprise est :

```
1) Word = c145
2) AND Story_Level = B
3) AND Student_Model_Prediction > 0.6
4) AND Help_Type = "SayWord"
      ⇨ Fluent (22/22)
```

La clause 1 spécifie que la règle s'applique sur les mots de classe « c145 », c'est-à-dire pour lesquels les types d'aide 1, 4 et 5 sont possibles. La clause 2 spécifie la difficulté de l'exercice (niveau B). La clause 3 spécifie que la probabilité que l'apprenant réponse correctement selon la technique de diagnostic doit être supérieure à 0,6. La clause 4 spécifie le type d'aide. La prédiction est que le mot est lu couramment (*fluent*) avec une confiance de 22/22 (22 essais satisfont cette règle sur 22 dans les traces d'apprentissage). Nous fixons comme limite pour la confiance 0,75 (par défaut dans Weka).

5.4 Utilisation d'un classificateur pour le choix d'un type d'aide

Une stratégie d'aide basée sur un des classificateurs appris ci-dessus fonctionne comme suit : choisir le type d'aide selon la règle ayant la confiance la plus élevée dans la situation donnée. Si plusieurs types sont possibles, choisir aléatoirement. Une solution alternative serait d'apprendre un classificateur par type d'aide, et choisir le type avec la probabilité la plus élevée de succès.

6 Résultats expérimentaux

Nous avons évalué notre méthode avec les traces de Reading Tutor (cf. section 4) en validation croisée. Nous avons séparé aléatoirement les traces en deux ensembles, 60 % pour apprendre les paramètres des techniques de diagnostic, 40 % pour apprendre et évaluer les classificateurs pour l'aide. Les classificateurs pour les stratégies d'aide sont appris sur le second ensemble (40 % des traces) en utilisant 10 *folds* (les traces

[1] http://www.cs.waikato.ac.nz/ml/weka

sont réparties en 10 sous-ensembles, 9 pour apprendre, le dernier pour tester ; la procédure est répétée dix fois pour que chaque sous-ensemble serve au test, et les résultats sont moyennés).

Nous évaluons au préalable la précision des techniques de diagnostic. Les résultats sont donnés table 3 (par ordre décroissant) à gauche. D'abord, la précision est définie comme le taux de bonnes prédictions de la prochaine performance (mot lu couramment ou non). Ensuite, le test AUC (*Area Under the ROC Curve*) mesure l'habilité des techniques à identifier les mots lus couramment et non couramment (en général, l'habilité à séparer des instances positives et négatives) [24]. Enfin, AIC (défini section 5.2) mesure la vraisemblance du modèle sur les traces d'apprentissage, pénalisée par le nombre de paramètres *k*. Ici *k* est le nombre de paramètres d'une technique, fois le nombre de connaissances, fois le nombre d'apprenants.

Table 3. Précision des techniques de diagnostic, puis prédiction du succès des stratégies d'aide basées sur les techniques. Intervalles de confiance à 95 % entre parenthèses.

Techniques de diagnostic	**Précisions des techniques**			**Prédiction du succès des aides**	
	Taux	**AUC**	**AIC**	**Précision**	**Signifiance**
Bayesian Knowledge Tracing	84 % (± 2,6 %)	0.68	5.1 E+4	75 % (± 4,1 %)	**
Control-based model	83 % (± 2,9 %)	0.67	7.2 E+4	73 % (± 4,4 %)	***
Performance Factor Model	81 % (± 3 %)	0.65	5.5 E+4	68 % (± 4,4 %)	***
Constraint-based model	80 % (± 2,8 %)	0.65	5.6 E+4	65 % (± 4,3 %)	

Signifiance du test de McNemar : ** $0,05 > p > 0,01$; *** $p \geq 0,01$

Chaque technique de diagnostic bat la classe majoritaire (76 % de mots lus couramment dans les traces). Ces résultats sont cohérents avec des évaluations passées du Knowledge Tracing [26], avec des précisions entre 72 % et 87 % (sur des traces différentes). Ces résultats montraient aussi des précisions en dessous de 35 % pour les mots lus non couramment, ce qui peut expliquer pourquoi les AUC sont de 0,68 au mieux dans nos résultats. AIC donne le Bayesian Knowledge Tracing premier, et pénalise le Control-based en raison du grand nombre de paramètres du réseau bayésien dynamique.

La table 3 montre à droite l'évaluation de chaque stratégie d'aide par la précision de la prédiction du succès d'une aide (prochaine lecture sur un même mot lu couramment après avoir reçu une aide). Nous montrons les résultats seulement pour les classificateurs appris par JRip, qui donne la meilleure précision (l'écart étant de moins de 2 %). Le Bayesian Knowledge Tracing donne la meilleure précision.

Les précisions de prédiction du succès d'une aide sont inférieures aux précisions des techniques de diagnostic utilisées. Prédire si un apprenant va lire un mot couramment est plus simple que de prédire si un type d'aide permettra de lire un mot couramment (succès). Une raison est les traces éparses (il peut ne pas y avoir beaucoup d'essais dans les traces pour chaque type d'aide et chaque situation).

Pour tester le degré de signifiance des différences entre les prédictions du succès des aides, nous utilisons le test de McNemar, qui évalue la signifiance de différences entre deux classificateurs C1 et C2. La formule est :

$$\chi^2 = (d1 - d2)^2 / (d1 + d2)$$

Ici, *d1* est le nombre d'instances classifiées comme positives par C1 et négatives par C2, et *d2* le nombre d'instances classifiées comme positives par C2 et négatives par C1. La somme *d1+d2* est supérieure à 80 dans nos données, la limite minimale définie par McNemar pour son test [15], qui peut être approximé par une distribution χ^2. Chaque ligne consécutive dans la table 3 diffère de façon significative (p<0.025), en assumant que l'effet des instances avec un même mot ou un même apprenant est négligeable.

Pour finir, nous calculons l'espérance d'amélioration du taux de succès des stratégies d'aide, c'est-à-dire le taux de mots lus couramment selon les stratégies d'aide. La différence entre le taux espéré et le taux réel (dans les traces) est rapportée table 4. La dernière ligne indique l'amélioration espérée en sélectionnant le type d'aide avec la plus grande probabilité dans les traces, sans utiliser de classificateur. Nous obtenons une espérance simulée, car calculée sur des traces précédemment collectées sans nouvelles expérimentations. Le succès espéré E est calculé comme suit :

$$E(Fluent \mid h^*, S, F)$$

Avec S la technique de diagnostic, F les éléments affectant le succès, et h* le type d'aide avec la probabilité de succès la plus élevée selon une stratégie d'aide pour une situation donnée :

$$h^* = \text{argmax}_h \, E(Fluent \mid h, S, F)$$

Table 4. Amélioration espérée du taux de succès des aides.

Technique de diagnostic	Amélioration espérée du succès de l'aide
Bayesian Knowledge Tracing	5,2 %
Control-based model	5,1 %
Performance Factor Model	4,7 %
Constraint-based model	4,5 %
Aucun classificateur	2,4 %

7 Conclusion

Cet article présente une méthode pour comparer des techniques de diagnostic et apprendre des stratégies d'aide. Les travaux précédents comparent les techniques de diagnostic basées sur un même modèle d'un point de vue prédictif. Nous comparons au contraire des techniques basées sur plusieurs modèles en fonction de leur impact sur une décision pédagogique, le choix d'une aide. Ces aspects sont nouveaux.

Notre méthode apprend une stratégie d'aide basée sur le résultat de techniques de diagnostic, les éléments impactant l'aide et des traces, afin de prédire les types d'aide ayant la plus forte probabilité de succès dans une situation donnée. Appliquées aux données du Reading Tutor, nous obtenons une amélioration estimée du succès des aides entre 4,5 % et 5,2 %.

Les limites de notre approche sont : la nécessité de traces où les types d'aide sont sélectionnés aléatoirement (comme dans le Reading Tutor), la dépendance au nombre de situations différentes observées pour chaque type d'aide, les résultats simulés et non obtenus par de nouvelles expérimentations. Une grande quantité de traces peut donc être requise pour éviter le sur-apprentissage ou des stratégies d'aide incomplètes. Ces limites sont le prix à payer pour un apprentissage automatique de plusieurs stratégies d'aide et une évaluation automatique de ces stratégies. Les perspectives incluent l'utilisation de techniques de diagnostic plus précises, comme LR-DBN [26], des classificateurs plus perfectionnés (comme les Support Vector Machines), des analyses sur l'impact des techniques sur le succès des aides, et des expérimentations pour mesurer en pratique le taux de succès des stratégies apprises.

Remerciement
Ce travail est supporté par une bourse de thèse et de mobilité accordée au premier auteur par la région Rhône-Alpes en France, et l'Institute of Education Sciences, U.S. Department of Education, via la subvention R305A080628 accordée à l'université Carnegie Mellon aux États-Unis. Nous remercions les enfants, les écoles et les membres du projet LISTEN pour les traces.

Références

1. Anderson, J.R., Gluck, K.: What role do cognitive architectures play in intelligent tutoring systems. In: Cognition and Instruction: Twenty-five years of progress, Mahwah, NJ: Lawrence Erlbaum, 227–262 (2001).
2. Balacheff, N., Gaudin, N.: Students conceptions: an introduction to a formal characterization. Cahier Leibniz 65, 1–21 (2002).
3. Barnes, T., Stamper, J., Lehman, L., Croy, M.: A pilot study on logic proof tutoring using hints generated from historical student data. In: Procs. of the 1st International Conference on Educational Data Mining, Montréal, Canada, 552–557 (2008).
4. Beck, J.E., Woolf, B.P., Beal, C.R.: ADVISOR: a machine-learning architecture for intelligent tutor construction. In: Procs. of the 17th AAAI Conference on Artificial Intelligence, Boston, MA, 552–557 (2000).
5. Cen, H., Koedinger, K., Junker, B.: Comparing two IRT models for conjunctive skills. In: Procs. of the 9th International Conference on Intelligent Tutoring Systems, Montréal, Canada, 796–798 (2008).
6. Chi, M., VanLehn, K., Litman, D., Jordan, P.: Inducing effective pedagogical strategies using learning context features. In: Procs. of the 18th International Conference on User Modeling, Adaptation, and Personalization, Big Island, HI, 147–158 (2010).
7. Cohen, W.W.: Fast Effective Rule Induction. In: Procs. of the 12th International Conference on Machine Learning, Tahoe City, CA, 115–123 (1995).
8. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modelling and User-Adapted Interaction 4, 253–278 (1995).

9. Desmarais, M.: Performance comparison of item-to-item skills models with the IRT single latent trait model. In: Procs. of the 19th International Conference on User Modeling, Adaption and Personalization, Girona, Spain, 75–86 (2011).
10. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In:Procs. of the 15th International Conference on Machine Learning, Madison, WI, 144–151 (1998).
11. Gertner, A.S., Conati, C., VanLehn, K.: Procedural help in Andes: Generating hints using a Bayesian network student model. In: Procs. of the 15th National Conference on Artificial Intelligence, Madison, WI, 106–111 (1998).
12. Gong, Y., Beck, J.E., Heffernan, N.T.: How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. International Journal of Artificial Intelligence in Education 21(1), 27–46 (2011).
13. Heiner, C., Beck, J., Mostow, J.: Improving the help selection policy in a Reading Tutor that listens. In: Procs. of the InSTIL/ICALL 2004 Symposium on NLP and Speech Technologies in Advanced Language Learning Systems, Venice, Italy, 195–198 (2004).
14. Le, N.-T., Pinkwart, N.: Can Soft Computing Techniques Enhance the Error Diagnosis Accuracy for Intelligent Tutors? In: Procs. of the 11th International Conference on Intelligent Tutoring Systems, Chania, Greece, 320–329 (2012).
15. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12(2), 153–157 (1947).
16. Minh Chieu, V., Luengo, V., Vadcard, L., Tonetti, J.: Student modeling in complex domains: Exploiting symbiosis between temporal Bayesian networks and fine-grained didactical analysis. International Journal of Artificial Intelligence in Education 20(3), 269–301 (2010).
17. Mitrovic, A., Koedinger, K., Martin, B.: A comparative analysis of cognitive tutoring and constraint-based modeling. In: Procs. of the 9th nternational Conference on User Modeling, Johnstown, PA, 313–322 (2003).
18. Mitrovic, A., Ohlsson, S.: Evaluation of a Constraint-Based Tutor for a Database. International Journal of Artificial Intelligence in Education 10(3-4), 238–256 (1999).
19. Mostow, J., Aist, G.: Evaluating tutors that listen: An overview of Project LISTEN. In: Smart Machines in Education: The coming revolution in educational technology, Cambridge, MA: MIT/AAAI Press, 169–234 (2001).
20. Ohlsson, S.: Constraint-based student modeling. NATO ASI Series F Computer and Systems Sciences 125, 167–189 (1994).
21. Pavlik, P. I., Cen, H., Koedinger, K.: Performance Factors Analysis–A New Alternative to Knowledge Tracing. In: Procs. of the 15th International Conference on Artificial Intelligence in Education, Auckland, New Zealand, 531–538 (2009).
22. Razzaq, L., Heffernan, N.T.: Scaffolding vs. Hints in the Assistment System. In: Procs. of the 8th International Conference Intelligent Tutoring Systems, Jhongli, Taiwan, 635–644 (2006).
23. Stamper, J., Barnes, T., Lehmann, L, Croy, M.: The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In: Procs. of the International 9th Conference on Intelligent Tutoring Systems Young Researchers Track, Montréal, Canada, 71–78 (2008).
24. Swets, J.A.: Measuring the accuracy of diagnostic systems. Science 240(4857), 1285–1293 (1988).
25. VanLehn, K.: Student modeling. In: Foundations of Intelligent Tutoring Systems, Mahwah, NJ: Lawrence Erlbaum, 55–78 (1988).
26. Xu, Y., Mostow, J.: Comparison of methods to trace multiple subskills: Is LR-DBN best? In: Procs. of the 5th International Conference on Educational Data Mining, Chania, Greece, 41–48 (2012).