Découverte de labels dupliqués par l'exploration du treillis des classifieurs binaires

Quentin Labernia, Victor Codocedo, Mehdi Kaytoue, Céline Robardet

▶ **To cite this version:**

Quentin Labernia, Victor Codocedo, Mehdi Kaytoue, Céline Robardet. Découverte de labels dupliqués par l'exploration du treillis des classifieurs binaires. 16 ème journées Francophones Extraction et Gestion des Connaissances, Jan 2016, Reims, France. pp.255–266. hal-01265202

HAL Id: hal-01265202
https://hal.science/hal-01265202

Submitted on 31 Jan 2016

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.

Découverte de labels dupliqués par l'exploration du treillis des classifieurs binaires

Quentin Labernia*, Victor Codocedo*, Mehdi Kaytoue*, Céline Robardet*

*Université de Lyon, CNRS, INSA-Lyon, LIRIS UMR5205, F-69621, France
prenom.nom@insa-lyon.fr

Résumé. L'analyse des données comportementales représente aujourd'hui un grand enjeu. Tout individu génère des traces d'activité et de mobilité. Lorsqu'elles sont associées aux individus, ou labels, qui les ont créées, il est possible de construire un modèle qui prédit avec précision l'appartenance d'une nouvelle trace. Sur internet, il est cependant fréquent qu'un utilisateur possède différentes identités virtuelles, ou labels doublons. Les ignorer provoque une grande réduction de la précision de l'identification. Il est ainsi question dans cet article du *problème de déduplication de labels*, et l'on présente une méthode originale basée sur l'exploration du treillis des classifieurs binaires. Chaque sous-ensemble de labels est classifié face à son complémentaire et des contraintes rendent possible l'identification des labels doublons en élaguant l'espace de recherche. Des expérimentations sont menées sur des données issues du jeu vidéo STARCRAFT 2. Les résultats sont de bonne qualité et encourageants.

1 Introduction

Les capteurs sont ancrés dans la vie quotidienne. Cachés dans les voitures, les smartphones, les objets connectés, ils enregistrent une multitude de mesures. Ces capteurs, qu'ils soient autonomes ou intégrés à un système plus complexe, génèrent des données comportementales riches. Correctement analysées, elles participent à la résolution de divers défis industriels et à la création de services et applications pour le grand public.

On s'intéresse ici à une technique d'identification d'individus se basant sur de données comportementales. De telles méthodes sont utiles d'un point de vue sécuritaire (détection de fraudes ou d'usurpations) ou de celui de la gestion des données privées (évaluation de techniques d'anonymisation de données). Il est vérifié dans de nombreux domaines qu'un individu peut être reconnu via les traces qu'il a générées : il est par exemple possible d'identifier de manière unique un individu à l'aide de quelques points d'intérêt dans l'espace et le temps (De Montjoye et al. (2013)). Encore, la manière d'interagir avec le clavier permet la reconnaissance d'un individu écrivant son mot de passe (Peacock et al. (2004)) ou encore jouant à un jeu vidéo (Yan et al. (2015)).

De nombreux utilisateurs possèdent de multiples identités virtuelles, appelées par la suite *labels doublons*, dont les relations sont à priori inconnues. Certaines problématiques de ciblage marketing concernent l'association de cookies provenant de périphériques différents à un

même individu (ICDM Contest (2015)). Dans cet article, on considère des données comportementales provenant de jeux vidéo, car ces données sont riches, anonymisées et disponibles librement sur internet. L'industrie vidéo-ludique possède en effet un besoin crucial en méthodes automatiques pour la détection d'éventuels tricheurs, usurpant un avatar (Von Eschen (2014)). Les structures de sport électronique cherchent également à identifier certains athlètes professionnels qui s'entraînent à l'aide d'avatars non officiels pour cacher leurs stratégies (Cavadenti et al. (2015)).

Il a été montré par Yan et al. (2015) que des modèles de prédiction basés sur la manière d'utiliser le clavier dans les jeux vidéos permettent d'identifier un joueur de manière précise. Cependant cette précision se dégrade fortement en présence d'avatars doublons : lorsque des joueurs utilisent de multiples identités virtuelles, le modèle apprend à classer ces identités en tant qu'individus différents. Par la suite, on se réfère à cette problématique en tant que *problème de déduplication de labels* : étant donné un ensemble de traces labellisées par des avatars, les ensembles en sortie correspondent aux avatars qui dénotent un même joueur.

Problème. On considère un ensemble de joueurs U et un ensemble d'identités ou labels L, aussi appelés avatars. Le problème de déduplication de labels consiste à découvrir la fonction $f : U \to \wp(L)$ correspondant à l'ensemble des identités assignées à chaque joueur. L'objectif est de créer une partition de L où chaque cluster représente un unique joueur, inconnu à priori.

Récemment, Cavadenti et al. (2015) a présenté une approche originale pour résoudre ce problème : en se basant sur l'analyse de la matrice de confusion d'une classification supervisée, elle exploite la confusion du classifieur en présence d'avatars doublons. Cette méthode présente des résultats intéressants, cependant elle a l'inconvénient d'opérer un post-traitement sur un modèle de classification unique et n'exploite pas la puissance des algorithmes de classification considérant une configuration particulière. En conséquence, certaines classes, surtout celles non équilibrées, ne pas peuvent être correctement apprises. Lorsqu'elles sont réunies avec leurs seuls doublons, elles peuvent cependant être correctement reconstituées.

L'approche proposée prend donc avantage du calcul d'un modèle particulier pour chaque cible à généraliser. Le treillis des classifieurs binaires est alors exploré, et chaque ensemble de labels – les possibles doublons – est évalué contre son complémentaire. Pour chaque nouvel ensemble généré, la matrice de confusion est comparée avec celle de tous ses sous-ensembles, de manière à trouver soit (a) que les labels des sous-ensembles –les exemples positifs – appartiennent à un même joueur, soit (b) qu'il est nécessaire d'élaguer l'espace de recherche en stoppant l'énumération des sur-ensembles. Pour se faire, on étudie à la fois l'évolution de (i) la F_1-mesure et de (ii) la distribution des labels des classes positives et négatives.

Le papier est organisé comme suit. La méthode est développée en Section 2. La Section 3 détaille l'implémentation suivie des expériences en Section 4, avant de conclure.

2 Méthode

Face à ce problème, une méthode originale basée sur le treillis des classifieurs binaires est suggérée, où chaque élément est un modèle construit à partir d'exemples positifs et négatifs – respectivement les instances d'un sous-ensemble de labels B et celles des labels complémentaires $\bar{B} = L \setminus B$. Ce treillis constitue l'espace de recherche du problème où chaque ensemble de labels correspond à un groupe potentiel d'avatars associés à un même individu. Son exploration est très coûteuse à cause de sa cardinalité et du coût d'évaluation de chaque nœud –

le temps nécessaire à l'exécution de l'algorithme de classification. Une méthode efficace est proposée pour éviter de considérer une grande partie de l'espace de recherche.

2.1 Treillis des classifieurs binaires

On considère un ensemble de traces T. À chaque trace $t \in T$ correspond un avatar, ou label, $l \in L$ via la relation $label(t) = l$. Soit un sous-ensemble quelconque de labels $B \subseteq L$. Le classifieur binaire correspondant ρ_B est construit sur la base des exemples positifs et négatifs. La classe positive est donnée par $B \subseteq L$ et la négative par $\bar{B} = L \backslash B$. Les instances de la classe positive correspondent ainsi à l'ensemble des traces étiquetées par $b \in B$, $\mathcal{I}_+(B) = \{t \in T \mid label(t) \in B\}$, tandis que celles de la classe négative sont les restantes, $\mathcal{I}_-(B) = T \setminus \mathcal{I}_+(B) = \{t \in T \mid label(t) \in \bar{B}\}$.

Définition 1 CLASSIFIEUR BINAIRE ET MATRICE DE CONFUSION. *Pour tout $B \subseteq L$, on définit le classifieur $\rho_B \colon T \longrightarrow \{+, -\}$. La matrice de confusion C^{ρ_B} associée est donnée en Table 1. Chaque score α_{ij}, avec $i, j \in \{+, -\}$, compte le nombre de traces de classe i classifiées en tant que j. On rappelle que α_{++} correspond aux vrais positifs, α_{+-} aux faux négatifs, α_{-+} aux faux positifs et α_{--} aux vrais négatifs – notés respectivement par la suite* VP, FN, FP *et* VN.

		Prédiction	
Réalité	C^{ρ_B}	$+$	$-$
	$+$	α_{++}	α_{+-}
	$-$	α_{-+}	α_{--}

TAB. 1 – *Matrice de confusion d'un classifieur binaire ρ_B. $+$ et $-$ correspondent resp. aux classes positives et négatives.*

Définition 2 SCORES D'UN CLASSIFIEUR BINAIRE. *Soit un sous-ensemble quelconque de labels $B \subseteq L$ et son classifieur binaire associé ρ_B. À partir de la matrice de confusion résultant de la classification des traces, on calcule deux scores $\varphi_B \in [0, 1]$ et $p_B \in \mathbb{N}$ tels que*

$$\varphi_B = \frac{2 \cdot \alpha_{++}}{(2 \cdot \alpha_{++}) + (\alpha_{+-}) + (\alpha_{-+})} \quad (1)$$

$$p_B = (\alpha_{++}) + (\alpha_{+-}) + (\alpha_{-+}) \quad (2)$$

Intuitivement, φ_B correspond à la F_1-mesure associée à la classification de l'ensemble des traces. Le score p_B comptabilise le nombre de traces similaires relativement au modèle, autrement dit, l'ensemble des traces susceptibles d'être classées VP.

Définition 3 TREILLIS DES CLASSIFIEURS BINAIRES. *$\mathcal{L} = (\wp(L), \subseteq)$ forme un treillis booléen ordonné par inclusion ensembliste, où chaque élément B est associé à un classifieur ρ_B.*

Notre méthode se base sur l'étude des scores φ_B et p_B lors de l'exploration de l'espace de recherche, depuis les singletons $\{\ell\}_{\ell \in L}$ jusqu'à l'ensemble L. L'objectif est de trouver les éléments maximaux respectant un ensemble de contraintes, qui assurent la validité de la relation $f(u) = B, u \in U$ pour chacun des éléments qui ont servis à leur construction.

2.2 Contraintes sur l'ensemble des classifieurs binaires

L'idée générale est d'explorer le treillis $\mathcal{L}$ en évaluant chaque élément pour savoir s'il représente un ensemble de labels doublons. Comme le nombre d'éléments du treillis est exponentiel au vu de $|L|$, il n'est pas acceptable de tous les parcourir. Deux contraintes sont introduites ; un élément du treillis doit les respecter pour qu'il puisse représenter un ensemble de labels doublons. Elles reposent sur l'évolution de la F_1-mesure et de la distribution des traces dans les classes positives et négatives, d'un ensemble par rapport à toutes ses parties. Il est donc nécessaire d'adopter une énumération par spécialisation – de $\varnothing$ à L – des éléments du treillis.

La première contrainte repose sur l'intuition que si $E \subseteq L$ est un ensemble de doublons, son classifieur ρ_E est plus robuste que celui de tous ses sous-ensembles. Soit $E = C \cup D$. S'il existe un ensemble C tel que $\varphi_C \geq \varphi_E$, cela signifie que la réunion de C et D doit être évitée. De manière plus formelle, un ensemble $E \subseteq L$ respecte la contrainte 1 s'il est valide.

Contrainte 1 *On considère un ensemble $E \subseteq L$ et un classifieur associé ρ_E. Si E est un ensemble valide de labels alors il respecte la contrainte suivante, $\forall C, D \subseteq E, E = C \cup D$,*

$$\varphi_E \geqslant max(\varphi_C, \varphi_D) \tag{3}$$

On note que le score φ_E n'est pas monotone, mais que la contrainte l'est.

Cette contrainte seule est nécessaire mais pas suffisante. En effet, il est possible d'obtenir un classifieur robuste, résultant de l'union d'autres classifieurs, mais qui n'est pas associé à un ensemble de labels doublons. Considérant les ensembles $C, D \subseteq E, E = C \cup D$, il est nécessaire d'introduire une contrainte supplémentaire pour s'assurer que les instances associées à E correspondent bien à l'ensemble de celles associées à C et D.

Il est possible d'observer directement ces instances, cependant le caractère de confusion serait sous-exploité. L'on propose plutôt de mettre une nouvelle fois à profit cette confusion pour déduire d'un ensemble $B \subset L$ une approximation du nombre d'instances associées à l'ensemble de labels doublons qui contient B. L'idée est d'exprimer un intervalle de validité de p_E à l'aide des scores p_C et p_D, de telle sorte que le classifieur ρ_E soit valide lorsque p_E est contenu dans cet intervalle.

Pour tous $B \subset L$, P_B est l'ensemble formé par les traces identifiées pour être des doublons (VP) et celles confondues par le classifieur ρ_B (FP et FN). L'intuition repose sur l'idée que deux traces confondues peuvent appartenir à deux labels doublons, mais parfois aussi à des labels différents – le classifieur peut se tromper en confondant deux traces ensembles parce qu'elles n'étaient finalement pas si ressemblantes, relativement aux autres. Ainsi, si l'ensemble $E = C \cup D$ est un ensemble de labels doublons, alors on a $P_E = (P_C \cup P_D) \cap \mathcal{E}$, avec $\mathcal{E}$ l'ensemble des traces confondues par ρ_C ou ρ_D mais pas par ρ_E. On a alors $|P_E| \leqslant |P_C \cup P_D|$, avec $|P_E| = p_E$. L'égalité $|P_C \cup P_D| = |P_C| + |P_D| - |P_C \cap P_D|$ est toujours vraie et permet l'estimation de la borne supérieure de l'intervalle de validité du score p_E. Il apparaît aussi clairement que l'ensemble P_E est au moins formé des éléments de P_C ou P_D, ce qui justifie la borne inférieure de cet intervalle de validité $|P_E| \geqslant \max(|P_C|, |P_D|)$.

Contrainte 2 *On introduit $\mu \colon \mathcal{P}(P)^2 \longrightarrow \mathbb{N}$ et $\theta \in [0, 1]$ tels que*

$$|P_C \cap P_D| = \mu(P_C, P_D) \cdot \theta \tag{4}$$

$x \in L$	$\{\ell_1\}$	$\{\ell_2\}$	$\{\ell_3\}$	$\{\ell_4\}$	$\{\ell_5\}$	$\{\ell_6\}$
$\lvert T_{\{x\}}\rvert$	40	50	20	50	120	40
$\varphi_{\{x\}}$	0.2	0.3	0.3	0.7	0.1	0.2
$p_{\{x\}}$	80	70	80	30	150	130
$\{x, y\} \in L^2$	$\{\ell_1, \ell_2\}$	$\{\ell_1, \ell_3\}$	$\{\ell_1, \ell_4\}$	$\{\ell_2, \ell_3\}$	$\{\ell_4, \ell_5\}$	$\{\ell_5, \ell_6\}$
$\lvert T_{\{x,y\}}\rvert$	90	60	90	70	170	160
$\varphi_{\{x,y\}}$	0.6	0.7	0.8	0.6	0.4	0.9
$p_{\{x,y\}}$	100	105	120	120	180	165

TAB. 2 – *Exemples de scores des classifieurs binaires ρ_B pour $|B| = 1$ et $|B| = 2$.*

où θ représente le taux de chevauchement entre P_C et P_D étant donné une mesure arbitraire μ. La contrainte suivante est ainsi définie, $\forall C, D \subseteq E$, $E = C \cup D$,

$$max(|P_C|, |P_D|) \leqslant |P_E| \quad \leqslant \quad |P_C| + |P_D| - \mu(P_C, P_D) \cdot \theta \tag{5}$$

$$\mu(P_C, P_D) \quad \leqslant \quad min(|P_C|, |P_D|) \tag{6}$$

Il est possible de choisir $\mu(P_C, P_D) = min(|P_C|, |P_D|)$ et $\theta = min(\varphi_C, \varphi_D)$. Ce choix est arbitraire et n'est pas approfondi dans la suite de cet article.

L'équation (4) fait apparaître un coefficient qui contrôle le chevauchement minimum requis entre les instances des ensembles P_C et P_D, pour être considérés comme assez similaires. Par exemple, un taux de chevauchement à zéro autorise la validité de E, même si P_C et P_D n'ont aucune instance commune. En revanche, plus ce taux augmente, plus la contrainte de similarité est forte. En pratique, il est nécessaire d'imposer une contrainte de similarité croissante car l'expérience montre que la confusion d'un classifieur singleton $\rho_{\{\ell\}, \ell \in L}$ est moins *précise* que celle d'un classifieur ρ_B avec B un ensemble de plus grand cardinal. Comme la première contrainte est croissante, elle est choisie pour définir θ.

Pour finir, deux associations sont définies $C_1, C_2 : L \to \{vrai, faux\}$. Elles indiquent si un ensemble de labels vérifie respectivement les contraintes 1 et 2. Un ensemble $B \subseteq L$ est donc valide si et seulement si $C_1(B) = C_2(B) = vrai$.

Exemple 1 *L'ensemble de labels $L = \{\ell_i \mid \forall i \in [1; 6]\}$ sert d'illustration à la méthode. Des exemples de scores sont donnés Table 2. Avec $\theta = 0$, on note que $\{\ell_1, \ell_2\}$ est valide, car $\{\ell_1\}, \{\ell_2\} \subset \{\ell_1, \ell_2\}$, $max(0.2, 0.3) \leqslant 0.6$ et $max(80, 70) \leqslant 100 \leqslant 80 + 70$. Au contraire, $\{\ell_1, \ell_4\}$ ne respecte pas la contrainte $120 \nleqslant 80 + 30$.*

2.3 Caractérisation du résultat

Le résultat attendu est un ensemble d'ensembles de labels, chacun correspondant à un joueur unique et à priori inconnu, car aucune information sur les joueurs n'est disponible. Cet ensemble constitue donc une partition de L. La construction des résultats se déroule comme suit. Il s'agit tout d'abord d'obtenir l'ensemble de tous éléments $B \subset L$ valides du treillis. Cet ensemble correspond à

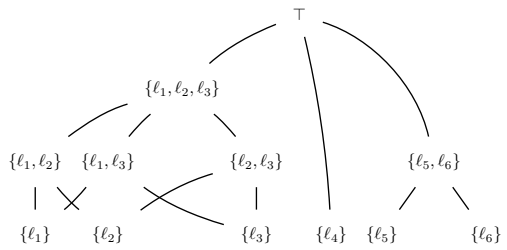
$$\mathcal{V} = \{\ B \subseteq L \mid (C_1(B) = vrai) \wedge (C_2(B) = vrai)\}$$

Le résultat final attendu est l'ensemble des éléments maximaux de $\mathcal{V}$ en regard de l'inclusion ensembliste :

$$\mathcal{R} = \{v \in \mathcal{V} |\ \nexists v' \in \mathcal{V}, v \subseteq v'\}$$

$\mathcal{R}$ est une antichaîne du treillis $\mathcal{L}$, ce n'est donc pas nécessairement une partition de l'ensemble L. $\mathcal{R}$ est ici une relation de tolérance, un ensemble d'ensembles qui couvrent L mais pas forcément de manière disjointe. Il serait possible d'imposer le fait que le résultat soit une partition, p. ex. en choisissant des éléments à supprimer, ou en ajoutant des contraintes sur la définition de l'ensemble $\mathcal{V}$. Cependant, étant donné les hypothèses initiales et le choix des contraintes, l'observation pratique montre que $\mathcal{R}$ est une partition. La seule explication possible pour ne pas avoir de partition correspond au cas d'un individu qui partagerait son compte avec d'autres joueurs : il sera éliminé tôt, au sens de l'exploration par spécialisation, et ne sera pas réuni avec les autres ensembles respectant les contraintes C_1 et C_2. Cette supposition est vérifiée dans les expérimentations. On peut prouver que $\mathcal{L} = ((\mathcal{V} \cup \{\emptyset, L\}), \subseteq)$ est un treillis et que $\mathcal{R}$ est une antichaîne composée de l'ensemble de des co-atomes.

Exemple 2 *On obtient le treillis $\mathcal{L}$ illustré ci-dessous. Les scores correspondants à l'ensemble L_3 sont $\varphi_{L_3} = 0.9$, $p_{L_3} = 120$. Le résultat final est $\mathcal{R} = \big\{\{\ell_1, \ell_2, \ell_3\}, \{\ell_4\}, \{\ell_5, \ell_6\}\big\}$.*



3 Algorithme

L'espace de recherche théorique est constitué des sous-parties de l'ensemble des labels L. L'exploration du treillis est effectuée par niveau, c.-à-d., par ensemble de cardinalité croissante, chacun des niveaux donnant lieu à une *passe*. Le pseudo-code de cette exploration est indiqué ci-après. Il s'agit tout d'abord de générer les classifieurs *singletons* $\rho_{\{\ell\}, \ell \in L}$, ce qui correspond à la passe 0 (l. 3 à 5). Ils constituent l'ensemble initial $Y \subseteq \mathcal{P}(L)$.

Tous les éléments de Y sont ensuite combinés deux à deux en prenant également en compte les singletons (l. 8). Chaque nouveau modèle est construit (l. 9) puis sa validité est testée (l. 10). S'il est valide, il est ajouté à la liste des éléments à être combinés lors de la prochaine passe (Z l. 11), tandis que les deux sous-ensembles sont supprimés du résultat – l'antichaîne $\mathcal{R}$. Si le classifieur n'est pas valide (l. 13 à 16), l'ensemble est marqué comme non pertinent de telle sorte qu'aucun de ses sur-ensembles ne puisse être considéré plus tard. L'algorithme enchaîne les passes tant qu'il est possible de faire l'union d'éléments. La complexité dans le pire des cas est $O(h \cdot 2^{|L|})$ où h dépend du classifieur ρ utilisé.

Entrées : L'ensemble des labels L et une méthode de classification ρ
Sorties : L'ensemble des résultats $\mathcal{R}$ (relation de tolérance)

$Y, R, S \leftarrow L$
$\mathcal{U} \leftarrow \emptyset$ `// contient les ensembles non pertinents`
pour tous les $y \in Y$ **faire** Construction de ρ_y et C^{ρ_y} `// passe 0`
tant que $|Y| > 1$ **faire** `// passe` $n \in \mathbb{N}^*$
  $Z \leftarrow \varnothing$
  pour $E \leftarrow C \cup D | (C \in Y, D \in Y \cup S, C \neq D, \forall u \in \mathcal{U}, u \nsubseteq E)$ **faire**
    Construction de ρ_E et C^{ρ_E}
    si $(C_1(E) \wedge C_2(E))$ **alors**
      $Z \leftarrow Z \cup \{E\}$
      $R \leftarrow (R \setminus \{C, D\}) \cup \{E\}$
    sinon
      $U \leftarrow \mathcal{U} \cup \{E\}$ `// ensemble E non valide`
      $R \leftarrow \{r \in R \mid \forall u \in \mathcal{U}, u \nsubseteq r\}$
  $Y \leftarrow Z$
$\mathcal{R} \leftarrow R$
retourner $\mathcal{R}$

4 Expérimentations

Cette section présente l'évaluation de l'approche à travers des expériences quantitatives et qualitatives. Elles se basent sur l'étude de données provenant du jeu vidéo STARCRAFT 2, c'est pourquoi les données sources sont celles présentes dans Cavadenti et al. (2015). Toutes les expériences ont été réalisées sur un 2,5 GHz Intel Core i7 doté de 8Go de Ram sous OSX. Différentes méthodes de classification supervisées du framework Weka ont été utilisées pour construire les modèles $\rho_{B \subseteq L}$(Hall et al. (2009)).

4.1 Données et configuration expérimentale

Collections de replays. Deux collections sont utilisées, $\mathcal{C}_1$ et $\mathcal{C}_2$. Un replay correspond à l'enregistrement d'une partie de jeu. Il contient l'ensemble des actions des joueurs – les traces associées à un avatar. La collection $\mathcal{C}_1$ est composée des 955 parties jouées par 171 joueurs professionnels pendant le *2014 World Championship Series*. Les règles de ce tournoi assurent qu'il n'existe aucun avatar doublon, $\mathcal{C}_1$ sert donc de base pour construire les vérités terrains, c.-à-d., créer des avatars doublons. La collection $\mathcal{C}_2$ se compose d'un ensemble de $10\ 108$ parties de jeu à *un contre un*, récupérées depuis un site internet spécialisé, et impliquant $3\ 805$ joueurs. Cette collection représente une configuration réelle.

Descripteurs et modèles de classifieurs. Les descripteurs utilisés pour la classification sont identiques à ceux de deux précédents travaux. Un bref récapitulatif est présenté, cependant le lecteur est invité à se référer aux travaux de Cavadenti et al. (2015) et Yan et al. (2015). Le jeu STARCRAFT 2 autorise le joueur à personnaliser l'usage de son clavier via l'association de fonctions aux touches 0 à 9. Des descripteurs supplémentaires comme le nombre d'actions réalisées par minute par le joueur sont utilisés. Lorsqu'aucun avatar doublon n'est présent dans une collection, il a été montré que ces descripteurs permettent de prédire un avatar avec une

précision supérieure à 95%. Les classifieurs IBk, J48, Perceptron multicouche, Bayes naïf, RandomForest et SMO, ont été testés via leurs implémentations *Weka* (Hall et al. (2009)). Finalement, on construit le modèle $\rho_{B,B\subseteq L}$ basé sur les exemples positifs et négatifs, et la matrice de confusion est créée à l'aide d'une validation croisée à facteur 10. Les scores φ_B et p_B sont alors calculés. Il est à noter que le choix de la méthode de classification est fixé lors de l'exécution de l'algorithme, c.-à-d., on ne mélange pas différents types de classifieurs lors de l'exploration du treillis.

Paramètres. Trois paramètres additionnels présents dans le travail de Cavadenti et al. (2015) sont considérés. Seules les $\tau \in \mathbb{N}$ premières secondes d'une partie de jeu sont prises en compte lors du calcul des différents descripteurs, car il a été montré que ce paramètre influe sur la phase d'apprentissage. De plus, certains labels ne sont représentés que par un faible nombre d'instances, ce qui a un impact négatif sur la précision de la classification. Une trace est retenue si son label associé possède au moins $\Theta \in \mathbb{N}$ instances, c.-à-d., $\forall \ell \in L, |T_{\{\ell\}}| \geqslant \Theta$. Il a été antérieurement montré qu'une bonne prédiction nécessite $\Theta \geq 10$. Finalement, un seuil $\Lambda \in [0;1]$ permet de retenir un élément $R \in \mathcal{R}$ si et seulement si $\varphi_R \geqslant \Lambda$. Cette sélection sur $\mathcal{R}$ permet l'amélioration de la précision des résultats.

Vérité terrain et évaluation. Étant donné un ensemble de labels, cette méthode s'attache à trouver la famille d'ensembles $\mathcal{R}$ pour laquelle chaque élément $R \in \mathcal{R}$ représente un ensemble de labels doublons. Comme il n'existe aucune vérité terrain – pour des questions d'anonymisation –, une artificielle est créée. On considère des données provenant de la collection $\mathcal{C}_1$. Les $\gamma \in \mathbb{N}$ premiers labels sont choisis parmi ceux qui possèdent le plus d'instances. Pour chaque label, l'ensemble de leurs instances sont partagées en différents groupes, représentant alors chacun un label doublon. En d'autres termes, chacun de ces γ labels est remplacé par p nouveaux labels $(\ell_i)_{i\in[1;p]}$, associés à une famille de *proportions* $(r_i)_{i\in[1;p]}$ tels que

$$\forall i \in [1;p], |T_{\{\ell_i\}}| = \frac{r_i \cdot |T_{\{\ell\}}|}{\sum_{j\in[1;p]} r_j} \tag{7}$$

On utilise la notation suivante dans les expériences : 1_1_2 signifie que chaque label ℓ est remplacé par trois labels ayant respectivement 25%, 25% et 50% des instances associées à ℓ.

Pour évaluer un résultat $\mathcal{R}$ par rapport à la vérité terrain $\mathcal{G}$, on procède comme suit. L'ensemble $\wp(L)$ est partagé entre les exemples positifs et négatifs, avec $\mathcal{G}^+ = \{X \subseteq G, \forall G \in \mathcal{G}\}$ et $\mathcal{G}^- = \wp(G) \setminus \mathcal{G}^+$. Un processus similaire est effectué pour partitionner les résultats observés, on a $\mathcal{R}^+ = \{X \subseteq R, \forall R \in \mathcal{R}\}$ ainsi que $\mathcal{R}^- = \wp(R) \setminus \mathcal{R}^+$. Il est ensuite possible de comparer la vérité terrain avec les résultats observés : VP, FP et FN sont définis de manière identique aux mesures classiques de précision, rappel et F_1-mesure.

Cette évaluation consiste à comparer deux partitions. Cependant, $\mathcal{R}$ n'est pas nécessairement une partition et peut être une relation de tolérance. Comme expliqué précédemment, ceci ne devrait pas arriver. De plus, des valeurs de précision et de rappel nulles pénalisent ces cas lors des expériences.

4.2 Résultats expérimentaux

Sélection des paramètres. On utilise la collection $\mathcal{C}_1$. Le paramètre γ est fixé à 10, Θ à 15. Les résultats de l'expérience sont agrégés pour tous les classifieurs, à l'exception de SMO qui rend de mauvais résultats. L'objectif est de déterminer la valeur du paramètre Λ, seuil appliqué

sur les éléments de $\mathcal{R}$. Cette valeur est prise comme étant le troisième quartile des φ_R pour $R \in \mathcal{R}$ FP. Le taux d'éléments VP de $\mathcal{R}$ éliminés est montré Figure 1 comme une fonction de τ. Le meilleur résultat correspond à $\tau = 200$ pour $\Lambda = 0.78$. La Figure 2 illustre la distribution des FP et VP pour cette valeur finale de Λ.

De plus, quatre méthodes de calcul de θ sont testées, comme une fonction de $\varphi_{C \subset L}$, $\varphi_{D \subset L}$: $\theta = 0$ (null), min, mean and max. Le tableau ci-contre montre l'agrégation des résultats avec les paramètres $\Gamma = 10$, $\Theta = 20$, $\tau = 200$: bien qu'ils possèdent une faible moyenne et un écart-type élevé, puisque des classifieurs bons et mauvais sont agrégés ensembles, il apparaît clairement que la méthode $\theta = 0$ donne les meilleurs résultats.

θ	Précision	Rappel
null	0.76 ± 0.28	0.69 ± 0.28
min	0.50 ± 0.50	0.22 ± 0.28
mean	0.39 ± 0.49	0.17 ± 0.26
max	0.35 ± 0.48	0.16 ± 0.26

Analyse des temps d'exécution et de l'occupation mémoire. Étant donnés les paramètres précédemment décrits, on constitue la vérité terrain $\mathcal{G}$ avec diverses proportions. La Figure 5 montre que le nombre d'éléments explorés du treillis est insignifiant par rapport à la taille de l'espace de recherche de 2^{171}. Cela indique que les contraintes mises en place permettent un élagage rapide ce qui rend l'exécution de cette méthode possible en pratique. Les temps d'exécutions associés, visibles Figure 4, sont raisonnables.

Analyse de l'efficacité. Il s'agit d'illustrer l'efficacité de la méthode. Les paramètres restent inchangés. La Figure 3 présente la précision et le rappel du résultat $\mathcal{R}$, ou ensembles résultants, par rapport à la vérité terrain $\mathcal{G}$. Le résultat principal est que le classifieur naïf de Bayes donne les meilleurs résultats, privilégie la précision au rappel, et présente une grande robustesse face aux classes non-équilibrées. Comme la méthode nécessite la validation des deux contraintes C_1 et C_2 pour tout sous-ensemble, elle favorise la précision. Il a été observé, lors d'une expérience non publiée, que le rappel est favorisé lorsqu'un élément du treillis n'a besoin de respecter les contraintes que vis-à-vis de deux éléments de sa couverture. Il s'avère que l'identification d'utilisateur se porte plus généralement sur la précision des résultats. Finalement, ces résultats sont comparés à ceux obtenus par la méthode DÉBROUILLE introduit par Cavadenti et al. (2015). Alors que les résultats obtenus pour des labels doublons de taille 2 et équilibrés sont assez similaires ou parfois meilleurs, la présente méthode offre des résultats de meilleure qualité lorsque les classes ne sont pas équilibrées et que le nombre d'éléments de chaque groupe de doublons est élevé – de taille 3 ou plus.

Expérience qualitative. Jusqu'à maintenant, l'objectif des expériences était d'étudier le comportement de la méthode sur la collection $\mathcal{C}_1$, sans doublon, après insertion d'une vérité terrain artificielle. Cette dernière expérience consiste à exécuter la méthode sur la collection $\mathcal{C}_2$ qui correspond à un cas réel. De nombreux paramètres ont été testés, seuls les premiers résultats sont présentés ici. La méthode de classification choisie est le classifieur naïf de Bayes, car il favorise la précision par rapport au rappel et se comporte de manière efficace face à des classes non équilibrées, ce qui est suspecté d'être le cas dans cette collection $\mathcal{C}_2$. Les paramètres sont fixés à $\tau = 200$, $\theta = \text{mean}(\varphi_C, \varphi_D)$ et $\Lambda = 0$ après différents essais. Les ensembles de labels résultants sont classés par rapport à la robustesse de leur classifieur, c.-à-d., $\mathcal{R} = \langle R_1, ..., R_n \rangle$, où $\varphi_{R_i} \geq \varphi_{R_{i+1}}$, for $1 \leq i \leq n-1$. Après une exécution de 1 017 secondes, sur les $|L| = 58$ avatars initiaux, 7 paires de doublons de taille 2 ont été trouvées. Pour quatre de ces paires, les avatars partagent le même identifiant de compte. Il s'avère également que l'avatar *EGStephanoRC*, célèbre et ancien joueur professionnel, est associé à *lllllllllll*,

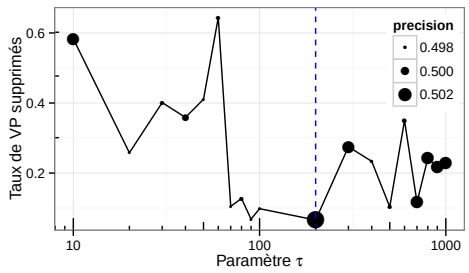


FIG. 1 – *Taux d'éléments* VP *de* $\mathcal{R}$ *supprimés en fonction du paramètre* τ *avec* Λ *le troisième quartile des* φ_R *pour* $R \in \mathcal{R}$ FP. *La ligne bleue indique la meilleure solution pour* $\tau = 200$.

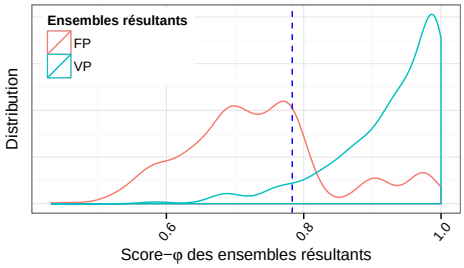


FIG. 2 – *Distribution des* φ_R *pour* $R \in \mathcal{R}$ VP *et* FP, *et* $\tau = 200$. *La ligne bleue indique le troisième quartile des* φ_R *pour* $R \in \mathcal{R}$ FP. *Cette solution n'implique que* 6% *de* VP *supprimés.*

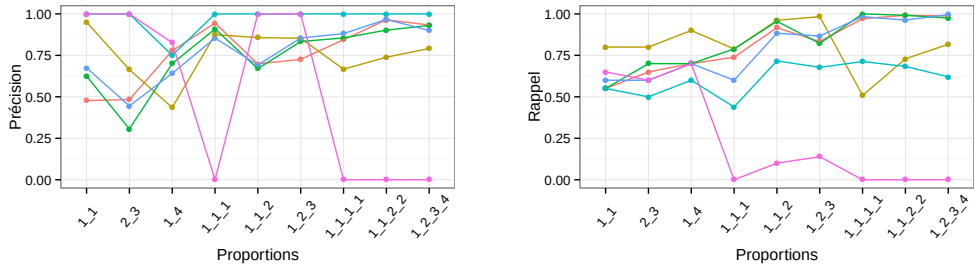


FIG. 3 – *Mesures des résultats en fonction des classifieurs et des proportions des ensembles de labels doublons de la vérité terrain. Les couleurs correspondent à celles Figure 4.*

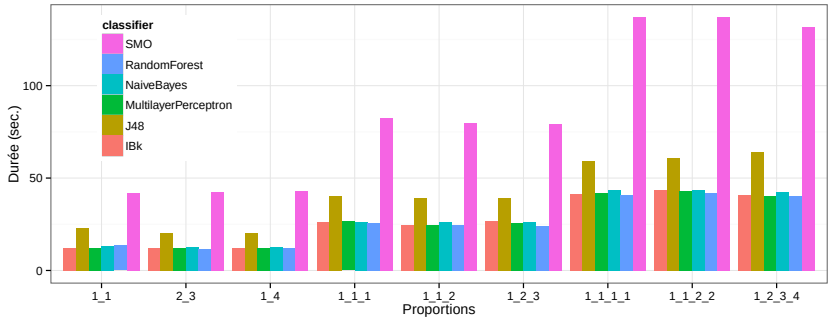


FIG. 4 – *Temps d'exécution de l'algorithme sur un processeur 2,5 GHz Intel Core i7 doté de 8Go de Ram sous OSX, en fonction des classifieurs et des proportions des ensembles de labels doublons de la vérité terrain.*

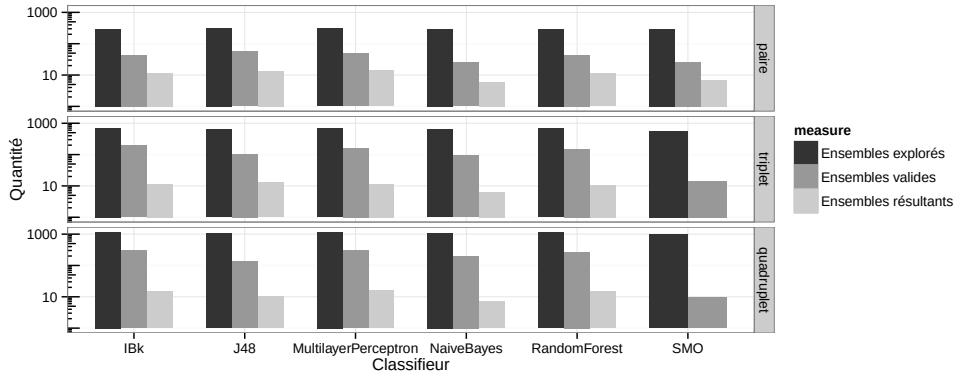


FIG. 5 – *Nombre d'éléments du treillis $\mathcal{L}$ explorés, valides et résultants.*

un pseudo non reconnaissable en l'état. Une paire d'avatars *LiquidHero*, joueur professionnel, partage un même pseudo sur des comptes différents. Il s'agit donc clairement d'un VP. Finalement, deux ensembles de labels FP sont présents, pour lesquels il n'est pas possible de vérifier l'association, mais seulement formuler l'hypothèse qu'un même individu correspond à chacune de ces paires d'avatars.

5 Conclusion

Il arrive souvent qu'un même individu possède différentes identités virtuelles. On peut alors faire face à un problème de déduplication de labels. Alors même que ce problème revêt des objectifs communs avec les techniques de résolution d'entité (Getoor et Machanavajjhala (2012), Mugan et al. (2014)), il est traité d'une nouvelle manière, en prenant en compte les comportements des joueurs cachés dans leurs traces de jeu. En effet, les données comportementales générées par les joueurs peuvent aider à construire des modèles de prédiction précis, qui ne confondent que les traces issues d'un même joueur. Il est proposé dans cet article une méthode qui prend avantage de cette idée, en générant un classifieur binaire pour chaque sous-ensemble possible de labels. Des contraintes permettent, lors de l'énumération de ces sous-ensembles, la découverte des ensembles de labels doublons. L'implémentation de l'approche a été expérimentée à l'aide de données issues du jeu vidéo STARCRAFT 2. Les résultats sont encourageants et globalement de meilleure qualité que ceux issus de Cavadenti et al. (2015), particulièrement lorsque la distribution des traces entre les avatars d'un même joueur n'est pas équilibrée. Ce travail constitue une première tentative en utilisant le treillis des classifieurs binaires pour l'identification des labels doublons. De nombreuses tâches restent encore à faire, comme des expériences supplémentaires sur des données d'autres domaines, et une comparaison plus fine des résultats avec le travail de Cavadenti et al. (2015). L'énumération des éléments du treillis constitue aussi un axe de recherche, de manière à permettre un meilleur passage à l'échelle de l'algorithme et étudier d'autres contraintes.

Remerciements. Cette recherche a été en partie financée par le Projet FUI AAP 14 Tracaverre 2012-2016 et VEL'INNOV (ANR INOV 2012).

Références

Cavadenti, O., V. Codocedo, J.-F. Boulicaut, et M. Kaytoue (2015). When cyberathletes conceal their game : Clustering confusion matrices to identify avatar aliases. In *International Conference on Data Science and Advanced Analytics (DSAA 2015)*.

De Montjoye, Y.-A., C. A. Hidalgo, M. Verleysen, et V. D. Blondel (2013). Unique in the crowd : The privacy bounds of human mobility. *Nature Scientific reports 3*(1376), 779–782.

Getoor, L. et A. Machanavajjhala (2012). Entity resolution : Theory, practice & open challenges. *PVLDB 5*(12), 2018–2019.

Hall, M. A., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten (2009). The WEKA data mining software : an update. *SIGKDD Explorations 11*(1), 10–18.

ICDM Contest (2015). Identify individual users across their digital devices. In *IEEE International Conference on data mining*.

Lourenço, A., A.-P. Alves, C. Carreiras, R.-P. Duarte, et A. Fred (2015). Cardiowheel : ECG biometrics on the steering wheel. In *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, LNCS (9286), pp. 267–270. Springer International Publishing.

Mugan, J., R. Chari, L. Hitt, E. McDermid, M. Sowell, Y. Qu, et T. Coffman (2014). Entity resolution using inferred relationships and behavior. In *IEEE International Conference on Big Data*, pp. 555–560.

Peacock, A., X. Ke, et M. Wilkerson (2004). Typing patterns : A key to user identification. *IEEE Security & Privacy 2*(5), 40–47.

Von Eschen, A. (2014). Machine learning and data mining in call of duty (invited talk). In *Eur. Conf. on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*.

Yan, E. Q., J. Huang, et G. K. Cheung (2015). Masters of control : Behavioral patterns of simultaneous unit group manipulation in starcraft 2. In *33rd Annual ACM Conf. on Human Factors in Computing Systems (CHI 2015)*, pp. 3711–3720. ACM.

Summary

Analysis of behavioral data represents today a big issue. Anyone generates activity and mobility traces. When traces are labeled by the user that generates it, models can be learned to accurately predict the user of an unknown trace. In online systems however, users may have several virtual identities, or duplicate labels. By ignoring them, the prediction accuracy drastically drops. In this article, we tackle this *duplicate labels identification problem*, and present an original approach that explores the lattice of binary classifiers. Each subset of labels is learned against the others, and constraints make possible to identify duplicate labels while pruning the search space. We experiment with data of the video game STARCRAFT 2. Results are of good quality and encouraging.