

Travaux pratiques sur l'apprentissage supervisé : Prédire qui se cache derrière un avatar d'un jeu-video

Sujet proposé par Mehdi Kaytoue – MEHDI.KAYTOUE@INSA-LYON.FR

Remise à niveau Séminaire Data Analytics – IF-5 – 2017-2018

1 Prise en main

Cet exercice permet de s'acclimater avec KNIME. On utilise le fichier de données IRIS.ARFF vu en cours et on construit un modèle de prédiction de l'espèce de la plante. Pour cela, on charge un projet KNIME préparé par nos soins. Vous devez comprendre ce que ce KNIME réalise et comment. Vous ne pourrez passer à la suite que lorsque vous aurez maîtrisé et adapté à votre convenance ce premier “*workflow*”. Le fichier à importer est TP-SUPERVISED-IRIS.KNWF.

2 Contexte



FIGURE 1 – Une finale d'esport.

Les tournois sont diffusés par des commentateurs à travers des TV sociales suivies en grand nombre [4, 2]. Comme tout “sport”, l'esport n'échappe pas aux problèmes de dopage² et de triche : la majeure partie des compétitions (et/ou phases qualificatives) se déroulent en ligne (*online*) et il est donc un problème de vérifier l'identité du joueur : *Est-ce bien la personne inscrite au tournoi qui se cache derrière son écran ?*.

Le “Sport électronique” : du sport et ... quelques tricheurs ! L'industrie du jeu vidéo a tiré grand profit de la digitalisation de notre société, générant des milliards d'euros par an et attirant des millions d'adeptes. Ce contexte a permis de révéler un phénomène : le jeu vidéo compétitif. Le *sport électronique* est caractérisé par la pratique extrême des jeux vidéos, où des cyber-athlètes s'affrontent lors de compétitions internationales sur site ou sur le Web (avec des récompenses atteignant le million de dollar)¹. Comme pour les sports classiques, ils sont entourés d'une équipe, d'entraîneurs et de sponsors.

3 Objectif

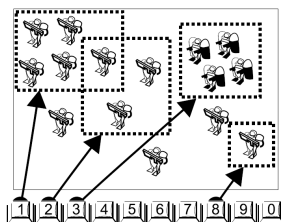


FIGURE 2 – Le système d'association entre touches et groupes de contrôle.

Existe-t-il des signatures comportementales ? Notre but est d'étudier s'il est possible d'apprendre un modèle qui serait capable de prédire quel est le joueur d'une partie dont on aurait enregistré toutes les actions. Ainsi, la manière d'enchaîner les actions aux claviers et souris serait-elle suffisante pour *apprendre le comportement* d'un joueur ? Peut-on expliciter une *signature numérique* ?

StarCraft 2. L'une des licences phares de l'esport est StarCraft II. Pendant le déroulement de la partie, les deux joueurs s'affrontent sur une carte de jeu (en vue aérienne), en contrôlant des bâtiments et des unités pour rassembler des ressources nécessaires à la construction de bâtiments et d'une armée afin de détruire la base adverse et ainsi gagner la partie³. Pour améliorer leur efficacité, les joueurs peuvent créer des *groupes de contrôle* : ils peuvent associer

1. [http://www.lequipe.fr/explore/generation-esport/#xtor=CS5-135-\[explore-generation-esport\]](http://www.lequipe.fr/explore/generation-esport/#xtor=CS5-135-[explore-generation-esport])

2. <http://rue89.nouvelobs.com/rue89-sport/2015/07/20/dopage-sinvite-partout-meme-sport-260340>

3. http://fr.wikipedia.org/wiki/Jeu_de_stratégie_en_temps_réel

aux touches 0 à 9 un groupe d'unités quelconque. Dès lors, l'appui d'une seule touche leur permet de sélectionner un groupe d'unité particulier. En fait, nous faisons l'hypothèse que la création, l'utilisation et la mise à jour de ces touches révèlent le comportement d'un joueur [5]. On se propose de le vérifier.

4 Données

Nous étudions un ensemble de 955 parties de joueurs professionnels qui se sont déroulées au cours du championnat WCS 2014. Nous avons accès au *logs* de ces parties, qui enregistrent chaque action, le joueur qui l'a opérée ainsi que son estampille temporelle. Chaque partie donne lieu à deux enregistrements (dits objets), un pour chaque joueur, et ces enregistrements ont les attributs (ou variables) suivants :

- USER : identifiant du joueur dans une partie
- RESULT : prend la valeur *win* si le joueur a gagné, *loss* sinon
- RACE : faction du joueur (type d'unités choisies) (*zerg*, *protoss*, *terran*)
- S : nombre de fois que le joueur a réalisé une sélection à la souris (unités, bâtiments, ...)
- BASE : nombre de fois qu'un bâtiment très spécial a été sélectionné
- SINGLEMINERAL : le nombre de fois que le joueur sélectionné un "carré de minerai".
- HOTKEYX0 : nombre de fois que le groupe de contrôle associé à la touche $X \in [0; 9]$ a été créé
- HOTKEYX1 : nombre de fois qu'un groupe de contrôle associé à la touche $X \in [0; 9]$ a été mis-à-jour
- HOTKEYX2 : nombre de fois qu'un groupe associé à la touche $X \in [0; 9]$ a été utilisé
- APM : NOMBRE D'ACTIONS PAR MINUTE MOYEN DU JOUEUR

5 Construire et évaluer un modèle de prédiction

Votre objectif est de construire et évaluer un modèle qui permet de prédire le joueur (USER) à partir des autres attributs. Pour cela, on suivra les instructions suivantes.

1. **Lecture et compréhension des données.** A l'aide du nœud ARFFREADER, on lira le fichier STARCRAFT-A.ARFF. Combien y-a-t-il d'instances (objets/enregistrement) ? Combien d'attributs (variables) ? Pour chacune, on donnera son type, son domaine de valeur ainsi que différentes informations jugées pertinentes (distributions des valeurs par exemple).
2. **Visualisation** On utilise maintenant les nœuds COLOR MANAGER et COLOR APPENDER pour colorier chaque instance en fonction de la classe à prédire USER. On pourra alors visualiser via des histogrammes ou nuages de points (nœuds SCATTER PLOT et SCATTER MATRIX) la répartition des classes en fonctions des attributs. Comme pour le jeu de données IRIS.ARFF, on pourra alors trouver des pistes pour la construction d'un modèle de prédiction.
3. **Jeu d'apprentissage et jeu de test.** On voudra construire le modèle sur une partie du jeu de données et on l'évalue sur le reste. Pour cela, on utilise le nœud PARTITIONNING.
4. **Construction du modèle.** On veut apprendre un arbre de décision. Pour l'apprentissage, et donc pour le jeu d'apprentissage créé en 3., on utilise le nœud DECISION TREE LEARNER. Il s'agit d'une version de l'algorithme de Hunt vue en cours. On laissera les paramètres par défaut dans un premier temps.
5. **Évaluation et amélioration du modèle.** Le modèle étant appris, nous pouvons l'utiliser pour prédire le joueur des enregistrements restants, ceux mis de côté en 3.. On demande donc au modèle de prédire la classe, même si on la connaît déjà : on peut alors vérifier comment le modèle s'est comporté. Pour cela, on utilise le nœud SCORER qui nous donne les taux de faux positifs, vrais positifs, ... ainsi que les mesures de précision, rappel et F-measure. Vous devez alors forger votre opinion sur la qualité du modèle produit et expliquer chacune de ces mesures. On pourra aussi afficher la matrice de confusion via une matrice de couleur (nœud JFREECHART HEATMAP). On cherchera à modifier les paramètres de l'algorithme de construction d'arbre de décision pour obtenir un meilleur score (par exemple le taux de classification réussies ou *accuracy*).
6. **Compréhension du modèle.** Un certain nombre de modèles sont explicatifs : ils donnent des règles simples qui expliquent chaque prédiction. Les arbres de décision en sont l'exemple par excellence. Afficher l'arbre appris, jugez de sa simplicité/complexité et des règles que l'on peut expliciter.
7. **Choix des attributs/variables.** A cette étape, vous devriez avoir obtenu un modèle qui fait peu d'erreur et dont on peut supposer qu'il sera bon avec de nouvelles données. Cependant, les variables pour l'apprentissage vous étaient données et construite à partir de données brutes et massives : chaque joueur réalise des milliers d'actions dans une partie et nous avons choisi de vous donner un agrégat particulier selon deux paramètres : θ le nombre minimum d'instance de chaque joueur,

et τ le nombre maximum de seconde depuis le début de la partie. Le jeu de données que vous avez utilisé jusque maintenant avait un paramétrage τ et θ particulier. Dans le dossier *starcraft-all* vous avez différents jeu de données en fonction de plusieurs pairs de paramètres. Chaque nom de fichier TYYTHETAZZ.ARFF donne un jeu de données pour lequel les variables sont mesurées pour les $\tau = YY$ premières secondes et pour lesquels seuls le joueurs USERS ayant au moins $\theta = ZZ$ instances dans les données brutes sont gardés (le nombre d'instances varie donc d'un fichier à l'autre). Vous allez pouvoir tester différents paramétrages et dresser une conclusion : quelles sont les meilleures conditions pour répondre à notre problématique de prédiction ? Pour cela, sélectionner votre flux sans le nœud de lecture et d'évaluation et grouper le en un *meta-nœud*. Vous pouvez alors copier ce meta-nœud plusieurs fois et lui connecter en entrée un fichier ARFF différent.

8. **Mise en situation plus réelle.** Pour cette question, on considère le jeu de données nommé STARCRAFT-B.ARFF seulement. Les objets et attributs sont toujours de la même nature. Tentez de construire un modèle à partir de ce nouveau jeu de données. Discuter les mesures de précision/rappel/F1-mesure. Le modèle est-il bon ? Observez la matrice de confusion. Que pouvez-vous conclure ? Imaginez une méthode qui pourrait solutionner le problème. L'étudiant intéressé pourra trouver quelques propositions récentes [5, 1, 3].
9. **Question optionnelle 1 : modèles explicatifs et boîte-noire.** Pour (i), vous pouvez changer les paramètres de l'algorithme d'apprentissage, ou même changer la méthode d'apprentissage. On pourra utiliser les nœuds KNN, PROP MLP LEARNER, MULTILAYERPERCEPTRON PREDICTOR, SVM LEARNER, SVM PREDICTOR. Pour (ii), on pourra mettre en place une technique de validation croisée (nœuds X-PARTITIONNER, X-AGGREGATOR, voir le tutoriel *002003* dans les exemples fournis par KNIME, onglet *Knime Explorer*). Cette question est optionnelle dans le cadre du TP.
10. **Question optionnelle 2 : Variation automatique des paramètres.** KNIME offre des nœuds de contrôle pour opérer des traitement itératifs ou encore conditionnels pendant l'exécution d'un flux. Il est donc possible de traiter de manière automatique tous les fichiers du dossier STARCRAFT-ALL.ZIP en récupérant les paramètres dans le nom de fichiers. Au final, on peut dresser un graphique où varie la précision/rappel/F-mesure en fonction des deux différents paramètres. On utilisera pour cela les nœuds LIST FILES, JAVA SNIPPET, TABLE ROW TO VARIABLE LOOP, LOOP END COLUMN APPEND, TRANSPOSE. Cette question n'est à traiter que par les passionnés.
11. **Encore ?** Mon challenge de prédiction KAGGLE réouvre bientôt !
<https://www.kaggle.com/c/starcraft-2-player-prediction>

Références

- [1] O. Cavadenti, V. Codocedo, J. Boulicaut, and M. Kaytoue. When cyberathletes conceal their game : Clustering confusion matrices to identify avatar aliases. In *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, pages 1–10. IEEE, 2015.
- [2] M. Kaytoue, A. Silva, L. Cerf, W. Meira Jr., and C. Raïssi. Watch me playing, i am a professional : a first study on video game live streaming. In *Proceedings of WWW'12 (Companion Volume)*, pages 1181–1188. ACM, 2012.
- [3] Q. Labernia, V. Codocedo, M. Kaytoue, and C. Robardet. Découverte de labels dupliqués par l'exploration du treillis des classifieurs binaires. In C. de Runz and B. Crémilleux, editors, *16ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2016, 18-22 Janvier 2016, Reims, France*, volume E-30 of *RNTI*, pages 255–266. Hermann-Éditions, 2016.
- [4] T. L. Taylor. *Raising the Stakes : E-Sports and the Professionalization of Computer Gaming*. MIT Press, 2012.
- [5] E. Q. Yan, J. Huang, and G. K. Cheung. Masters of control : Behavioral patterns of simultaneous unit group manipulation in starcraft 2. In B. Begole, J. Kim, K. Inkpen, and W. Woo, editors, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015*, pages 3711–3720. ACM, 2015.