IHM

Gamification adaptative

Marion Vertessen et Yohan Michelland

Introduction

Ce rapport se rapporte à un travail pratique sur la gamification adaptative. Il est basé sur les travaux menés dans le cadre du projet e-FRAN LudiMoodle et de la thèse de Stuart Hallifax. Il s'agit d'une application numérique permettant de réaliser des exercices de mathématiques en classe de collège.

Première partie : Recommandation à partir de profil

Étape 1 :

Dans cette partie, on souhaite décrire deux des matrices de résultats de l'analyse PLS (Partial least squares analysis). Nous avons décidé de prendre la matrice Timer pour le profil Hexad représenté dans le tableau 1.

	achiever	player	socialiser	freeSpirit	disruptor	philanthropist
MIVar	0.47550491	0.71752931	0.85232795	0.76701909	0.71887501	0.651149167
MEVar	0.18603618	0.75388051	0.28305879	0.43322525	0.94630133	0.110116577
AMOTVar	0.05963662	0.90661762	0.45473599	0.27640043	0.82734347	0.036860063

Tableau 1 : Hexad pour les TimerpValues

On remarque que la mise en place de timer à une influence sur l'AMotivation. L'AMotivation correspond au niveau de l'absence de motivation liée au sentiment de ne pas se sentir capable de réaliser une tâche. Cette variation d'AMotivation est notable chez les profils de joueurs "achiever" et "philanthropist".

On utilise le tableau 2 pour déterminer si cet impact est négatif ou positif.

	achiever	player	socialiser	freeSpirit	disruptor	philanthropist
MIVar	0.1984098	0.07716076	0.0422975	0.06440708	-0.0673409	-0.128333302
MEVar	0.3607882	-0.0650107	0.2391139	-0.1662927	0.01224112	-0.448526193
amotVar	0.4904743	-0.0228988	0.1561808	0.21846641	-0.0373533	-0.559363613

Tableau 2 : Hexad pour les TimerPath

On remarque que le profil de joueur "achiever" est influencé positivement au niveau de l'amotivation. En effet, il y a une augmentation de l'amotivation. On a donc une perte de motivation chez les joueurs de type "achiever" lorsque l'on utilise un timer.

De plus, l'amotivation du profil de joueur "philantropist" est influencée négativement par l'ajout d'un timer. On a donc une diminution de l'amotivation au cours du temps.

On s'intéresse maintenant à la matrice de motivation pour la mise en place d'un avatar par rapport au niveau de motivation initiale.

	MI	ME	amotI
MIVar	0.223024943	0.150608651	0.004673462
MEVar	0.03813866	0.230983359	0.588496236
AMOTVar	0.228728342	0.256401288	0.776439204

Tableau 3 : P-values des motivations pour les avatars

On remarque que plus le joueur est amotivé initialement, plus la variation de la motivation intrinsèque est importante. De plus, on remarque également que plus le joueur est motivé intrinsèquement initialement, plus la variation de la motivation extrinsèque est importante.

On utilise le tableau 4 pour déterminer si cet impact est positif ou négatif.

Colonne1	MI	ME	amotI
MIVar	-0.18754257	0.198529314	0.42314114
MEVar	0.407912547	0.207203747	-0.09716436
amotVar	0.24758242	0.208581957	0.054161957

Tableau 4 : P-values des motivations pour les avatars

On remarque que plus l'amotivation est importante au début, plus il va y avoir le développement de la motivation intrinsèque lorsque l'on ajoute l'avatar au joueur. De plus, plus le joueur est motivé intrinsèquement initialement, plus sa motivation extrinsèque va être importante lorsque l'avatar est ajouté à l'environnement.

Étape 2 :

On souhaite construire un vecteur d'affinité. Ce tableau est de taille six et chaque case correspond à un élément ludique. Les éléments ludiques sont ordonnés du plus au moins adaptés. De plus, chaque élément ludique est associé à un score d'affinité. Un élève possède deux vecteurs d'affinité :

- un selon son profil Hexad
- un selon son profil de motivation (motivation initiale)

Ces deux vecteurs sont calculés à partir d'un fichier contenant différents élèves. On prend en considération les différentes influences de ce fichier.

En effet, un même élève possède une valeur pour chacun des profils différents :

- Achiever
- Player
- Socialiser
- FreeSpirit
- Disruptor
- Philanthropist

Par exemple, l'élève bf01, possède les valeurs présentes dans le tableau 5.

Achiever	Player	Socialiser	FreeSpirit	Disruptor	Philanthropist
0	6	4	5	1	2

Tableau 5 : Valeur Hexad pour l'élève bf01

De fait, afin de calculer le vecteur d'affinité, nous avons décidé de calculer un score pour chaque élément ludique. Ce score est calculé à partir de la variation de motivation intrinsèque, extrinsèque et de l'Amotivation présent dans les fichiers Hexad qui nous ont été fournis.

Tout d'abord, on considère un élément ludique pour lequel on va regarder les p-values. Si la p-value est inférieur à 0.1, on considère que son influence est significative, sinon on considère qu'elle ne l'ai pas et on ne la prend pas en compte. Enfin on fait une somme pondérée selon les valeurs de différents profils de l'élève.

Dans un premier temps, on s'intéresse au profil de motivation initiale des élèves.

La motivation intrinsèque initiale et la motivation extrinsèque initiale sont données dans trois colonnes différentes. Cependant, l'amotivation est décrite dans une seule colonne. De fait, on fait la somme des trois colonnes. On obtient donc :

- $MI_{initial} = micoI + miacI + mistI$
- $ME_{initial} = meidI + meinI + mereI$
- $AMot_{initial} = amotI$

Enfin, à l'aide des données Path contenant la variation des motivations, on peut en déduire le score que l'on calcule de la manière suivante :

$$score = MI + ME - AM$$

Avec :

- MI : la motivation intrinsèque qui résulte de la multiplication de $MI_{initial}$ et de la valeur du path correspondante si $p - value < 0.1$
- ME : la motivation extrinsèque qui résulte de la multiplication de $ME_{initial}$ et de la valeur du path correspondante si $p - value < 0.1$
- AM : l'amotivation qui résulte de la multiplication de $AMot_{initial}$ et de la valeur du path correspondante si $p - value < 0.1$

Dans un second temps, afin de calculer le vecteur d'affinité correspondant au profil du joueur Hexad, on prend en compte les valeurs pour chaque profil du joueur ainsi que les valeurs du path si $p - value < 0.1$ et on utilise le même raisonnement qu'à la question précédente, mais avec les types de joueurs.

Avec cette formule, le score est calculé de manière à prendre en compte le souhait d'augmenter la motivation intrinsèque et extrinsèque et de réduire l'amotivation.

Par exemple pour l'élève bf01, on obtient le vecteur d'affinité concernant la motivation initiale :

```
[['Score', 4.2518780978531866], ['Badge', 1.7064270997569597], ['Avatar',
-0.016315077944072165], ['Ranking', -2.8223681009049795], ['Progress',
-3.220658536253221], ['Timer', -6.1668490461433905]]
```

Et le suivant concernant le profil Hexad :

```
[['Avatar', 2.994505106494738], ['Score', 1.9418581276818219], ['Timer',
1.118727225895414], ['Ranking', 0], ['Progress', -0.08161848109909603],
['Badge', -0.11095747605729511]]
```

Deuxième partie : Algorithme d'adaptation

Cette partie a pour but de définir un algorithme qui permet des recommandations spécifiques à différents profils. Cet algorithme devra à la fois prendre en compte les recommandations pour le profil de joueur Hexad d'un élève et les recommandations pour sa motivation initiale.

Étape 1 :

On réfléchit dans un premier temps sur une série d'exemples afin de lister les différents cas à considérer.

Eleve	Vecteur Hexad	Vecteur Motivation
elevebf01	Avatar : 2.99 Score : 1.94 Time : 1.11 Ranking : 0 Progress : -0.08 Badge : -0.11	Avatar : 16.87 Score : 7.2 Timer : 6.16 Ranking : 2.82 Badge : -1.7 Progress : -12.46
elevebg01	Timer : 7.2 Ranking : 0 Progress : -0.13 Badge : -0.19 Avatar : -4.16 Score : -10.4	Avatar : 16.5 Timer : 9.86 Score : 5.1 Ranking : 4.5 Badge : -2.7 Progress : -10.22

elevebg04	Ranking : 0 Badge : -0.08 Avatar : -1.86 Timer : -2.24 Progress : -2.6 Score : -3.45	Avatar : 23.3 Score : 10.5 Timer : 2.46 Ranking: 1.12 Badge : -0.68 Progress : -20.38

Tableau 6 : Vecteurs d'influences pour différents types d'élèves

On retrouve dans le tableau 6, les vecteurs d'influences calculées à la question précédente. On retrouve différents cas dans ce tableau :

- L'élève bf01 a une recommandation très élevée de l'élément ludique "Avatar" par le vecteur de motivation et par le vecteur d'Hexad. De fait, dans un cas similaire, il semble pertinent de recommander cet élément ludique.
- L'élève bg01 a pour recommandation par le vecteur d'affinité Hexad, l'utilisation ludique de l'élément "Timer". Cependant, le vecteur d'affinité relié à la motivation propose plutôt l'utilisation de l'avatar. Le timer est positionné au second rang pour le vecteur de motivation tandis que l'avatar est positionné en avant-dernière position pour le vecteur Hexad. De fait, dans un tel cas, il semble plus pertinent de recommencer l'élément ludique Timer puisqu'il semble satisfaire le profil Hexad et de motivation de l'élève.
- L'élève bg04 ne possède que des scores négatifs ou nul sur la recommandation d'un élément ludique pour son profil Hexad. Son profil de motivation lui recommande l'utilisation de l'avatar. On remarque que l'avatar a un score négatif qui reste néanmoins acceptable. On pourrait donc sélectionner cet élément dans la recommandation.

Étape 2 :

À l'aide de cette étude, nous allons maintenant pouvoir choisir une stratégie de compromis qui permette de recommander un élément de jeu approprié à chaque élève. Nous avons imaginé construire un algorithme qui suit les règles suivantes :

- Si un ou plusieurs éléments ludiques ont un score positif pour le vecteur d'affinité Hexad et de motivation, on calcule la somme des scores et on prend l'élément ludique ayant le plus grand.
- S'il n'y a aucun élément ludique se trouvant dans les deux vecteurs avec un score positif. On prend l'élément qui possède le score le plus important lorsque l'on somme le score Hexad et de motivation.
- Si aucun élément ludique n'est présent avec un score positif, on fait la somme des scores et on prend celui qui a une somme le plus élevé.

On peut donc réaliser le pseudo-code suivant :

```
LISTE_SCORE_POSITIF = []
LISTE_SCORE_MEDIUM = []
LISTE_SCORE_NEGATIF = []
FOR ELEMENT in ELEMENT_LUDIQUE :
IF ELEMENT a deux scores positifs :
```

```
      Ajout ELEMENT à LISTE_SCORE_POSITIF
      IF ELEMENT a un score positif et un score negatif
            Ajout ELEMENT à LISTE_SCORE_MEDIUM
      IF ELEMENT a deux scores négatifs :
            Ajout ELEMENT à LISTE_SCORE_NEGATIF
IF LISTE_SCORE_POSTIF est non vide :
      FOR ELEMENTP in LISTE_SCORE_POSITIF:
            score = score_hexad(ELEMENTP) + score_motivation(ELEMENTP)
      On propose élément possédant le plus grand score
ELSE IF LISTE_SCORE_MEDIUM est non vide :
      FOR ELEMENTM in LISTE_SCORE_MEDIUM:
            score = score_hexad(ELEMENTM) + score_motivation(ELEMENTM)
On propose élément possédant le plus grand score
ELSE IF LISTE_SCORE_NEGATIF est non vide :
      FOR ELEMENTN in LISTE_SCORE_NEGATIF:
            score = score_hexad(ELEMENTN) + score_motivation(ELEMENTN)
On propose élément possédant le plus grand score
```

Étape 3 :

On écrit le pseudo-code présenté précédemment dans la fonction algo_adaptation. Elle retourne l'élément de jeu qui a été recommandé par l'algorithme.

Étape 4 :

Dans cette étape, on cherche à évaluer la pertinence de notre algorithme d'adaptation.

Dans un premier temps, nous avons séparé les élèves en deux groupes :

- Le premier contient les élèves pour lequel l'élément ludique est celui prédit par notre algorithme de prédiction
- Le deuxième groupe contient les élèves pour lesquels l'élément ludique n'est pas celui prédit par notre algorithme.

Afin de déterminer la pertinence de notre algorithme d'adaptation. Nous avons décidé de faire la moyenne d'une série de ratios selon la formule suivante :

$$satisfaction = (ratio_{question} + ratio_{temps} + ratio_{qizz} + ratio_{motivation}) / 4$$

Avec :

- $ratio_{question} = \frac{CorrectCount}{QuestionCount}$
- $ratio_{temps} = \frac{temps\ passé\ sur\ l'application}{Max\ des\ temps\ passé}$
- $ratio_{quizz} = \frac{PassedQuizCount}{QuizCount}$
- $ratio_{motivation} = \frac{(micoVar + miacVar + mistVar + meidVar + meinVar + mereVar - amotVar) + 46}{94}$

En effet, la valeur -46 correspond à la somme des variations minimale et 94 correspond à la somme en valeur absolue de la valeur de la somme minimale et maximale sur l'ensemble du dataset. (valeurs trouvés à l'aide d'excel)

Cependant, en utilisant une telle formule et le t-test, on obtient une valeur de p-value égale à 0.64. Ce résultat est catastrophique et ne permet donc pas de confirmer la validité de notre algorithme.

Néanmoins, si on utilise uniquement la somme des variations de la motivation intrinsèque, extrinsèque et de l'amotivation, on obtient une p-value de 0.16. Cela confirme que notre algorithme semble plutôt bien fonctionner. En effet, la variation des motivations prouve bien que l'élément ludique adapté a eu un impact sur l'apprentissage des élèves.

De plus, cette variation est plus encourageante que pour les élèves ayant eu un élément ludique non adapté.

En effet, la variation moyenne de la variation des motivations pour les élèves ayant eu l'élément recommandée est de -4.1 tandis que pour les autres elle est de -7.1.

Conclusion

Nous n'avons pas réussi à obtenir des résultats notables. En effet, nous avons une p-value qui vaut 0.16 tandis qu'il faudrait qu'elle soit inférieur à 0.05 pour que la différence soit notable. Cette imprécision peut provenir de différents facteurs. Voici nos hypothèses :

- La prise en compte de valeur non significative pour l'élaboration puisque pour avoir un nombre de données assez important, nous avons pris comme étant significative les variations ayant une p-value < 0.1 et non à 0.5.
- Notre algorithme de recommandation pourrait ne pas être correct et ne permettrait donc pas d'évaluer correctement l'élément ludique adapté.
- La construction de notre vecteur d'affinité pour les différents types de profils pourrait ne pas répondre convenablement à la problématique.