

Modélisation des usages utilisateurs pour le Crowdsourcing à grande échelle



Mémoire de fin d'étude

Master *Sciences et Technologies*,

Mention *Informatique*,

Parcours : DONNÉES, CONNAISSANCES ET LANGAGE NATUREL
(DECOL)

Auteur

Yonatan Carlos CARRANZA ALARCON

Superviseurs

Esther PACITTI

Alexis JOLY

Maximilien SERVAJEAN

Lieu de stage

LIRMM UM5506 - CNRS, Université de Montpellier

Résumé

Les services de *Crowdsourcing* sont de plus en plus utilisés aujourd’hui car ils permettent la réalisation de tâches à faible coût par des utilisateurs d’internet. Ces services nous offrent des processus, ou étapes, pour assurer la qualité du travail tels que l’assignation des tâches, le calcul des compétences du travailleur et l’évaluation des résultats.

Cette étude s’est focalisée sur l’approfondissement de l’étape d’évaluation des résultats en proposant un modèle du comportement utilisateur qui permettra d’évaluer les solutions de crowdsourcing existantes et futures. Ainsi donc, nous nous sommes servis de la *Matrice de Confusion* afin de représenter la connaissance de l’utilisateur sur un problème de classification pour ensuite modéliser quatre différents profils d’utilisateurs (e.g. expert et amateur). Ces profils suivront une loi de probabilité discrète (e.g. loi logarithmique) et la connaissance de l’utilisateur sera générée à partir de cette loi et une *simulation de Monte-Carlo*.

Nous avons aussi exploré un autre horizon en analysant les données réelles du site web Tela-Botanica dans le but d’extraire des profils réalistes.

Enfin, avec l’aide de ce modèle, nous avons effectué un ensemble de simulations aléatoires pour valider nos profils et pour évaluer deux méthodes d’inférence de solutions de *Crowdsourcing*.

Mots clés: Crowdsourcing, contrôle de la qualité, matrice de confusion, simulation Monte-carlo

Abstract

Nowadays, Crowdsourcing services are increasingly used in a variety of applications, because they allow to publish small tasks to be performed by a large group of networked people at low-cost.

Those services are structured in processes, or stages, to ensure the quality of work, such as assignment of tasks, workers’ skills estimation and quality estimation.

This study focused on deepening the quality estimation stage of Crowdsourcing by proposing a model of the users behaviors enabling us to evaluate state-of-the art *Crowdsourcing* solutions.

Therefore, we used the *Confusion Matrix* to represent the knowledge of the user in a multi-class classification problem and then model four different user profiles (e.g. expert and amateur). These profiles follow a discrete probability distribution (e.g. logarithmic distribution) and knowledge of the user will be generated from this distribution and *Monte-Carlo Simulation*.

We also explored another horizon by analyzing the real data of the Tela-Botanica web site in order to extract realistic profiles.

Thus, with the help of this model, we will perform a set of random simulations in order to validate our profiles and to evaluate two methods, or inferences, of *Crowdsourcing* solutions.

Keywords: Crowdsourcing, quality control, confusion matrix, Monte-carlo simulation

Table des matières

Table des matières	v
Table des figures	vi
Remerciements	1
1 Introduction	3
2 État de l'art	7
2.1 Inférence	7
2.2 Méthodes d'estimation des tâches structurées	8
2.3 Méthodes d'estimation des tâches non-structurées	9
3 Modélisation	11
3.1 Approche générale	13
3.2 Approche Réaliste	21
4 Résultats et Conclusion	25
4.1 Validation des profils	25
4.2 Evaluation des solutions Crowdsourcing	27
4.3 Conclusion et Overtures	28
Bibliographie	29

Table des figures

1.1	Exemple de classification à choix multiple	3
1.2	Schéma trivial du Crowdsourcing	4
1.3	Schémas de Crowdsourcing inspiré de [1]	4
1.4	Classification d'une espèce de plante sur Pl@ntNet.	5
2.1	Matrices de confusion	9
2.2	Schéma de validation de qualité	10
3.1	Schéma détaillé d'évaluation d'usage des systèmes de Crowdsourcing	11
3.2	Représentation du modèle de simulation artificielle.	12
3.3	Schéma des processus à accomplir pour la simulation artificielle	13
3.4	Exemple d'obtention de la mesure de qualité	14
3.5	La distribution des probabilités du profil expert	15
3.6	La distribution des probabilités du profil amateur	15
3.7	La distribution des probabilités du profil novice	16
3.8	La distribution des probabilités du profil spammeur	17
3.9	Génération de courbe plus réaliste	19
3.10	<i>Visualisation de la matrice de confusion en 3D.</i>	20
3.11	<i>Exemple de calcul de réponse d'un utilisateur</i>	20
3.12	Schéma d'extraction des matrices de confusion plus réalistes	21
3.13	Site web Tela-botanica, publication vérifiée d'une espèce de plante.	22
3.14	Exemple de relation transitive dans un graphe orienté	23
3.15	Visualisation des utilisateurs de Tela-Botanica regroupés.	24
4.1	Méthode d'inférence de Dawid et Skene (I)	25
4.2	Méthode d'inférence de Dawid et Skene (II)	26
4.3	Visualisation des utilisateurs par profil	26
4.4	Évaluation des solutions de <i>Crowdsourcing</i>	27

List of Algorithms

1	Simulation naïf	12
2	Génération des matrices de confusion	18
3	Normalisation de la loi de probabilité discrète	18
4	Génération de courbe réaliste par Monte-Carlo	19
5	Remplissage de la matrice de confusion	23
6	Trouver les relations transitives.	24



Remerciements

Je tiens à remercier mes encadrants Alexis JOLY, Esther PACITTI et Maximilien SERVAJEAN qui m'ont enseigné, guidé, suivi et orienté et qui ont répondu à toutes mes questions et autres problèmes rencontrés lors de ce modeste travail. Je tiens aussi à remercier à mes parents qui m'ont toujours soutenu, encouragé et aidé.

Introduction

Qu'est-ce que le Crowdsourcing ?


La croissance et l'évolution d'Internet nous ont amenés à rechercher de nouvelles méthodes sur le traitement des données massives (e.g BigData). L'une d'entre elles, est le *Crowdsourcing*, le terme ayant été inventé par Howe dans [2]. *Il correspond à l'externalisation de micro-tâches spécifiques d'une organisation envers un grand groupe de personnes connectées à Internet sous la forme d'appels ouverts*. Le *Crowdsourcing* est aujourd'hui de plus en plus utilisé dans de nombreux domaines scientifiques.

Les services tels que Mechanical Turk Amazon¹ ainsi que Zooniverse² proposent des plateformes ergonomiques peu coûteuses permettant de poster des micro-tâches simples (*Tâches d'Intelligence Humaine (TIHs)*³) et pouvant être résolues en quelques secondes par toute personne ayant accès à Internet (désormais nommé *travailleur*). Ces micro-tâches sont parfois faites bénévolement, récompensées par la notion de réputation (e.g. points, badges), ou bien même payées. Voici des exemples de micro-tâches : la reconnaissance de l'écriture manuscrite, la vérification de l'adresse d'une société, la conception graphique de logos, l'identification et l'étiquetage d'images (e.g [3], [4], [5]) ou de données (e.g. Fig. 1.1, [6] et [7]).

Surigao del Sur: relief goods infant needs #pabloPH -
<http://t.co/fpEEyFwO> #ReliefPH

Picture: link points to picture(s) of general damage/flooding
 Video: link points to video(s) of general damage/flooding
 Location: tweet/picture/video includes reference to location, place, town etc.
 Other: link **does not point** to picture(s) of general damage/flooding

Surigao del Sur



Approximate damage/impact location: Longitude: -, Latitude: -

FIG. 1.1: **Tâche de classification à choix multiple** du site web *Crowdcrafting*^a : Étant donné le tweet de la figure : *Surigao del Sur : relief goods infant needs #pabloPH http://t.co/fpEEyFwO #ReliefPH*, Veuillez étiqueter le tweet suivant par la ou les catégories qui décrivent le mieux les liens qu'il contient. La réponse à cette tâche est que le lien correspond parfaitement à un lieu.

^awww.crowdcrafting

¹<https://www.mturk.com>

²<https://www.zooniverse.org>

³TIHs sont des tâches individuelles que l'être humain peut résoudre facilement.

Quelles sont les étapes à accomplir en Crowdsourcing ?

Assurer l'efficacité et l'efficacité d'une tâche effectuée par un travailleur en suivant le schéma, de la figure 1.2, continuent d'être un grand défi pour les entreprises et les chercheurs. Kittur nous propose ainsi, dans [1], un schéma plus détaillé (e.g. des utilisateurs classés par compétences), collaboratif (e.g. la communication parmi les utilisateurs) et durable, illustré dans la figure 1.3.

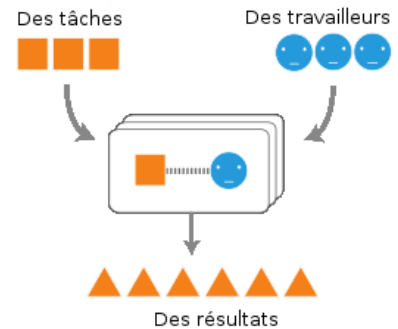


FIG. 1.2: Schéma trivial du Crowdsourcing

Le schéma de Crowdsourcing présenté dans la figure 1.3 souligne trois concepts essentiels à approfondir :

1. *Calcul des compétences de l'utilisateur*, qui est basé sur la réputation de ses tâches antérieures, attribuées par les demandeurs.
2. *Assignment des tâches*, qui est liée à plusieurs facteurs d'un travailleur, par exemple : l'intérêt, la réputation, les compétences, les mesures incitatives, etc.
3. *Évaluation des résultats*, qui est liée aux réponses des travailleurs ainsi qu'à leur fiabilité.

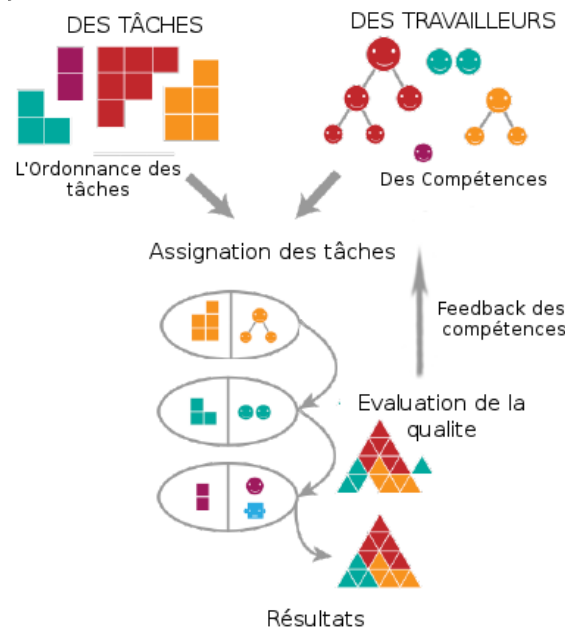


FIG. 1.3: Schémas de Crowdsourcing inspiré de [1]

Cas d'étude : Pl@ntNet

*Pl@ntNet*⁴ est une application d'aide à l'identification interactive des espèces de plantes. L'application retourne des plantes semblables grâce à l'existence d'un algorithme

⁴<http://www.plantnet-project.org>

supervisé de classification qui identifie les espèces potentielles. Ces propositions sont ensuite recommandées à la communauté de biologistes afin que celle-ci puisse valider au travers de notes l'identification correcte. Voici un exemple :

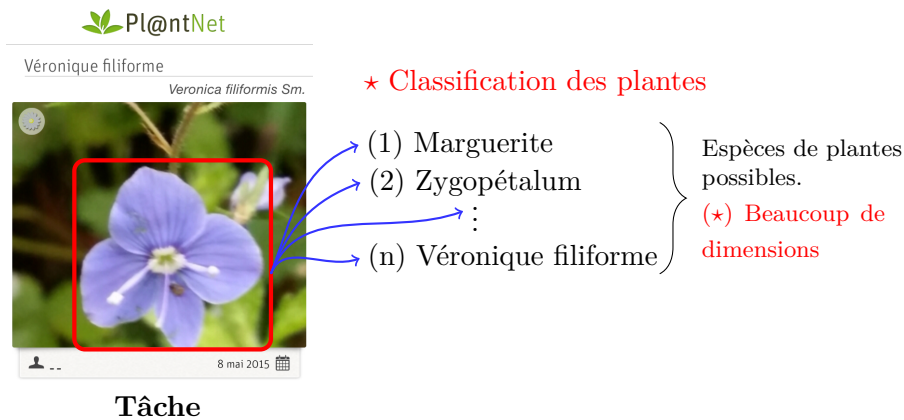


FIG. 1.4: Classification d'une espèce de plante sur Pl@ntNet.

Il est important de remarquer que l'ensemble des espèces possibles est de très grande taille d'un point de vue utilisateur.

Quels problèmes existent autour des plateformes de Crowdsourcing ?

Les problèmes suivants ont été identifiés :

1. Les plateformes de *Crowdsourcing* ont toujours été évaluées à petite échelle, c'est-à-dire peu (eg. Fig. 1.2)
2. Il n'existe pas de modélisation afin de simuler les solutions de *Crowdsourcing*, que ce soit à petite échelle ou à grande échelle.
3. Dans le cadre de notre cas d'étude, les applications telles que *Pl@ntNet* manipulent un grand nombre de catégories, et les solutions actuelles de *Crowdsourcing* ne peuvent pas être mise en place
4. Il n'existe pas de Benchmark pour évaluer les solutions de *Crowdsourcing* à grand échelle.

Quelles solutions proposer ?

L'objectif de notre travail se focalise à comprendre les différentes étapes du *Crowdsourcing* et à modéliser (i.e de proposer un modèle statistique) les différents comportements d'utilisateurs dans le but d'évaluer les différentes solutions de *Crowdsourcing* existant dans la littérature (e.g. [7], [8]) ainsi que les contributions futures.

En résumé, le reste de ce memoire est organisé de la manière suivante. Le chapitre 2 décrit l'état de l'art, c'est-à-dire les différents solutions d'évaluation des résultats de *Crowdsourcing*. Le chapitre 3 aborde la modélisation des usages utilisateurs. Enfin, le chapitre 4 montrera les résultats obtenus, la conclusion ainsi que les futurs travaux.

État de l'art

2.1 Inférence

L'inférence, à savoir l'opération consistant à déterminer la classification se focalise en particulier sur le problème de la qualité d'une tâche qui a été effectuée par plusieurs *travailleurs en ligne* ¹. Ce problème d'étude de la qualité d'une tâche remonte aux années 50 lorsque le père de la qualité totale *Walter A. Shewhart* résume que lorsque les organisations se concentrent sur les coûts, la qualité tend à diminuer à long terme, mais en revanche, dans le cas inverse, s'ils se concentrent sur la qualité, celle-là augmente et les coûts baissent [9]. Le lien existant parmi les plates-formes de *Crowdsourcing* est la possibilité de poster des tâches à coûts faibles mais avec une incertitude élevée sur la qualité, laquelle servira par la suite comme un indicateur pour les travailleurs.

Exemple 2.1.1 *Considérons la tâche de classification suivante : un travailleur doit observer avec attention un site web et ensuite décider s'il y a des contenus pour adultes sur la page. Le travailleur doit classer la page dans l'une des deux catégories : G pour aucun contenu pour adultes et R pour contenu pour adultes. Le taux d'achèvement, lorsqu'elle est effectuée par un stagiaire qualifié était de 250 site Web par heure, avec un coût de \$15/hr. Sur Amazon Mechanical Turk, le taux de classification est arrivé jusqu'à 2500 site Web par heure et le coût global est resté le même, y compris avec une amélioration de la productivité mais avec une qualité incertaine (pouvant être vérifiée par un examinateur mais avec une augmentation du coût) [10].*

Le problème réside en plusieurs facteurs liés au travailleur : il peut – même avec de bonnes compétences – faire des erreurs en répondant à la tâche [7], et ne garantit pas toujours d'avoir les capacités suffisantes pour atteindre un niveau de qualité satisfaisant [8]. L'existence de spammeurs répondant au hasard aux questionnaires [10] et aux incitations peu motivantes (e.g. un paiement moyen de 2 \$/heure ²) peuvent aussi influencer négativement la qualité [1].

D'après Y. Baba et al [8], il existe actuellement plusieurs approches dans le but de vérifier la qualité qui peuvent être classées en deux groupes : les approches *supervisées* et *non supervisées*. L'approche supervisée utilise l'ensemble des données justes (i.e. gold

¹Aussi appelé en anglais : *Crowd Worker*

²<http://www.behind-the-enemy-lines.com/2010/07/mechanical-turk-low-wages-and-market.html>

standard) afin d'estimer les capacités du travailleur en faisant tout d'abord des tests au hasard sur ces derniers et en vérifiant par la suite leurs réponses sur la base de données des gold standards. L'approche non-supervisée utilise la *redondance des réponses*³ au lieu des ensembles de données gold standard pour trouver les meilleurs travailleurs avec des méthodes tels que *Majority Voting* (e.g. [11]), *Maximum de vraisemblance* (e.g. [7] et [4]), *estimation de la qualité statistique non supervisée* (e.g. [8]) et *des techniques d'agrégation statistiques sophistiquées* [11].

2.2 Méthodes d'estimation des tâches structurées

Afin de pouvoir mesurer la qualité des tâches structurées publiées sur la plateforme *Crowdsourcing*, telles que des questions de logique binaire (e.g. questions de oui ou non) et les questions à choix multiples (e.g. notation de cinq point). Certaines de ces stratégies sont présentées plus en détails par la suite :

Majority voting

Majority voting [7, 10, 11] est la méthode la plus fréquemment utilisée et facilite la mise en pratique des étiquetages multiples. Du fait que celle-ci consiste à prendre l'étiquette choisie par la majorité par exemple. Ainsi, étant donné un problème de *classification binaire* (aussi utilisé pour *classification multi-classe*) avec un ensemble d'apprentissage $D = \{(x_i, y_i)\}_{i=1}^N$ contenant N instances, où $x_i \in X$ est une instance et $y_i \in Y$ est l'étiquette correspondantes. Dans l'exemple précédent, la dimension est deux : $y_i = \{1, 0\}$ et la bonne réponse peut être évaluée de la manière suivante :

$$\hat{y}_i = \begin{cases} 1 & \text{Si } (1/M) \sum_{j=1}^M y_i^j \geq 0.5 \\ 0 & \text{Si } (1/M) \sum_{j=1}^M y_i^j < 0.5 \end{cases}$$

D'après V. Raykar et al [4], un *majority voting* sans ensemble de données gold standard ne conduit pas à de bons résultats. Par exemple, dans le contexte du *Crowdsourcing*, nous avons des annotateurs (les travailleurs en ligne) distribués de la façon suivante ; un expert et plusieurs débutants, ainsi qu'un ensemble d'extraits de documents classés par catégories. A cause du nombre de débutants, la probabilité que l'extrait du document soit mal classé est très élevée. Cependant, nous pouvons augmenter la précision de la mesure en calculant un poids de qualité pour chaque annotateur avec l'aide des ensembles de données gold standard.

Matrice de confusion

La *matrice de confusion* (MC) est un instrument qui a pour but de mesurer la précision d'un classifieur à N étiquettes, avec une matrice carrée de N étiquettes réelle par N étiquettes estimée. Autrement dit, nous calculons la probabilité que l'utilisateur mette i lorsque la bonne réponse est j (c-à-d $\Pr[\text{estimée} = i \mid \text{réelle} = j]$). Cette matrice a été

³C'est-à-dire, une même tâche est accompli par plusieurs travailleur. Il faudra également faire remarquer qu'à plus des tâches publié sur la plate-forme, le service coûtera donc plus cher.

"mentionnée" pour la première fois par Dawid et Skene [7] en tant que matrice *Error-rates probabilities*. V. Raykas et al [4] les relie aux termes de *sensibilité* et de *spécificité*, dans le cas d'une classification binaire. Les figures ci-dessous illustrent la matrice de confusion :

$$\begin{pmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad \begin{pmatrix} p & p & \dots & p \\ p & p & \dots & p \\ \vdots & \vdots & \ddots & \vdots \\ p & p & \dots & p \end{pmatrix}$$

(a) Probabilités de la MC (b) Meilleur des cas (c) Pire des cas

FIG. 2.1: Matrices de confusion

La somme de probabilités de chaque ligne de la matrice 2.1a est la distribution de la probabilité qu'une étiquette réelle soit classée par rapport à une étiquette estimée, c'est-à-dire que leur somme sera égale à 1 ($\sum_{j=1}^N p_{ij} = 1$, $\forall i \in \{1, 2, 3, \dots, N\}$). Nous pouvons également souligner que la matrice 2.1b représente le meilleur des cas grâce à une probabilité estimée/réelle de 1 sur la diagonale (les objets ont donc systématiquement été bien classés) et la matrice 2.1c représente le pire des cas avec une probabilité estimée/réelle p dans tous les cas (les objets ont été affectés aléatoirement, ce qui ne permet pas d'inférer le moindre résultat).

Et enfin, si la matrice de confusion représente l'estimation d'un annotateur, nous pouvons alors inférer, étant donné un objet et pour chaque classe, la probabilité que ce soit la bonne réponse, la probabilité d'une allocation correcte de l'annotateur (avec la multiplication de la diagonale) et le taux d'erreur total de l'annotateur (1 - probabilité d'allocation correcte) [7, 5, 10].

2.3 Méthodes d'estimation des tâches non-structurées

Dans cette section, nous analyserons les méthodes d'évaluation de la qualité des tâches non-structurées, telles que la rédaction d'un article de journal ou la conception d'un logo, éléments qui ne peuvent pas facilement être classés comme mauvais ou bon [8].

Méthode statistique non-supervisée

Y. Baba nous propose dans [8] une méthode pour traiter ces cas ; il sépare l'analyse en deux étapes : *l'étape de création* de *l'étape de révision* de la manière illustrée dans la figure ci-dessous 2.2.

1. **L'étape de création** ; plusieurs travailleurs (appelés des auteurs) sont assignés à plusieurs tâches non-structurées en créant des artefacts (e.g. création d'un logo). Cette étape envisage ; (1) la capacité du travailleur et (2) la performance de la tâche (e.g. quelqu'un sans une grande compétence en traduction pourrait-il traduire un extrait d'un document technique de l'anglais vers le français sans problème).



FIG. 2.2: Schéma de validation de qualité

2. **L'étape de révision**; les artefacts achevés passent par cette étape où chacun d'entre d'eux sont révisés par plusieurs travailleurs (appelés des examinateurs), les tâches révisées sont souvent notés par une question à choix multiples (e.g. excellent, bien, moyenne, juste, pauvre). Cette étape envisage; (1) un biais basé sur chaque examinateur qui est directement proportionnelle par rapport au grade de qualité que l'examinateur a donné à l'artefact crée auparavant, (2) ainsi que les préférences de l'examinateur car il pourrait préférer réviser des tâches courtes et non des tâches longues.

Et enfin, cette méthode surpasse celles comme majority voting et le modèle logit ordinal[6](ou régression logistique ordinale). Cependant, nous pouvons constater que la méthode ne prend pas en compte le fait que le travailleur ou l'examinateur peut se tromper en soumettant la réponse, même en ayant de bonnes compétences et de bonnes intentions.

Modelisation

Après avoir examiné l'état de l'art des méthodes existantes correspondants à l'étape d'évaluation, ou d'inférence, des solutions de *crowdsourcing*. Nous allons proposer un modèle statistique afin de *modéliser l'usage des utilisateurs dans les systèmes de Crowdsourcing* en tenant compte du comportement aléatoire des types de travailleurs et de leurs compétences.

Nous allons tout d'abord présenter un schéma détaillé du déroulement des étapes de Crowdsourcing et où notre modèle de simulation participe, dans la figure 3.1, pour mieux comprendre le contexte du problème

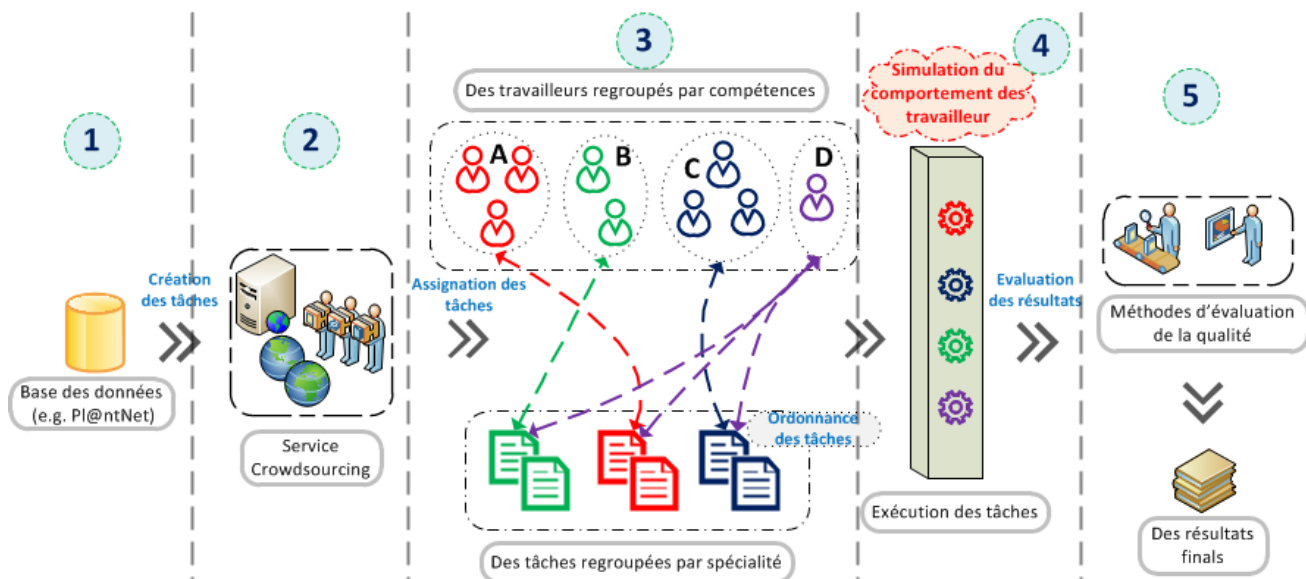


FIG. 3.1: Schéma détaillé d'évaluation d'usage des systèmes de Crowdsourcing. Le schéma est divisé en 5 étapes indispensables.

Sur la figure 3.1, le processus débute à l'étape 1 par la nécessité de résoudre de petites tâches d'une organisation (e.g. l'étiquetage du contenu d'une image ou classification des espèces de plantes). Ensuite, lors de l'étape 2, nous cherchons un service de *Crowdsourcing* dans le but de les y poster (i.e. creation et mise en ligne des tâches sur l'internet). Ces tâches sont ensuite regroupées par "spécialité" ou par "difficulté" (e.g. des tâches rouges, vertes ou bleues) et ensuite assignées aux travailleurs qui sont aussi regroupés par compétences (e.g profil A, B, C ou D). Au cours de l'étape suivante, les

utilisateurs répondront aux tâches assignées et c'est là que nous proposerons un modèle qui va s'appuyer sur la définition de matrice de confusion pour simuler les réponses des utilisateurs. Enfin, à l'étape 5, nous utiliserons des algorithmes de l'état de l'art pour évaluer la qualité de réponse des utilisateurs et pouvoir obtenir des résultats finaux.

Algorithme Général

L'algorithme 1 simule l'étape 4 de la figure 3.1 : il prend comme arguments d'entrée un ensemble de tâches et un ensemble d'utilisateurs regroupés par profil. Ensuite le traitement des tâches par utilisateur, l'algorithme retourne l'ensemble de réponses estimées $\{\hat{U}_1, \hat{U}_2, \dots, \hat{U}_n\}$.

Algorithm 1 Simulation naïf

Entrée: Ensemble des tâches T_i

Entrée: Ensemble des utilisateurs regroupés par profil P_i

Sortie: Ensemble des réponses estimées \mathbf{U}

```

1: for each  $T_i \in \{T_1, T_2, \dots, T_n\}$  do
2:   for each  $P_i \in \{P_1, P_2, \dots, P_m\}$  do
3:     /* Utilisateur  $u_i$  appartenant au profil  $P_i$  */
4:     for each  $u_i \in P_i$  do
5:        $\hat{U}_i = \text{response\_estimated}(T_i, u_i)$ 
6:     end for
7:   end for
8: end for
9: return  $\hat{U}_i \in \{\hat{U}_1, \hat{U}_2, \dots, \hat{U}_n\}$ 

```

La représentation graphique de l'algorithme 1 se trouve sur la figure 3.2.

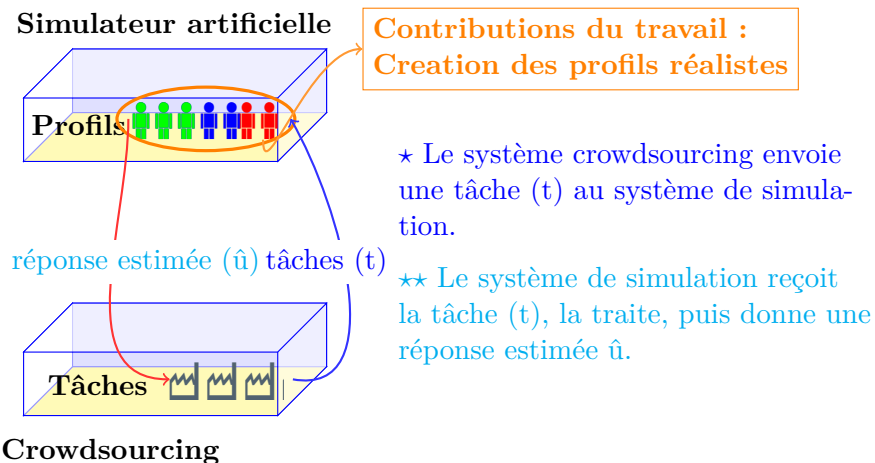


FIG. 3.2: *Représentation du modèle de simulation artificielle.* Sur la boîte du simulateur artificiel, il y a de petites poupées qui représentent les différents utilisateurs, chacun avec une couleur représentant un profil.

Nous tenons à souligner, dès à présent, que notre contributions se focalise sur *la construction de profils des utilisateurs* dans le but de simuler des réponses réelles d'un utilisateur.

3.1 Approche générale

Les méthodes existantes correspondant à l'étape d'évaluation des solutions du Crowdsourcing, étudiées dans l'état de l'art, nous ont permis de découper notre travail dans un schéma composé de plusieurs processus nécessaires à l'accomplissement de la simulation des usages utilisateurs. La figure 3.3 ci-dessous schématise ces processus.

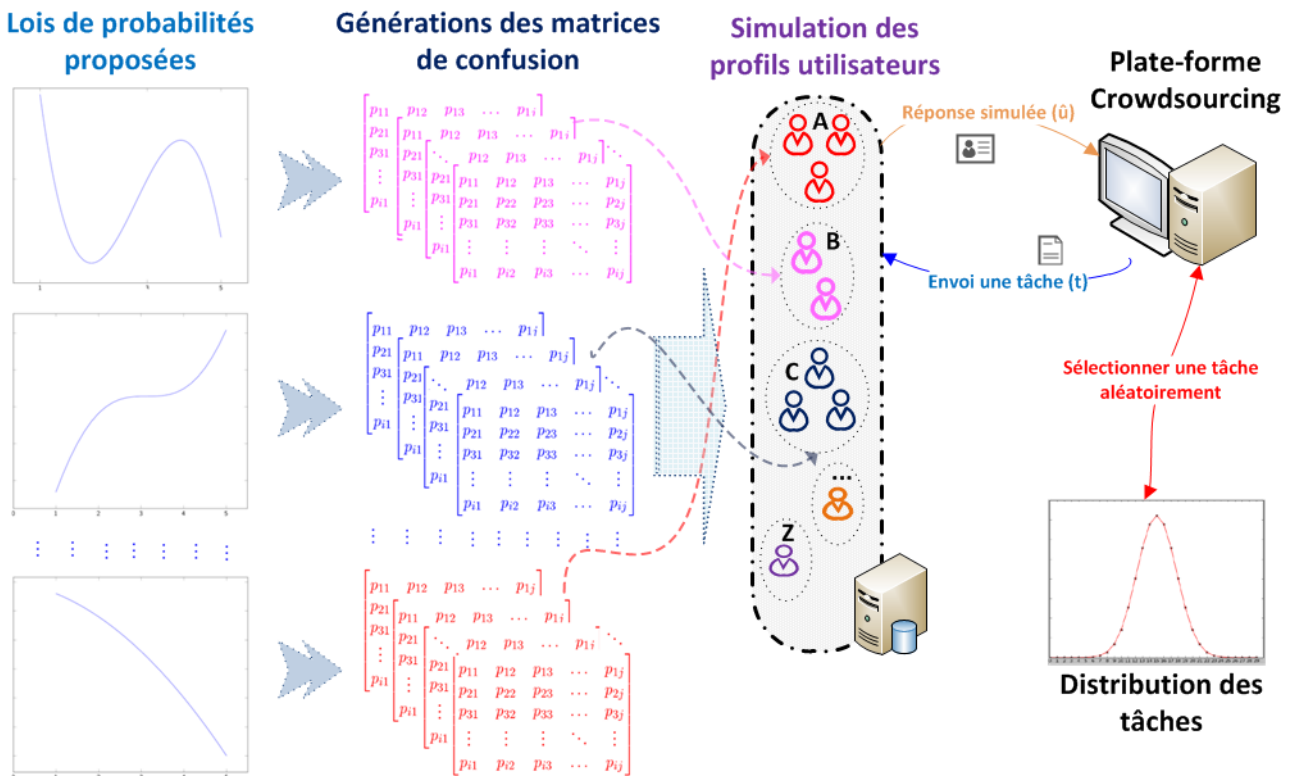


FIG. 3.3: *Schéma des processus à accomplir pour la simulation artificielle.* En gros, le schéma débute en proposant des lois de probabilité par profil, ensuite c'est la génération des matrices de confusion, et finalement, c'est la simulation artificielle des utilisateurs.

Formulation du problème

Étant donné un scénario typique d'apprentissage supervisé consistant à entraîner des utilisateurs sur des données d'apprentissage avec une méthode de classification ayant T classes. Nous nous intéressons à l'estimateur $\hat{\mathbf{u}}$ qui mesure la qualité qu'un utilisateur propose la classe \mathbf{y}_i en sachant que nous connaissons à priori la vraie classe \mathbf{y}_{true} (i.e $\mathbf{Pr}[\mathbf{y}_i | \mathbf{y}_{\text{true}}] = \hat{\mathbf{u}}$). Ainsi donc, nous nous appuyerons sur l'outil existant nommé *Matrice de Confusion* qui sert à mesurer la qualité d'un système de classification, qui

a déjà été abordé dans l'état de l'art en sous-section 2.2. Voici ci-dessous un exemple. Dans cette perspective, nous aurons une matrice de confusion par utilisateur représen-

		Réponse estimée					
		c_1	\dots	c_j	\dots	c_t	
Vraie réponse	c_1	p_{11}	\dots	c_{1j}	\dots	p_{1t}	\Rightarrow La mesure de qualité $\hat{u}_{i,j}$ est de 98% de certitude par rapport à la réponse estimée c_j en sachant la vraie réponse c_i . C'est-à-dire, qu'il y a de fortes chances que c_j soit la vraie classe $c_i \approx c_j$
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
	c_i	0.01	\dots	0.98	\dots	0	
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	
	c_t	p_{t1}	\dots	c_{tj}	\dots	p_{tt}	

FIG. 3.4: Exemple d'obtention de la mesure de qualité

tant leurs mesures de qualité de la réponse d'avoir bien classifiée une classe. Ainsi donc, nous aurons par chaque utilisateur une matrice carrée de T classes où chaque ligne i représente la vraie classe, chaque colonne j représente la classe estimée, chaque cellule de la matrice représente la mesure de l'estimateur $\hat{u}_{i,j}$ et où la somme des estimateurs $\sum_{j=1}^T \hat{u}_{i,j}$ d'une ligne est toujours égale à 1. Voici la représentation de la matrice carrée de confusion par utilisateur :

		Réponses estimées						
		c_1	c_2	\dots	c_j	\dots		c_t
Vraies réponses	c_1	p_{11}	p_{12}	\dots	p_{1j}	\dots	p_{1t}	$c_i \begin{bmatrix} \hat{u}_{i1} & \hat{u}_{i2} & \dots & \hat{u}_{ij} & \dots & \hat{u}_{it} \end{bmatrix}$ Donc, si l'indice i est la vraie reponse et $\forall j \in \{1, 2, 3, \dots, T\}$ $\sum_{j=1}^T \Pr [c_j c_i] = \sum_{j=1}^T \hat{u}_{i,j} = 1$
	c_2	p_{21}	p_{22}	\dots	p_{2j}	\dots	p_{2t}	
	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	
	c_i	p_{i1}	p_{i2}	\dots	p_{ij}	\dots	p_{it}	
	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	
c_t	p_{t1}	p_{t2}	\dots	p_{tj}	\dots	p_{tt}		

Nous pouvons donc à présent conclure que **chaque ligne de la matrice exprime une loi de probabilité discrète** et c'est grâce à cette conclusion que nous pourrons proposer des lois de probabilités existantes dans le but de modéliser le système de simulation des usages utilisateurs.

Propositions des lois de probabilités

Afin de valider notre approche générale et de continuer à affiner notre modèle de simulation, nous allons proposer quatre profils différents qui suivront chacun une loi de probabilité. Les profils proposés sont : *Expert*, *Amateur*, *Novice* et *Spammeur*.

Profil expert

Les utilisateurs avec ce profil auront une forte chance de bien répondre aux tâches. Autrement dit, la probabilité de réponse de l'utilisateur sur la classe c_i en sachant que

la vraie classe est aussi \mathbf{c}_i , doit être beaucoup plus forte que les autres probabilités ; $\Pr[\mathbf{c}_j | \mathbf{c}_i] \gg \Pr[\mathbf{c}_j | \mathbf{c}_i]$. En outre, les valeurs de la courbe doivent s'éloigner rapidement de la vraie classe et doivent avoir un comportement décroissant. Ainsi donc, la loi de probabilité discrète proposée pour modéliser ce profil est la loi logarithmique. Voici, ci-dessous sur la figure 3.1, un exemple de la distribution des probabilités du profil expert, étant donné que la vraie classe est la première. Notons que nous avons émis l'hypothèse que deux classes proches dans la matrice de confusion seront proches sémantiquement.

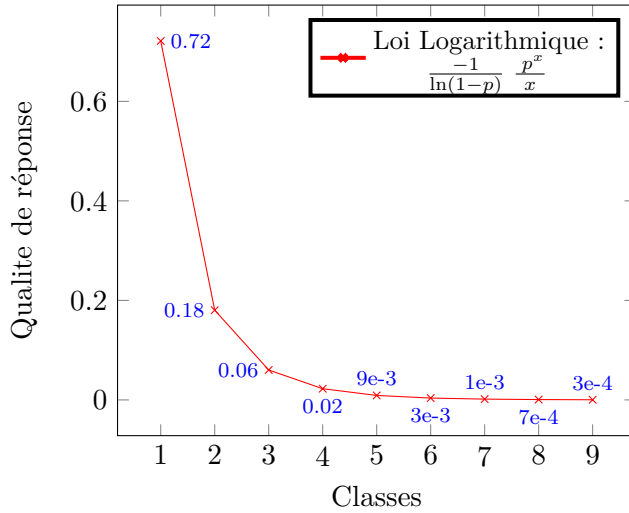


FIG. 3.5: La distribution des probabilités du profil expert étant donné que la vraie classe est la première

Profil amateur

Les utilisateurs avec ce profil auront une probabilité plus forte de se tromper. Autrement dit, les probabilités de réponse d'utilisateur seront proche l'une de l'autre ; e.g $\Pr[\mathbf{c}_j | \mathbf{c}_i] \gg \Pr[\mathbf{c}_{j+1} | \mathbf{c}_i]$. Ainsi donc, la loi de probabilité discrète proposée pour modéliser ce profil est la loi de Laplace. La figure 3.1 illustre un exemple de la distribution des probabilités du profil amateur.

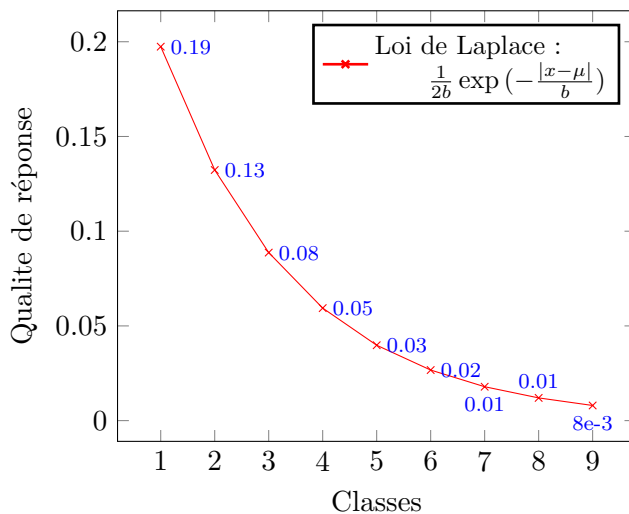


FIG. 3.6: La distribution des probabilités du profil amateur étant donné que la vraie classe est la première.

Profil novice

Les utilisateurs avec ce profil auront une chance aléatoire uniforme de bien répondre aux tâches assignées. Autrement dit, les profils novices suivront une loi de probabilité uniforme avec un générateur de nombre pseudo-aléatoires. Lorsqu'on générera la matrice, cela consistera à tirer aléatoirement les valeurs entre 0 et 1 : $\Pr[c_j | c_i] = U_{random} \forall U \in [0, 1]$. La figure 3.1 illustre un exemple de la distribution des probabilités du profil amateur.

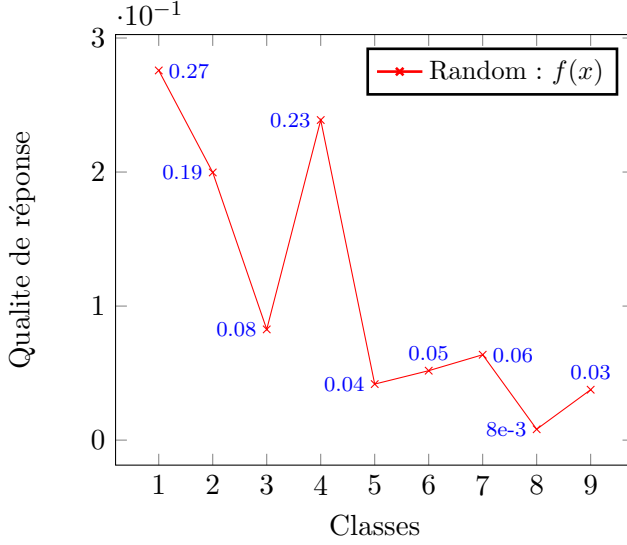


FIG. 3.7: La distribution des probabilités du profil novice étant donné que la vraie classe est la première.

Profil spammeur

Les utilisateurs avec ce profil auront autant de chance de se tromper que de bien répondre aux tâches assignées. Autrement dit, les probabilités de réponse de l'utilisateur seront tellement proche l'une de l'autre qu'il sera possible de confondre la probabilité c_i avec leurs ultérieures c_{i+1} et leurs antérieures c_{i-1} ; $\Pr[c_{j-1} | c_i] \approx \Pr[c_j | c_i] \approx \Pr[c_{j+1} | c_i]$. Ainsi donc, la loi de probabilité discrète proposée est :

$$\Pr[c_j | c_i] = \begin{cases} \frac{1}{n} + \beta & \text{Si } j \text{ est pair} \\ \frac{1}{n} - \beta & \text{Si } j \text{ est impair} \end{cases}$$

La figure 3.1 illustre un exemple de la distribution des probabilités du profil amateur.

Génération des matrices de confusion

Après avoir défini une loi de probabilité discrète par profil, nous allons donc décrire dans cette sous-section les pseudo-algorithmes génériques nécessaires pour générer des ensembles d'utilisateurs par profil, autrement dit, des matrices de confusion par profil dont une matrice est définie comme suit :

Définition 3.1.1 *Étant donné le profil $n \in \{1, 2, \dots, N\}$. La qualité de réponse d'un utilisateur $u_{n,k} \forall k \in \{1, 2, 3, \dots, K\}$ est représentée par une matrice de confusion de T*

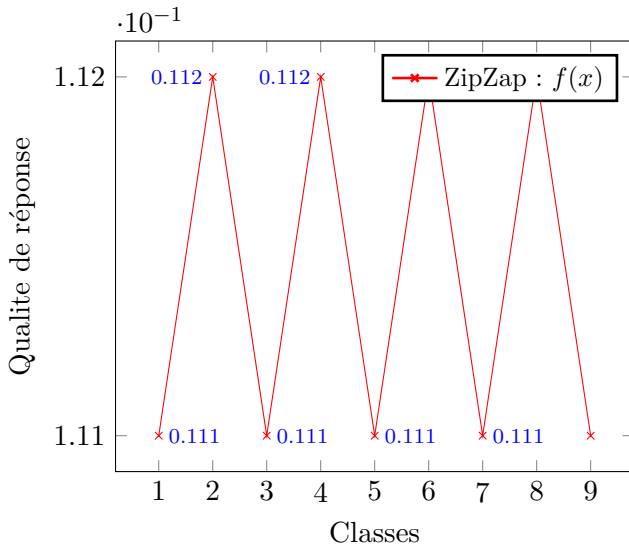


FIG. 3.8: La distribution des probabilités du profil spammeur étant donné que la vraie classe est la première.

classes, $c_i \forall i \in \{1, 2, 3, \dots, T\}$ où chaque cellule de la matrice a une probabilité p_{ij} étant la mesure de qualité qu'un utilisateur puisse estimer une classes j correctement lorsque la vraie classe est i . La représentation en probabilité est $P(\text{réponse} = j \mid \text{vraie} = i) = p_{ij}$. Voici ci-dessous la matrice de confusion de l'utilisateur k appartenant au profil n .

		Réponse de l'utilisateur				
		c_1	c_2	c_3	\dots	c_t
La vraie réponse	c_1	p_{11}	p_{12}	p_{13}	\dots	p_{1t}
	c_2	p_{21}	p_{22}	p_{23}	\dots	p_{2t}
	c_3	p_{31}	p_{32}	p_{33}	\dots	p_{3t}
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
	c_t	p_{t1}	p_{t2}	p_{t3}	\dots	p_{tt}

Nous allons à présent introduire l'algorithme 2 de génération des matrices de confusion. Celle-ci reçoit comme argument d'entrée un ensemble de probabilités qui suit une loi de probabilité d'un profil (i.e. $P \sim \Phi_{profil}(x)$ et $\sum_{j=1}^T P_j = 1$). Ensuite, afin de réutiliser la loi de probabilité adaptée à la première vraie classe c_1 de la matrice de confusion et d'en propager pour toutes les autres vraies classes c_i de la matrice de confusion, différente à la première ligne, nous utiliserons une astuce en faisant bouger la probabilité principale \mathbf{P}_0 sur la diagonale de la matrice (ligne 6 de l'algorithme 2) : $\mathbf{M}_{k,i,j} = \mathbf{P}_{|j-i|} \forall i \in \text{ligne et } j \in \text{colonne}$. Le tableau 3.1 illustre la représentation de la matrice et ses probabilités.

L'algorithme 4 a pour but de prendre la loi de probabilité discrète proposée par profil $\{P_0, P_1, P_2, \dots, P_t\} \sim \Phi_{profil}$ et de générer une autre loi de probabilité nommée réaliste par utilisateur $\{\hat{P}_0^r, \hat{P}_1^r, \hat{P}_2^r, \dots, \hat{P}_t^r\} \sim \Phi_{profil}^r$ en se servant de la méthode de simulation de Monte-Carlos. Ainsi donc, nous allons utiliser la méthode de transformation inverse [12, p. 16] qui repose sur le résultat suivant :

Théorème 1 Soit $F_X(x)$ une fonction de répartition de la loi de probabilité Φ_{profil}

	1	2	3	...	j
1	$P_{ 1-1 }$	$P_{ 2-1 }$	$P_{ 3-1 }$...	$P_{ j-1 }$
2	$P_{ 1-2 }$	$P_{ 2-2 }$	$P_{ 3-2 }$...	$P_{ j-2 }$
3	$P_{ 1-3 }$	$P_{ 2-3 }$	$P_{ 3-3 }$...	$P_{ j-3 }$
⋮	⋮	⋮	⋮	⋮	⋮
i	$P_{ 1-i }$	$P_{ 2-i }$	$P_{ 3-i }$...	$P_{ j-i }$

TAB. 3.1: *Loi de probabilité adaptée à la première vraie classe (i.e la première ligne) : $\{P_0, P_1, P_2, P_3, \dots, P_T\} \sim \Phi_{profil}$.*

Algorithm 2 Génération des matrices de confusion

Entrée: Ensemble des probabilités du profil proposé \mathbf{P}

Entrée: K nombre des utilisateurs, T nombre des classe

Sortie: Matrice de confusion $\mathbf{M}_{\mathbf{N} \times \mathbf{T} \times \mathbf{T}}$

```

1: for k :=1 to K do
2:   for i :=1 to T do
3:     for j :=1 to T do
4:        $M_{k,i,j} := P_{|j-i|}$ 
5:     end for
6:      $M_{k,i} := \text{génération\_courbe\_réaliste}(M_{k,i}, 10^5)$ 
7:      $M_{k,i} := \text{normalisation\_courbe}(M_{k,i})$ 
8:   end for
9: end for
10: return  $M_{T \times T}^K \in \{M_1, M_2, \dots, M_k\}$ 

```

définie sur un intervalle $[a, b]$, de fonction inverse :

$$F^{-1}(u) = \inf\{z \in [a, b] : F(z) \leq u\}$$

Si U est une variable de loi uniforme sur $[0, 1]$, alors $Z = F^{-1}(U)$ est distribuée suivante F et l'histogramme Z génère la loi de probabilité réaliste.

Algorithm 3 Normalisation de la loi de probabilité discrète

Entrée: Ensemble des probabilités \mathbf{P}

Sortie: Ensemble des probabilités normalisés \mathbf{P}^n

```

1:  $P_{total} := \sum_{j=1}^T P_j$ 
2: error :=  $P_{total} - 1$ 
3: if |error| ≥ 1e-5 then
4:   ratio = error /  $P_{total}$ 
5:   for each  $P_i \in \{P_1, P_2, \dots, P_t\}$  do
6:      $P_i^n := P_i - P_i * ratio$ 
7:   end for
8: end if
9: return  $\hat{P}_i^n \in \{\hat{P}_0^n, \hat{P}_1^n, \hat{P}_2^n, \dots, \hat{P}_t^n\}$ 

```

L'algorithme 3 a pour but de normaliser la courbe afin que la somme de l'ensemble des classes donne une probabilité de 1.

Algorithm 4 Génération de courbe réaliste par Monte-Carlo**Entrée:** Ensemble des probabilités $\mathbf{P}_k \in \{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_t\} \sim \Phi_{profil}$ **Entrée:** Nombre de simulations \mathbf{N} **Sortie:** Ensemble des probabilités réalistes $\hat{\mathbf{P}}^r$

```

1:  $H := array[N]$ 
2: for  $i := 1$  to  $\mathbf{N}$  do
3:    $x := U_{random_{[0,1]}}$ 
4:   /*  $F_X(x)$  Fonction de répartition de la loi de probabilité  $P_k$ . */
5:   if  $F_X(k) \leq x < F_X(k + 1)$  then
6:      $H_i := k$ 
7:   end if
8: end for
9:  $\hat{\mathbf{P}}^r := histogramme(H)$ 
10: return  $\hat{\mathbf{P}}_i^r \in \{\hat{\mathbf{P}}_0^r, \hat{\mathbf{P}}_1^r, \hat{\mathbf{P}}_2^r, \dots, \hat{\mathbf{P}}_t^r\}$ 

```

L'exécution des trois algorithmes (i.e. des algorithmes 2, 4 et 3) sur l'un des profils proposés nous produira un ensemble de matrices de confusion différentes par profil, tout en respectant la loi de probabilité du profil. Prenons l'exemple du profil expert (i.e loi logarithmique), sur la figure 3.9, étant donné que la vraie reponse est la vingtième classe.

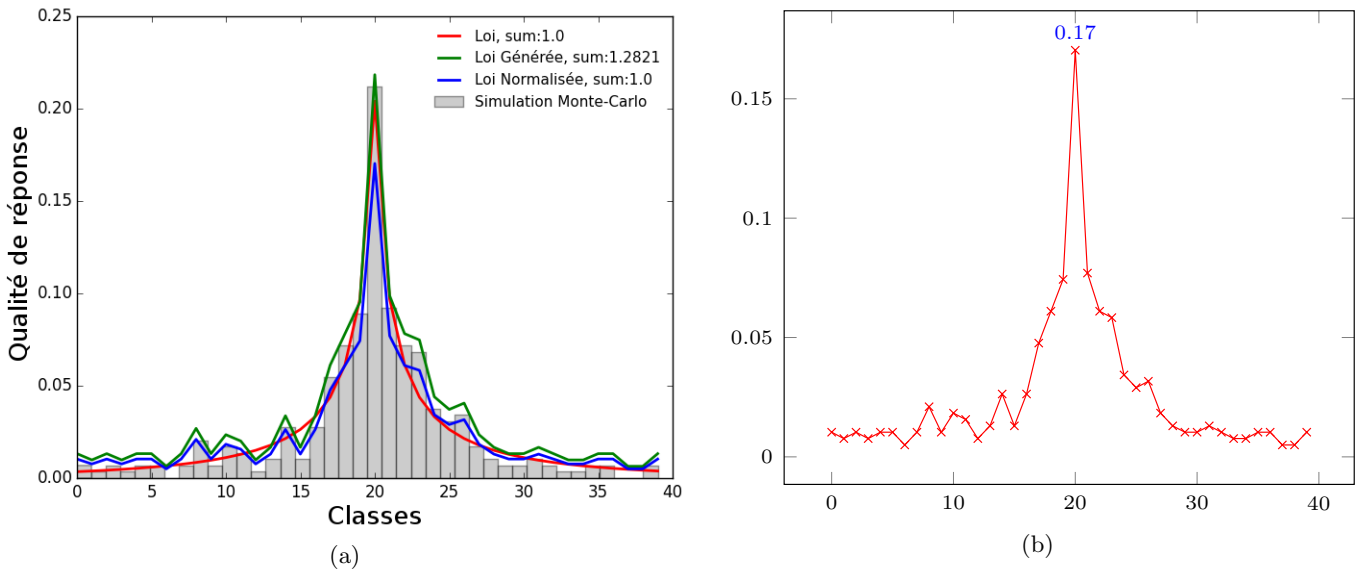


FIG. 3.9: **Génération de courbe plus réaliste.** (a) Le traitement de la loi de probabilité du profil Φ_{profil} : la courbe rouge suit la loi de probabilité du profil, la courbe verte est générée grâce à la simulation de Monte-Carlos (i.e méthode de la transformation inverse) et la courbe bleue est la loi normalisée à 1. (b) La courbe des probabilités finale normalisée à 1.

La figure 3.10 illustre une matrice de confusion d'un utilisateur de profil expert en 3D.

Matrice de confusion (6x6) :

		Réponses estimées					
		c_1	c_2	c_3	c_4	c_5	c_6
Vraies réponse	c_1	0.46	0.12	0.01	8e-3	8e-3	0.01
	c_2	0.38	0.40	0.26	0.05	6e-3	0.03
	c_3	0.09	0.34	0.28	0.23	0.06	0.07
	c_4	0.02	0.09	0.32	0.44	0.19	0.07
	c_5	0.02	0.02	0.08	0.18	0.61	0.16
	c_6	5e-3	0.01	0.03	0.06	0.11	0.65

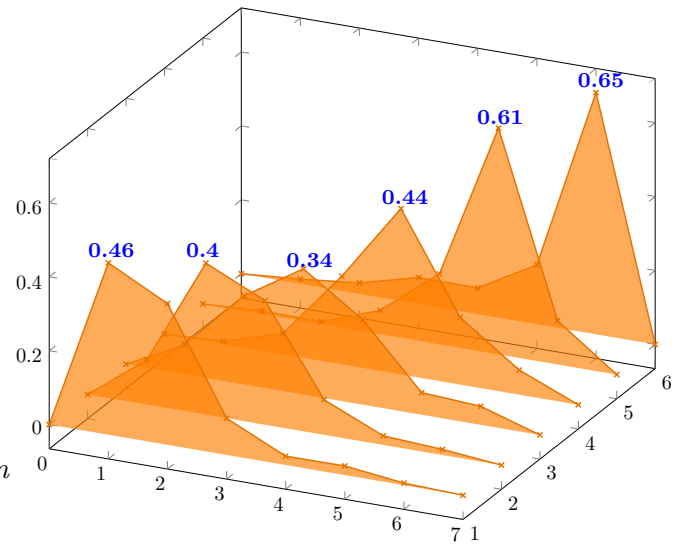


FIG. 3.10: Visualisation de la matrice de confusion en 3D.

Simulation des usages utilisateurs

Après avoir créé l'ensemble des matrices de confusion par profils, nous allons donc discuter à cette occasion du tirage aléatoire sur la fonction cumulée (i.e. fonction de répartition) de la loi discrète de la matrice de confusion étant donné une vraie classe afin de calculer la réponse de l'utilisateur aléatoirement.

Définition 3.1.2 Étant donné T classes, la matrice de confusion $M_{k,t,t}$ de l'utilisateur k , une vraie classe $v \in \{0, 1, 2, \dots, T\}$ et un nombre aléatoire uniforme $u \sim \mathcal{U} \in [0, 1]$, nous cherchons donc calculer la réponse de l'utilisateur k . Ainsi donc, nous allons générer la fonction de répartition $F_X(x)$ de la loi de probabilité de la vraie classe P_v (i.e. $F_X(x) = \sum_{j=0}^x P_{v,j}$) et cherche la réponse $k \in \{0, 1, 2, \dots, T\}$ tel que $F_X(k) \leq u < F_X(k+1)$.

Voici ci-dessous un exemple sur la figure 3.11.

Exemple :

Étant donné la fonction de répartition $F_X(x)$ de la vraie classe N° 3 de la matrice de confusion (i.e. ligne 3) sur la figure 3.10 et un nombre aléatoire uniforme $u = 0.79$ (i.e. $u \sim \mathcal{U} \in [0, 1]$).

Nous devons calculer $k \in \{0, 1, \dots, T\}$ tel que $F_X(k) \leq u < F_X(k+1)$.

Ainsi donc, dans notre exemple k est égal à 3.

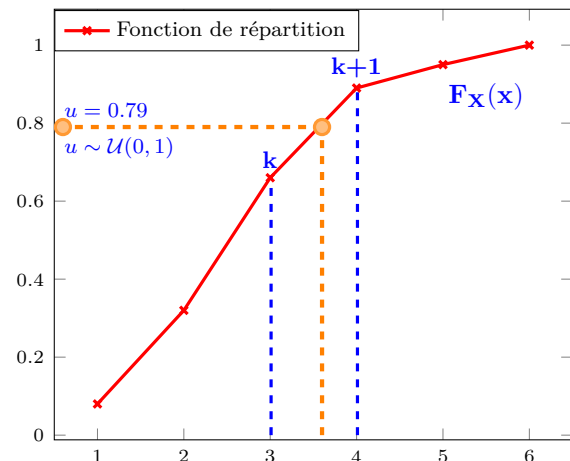


FIG. 3.11: Exemple de calcul de réponse d'un utilisateur

3.2 Approche Réaliste

Les profils proposés dans la section précédente ont été créés de façon synthétique afin de valider notre approche d'exploitation de la matrice de confusion qui sert à mesurer la qualité de réponse d'un utilisateur.

Nous allons présenter une approche encore plus réaliste en utilisant des données réelles d'un site web (e.g. Tela-botanica ¹) dans le but de générer des matrices de confusion encore plus réaliste. La figure 3.12 ci-dessous schématise les processus nécessaires à l'accomplissement de l'extraction des matrices de confusion.

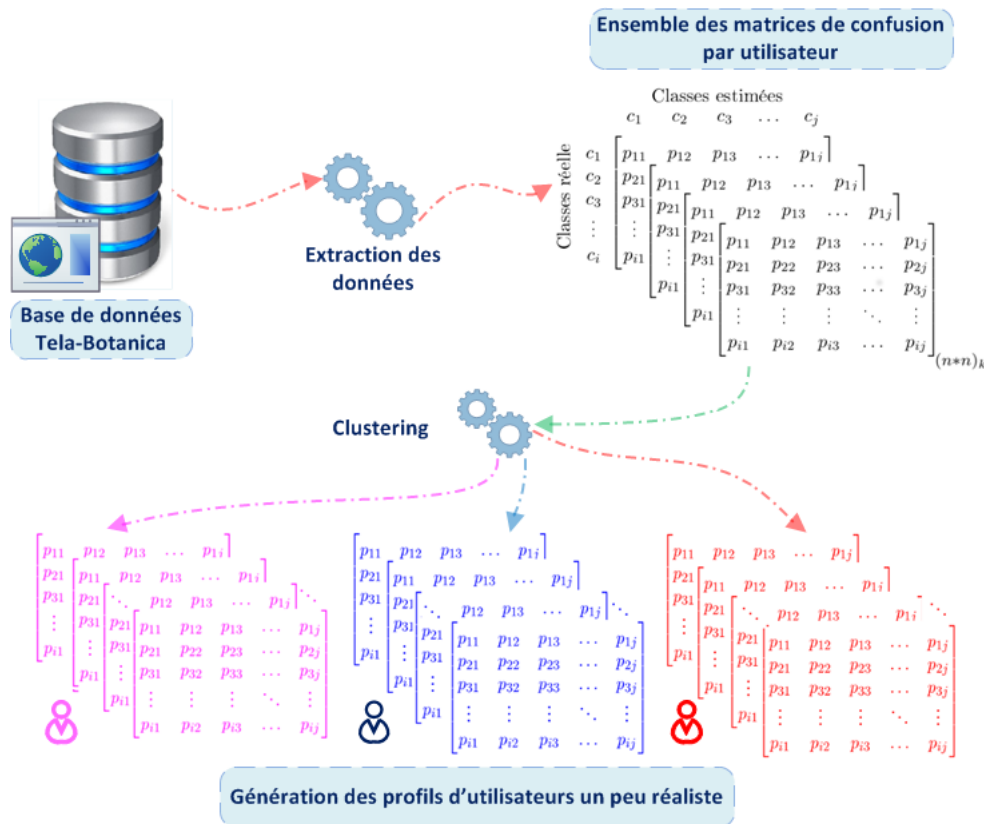


FIG. 3.12: Schéma d'extraction des matrices de confusion plus réalistes

Extraction de données

Nous allons d'abord analyser le site web *Tela-Botanica* en vue de connaître quelles sont les données dont nous allons avoir besoin. La figure 3.13 illustre la publication d'une espèce de plante faite par quelqu'un et associée à des commentaires ainsi qu'à des propositions d'autres personnes, en outre, des votes pour ou contre par proposition et le plus important est la vraie espèce de plante vérifiée. (i.e. la vraie réponse).

¹www.tela-botanica.org

Fuligo septica proposé par Marc CHOUILLOU le 07/02/2014

Votes Pour **100,00%** Votes Contre **0,00%**

Françoise CARLE 16/02/2014

☹ Ces votes permettent de confirmer ou non une détermination proposée par un membre du réseau. Vous pouvez changer à tout moment votre vote à l'aide de ou . Une pondération s'opère pour le calcul des votes : vote en tant que membre identifié (3 points) / non identifié (1 point).

■■■ DÉTERMINATION / CONFIRMATION

Proposer une détermination Ajouter un commentaire Suivre cette observation

La vraie réponse

Fuligo septica

Détermination proposée par Marc CHOUILLOU le 07/02/2014 Score Voter **3**

Fuligo septica est un champignon qui fait partie des Myxomycètes. C'est un groupe très difficile ; il faut le microscope pour les identifier avec certitude. Donc ma proposition est sans garantie !

Répondre

Françoise CARLE le 11/02/2014

Merci, pas grave, si c'est pas ça personne ne saura que c'est vous qui m'avez dit le nom, et ceux à qui je montrerai mes photos s'en foutent du nom, mais il faut quand même en donner un.

Répondre

Phra

Détermination originale par Françoise CARLE le 05/02/2014 Score Voter **0**

Répondre

Proposer une détermination Ajouter un commentaire

FIG. 3.13: Site web Tela-botanica, publication vérifiée d'une espèce de plante.

Nous allons donc extraire par publication vérifiée $^2 P_i^v \forall i \in \{1, 2, \dots, V\}$ la vraie espèce de plante proposée $C_v^{P_i^v}$ et l'ensemble de mauvaises espèces de plantes proposées $\{C_i^{P_i^v} : i \neq v \text{ et } C_i^{P_i^v} \in P_i^v\}$, afin de créer une matrice de confusion $M_{N \times N}^k$ tel que $N = |\{C_j : C_i \neq C_j \text{ et } i \neq j \text{ et } C_i, C_j \in \{P_1^v, P_2^v, \dots, P_V^v\}\}|$, par utilisateur k . Ensuite, nous procéderons au remplissage de la matrice avec l'algorithme 5.

Définition 3.2.1 La relation de transitivité binaire s'écrit ; étant donné une relation \mathcal{R} définie sur l'ensemble $E : \forall x, y, z \in E, \{(x\mathcal{R}y \wedge y\mathcal{R}z) \Rightarrow x\mathcal{R}z\}$.

Afin de remplir encore plus de confusions dans la matrice de confusion d'un utilisateur, nous nous servons du concept mathématique ancien mais puissant nommé « relation transitive binaire » (défini dans 3.2.1). Ce concept sera appliqué en utilisant un graphe orienté $\mathcal{G} = (V, E)$ tel que $V = \{C_1, C_2, \dots, C_n : C_i \neq C_j \text{ et } i \neq j \text{ et } C_i, C_j \in \{P_1^v, P_2^v, \dots, P_V^v\}\}$ et $E = \{(C_v^{P_i^v}, C_{adj}, W_{adj}) : (C_v^{P_i^v} \mathcal{R} C_{adj}) \text{ et } C_v^{P_i^v}, C_{adj} \in P_i^v \text{ et } P_i^v \in \{P_1^v, P_2^v, \dots, P_V^v\}\}$ où C_i est une espèce de plante et $C_v^{P_i^v}$ la vraie espèce de plante vérifiée de la publication vérifiée P_i^v . L'objectif est de trouver toutes les relations binaires

²Les étapes d'une publication sont trois ; "à déterminer" lorsqu'elle a été créée avec une espèce de plante proposée par l'auteur de la publication, commentée par d'autres personnes, votée pour ou contre, et affectée par d'autres propositions d'espèces de plantes, "à confirmer" lorsqu'elle a assez de votes pour confirmer une proposition, "vérifiée" lorsqu'une proposition a été choisie pour de bonne.

possibles, et pour cela, nous ferons tourner l'algorithme 6 de base. La figure 3.14 illustre les étapes de trouver des relations transitives dans un graphe orienté.

Algorithm 5 Remplissage de la matrice de confusion

Entrée: Ensemble des publications vérifiées \mathbf{P}^v

Sortie: Matrices de confusion $\mathbf{M}_{N \times T \times T}$

```

1: for each  $P_i^v \in \{P_1^v, P_2^v, \dots, P_v^v\}$  do
2:    $C_v^{P_i^v} := \text{extraire\_vraie\_réponse}(P_i^v)$ 
3:    $v := \text{indice}(C_v^{P_i^v})$ 
4:   /* Parcourir les propositions de la publication  $P_i^v$ . */
5:   for each  $(C_j, U_k) \in P_i^v = \{(C_1, U_a), (C_2, U_b), \dots, (C_c, U_r)\}$  do
6:      $j, k := \text{indice}(C_j), \text{indice}(U_k)$ 
7:     /* Parcourir les votes de la proposition  $C_j$ . */
8:     for each  $(V_z, U_{k'}) \in C_j = \{(V_1, U_{a'}), (V_2, U_{b'}), \dots, (V_m, U_{r'})\}$  do
9:        $k' := \text{indice}(U_{k'})$ 
10:       $M_{k',v,j} := M_{k',v,j} + 1$ 
11:    end for
12:     $M_{k,v,j} := M_{k,v,j} + 1$ 
13:  end for
14: end for
15: return  $M_{N \times N}^K \in \{M_1, M_2, \dots, M_k\}$ 

```

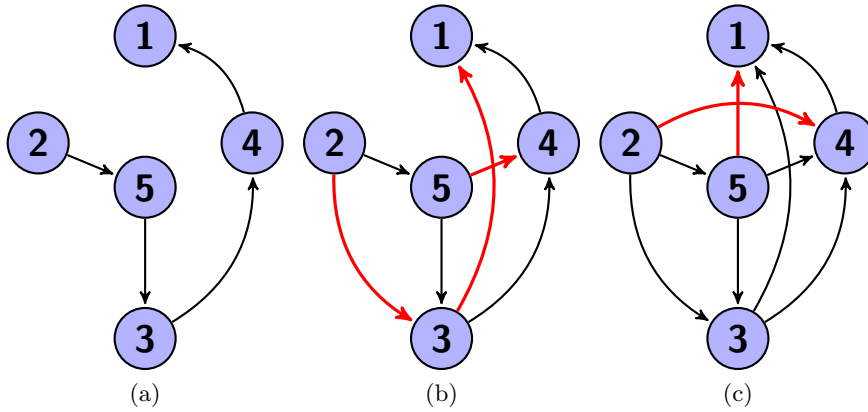


FIG. 3.14: **Exemple de relation transitive dans un graphe orienté.** (a) Le graphe initial. (b) Les arêtes en rouge sont les relations de transitivité trouvées à la première itération de l'algorithme (e.g. $(2 \mathcal{R} 5) \wedge (5 \mathcal{R} 3) \Rightarrow (2 \mathcal{R} 3)$) (c) La deuxième itération trouve encore des relations transitives et après il n'y en a plus.

En faisant d'abord tourner l'algorithme 6 et après l'algorithme 5, nous tiendrons une matrice de confusion par utilisateur encore moins creuse.

Fouille de données

Dans cette sous-section, nous allons appliquer un algorithme de regroupement nommé *K-means* à l'ensemble des matrices de confusions obtenues dans la sous-section précédente afin de trouver des profils encore plus réalistes avec des données réelles.

Algorithm 6 Trouver les relations transitives.

Entrée: Ensemble des publications vérifiées \mathbf{P}^v

Sortie: Le graphe avec relations transitives $\mathcal{G}^t = (V^t, E^t)$

```

1: while  $\exists$  au moins une relation transitive do
2:   for each  $C_v^{P_i^v} \in \{C_v^{P_1^v}, C_v^{P_2^v}, \dots, C_v^{P_V^v}\}$  do
3:     for each  $C_j^b \in \{\text{Ensemble des noeuds adjacents de } C_v^{P_i^v}\}$  do
4:       /* Verifier si  $C_j^b$  a des confusion (i.e des noeuds adjacents). */
5:        $C_v^{P_j^v} := \text{verifié\_vraie\_espèce}(C_j^b)$ 
6:       for each  $E_l \in \{\text{Ensemble des arêtes de } C_v^{P_j^v}\}$  do
7:         if  $E_l.C_{adj} \notin \{\text{Ensemble des noeuds adjacents de } C_v^{P_i^v}\}$  then
8:           /*  $(C^{P_i^v} \mathcal{R} C^{P_j^v}) \wedge (C^{P_j^v} \mathcal{R} E_l.C_{adj}) \Rightarrow (C^{P_i^v} \mathcal{R} E_l.C_{adj})$  : transitivité. */
9:            $\text{ajouter\_arête\_transitivité}(C^{P_i^v}, E_l.C_{adj}, E_l.W_{adj})$ 
10:        else
11:          /* S'il existe la relation transitive,
12:            le mettre à jour avec le minimum valeur de confusion. */
13:           $\text{mettre\_à\_jour\_arête}(C^{P_i^v}, E_l.C_{adj}, \min(E_l.W_{adj}, W_{(C^{P_i^v}, C_{adj})}))$ 
14:        end if
15:      end for
16:    end for
17:  end for
18: end while
19: return  $\mathcal{G}^t = (V^t, E^t)$ 

```

L'implementation de l'algorithme *K-means* a été prise de [13, p. 67], avec une configuration de 3 clusters et 197 utilisateurs, et les résultats obtenus ont été transformés dans un espace à deux dimensions en utilisant l'algorithme d'Analyse en composantes principales (ACP) ([13, p. 62]). Voici ci-dessous les résultats.

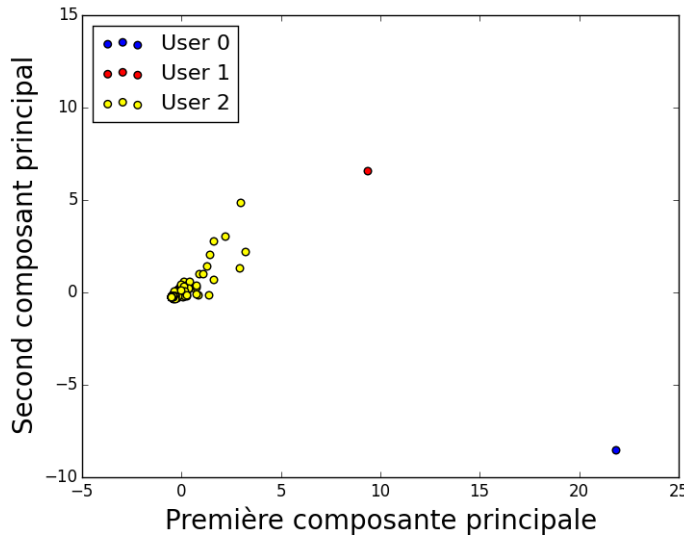


FIG. 3.15: Visualisation des utilisateurs de Tela-Botanica regroupés.

Résultats et Conclusion

Dans ce chapitre, nous présenterons les résultats expérimentaux de simulation des solutions d'évaluation de qualité abordée dans l'état de l'art avec les profils des utilisateurs modélisés dans le chapitre précédent.

Nous allons découper ce chapitre en trois sections : Validation des profils, Évaluations des solutions *Crowdsourcing*, et finalement, Conclusion et travaux futurs.

4.1 Validation des profils

Pour valider les comportements différents de nos profils modélisés dans le chapitre 3, nous allons d'abord créer une simulation avec une configuration équilibrée du nombre des utilisateurs par profil. Voici le tableau de configuration :

Profils	Nb. Users	Nb. Classes	Nb. Tâches
Experts	[10, 20, ..., 1000]	150	50
Amateurs	[10, 20, ..., 1000]	150	50
Novices	[10, 20, ..., 1000]	150	50
Spammers	[10, 20, ..., 1000]	150	50

TAB. 4.1: Configuration des profils utilisateurs

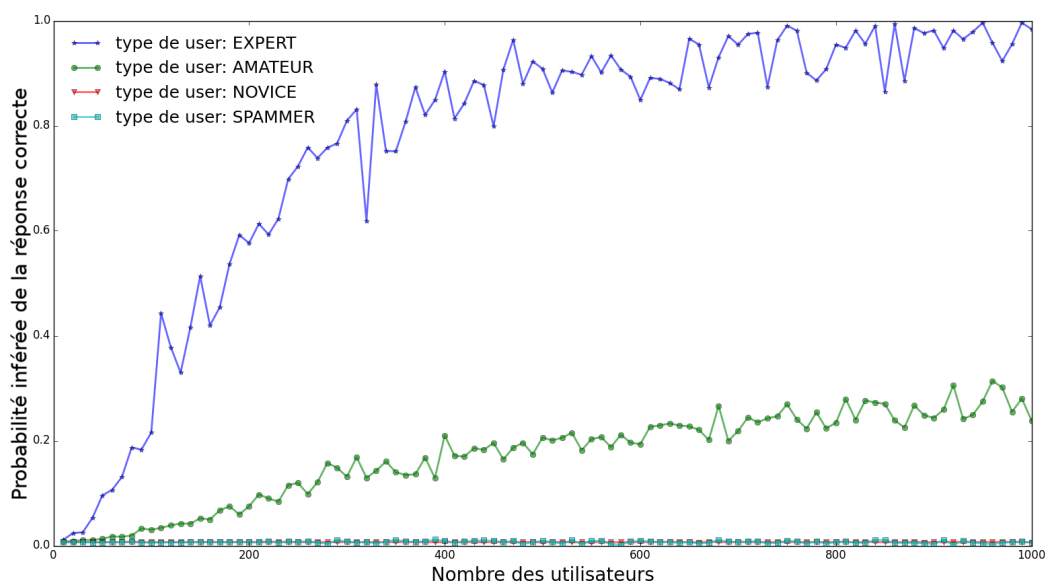


FIG. 4.1: Méthode d'inférence de Dawid et Skene (I)

La figure 4.1 illustre les résultats de la simulation afin d'évaluer la qualité des réponses des utilisateurs par profil conformément à la configuration du tableau 4.1 et la méthode d'inférence de Dawid and Skene [7]. Nous pouvons d'abord remarquer une différence importante de compétences entre les profils Expert/Amateur et les profils Novice/Spammeur. Le profil expert nous montre aussi une tendance à mieux répondre les tâches que le profil amateur.

Quant aux profils novice et spammeur illustre sur la figure 4.2, nous pouvons vérifier le comportement des compétences fiables qui n'arrivent même pas à 1% de qualité de réponse.

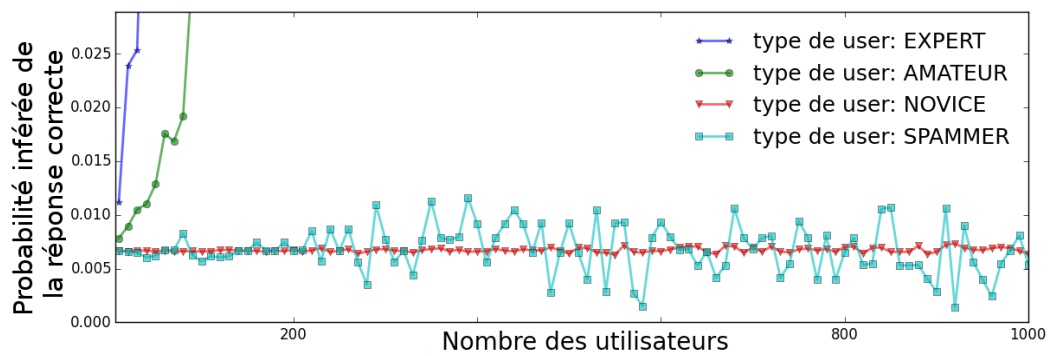


FIG. 4.2: Méthode d'inférence de Dawid et Skene (II)

Nous avons aussi utiliser l'algorithme *d'Analyse en composantes principales (ACP)* ([13, p. 62]) dans le but de visualiser les profils d'utilisateurs bien définis et regroupés séparément sur la figure 4.3.

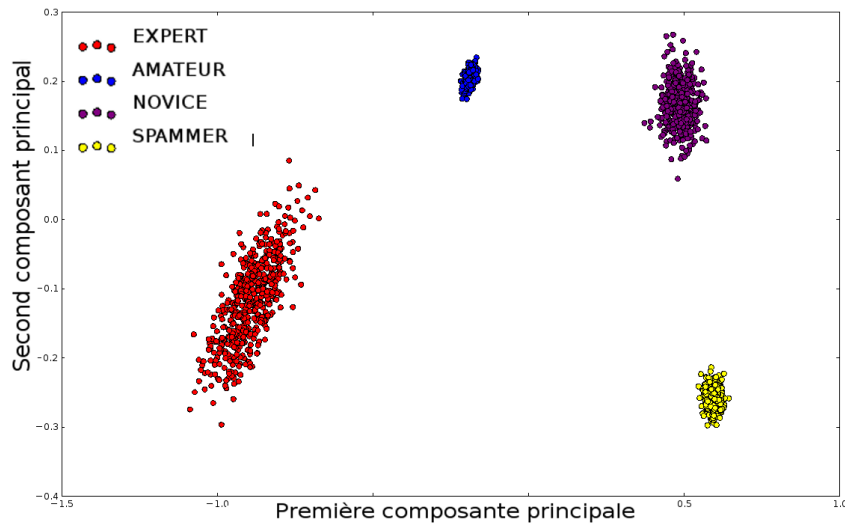


FIG. 4.3: Visualisation des utilisateurs par profil

4.2 Evaluation des solutions Crowdsourcing

Dans cette section, nous allons évaluer deux méthodes d'inférence utilisées sur les plateformes de *Crowdsourcing* et déjà abordées dans l'état de l'art ; *Majority voting* et *Méthode d'inférence de Dawid and Skene*. Le nombre d'utilisateur à cette occasion sera dans l'intervalle de $[10, 20, \dots, 1000]$ utilisateurs et répartis en différents pourcentage par profil selon le tableau de configuration ci-dessous.

Profils	Nb. Users	Nb. Classes	Nb. Tâches
Experts	20%	150	50
Amateurs	30%	150	50
Novices	30%	150	50
Spammers	20%	150	50

TAB. 4.2: Configuration des profils utilisateurs

La figure 4.4 illustre les résultats de simulation des deux algorithmes d'inférence (i.e. Majority voting et Méthode d'inférence Dawid and Skene). Nous pouvons remarquer que la méthode d'inférence de Dawid and Skene a une tendance à obtenir de meilleurs résultats que le Majority voting, lorsque nous augmentons les nombres d'utilisateurs. Ces résultats se vérifient, d'après la littérature, que les profils modélisés respectent le fait que la méthode d'inférence obtient de meilleurs résultats que le Majority voting.

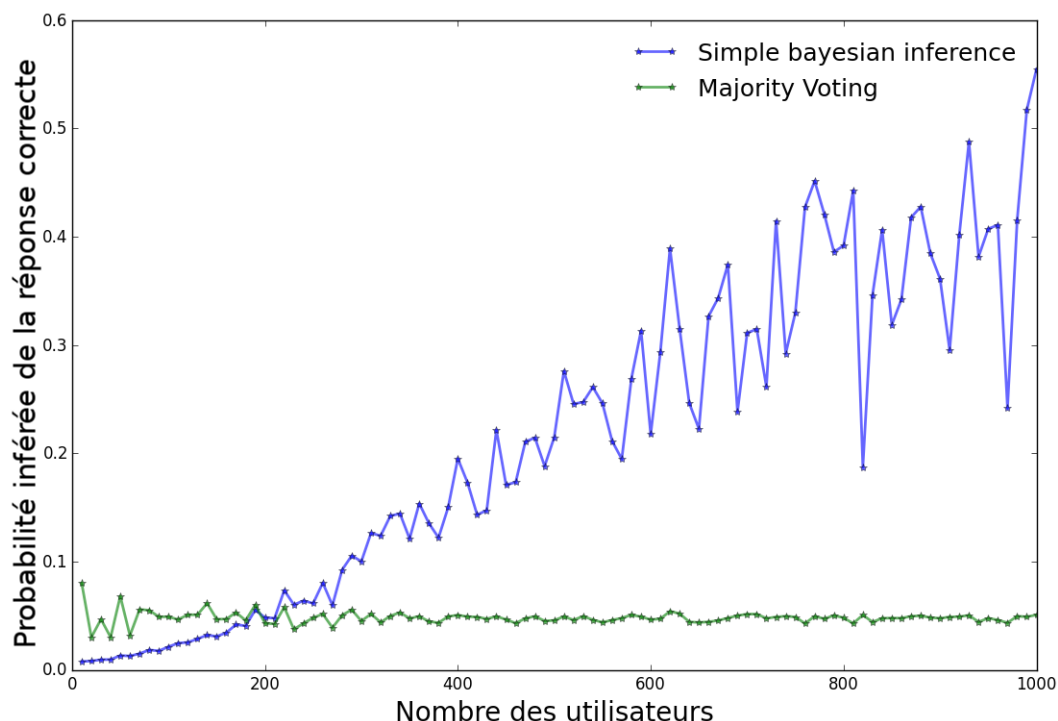


FIG. 4.4: Évaluation des solutions de *Crowdsourcing*

4.3 Conclusion et Overtures

Bilan du travail

Ainsi, ce travail a été plutôt difficile car les différentes techniques et concepts que j'ai appliqués, n'ont pas été évidents à assimiler au premier abord. Il m'a fallu en certain temps de recherche pour bien m'imprégner du sujet. Ainsi donc, j'ai réussi à respecter au maximum les consignes demandées pour ce projet malgré la contrainte du temps.

Après cette première étape importante, je suis parvenu à appliquer la matrice de confusion en tant que connaissance d'un utilisateur, afin de simuler le problème de classification.

Ensuite, j'ai pris du temps à analyser les possibles lois de probabilités discrètes dans le but d'en proposer quelques-unes. Puis, j'ai commencé à réviser les méthodes de simulation de monte-carlo afin d'en appliquer une sur les lois de probabilité. À la fin de la modélisation, j'ai exploré d'autres horizons afin de proposer des profils encore plus réalistes en me servant des données réelles de Tela-Botanica.

Enfin, j'ai mis en place des simulations avec les profils modélisés et les algorithmes d'inférences implémentés.

Bilan personnel

Ce travail m'a permis d'approfondir encore plus mes connaissances en statistique, ou plus précisément en apprentissage supervisé au niveau de la classification sans ou avec un ensemble de données juste. J'ai aussi pu apprendre quelques concepts sur le domaine de la simulation de monte-carlo, comme le fait qu'une distribution de probabilité peut se transformer à partir d'une distribution uniforme.

Overtures

La manipulation de certaines propriétés de la loi de probabilité telle que l'écart type, auraient été intéressante à l'analyser afin de trouver d'autres profils.

L'implémentation des méthodes d'inférences, existant dans la littérature, de classification non vues représenteraient un plus à la validation de nos profils modélisés.

L'analyse des confusions du site web Tela-Botanica me semble être très prometteuse dans le but de trouver des matrices de confusion réalistes.

Enfin et le plus intéressant serait de modéliser l'apprentissage d'un utilisateur au fur et à mesure qu'il répond aux tâches, afin d'avoir les compétences des utilisateurs qui évoluent au cours du temps.

Bibliographie

- [1] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, pages 1301–1318, New York, NY, USA, 2013. ACM.
- [2] Jeff Howe. *Crowdsourcing : Why the Power of the Crowd Is Driving the Future of Business*. Itzy, 2008.
- [3] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning : Training object detectors with crawled data and crowds. *Int. J. Comput. Vision*, 108(1-2) :97–114, May 2014.
- [4] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *J. Mach. Learn. Res.*, 11 :1297–1322, August 2010.
- [5] Padhraic Smyth, Usama M. Fayyad, Michael C. Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *NIPS*, pages 1085–1092. MIT Press, 1994.
- [6] Vikas C. Raykar and Shipeng Yu. Ranking annotators for crowdsourced labeling tasks. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 1809–1817, 2011.
- [7] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1) :20–28, 1979.
- [8] Yukino Baba and Hisashi Kashima. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13*, pages 554–562, New York, NY, USA, 2013. ACM.
- [9] W. A. Shewart. *Economic control of Quality of Manufactured Product*. Van Nostrand Reinhold Co., New York, 1931.
- [10] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10*, pages 64–67, New York, NY, USA, 2010. ACM.

- [11] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 614–622, New York, NY, USA, 2008. ACM.
- [12] George Casella Christian P. Robert. *Méthodes de Monte-Carlo avec R*. Springer, 2011.
- [13] Guillermo Moncecchi Raul Garreta. *Learning scikit-learn : Machine Learning in Python*. Packt, 2013.