

**CLASSIFICATION AUTOMATIQUE DE COMPORTEMENTS D'ÉLÈVES
APPRENANT L'ALGÈBRE**

David Renaudie,

Doctorant en informatique, équipe Apprentissage

david.renaudie@imag.fr, + 33 4 76 57 48 13

Adresse professionnelle

Laboratoire Leibniz-IMAG, 46 av. Félix Viallet, 38031 Grenoble Cedex 1

Résumé : Nous nous intéressons à l'analyse de comportements d'élèves effectuant des exercices d'algèbre, basée sur des données recueillies lors de l'utilisation d'un EIAH (Environnement Informatique pour l'Apprentissage Humain) pour l'algèbre, le logiciel Aplusix. Toutes les actions des élèves ont été enregistrées et constituent notre corpus de données initial. Par l'application de techniques de classification non supervisée à ces données préalablement « vectorisées », nous construisons des catégories d'élèves dont la réponse aux problèmes posés est comportementalement homogène. Nous présentons ici les résultats obtenus avec les différents prétraitements employés, ainsi que les variables de représentation retenues lors de l'analyse automatique de données réelles.

Summary : We focus on analysing the behaviors of students learning algebra, based on data gathered while using a learning environment software called Aplusix. Every single action performed by each student (key pressed, mouse moves, etc...) was recorded during sessions organized in French high schools, and constitute our base corpus. By applying unsupervised classification techniques to the data, which is pre-processed into a vector set, we discover clusters of students which have homogeneous behaviors. We present here the results obtained by using various pre-processing techniques and feature sets during the automatic analysis of real data.

Mots clés : représentation de l'information, fouille de données, classification non supervisée, clustering, EIAH.

CLASSIFICATION AUTOMATIQUE DE COMPORTEMENTS D'ÉLÈVES APPRENANT L'ALGÈBRE

Nous nous intéressons à l'analyse de comportements d'élèves effectuant des exercices d'algèbre, basée sur des données recueillies lors de l'utilisation d'un EIAH (Environnement Informatique pour l'Apprentissage Humain) pour l'algèbre, le logiciel Aplusix. Toutes les actions des élèves ont été enregistrées et constituent notre corpus de données initial. Par l'application de techniques de classification non supervisée à ces données préalablement « vectorisées », nous construisons des catégories d'élèves dont la réponse aux problèmes posés est comportementalement homogène. Nous présentons ici les résultats obtenus avec les différents prétraitements employés, ainsi que les variables de représentation retenues lors de l'analyse automatique de données réelles.

1 – INTRODUCTION

L'analyse de comportements d'élèves, étape centrale de toute étude didactique, est souvent manuelle et individuelle. Il en ressort une description « détaillée » des phénomènes mais au prix d'un temps considérable. Ainsi, les outils statistiques (Bronner, 2003) apportent une aide précieuse au didacticien en synthétisant le résultat des observations. Notre approche consiste à étendre ce type de démarche en appliquant des méthodes d'apprentissage machine afin d'automatiser davantage une partie de ces analyses, et ainsi obtenir des résultats de plus haut niveau faisant apparaître des catégories de comportements d'élèves homogènes.

La modélisation de l'élève par apprentissage machine est un domaine très actif, comme en témoigne la riche bibliographie s'y rapportant. Ainsi, Sison (1998) dresse un éventail des techniques employées et des systèmes développés. Certains de ces systèmes sont capables de construire une représentation des connaissances de l'élève sur un domaine donné à partir d'une seule prestation de cet élève (LISP Tutor (Anderson, 1985)); Proust (Johnson, 1990)), mais ils s'appliquent

seulement dans la cas où la représentation du résultat est suffisamment structurée pour le faire, à savoir des programmes informatiques. D'autres systèmes nécessitent plusieurs cas pour construire le modèle de l'élève, et s'appliquent à des domaines où la nature du résultat est plus factuelle (DEBUGGY (Burton, 1982) ; ACM, (Langley 1984) ; ASSERT (Baffes, 1996)). Ces systèmes sont conçus à partir de techniques d'apprentissage machine symboliques (apprentissage par induction supervisé ou non supervisé), et nécessitent souvent la constitution préalable d'une large base de connaissances par des experts du domaine pour fonctionner. Dans un contexte proche de notre travail, Beck (2000) utilise l'apprentissage par renforcement, mais cherche à prédire la justesse de la réponse et le temps de réaction de l'étudiant modélisé, alors que nous cherchons à découvrir et expliciter des descriptions de groupes d'élève.

Dans notre approche, pour décrire le comportement des élèves nous disposons à la fois de descripteurs quantitatifs décrivant leurs performances (temps, résultats, ...) mais aussi de descripteurs modélisant les connaissances symboliques caractérisant les manipulations mathématiques effectuées par les élèves sur les expressions algébriques. Sur l'ensemble de ces descripteurs nous appliquons des techniques de classification non supervisée, dans un but exploratoire, afin d'obtenir des groupes d'élèves présentant des comportements homogènes.

Le contexte de travail : le projet « Cognitive-Algèbre » et Aplusix

Notre travail se déroule dans le cadre du projet « Modélisation cognitive de comportements d'élèves en algèbre », une ACI financée par le ministère de la recherche. Il repose sur l'utilisation d'un EIAH en algèbre, le système Aplusix, dont la présentation détaillée peut être trouvée dans (Bouhineau, 2001). Ce logiciel permet à l'élève de manipuler facilement des expressions mathématiques, tout en « découpant » son travail en plusieurs étapes de calcul (il y en a 3 dans la Figure 0).

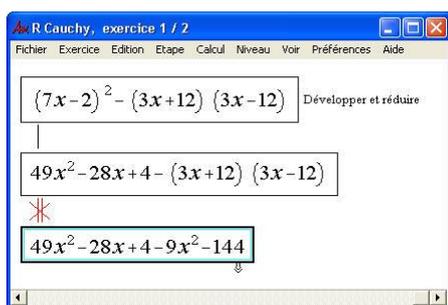


Figure 0 : le logiciel Aplusix

Chaque prestation d'élève utilisant Aplusix est systématiquement enregistrée dans un fichier recueillant la séquence des actions effectuées (frappes de touches, clics souris ...) ainsi que leurs instants d'occurrence, l'expression mathématique qui était alors affichée, et quelques renseignements additionnels. Un tel fichier, contenant tout l'historique des interactions élève-Aplusix, s'appelle un protocole. Enfin, Aplusix contient un module capable de « rejouer » un protocole, action par action ou étape par étape : le magnéscope. C'est l'outil d'analyse manuelle de protocoles.

En Septembre 2002, des enseignants en mathématiques d'une dizaine de collèges et lycées ont mis en place des sessions d'utilisation du logiciel Aplusix (voir (Bronner 2003)). Plusieurs listes d'exercices ont été mises au point pour chaque niveau (3e, 2nde, 1ère), chacune contenant une trentaine d'exercices de difficultés croissantes. Chaque exercice est traité par un nombre très variable d'élèves : de plus de 200 élèves sur les premiers exercices d'une série à moins de 10 pour les derniers. Notre corpus initial est constitué d'environ 11000 couples « élève/exercice ».

2 – LA CHAÎNE DE TRAITEMENTS

2.1 – La complétion de protocoles

Les protocoles bruts contiennent une information purement syntaxique (la succession des actions élémentaires et les expressions algébriques produites) qu'il est difficile d'exploiter directement. Un module de complétion de protocoles a été mis au point par nos partenaires au sein du projet Cognitique-Algèbre : à partir d'une analyse de l'expression algébrique, il explicite et déduit des informations de plus haut niveau. Les

fichiers obtenus sont désignés par protocoles complétés.

Parmi les champs rajoutés, nous avons ceux qui apportent des compléments d'information sur la justesse des expressions (par exemple l'équivalence mathématique entre l'expression en cours et la dernière expression de l'étape précédente), utiles pour les calculs des variables servant à la classification ou pour l'interprétation des résultats. D'autres descripteurs concernent la « qualification » (Gélis 1994). Cela consiste à identifier le type de transformation effectuée entre une expression initiale et une expression finale donnée, indépendamment de sa justesse mathématique. Ex : « $(10x=7x-4 \rightarrow 3x=4)$: *Reduction Additive Equivalente* » signifie qu'entre les expressions situées de part et d'autre du symbole « \rightarrow » a eu lieu une transformation qualifiée par « *Reduction Additive* », ici mathématiquement équivalente. Comme nous le verrons plus loin, ces champs additionnels vont nous être d'une grande utilité pour mieux identifier des groupes d'élèves aux comportements homogènes.

2.2 – Construction du jeu de données

Afin de pouvoir utiliser un algorithme de classification automatique, il nous faut « recoder » les informations des protocoles sous la forme d'une matrice lignes (élèves) colonnes (variables), donc décrire chaque protocole par un unique vecteur de taille fixe. Dès lors, il devient difficile de conserver l'aspect séquentiel des protocoles, et l'utilisation de descripteurs rendant compte de cette séquentialité pourrait être envisagée. Dans une première approche, les variables descriptives qu'il était possible de calculer directement à partir d'un protocole sont résumées dans la table 1.

Famille de variable	Nbre	Type	Explication
Nombre d'actions	14	Entiers naturels	Décomptes de nombres d'actions effectués par l'élève. Ex : actions dont l'expression résultante est équivalente, mal formée, etc...
Nombres d'étapes	10	Entiers naturels	Décomptes des nombres d'étapes, distingués selon l'état de l'expression au moment de la création.
Temps	9	Réels	Temps passés devant des expressions équivalentes, non équivalentes, etc...

Qualifications	30	Entiers naturels	Nombre d'instances de chaque qualification dans l'exercice, distribués dans des champs différents selon la justesse de la transformation (équivalente ou non)
----------------	----	------------------	---

Table 1 : Variables initialement retenues pour caractériser, de manière statistique, le comportement d'un élève sur un exercice donné.

De plus, nous normalisons ces variables car elles évoluent sur des domaines de valeurs très différentes, et cela a une influence considérable sur nos résultats vu que nous utilisons une distance euclidienne. Nous centrons puis réduisons chacune des dimensions selon les formules suivantes :

Soit $x_{i,j}$ la valeur de la variable numéro j de la donnée numéro i . La variable centrée-réduite $z_{i,j}$ correspondant vaut :

$$z_{i,j} = \frac{x_{i,j} - \bar{x}_j}{\sigma_j} \quad (1) \quad \text{avec} \quad \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

$$\sigma_j = \frac{1}{n} \sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2$$

2.3 – La sélection des variables

Toutes les variables précédemment collectées ne sont pas de bons descripteurs représentatifs de la prestation de l'élève. Ainsi, Parmi les 63 variables présentées dans la table 1, nous avons choisi, dans un premier temps, d'en retenir 12. En effet, notre matrice de données est creuse, car certains de nos descripteurs initiaux ne sont calculés que lors d'évènements relativement rares, par exemple quand l'expression est indéfinie (ex : $\text{rac}(-1)$). Voici les 12 premières variables retenues

- #a_eqenc : nbe d'exp équiv à l'énoncé
- #a_equiv : nbe d'exp équiv à l'étape préc
- #a_expmf : nombre d'exp mal formées
- #a_modif : nbe d'actions modifiant l'exp
- #a_noneq : nbe d'exp non équivalentes
- #a_total : nombre total d'exp
- #e_suppr : nombre d'étapes supprimées
- #e_total : nombre d'étapes créées
- ~t_equiv : tps passé dans des exp équiv
- ~t_expmf : tps passé sur des exp mal formées

- ~t_noneq : tps passé sur des exp non équiv

- ~t_total : tps total passé sur l'exercice

(« exp »=expression, « equiv »=équivalentes, « tps »=temps).

De plus, parmi les 30 différentes variables portant sur les qualifications, nous avons choisi de ne retenir que celles portant sur des transformations fausses, et parmi ces dernières, seulement les 6 plus fréquentes sur l'exercice analysé (cf d.):

- #q_epcne_ANTIEFFONDREMENT : hausse subite du degré formel. Exemple : $x+3 \rightarrow x^2+3$

- #q_epcne_CALCULNUMERIQUE.

Exemple : $8+3 \rightarrow 12$

- #q_epcne_DEVELOPPEMENT.

Exemple : $-54x^2+18x+30x \rightarrow -54x^2+48$

- #q_epcne_DEVELOPPEMENTPOLY.

Exemple : $(9x-5)(-6x+2) \rightarrow -54x^2+18+30x-10$

- #q_epcne_EFFONDREMENT : baisse subite du degré formel. Exemple : $x^2+x-2 \rightarrow 3x-2$

- #q_epcne_REDUCTIONADDITIVE.

Exemple : $3x+4x \rightarrow 8x$

Le choix de ces variables est une phase importante de notre travail, et découle d'une expertise manuelle que nous avons acquise en manipulant les données. Nous travaillons actuellement pour confirmer notre analyse par d'autres mécanismes de sélection de variables, notamment basés sur des techniques d'apprentissage supervisé que l'on peut trouver dans la boîte à outils WEKA (Witten 1999). Nous présentons en partie 3.d quelques premiers travaux visant à étudier l'impact du choix de variables sur la classification obtenue.

2.4 – Choix du corpus d'apprentissage

Chaque élève ayant traité plusieurs exercices, nous disposons de plusieurs vecteurs numériques pour chaque élève. Lors d'expériences préliminaires, nous avons effectué des classifications en considérant comme représentant d'un élève un vecteur moyen sur tous les exercices traités. Cela menait à de mauvais résultats. Ceci est peu surprenant vu que cela revenait à mélanger les disparités de compréhension de l'élève (faiblesses dans tels types d'exercices et bon niveau dans d'autres).

En effet, un élève pouvait très bien avoir compris une technique (comme la factorisation) et être en difficulté sur une autre (comme le développement) : la nécessité de caractériser des groupes d'élèves sur des types de tâches différents est donc devenue incontournable ; et cela nous permettait, en outre, de décrire plus finement son profil. Ainsi, dans un premier temps, nous nous sommes concentrés sur un exercice précis et avons effectué notre classification sur tous les élèves l'ayant traité, avec l'objectif ultérieur de réitérer ce procédé sur d'autres exercices. Son choix a été guidé par deux facteurs : le nombre d'élèves l'ayant traité devait être le plus grand possible (afin d'avoir des statistiques significatives) et non trivial (un exercice trop facile donne peu d'indications sur le niveau de l'élève). Ainsi, nous avons choisi un exercice de 2^{nde}, que nous nommerons « exercice 1204 ». Son intitulé exact est : « développer puis réduire $(9x-5)(-6x+2)$ ». 218 élèves l'ont effectué, d'où un jeu de données de 218 vecteurs.

2.5 – Les outils de classification

Après avoir étudié plusieurs algorithmes de classification, nous avons retenu le SuperParamagnetic Clustering (SPC) développé par Blatt (1997). Inspiré par une analogie issue de la physique, cette technique présente l'avantage du multi-échelle : par opposition à beaucoup d'approches classiques (du type K-means (Diday, 1973) (Duda 2001), PAM/CLARANS (Kaufman 1990), DBSCAN (Ester 1996)...), le résultat fourni par le SPC n'est pas une partition des données en classes, mais plusieurs partitions ordonnées selon un paramètre dit de « température » (il n'est pas nécessaire de connaître ici le fonctionnement précis de cet algorithme. Le lecteur intéressé pourra se référer à (Blatt, 1997) pour plus de détails). C'est l'utilisateur qui choisit l'échelle à laquelle il veut observer les classes obtenues, à la manière des approches hiérarchiques (Lebart 2000). Cependant, cet algorithme surpasse ces approches hiérarchiques classiques en terme de temps de calcul et de résistance au bruit; de plus, son mode de fonctionnement par densité lui permet de détecter des clusters de formes arbitraires (Berkhin, 2002).

2.6 – La visualisation des résultats

Nous utilisons un module Matlab pour visualiser le résultat d'une classification par SPC. La courbe de la Figure 2 représente les tailles des différents clusters, pour chaque valeur du paramètre. Les paliers observés sont le signe de l'existence de groupements stables sur une plage de variation du paramètre : cela est caractéristique d'une classification de bonne qualité.

En pratique, nous choisissons la valeur de température à laquelle nous observons des paliers mettant en jeu des classes avec un effectif « convenable », en nous basant sur les cardinaux des groupes détectés : trop élevés, ils sont synonymes de faible homogénéité, et trop faibles, ils sont peu informatifs.

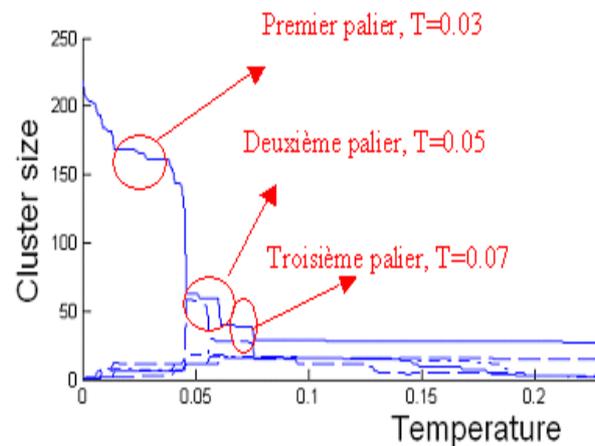


Figure 0 : Exemple de courbe fournie par le SPC. Le premier palier correspond à un seul groupe comprenant quasiment tous les individus. Le deuxième et le troisième palier sont plus intéressants car il y a plus de groupes de plus petite taille.

3 - RESULTATS

3.1 - Résultats avec les protocoles bruts, sans traits de qualification

La première de nos expérimentations avait pour objectif de constituer un résultat-témoin, auquel nous pourrions nous référer par la suite pour constater les améliorations. Nous n'avons donc utilisé que les 12 premières variables.

On obtient 4 principaux groupes totalisant 101 élèves :

a) Les succès rapides (48) : on constate, a posteriori, et ce sans avoir inclus la variable « résolu » dans nos 12 variables, qu'une majorité des élèves de ce groupe a fait juste, en un temps bien inférieur à la moyenne et en interagissant relativement peu avec le logiciel

b) Les succès actifs (15) : légèrement plus lents que le groupe a), ces élèves se caractérisent par un fort niveau d'interaction avec le logiciel, et leurs manipulations concernent essentiellement des expressions équivalentes à l'énoncé.

c) Les échecs pour faute de parcours (20) : ce groupe est très similaire au a) en termes de temps et d'actions, mais il se distingue par un très faible nombre d'expressions équivalentes. Les élèves de ce groupe ont souvent fait une erreur en cours de résolution

d) Les échecs lents (18) : ceux ci passent la majorité de leur temps devant des expressions mal formées ou non équivalentes. Ils font beaucoup d'erreurs (nombre d'actions équivalentes faible)

En examinant de manière plus détaillée le travail des élèves de chaque groupe avec le « magnétoscope » (cf. 1) on a constaté de fortes inhomogénéités comportementales dès que l'on s'éloignait des centroïdes (notamment dans le groupe a), c'est-à-dire que la présence de certains élèves était difficile à expliquer au sein de ces groupes. Cela est le signe de variables mal choisies.

En effet, en y regardant de plus près, on voit que les variables renseignées à partir des protocoles bruts tiennent compte de toutes les lignes, y compris celles relatives à des actions temporaires vides de sens algébrique. Par exemple, pour écrire $21x^2$, l'élève va frapper successivement sur les touches 2, 1, x, ^, 2 et passer par les expressions intermédiaires suivantes : $2 \rightarrow 21 \rightarrow 21x \rightarrow 21x^ \rightarrow 21x^2$. Certaines de ces expressions sont mal formées et la majorité contient peu d'informations, donc toutes les variables de décomptes d'actions sont perturbées. Tous les élèves passent par ce style d'expressions transitoires : ceci explique la plus grande difficulté de cette expérience à dégager de nettes catégories d'élèves.

Aussi, nous avons introduit une étape de prétraitement qui vise à « condenser » les protocoles bruts de manière à supprimer toutes

ces actions intermédiaires et à garder de manière automatique des expressions plus pertinentes. Cela nous a conduit à concevoir un programme PERL effectuant un résumé de protocoles, afin d'effectuer une classification sur des données plus significatives.

3.2 – Mécanismes de résumé

Nous avons mis au point des heuristiques de transformations qui remplacent une ou plusieurs lignes de protocole en une nouvelle ligne (on parle alors de condensation) éventuellement vide (on parle alors de nettoyage). Ces transformations sont appliquées successivement sur un protocole brut, dans l'ordre précisé ci-après.

1. Nettoyage des actions non modifiantes (clics souris, déplacements de curseur...)

2. Nettoyage des « mono-backspaces » : lorsqu'un élève fait une erreur de frappe immédiatement corrigée, on ne garde que le résultat de sa correction.

Exemple : 8, 4, Bcksp, 2 -> 8, 2

3. Condensation des frappes successives : agrégation d'une série ininterrompue d'actions considérées comme frappes additives.

Exemple : 2, 1, x, ^, 2 -> « $21x^2$ »

4. Condensation des répétitions : agrégation d'une répétition d'actions identiques. Exemple : Backspace, Backspace, Backspace -> « Backspace (x3) »

5. Nettoyage des séries de Backspaces entre deux séries de frappes successives : celles-ci produisant systématiquement des expressions intermédiaires (souvent mal formées) constituant des points de départ pour de nouvelles séries d'éditations, elles ne sont pas significatives d'un point de vue algébrique.

L'application successive de ces 5 transformations à un protocole brut produit un protocole résumé. En moyenne, sur toutes les données traitées, on constate une réduction de la taille du protocole à environ un cinquième de la longueur de l'original.

3.3 - Résultats et analyse avec les protocoles résumés

En n'utilisant que les 12 premières dimensions, mais cette fois sur les protocoles résumés au lieu des protocoles bruts, on obtient 5 groupes

(totalisant 96 élèves) beaucoup plus homogènes en termes de comportements :

a) Les succès rapides (27) : Ceux-là sont plus rapides que le groupe du même nom trouvé en 3.a, et passent par moins d'expressions intermédiaires non équivalentes. Ils sont moins nombreux et beaucoup plus homogènes.

b) Les succès lents (12) : ces élèves agissent plus et sont plus lents que a), mais finissent par faire juste. Ils n'étaient pas détectés précédemment, ce qui conforte l'utilité du procédé de résumé de protocole.

c) Les succès actifs (16) : la description est identique à 3.2.1

d) Les échecs par faute de parcours (26)

e) Les échecs lents (16)

Pour caractériser les classes, nous avons visionné au « magnétoscope » les protocoles de quelques élèves de chaque catégorie obtenue. Dans le cas de l'ensemble d), on constate que dans un même groupe d'élèves ayant fait faux, les élèves font des erreurs radicalement différentes. Exemples :

- $9x^* - 6x \rightarrow -64x^2$ (erreur de calcul)
- $9x^* - 6x \rightarrow -54x$ (oubli de la puissance)
- $9x^* - 6x \rightarrow -54$ (oubli de la variable)

Or, en introduisant des variables qualitatives portant sur les transformations effectuées par l'élève on peut facilement améliorer la classification, en faisant apparaître de nouvelles catégories correspondant à des groupes d'élèves aux comportements plus homogènes. Dans cette optique, les 6 variables de qualifications supplémentaires décrites en 2.c constituent de bons candidats.

3.4 – Résultats obtenus avec protocoles résumés et variables de qualification

Cette fois-ci, nous utilisons pour la classification la totalité des 18 variables décrites en 2.c. On obtient un découpage encore plus fin et homogène que le précédent, concernant plus d'élèves (129), bien que ce chiffre reste encore largement améliorable :

a) Les succès rapides (29)

b) Les succès actifs (16)

c) Les succès lents (10)

d) Les échecs pour erreurs diverses (28)

e) Les échecs pour erreur de développement polynômial (17)

f) Les erreurs de développement (11)

g) Les erreurs d'antiefonnement (11)

h) Les erreurs de calcul (7)

On constate que les groupes « élèves ayant fait une erreur de parcours » et « mauvais laborieux » (groupes d et e du 3.4) se retrouvent maintenant répartis dans les 5 groupes d'élèves ayant échoué détectés ici (d, e, f, g, h), et que cette classification concerne aussi de nouveaux élèves, grâce à l'ajout des nouveaux traits. On voit donc émerger des groupes encore plus homogènes en termes de comportements, pour lesquels le type d'erreur commise est identifié.

Par ailleurs, nous avons étudié l'influence des traits de qualification sur la stabilité des catégories détectées, en comparant les résultats obtenus en utilisant ou non ces traits lors de la classification. Pour cela, nous avons utilisé la matrice des cardinaux des intersections ensemblistes des deux familles de clusters : chaque case (i,j) de cette matrice contient le nombre d'éléments communs entre le cluster i de la 1ère classification et le cluster j de la 2e. Nous constatons que le groupe des bons (a) est bien détecté dans les deux cas (93% d'élèves ensemble en utilisant les qualifications se retrouvent ensemble sans les utiliser) et que les bons actifs sont relativement stables (50%), de même que les élèves ayant fait faux pour des raisons diverses (82%). Par contre, les groupes d'élèves ayant fait des erreurs identifiées (e,f,g,h) sont totalement absents d'une classification sans traits de qualification, ce qui montre l'utilité de ces traits pour notre objectif de classification comportementale.

Cependant, il faut remarquer que l'on mélange ici des variables quantitatives (nombre d'actions, temps...) avec des variables plus qualitatives (qualifications). En agissant de la sorte, et ce à cause du fonctionnement même des algorithmes de classification (basés sur des distances euclidiennes dans des espaces de haute dimension), des individus sont regroupés dans une même catégorie s'ils ont le même profil en termes à la fois d'utilisation d'Aplusix (nombre d'actions, d'étapes), de temps (rapides, lents), et d'erreurs effectuées (qualifications fausses). Par conséquent, si d'un côté on observe un gain certain dans

l'ajout de ces variables de « qualifications fausses » (une structuration plus forte des données), il y a aussi un effet négatif : un reclassement brutal de certains élèves passant temporairement par une expression erronée, mais dont la qualification est identifiée et peu fréquente.

4 – CONCLUSIONS ET PERSPECTIVES

Nous avons présenté les résultats de travaux visant à constituer automatiquement des catégories d'élèves à partir de séquences d'actions. Le travail effectué vise en grande partie à transformer ces séquences afin de faciliter l'analyse et de pouvoir s'appuyer sur des techniques de classification automatique. Nous avons réussi à améliorer la pertinence des catégories produites en résumant les protocoles et en introduisant des variables plus qualitatives qui permettent de caractériser les transformations faites par les élèves. Plusieurs directions de recherche se dessinent.

Premièrement, nous comptons tester l'influence de la granularité avec laquelle on regarde la prestation de l'élève. Toute la difficulté réside dans la détection (et la prise en compte) des expressions clés de l'élève. La condensation effectue un premier filtrage, mais est peut-être encore trop « laxiste » en conservant certaines lignes « brouillons ». D'autre part, si on ne regarde que les transformations d'étape en étape (un grain plus gros), on risque de rater des expressions révélatrices de conceptions erronées de l'élève. En nous munissant d'un évaluateur de la qualité de la condensation, nous pourrions affiner nos mécanismes de condensation.

Deuxièmement, nous comptons accentuer nos efforts dans la constitution des variables servant à la classification. En premier lieu, utiliser des ensembles de variables descriptives différents selon le type d'exercice nous paraît désormais une nécessité, car les types d'erreurs à identifier dépendent fortement de la tâche à effectuer. De plus, les didacticiens du projet ont travaillé à la constitution d'une base de règles erronées (elles correspondent à des conceptions fausses connues du type : $(a+b)^2 \rightarrow a^2 + b^2$), qui est désormais intégrée à la base de règles correctes existant dans Aplusix. Ainsi, la phase de complétion est en train de s'enrichir d'un nouvel outil (encore à l'étude), capable de construire (si possible)

un diagnostic, i.e. une séquence de règles correctes ou erronées permettant « d'expliquer » le passage d'une expression initiale à une expression finale, et ce de manière plus précise que les qualifications. Des expériences préliminaires en cours, où nous avons basé notre classification sur des variables issues de ces diagnostics, donnent des résultats encourageants. En effet, de tels indicateurs à forte sémantique mathématique et didactique, semblent faire ressortir de plus nettes catégories d'élèves.

Notre but est de constituer à partir de notre corpus de données des « cartes topologiques d'élèves », différentes selon le type de tâche (développer, factoriser...), permettant de situer tout nouvel élève en fonction des difficultés rencontrées. Nous pensons que la classification automatique est un outil exploratoire idéal pour découvrir les variables pertinentes permettant de bien détecter les catégories d'élèves aux comportements homogènes, et ce pour chaque type d'exercice. Ces analyses fourniraient une base de connaissances servant de support à un tuteur artificiel, qui proposerait à l'élève de nouveaux exercices ciblés en fonction de son appartenance à une catégorie identifiée. Une application en e-learning sera envisageable à plus long terme, si nos travaux permettent d'identifier avec un bon taux de réussite le groupe de niveau de connaissances auquel appartient l'élève, et si des réponses didactiques appropriées sont mises au point pour chaque type de difficulté rencontrée.

BIBLIOGRAPHIE

- Anderson, J.R., Reiser, B. (1985), "The LISP Tutor". *Byte n°10*, p. 159-175.
- Baffes, P., Mooney, M. (1996), "Refinement-based student modeling and automated bug library construction". *Journal of AI in Education, Vol 7(1)*, p. 75-116.
- Beck, J. E., Woolf, B.P. (2000), "High-level Student Modeling with Machine Learning". *Intelligent Tutoring Systems*, p. 584-593.
- Berkhin, P. (2002), "Survey of Clustering Data Mining Techniques". *Technical Report*.

- Bouhineau, D., Nicaud, J.F., Pavard, X., Sander, E. (2001), "Un micromonde pour aider les élèves à apprendre l'algèbre". *6e journées francophones Environnements Interactifs d'Apprentissage avec Ordinateur*.
- Blatt, M., Domany, E., Wiseman, S. (1997), "Data Clustering using a model granular magnet". *Neural Computation* 9.
- Bronner, A., Bouhineau, D., Chaachoua, H., Huguet, T. (2003), "Analyse didactique de protocoles obtenus dans un EIAH en algèbre ». *Actes de EIAH Strasbourg*, p. 79-90.
- Burton R. (1982), "Diagnosing Bugs in a simple procedural skill». *Intelligent Tutoring Systems*, Eds D. Sleeman & L. Brown, London Academic Press.
- Diday E. (1973), "The dynamic cluster method in non-hierarchical clustering ». *Journal of Comput. Inf. Sci.* 2, p. 61-88.
- Duda, R.O., Hart, P.E., Stork, D.G. (2001), *Pattern Classification (2nd edition)*. John Wiley & Sons, p. 561-562
- Ester, M., Kriegel, H.P., Sander, J., Xu, X. (1996), "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". *2nd Int. Conf. on Knowledge Discovery and Data-mining*.
- Gélis, J.M. (1994), *Éléments d'une théorie cognitive et computationnelle de l'algèbre*. Thèse à l'Université Paris-Sud/ Orsay.
- Johnson, WL (1990), "Understanding and debugging novice programs". *AI n°42*, p. 51-97.
- Kaufman, L., Rousseuw, P.J. (1990), *Finding Groups in Data : An introduction to Cluster Analysis*. John Wiley & Sons.
- Langley, P., Ohlsson, S. (1984), "Automated cognitive modeling". *Second National Conference on AI*.
- Lebart, L., Morineau, A., Piron, M. (2000), *Statistique exploratoire multidimensionnelle* (chap 2.2). Dunod, Paris (3e éd).
- Sison, R., Shimura, M. (1998), "Student Modeling and Machine Learning". *Int. Journal of AI in Education*, n°9, p. 128-158.
- Witten, I.H., Frank, E. (1999), "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann.

Remerciements: Bisson G, Gordon M.B, Nicaud JF