Extraction d'un vocabulaire de surprise par mélange de filtrage collaboratif et d'analyse de sentiments

Mickaël Poussevin* — **Vincent Guigue*** — **Patrick Gallinari***

* *UPMC, PRES Sorbonne-Universités Lab. d'Informatique de Paris 6, UMR 7606, CNRS – 4 Place Jussieu, Paris, France – Mail: Prénom.Nom@lip6.fr*

RÉSUMÉ. *L'informatique subit actuellement une mutation profonde: les améliorations matérielles et les grandes quantités de données disponibles fournissent un terrain fertile à la recherche en apprentissage automatique. Dans ce contexte, le principal défi est de tenir compte des préférences des utilisateurs pour proposer un accès personnalisé à l'information. Les systèmes de recommandation créent des profils utilisateurs et objets en utilisant les revues utilisateurs, et ces profils reflètent les préférences des utilisateurs et les caractéristiques des objets. Nous proposons ici une analyse par combinaison de systèmes de recommandation et classifieurs de polarité qui met en évidence le vocabulaire de la surprise. En effet, la recommandation analyse le passé et anticipe les attentes d'un utilisateur tandis que le classifieur de polarité prend en entrée une revue déjà écrite par l'utilisateur: nous montrons que l'écart entre l'expérience attendue et le retour réel sur un objet permet de construire un lexique de la surprise.*

ABSTRACT. *Computer science is undergoing a profound transformation: gains in computing power and the availability of large datasets provide a fertile ground for machine learning research. One challenge in this context is to take account of user preferences to provide personalized access to information.Recommender systems create user and item profiles using past user reviews, profiles that reflect user preferences and item characteristics. We propose here an analysis of a combination of recommendation systems and polarity classifiers which highlights a vocabulary of surprise: words indicating a gap between the expectations of users and their actual experience with an item. Indeed, recommender systems analyze past ratings to predict user preferences while sentiment classifiers use existing reviews as input: we show that the difference between expectations and observations enables the construction of a surprise lexicon.*

MOTS-CLÉS : *Filtrage collaboratif, Analyse de sentiments, Surprise.*

KEYWORDS: *Collaborative filtering, Sentiment analysis, Surprise.*

1. Introduction

L'informatique et les systèmes d'information subissent aujourd'hui une mutation, médiatiquement nommée *Big Data*, qui repose sur la croissance conjointe des capacités de calculs et de stockage des infrastructures informatiques ainsi que du nombre de sources de données numériques (traces numériques, géolocalisation, capteurs en tous genres, textes...). Cette évolution fournit un terreau propice au développement rapide de la recherche en apprentissage automatique. D'une part, les capacités de calculs permettent l'apprentissage de modèles complexes mettant en jeu un grand nombre de paramètres libres et gérant des données hétérogènes. D'autre part, les données disponibles et leur masse importante ouvrent de nouvelles applications et de nombreuses perspectives, notamment dans l'extraction de profils utilisateur et la personnalisation de l'accès à l'information. Les modèles récents comme la factorisation matricielle et les réseaux de neurones profonds sont très complexes (en nombre de paramètres), ils doivent être couplés avec des techniques de contrôle de la complexité ou de sélection de modèles pour éviter le sur-apprentissage. Mais de manière générale, plus un modèle est complexe, plus il requiert de données pour l'entraînement, car le sur-apprentissage est alors mécaniquement limité, l'échantillon observé étant plus exhaustif. La puissance de calcul permet donc de traiter le grand nombre de paramètres ainsi que les données nécessaires à leurs apprentissages. Les bases de données textuelles que nous utilisons dans cet article, en particulier issues de (McAuley et Leskovec, 2013a ; McAuley et Leskovec, 2013b ; Jindal et Liu, 2008), sont de l'ordre du million d'exemples.

La première opportunité offerte par ces nouvelles approches est d'étendre les analyses et modèles traditionnels. En particulier, le développement de l'apprentissage de représentations propose d'utiliser des algorithmes d'apprentissage automatique pour remplacer la tâche fastidieuse de création de caractéristiques pertinentes. En image par exemple, des procédures automatisées d'extraction de descripteurs comme SIFT (Vedaldi, 2007) ont été proposées. Aujourd'hui, les réseaux de convolution (Lawrence *et al.*, 1997) peuvent tirer profit des larges quantités d'images disponibles pour extraire automatiquement des descripteurs qui se révèlent plus efficaces (Krizhevsky *et al.*, 2012) que les classiques SIFT. Le même phénomène se retrouve dans d'autres tâches comme la reconnaissance d'écriture manuscrite ou de parole, avec des réseaux récurrents (Graves, 2012 ; Dorffner, 1996). A moyen terme, le but est de proposer des chaînes de traitement totalement automatisées, de l'acquisition de données au résultat final.

La seconde opportunité est d'enrichir l'application visée en incorporant de nouveaux axes d'analyses. En particulier, la modélisation des comportements et préférences des utilisateurs (Fürnkranz et Hüllermeier, 2010) permet de mieux répondre à plusieurs tâches comme la recommandation (Kantor *et al.*, 2011), les moteurs de recherche personnalisés (Teevan *et al.*, 2005), le ciblage publicitaire (Chakraborty *et al.*, 2010), la classification de sentiments (Poussevin *et al.*, 2014) ou même les sites de rencontre (Diaz *et al.*, 2010).

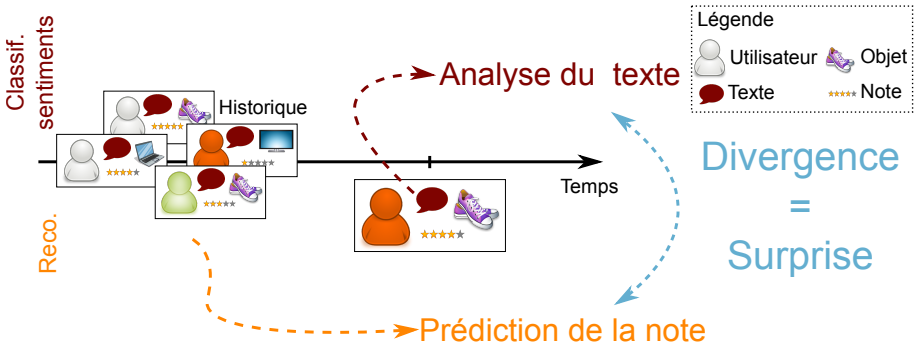

Figure 1. *Notre contribution est d'utiliser un système de filtrage collaboratif, exploitant l'historique des notes, pour contextualiser un classifieur de polarité, utilisant seulement les textes. L'utilisation des profils utilisateurs et objets extraits par le premiers et les poids donnés aux mots par le second permettent d'extraire un vocabulaire de surprise des commentaires. Coupler ces deux informations permet aussi une amélioration des performances en classification de polarité.*

Les travaux réalisés dans cet article se placent à l'intersection de l'apprentissage de représentation, de l'extraction de profils utilisateurs et de l'analyse de sentiments (cf figure 1). Nous commençons par développer un système de recommandation, dans le cadre défini par le filtrage collaboratif, pour apprendre des représentations pertinentes des utilisateurs. Nous utilisons ensuite ce système pour contextualiser les classifieurs de sentiments à la manière de (Poussevin *et al.*, 2014). Mais ce qui nous intéresse en particulier concerne les différences entre le comportement prédit -extrait des notations passées et issu du système de recommandation- et la revue effectivement produite par l'utilisateur -analysée par le classifieur de sentiments-. Lorsque les deux systèmes divergent, cela signifie que la note attendue est différente de la note réelle : nous faisons alors l'hypothèse que l'utilisateur a été surpris (positivement ou négativement) et que la revue écrite doit contenir un vocabulaire spécifique. Cette étude s'attache à caractériser les différences de notation entre les systèmes, elle présente ensuite un système de correction d'erreur améliorant les performances en classification de sentiments et un module d'extraction du vocabulaire de surprise.

Après avoir récapitulé l'état de l'art en section 2, nous détaillons les modèles utilisés en section 3 et présentons nos expériences en section 4.

2. État de l'art

Nos travaux se placent à l'intersection du filtrage collaboratif, une des façons d'aborder la recommandation d'objets, et de classification de polarité. Nous détaillons les états de l'art propres à chacun de ces domaines.

2.1. *Recommandation de produits*

Les systèmes de recommandation sont l'archétype des modèles de personnalisation de l'accès à l'information. Ils sont centrés sur les utilisateurs afin de pouvoir apprendre les préférences individuelles de chacun et sont aujourd'hui omniprésents sur Internet et beaucoup les utilisent quotidiennement parfois même sans s'en rendre compte. Dans le contexte des moteurs de recherche, ils permettent de trier les résultats d'une requête en fonction des goûts et habitudes de chacun (Teevan *et al.*, 2005). Dans le contexte des sites de rencontre, ils mettent en relation des individus dont les profils correspondent (McFee et Lanckriet, 2010 ; Diaz *et al.*, 2010). Dans le contexte des sites marchands, comme Amazon, ils proposent aux visiteurs des produits en fonctions de l'historique de leurs visites (Schafer *et al.*, 1999). Nous nous plaçons dans ce dernier contexte, celui de la recommandation de produits, qui est le plus étudié dans la littérature, du fait de la disponibilité des revues utilisateurs sur le web 2.0. Les revues constituent des traces polarisées, contenant des avis de personnes sur les biens qu'elles possèdent, ou les services qu'elles consomment, et se composent de quatre éléments :

Utilisateur La plupart du temps, les sites forcent leurs utilisateurs à se connecter pour écrire des critiques, par conséquent, elles sont associées à un utilisateur et il est possible de suivre les avis de chaque utilisateur au fil du temps. Ainsi, il est possible de centrer l'analyse sur les utilisateurs, d'extraire leurs goûts et habitudes, de suivre leurs évolutions au cours du temps.

Objet Il peut s'agir d'un produit, au sens large sur Amazon qui propose des biens allant du livre à l'électro-ménager, ou restreints à une certaine catégorie comme les bières sur Ratebeer, ou encore d'un service comme les restaurants sur Yelp !. Ils font partie du catalogue du site et l'enjeu de la tâche de recommandation est de proposer des items pertinents à chaque utilisateur.

Note Elle résume l'appréciation globale d'un produit par un utilisateur. Elle dépend à la fois de l'utilisateur (objectif, partial, sévère ...) et du produit (qualité, popularité, marketing...). Elle est traditionnellement représentée par un entier de 1 à 5.

Texte Il décrit, avec plus de détails que la note, l'appréciation de l'objet par l'utilisateur. Notre hypothèse est que le texte, qui est un moyen riche de communiquer son ressenti, permet de raffiner les profils utilisateurs extraits par les systèmes de recommandation et de proposer des produits plus pertinents encore. La manière d'écrire d'un utilisateur est un élément de son profil.

Le problème de la recommandation d'items est défini de deux manières : il s'agit soit d'une prédiction de notes, soit d'une génération de listes ordonnées. La première estime la note que donnerait un utilisateur à un objet (modélisant ainsi son intérêt pour ce dernier), la qualité de l'estimation est mesurée par la proximité entre la note prédite et la note réelle (e.g. erreur des moindres carrés). C'est le paradigme utilisé par le moteur GroupLens (Resnick *et al.*, 1994) ou dans le cadre du challenge Netflix

(Bennett et Lanning, 2007 ; Koren, 2008) et c'est aussi l'approche que nous avons retenue. Pour l'ordonnancement, la recommandation est une liste d'items ordonnée par ordre d'intérêt pour l'utilisateur et seul les K items les plus pertinents sont renvoyés vers l'utilisateur (Breese *et al.*, 1998 ; McLaughlin et Herlocker, 2004). Il s'agit de l'implémentation utilisée sur les sites de e-commerce, qui recommandent un nombre fixe d'objets à chaque fois et le modèle est alors évalué à l'aide de mesures TopK (rappel et précision) ou d'aire sous la courbe ROC.

Quatre principales approches de recommandation sont généralement distinguées dans la littérature : *content based, knowledge based, collaborative filtering* ainsi que les modèles hybrides. Plusieurs possibilités d'hybridation entre les différents systèmes sont présentés dans (Burke, 2002). Les méthodes s'appuyant sur le contenu (*content based*) utilisent l'historique utilisateur et des descripteurs sur les objets (Pazzani et Billsus, 2007), comme des mots-clés (Adomavicius et Tuzhilin, 2005). Ces méthodes permettent de prendre plus facilement en compte les méta-données mais connaissent cependant des limites : il est impossible de proposer une recommandation personnalisée à un nouvel utilisateur (*cold start*) et ces modèles sont intrinsèquement très dépendants de la qualité des descripteurs fournis. Les méthodes dites *knowledge-based* utilisent généralement un ensemble de connaissances sur l'objet (restaurant, voiture, produit financier) à recommander (Burke, 2000). Les méthodes de filtrage collaboratif exploitent quant à elles les similarités de notations entre utilisateurs : les personnes ayant des notations en commun sont proches et leurs profils peuvent alors se compléter l'un l'autre. Les méthodes de filtrage collaboratif et basées sur le contenu souffrent toutes les deux du problème de démarrage à froid (*cold start*) : lorsque peu d'historique (ou pas du tout) est disponible, il leur est difficile (voire impossible pour les *content based*) de recommander des objets pertinents (Schafer *et al.*, 1999). En utilisant un ensemble de procédés établis à l'avance, les systèmes *konwledge-based* permettent de proposer des recommandations dans de tels cas (Burke, 2000), mais nous ne traiterons pas ce problème de démarrage à froid dans nos travaux, quoique cela puisse être une extension intéressante.

Les objets sur lesquels nous travaillons sont divers et nous ne disposons pas toujours de descriptifs, par contre, nous avons les avis de nombreux utilisateurs. Notre travail se place donc dans le cadre du filtrage collaboratif qui s'appuie uniquement sur l'historique des revues utilisateurs. Cette approche a été proposée pour la première fois dans le cadre de la plate-forme GroupLens (Resnick *et al.*, 1994). Le filtrage collaboratif s'est ensuite progressivement divisé en deux sous-catégories : les approches à mémoire (*memory based*) et les approches à modèles (*model based*). La première, s'appuie sur des algorithmes de k-plus proches voisins (Koren, 2008), la seconde, sur des modèles de prédictions appris sur des données d'apprentissage (Bennett et Lanning, 2007). Les algorithmes les plus utilisés pour cela sont la factorisation matricielle et les RBM (*Restricted Boltzman Machine*) (Koren, 2008 ; Koren et Bell, 2011). La factorisation matricielle extrait simultanément des profils utilisateurs et objets (Koren *et al.*, 2009) et permet l'ajout de biais utilisateurs, objet et globaux pour prendre en compte les caractéristiques individuelles de chaque utilisateur, objet et de la base de données et est la méthode que nous avons choisie. Nos contributions proposent d'ajou-

ter, aux profils extraits sur les notes, des profils extraits sur les textes afin de les compléter.

2.2. *Analyse de sentiments*

La tâche de classification de polarité consiste à décider si les textes expriment un avis positif ou négatif, en évaluant la polarité des mots qui le composent. Comme décrit dans (Pang et Lee, 2008), les textes des revues utilisateurs sont une ressource importante pour cette tâche : ils fournissent d'énormes quantités de textes polarisés et étiquetés (grâce à la note donnée par l'utilisateur). Ces données sont exploitées pour concevoir des modèles de classification capables de généraliser à d'autres types de documents. L'analyse des textes peut bénéficier de l'information contextuelle fournie par l'analyse des préférences des utilisateurs et de leur style d'écriture (Poussevin *et al.*, 2014). Par exemple, l'utilisation de l'ironie est très personnelle, dépend du style de l'auteur et peut changer la polarité d'un texte : c'est un cas très difficile en analyse de texte, mais beaucoup plus abordable en exploitant les informations de profils.

Les notes des revues utilisateurs sont généralement des notes de 1 à 5 (*star rating*). Nous allons maintenant décrire la méthode traditionnelle de conversion de ces notes en étiquettes binaires, présentée également dans (Pang et Lee, 2008) : les notes 1 et 2 sont converties en exemples négatifs $y = -1$; les notes 4 et 5 sont converties en exemples positifs $y = +1$; les notes égales à 3 sont ignorées car ambiguës : leurs significations changent en fonction du domaine et des utilisateurs. La quantification des aspects positifs/négatifs est donc prise en compte en recommandation mais pas en classification. (Pang et Lee, 2008) référence plusieurs études montrant que la distinction entre les textes notés 1 ou 2 d'une part et 4 ou 5 d'autre part est particulièrement ardue : les modèles actuels ne sont pas significativement meilleurs que l'aléatoire sur cette tâche.

La difficulté principale vient du glissement dans le vocabulaire utilisés dans différents domaines : les marqueurs de polarité ne sont pas les mêmes pour de la musique, des restaurants ou des ordinateurs. La construction d'un modèle générique efficace est encore un problème ouvert. La problématique du transfert en apprentissage (Blitzer *et al.*, 2007) étudie comment transférer des connaissances apprises dans un certain contexte à d'autres situations et propose des pistes pour la classification de sentiment. Par exemple, il est possible d'utiliser des réseaux neuronaux profonds pour généraliser à plusieurs domaines en utilisant de grandes quantités de textes non-étiquetés (Glorot *et al.*, 2011). Également, dans le récent (Le et Mikolov, 2014), les auteurs de word2vec (Mikolov *et al.*, 2013) étendent leur modèle pour permettre la classification de polarité. Word2vec utilise de grandes quantités de textes, sans étiquettes, pour apprendre des représentations des mots dans un espace où la distance est représentative de la différence syntaxique. Leur extension, Paragraph Vector, (Le et Mikolov, 2014), propose d'apprendre en plus des représentations des documents sur lesquelles peuvent opérer des classifieurs. Toutes ces approches visent à créer un modèle générique pour classer tous les textes mais ne prennent pas en compte l'historique des utilisateurs. Nous

proposons ici de contextualiser un modèle de classification en utilisant un système de recommandation.

2.3. *Approches mixtes*

Les approches mixtes, exploitant l'historique de notation et les données textuelles sont rares. Les premiers travaux concernent l'amélioration des systèmes de recommandation en extrayant des sous-domaines d'intérêt non explicite (Ganu *et al.*, 2009). Un restaurant, par exemple, reçoit une note globale d'un utilisateur mais elle correspond à la fois au cadre (la salle, la table), au service, à la qualité des plats... L'analyse du texte permet d'identifier ces catégories et de raffiner la notation. L'extension à la recommandation est directe : les utilisateurs qui partagent les mêmes sous-domaines d'intérêt sont les plus indiqués pour construire une recommandation fiable. Ce premier modèle était essentiellement construit à la main, (McAuley et Leskovec, 2013b) ont ensuite proposé une extension basée sur une intégration de l'algorithme de clustering (LDA) dans le processus de recommandation. Cette formulation unifiée a donné de bonne performance. Encore plus récemment, (Poussevin *et al.*, 2014) ont montré l'intérêt d'utiliser le texte brut des revues dans le processus de recommandation pour saisir le style des différents utilisateurs et l'intégrer dans la construction des profils. Le travail présenté dans notre article s'appuie sur ces techniques hybrides mais la finalité est différente : nous nous intéressons exclusivement à la classification de polarité et aux effets de surprise, non à la recommandation.

3. Modèles

Nous décrivons ici un modèle qui combine un système de recommandation et un modèle de classification de sentiments. Nos données sont les revues utilisateurs, des quadruplets $(u, i, r_{ui}, \mathbf{d}_{ui})$ composés d'un utilisateur u, d'un objet i, de la note donnée par u à i, r_{ui}, et du texte commentant l'opinion de u sur l'objet i, $\mathbf{d}_{ui}$. Nous disposons de m_R revues correspondant à m_I items et m_U utilisateurs. Le système de recommandation apprend des profils pour les items et les utilisateurs qui permettent l'estimation des notes inconnues. Nous exploitons l'écart entre cette estimation et la note réelle pour entraîner le modèle de surprise. Il s'agit d'un régresseur sur l'écart entre les notes réelles et prédites opérant sur les textes des revues utilisateurs. Ce sont les poids saillants de ce régresseur qui permettent d'extraire un vocabulaire de surprise. Du point de vue de l'implémentation ce régresseur s'apparente à un classifieur de sentiments.

3.1. *Filtrage collaboratif*

Pour la recommandation, nous utilisons la factorisation de la matrice des notes $\mathbf{R} \in \mathbb{R}^{m_U \times m_I}$. Cette dernière combine plusieurs termes : le produits des profils latents

des utilisateurs et objets ainsi que des biais. Pour un utilisateur u et un objet i donnés, la prédiction du modèle $g(u,i)$ est calculée comme suit :

$$g(u,i) = \mu + b_u + b_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle, \quad \mathbf{p}_u, \mathbf{q}_i \in \mathbb{R}^k \qquad [1]$$

où k est un hyper-paramètre désignant la dimension de l'espace latent. La factorisation matricielle s'appuie sur trois biais. Le premier, μ, est le biais global. C'est une constante égale à la moyenne des notes sur l'ensemble d'entraînement. Les deux autres, b_u et b_i sont respectivement le biais de l'utilisateur u et de l'objet i. Ils prennent en compte les habitudes de notations de chacun dans la base et sont modifiés lors de l'apprentissage du modèle. Le dernier terme de [1] est le produit scalaire entre le profil latent de l'utilisateur u, et celui de l'objet i. Ces profils sont les lignes des matrices $\mathbf{P}_U \in \mathbb{R}^{m_U \times k}$ pour les utilisateurs et $\mathbf{Q}_I \in \mathbb{R}^{m_I \times k}$ pour les objets.

L'entraînement du modèle $\theta = \{\mathbf{P}_u, \mathbf{Q}_i, b_u, b_i\}$ correspond à l'optimisation du critère des moindres carrés régularisé pour prédire au mieux les m_R notes observées :

$$\theta^\star = \underset{\mathbf{P}_u, \mathbf{Q}_i, b_u, b_i}{\operatorname{argmin}} \frac{1}{m_R} \sum_{(u,i,r_{ui})} (g(u,i) - r_{ui})^2 + \lambda_R(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) \qquad [2]$$

La régularisation, dont l'influence est réglée par le paramètre λ_R, est nécessaire pour prévenir le sur-apprentissage dans ce cadre où le nombre de paramètres est très important. Régulariser les termes de biais est aussi essentiel, ces termes récupérant une grande partie de l'énergie du signal (Koren *et al.*, 2009). Nous avons choisi de résoudre le problème d'optimisation défini par l'équation 2 en utilisant une descente de gradient stochastique. Nous utilisons l'implémentation proposée par (Low *et al.*, 2010).

3.2. *Régression de polarité*

Comme préconisé dans (Pang et Lee, 2008), nous avons retenu les machines à vecteur de support linéaires opérant sur des sacs de mots présentiels pour la tâche de classification de polarité. Néanmoins, nous proposons ici une approche originale en entraînant ce modèle non pas comme un classifieur mais comme un régresseur sur l'écart entre la note réelle d'un texte et la note prédite par notre système de recommandation. Par exemple, considérons la revue dont le document est $\mathbf{d}_{ui}$ ="J'ai adoré les trois premiers mais celui me déçoit" et la note $r_{ui} = 2$. En utilisant l'information passée, il est probable que, comme l'utilisateur a apprécié les premiers films de cette série, la note prédite par notre système de recommandation soit élevée, mettons $g(u,i) = 4$. Nous désirons que notre modèle de régression de sentiment utilise la présence de mots comme "mais" ou "deçoit" pour corriger la note. Sa prédiction doit alors être $h(\mathbf{d}_{ui}) = -2$. Ainsi, notre régresseur pourra mettre en évidence un vocabulaire de surprise.

Vues la dimension des données (sacs de mots) et la quantité de textes, nous optimisons directement un modèle linéaire h avec un vecteur de poids $\mathbf{w}$ et un biais b. Par

rapport à un SVM classique, nous nous distinguons en optimisant un coût au sens des moindres carrés.

$$\mathbf{w}^\star b^\star = \underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{m_V} \sum_{(u,i,r_{ui},\mathbf{d}_{ui})} (h(\mathbf{d}_{ui}) - (r_{ui} - g(u,i)))^2 + \lambda_C \|\mathbf{w}\|^2 \quad [3]$$

$$h(\mathbf{d}_{ui}) = \sum_{j \in Vocab.} d_{ui}^{(j)} w_j + b,\ d_{ui}^{(j)} \in \{0,1\} \text{ présence du mot } j \text{ dans } \mathbf{d}_{ui} \quad [4]$$

Attention, ce modèle n'est pas appris sur l'ensemble d'entraînement mais sur les m_V revues de l'ensemble de validation. En effet, les performances du système de recommandation sur le premier ensemble sont très bonnes (avec le sur-apprentissage) et les prédictions souvent correctes. Sur cet ensemble, l'écart entre note prédite et note réelle n'est pas représentatif et il ne reste rien à apprendre.

4. Expériences

Cette section présente les résultats de nos expériences. Nous décrivons dans un premier temps les données et le pré-traitement de ces dernières. Nous comparerons ensuite nos modèles en tant que modèles de classification de polarité en utilisant le taux d'erreur en classification. Pour finir, nous étudierons les poids du régresseur entraîné en sortie du système de recommandation.

4.1. *Données*

Nos ensembles de données sont des collections de revues utilisateurs composées d'un texte (souvent court) et d'une note, issus des sites qui suivent :

Yelp Ce jeu de données est mis à disposition de la communauté scientifique par Yelp ! et contient un million de revues.

RateBeer RateBeer.com est un site internet proposant aux amateurs de bières de partager leur avis sur les multiples bières du monde. 2.9M de revues ont été extraites par (McAuley et Leskovec, 2013a ; McAuley et Leskovec, 2013b).

Movies Il s'agit d'une collection de 7.9M de critiques de films sur Amazon.com, également récupérées par (McAuley et Leskovec, 2013a ; McAuley et Leskovec, 2013b).

Amazon Un ensemble de 5.8M revues sur différents types de produits vendus par Amazon.com collectées par (Jindal et Liu, 2008).

Nous proposons ici l'extraction d'un vocabulaire de surprise composé des mots utilisés par les utilisateurs pour exprimer une différence entre ce qu'ils attendaient de l'objet et leur expérience réelle. La première étape de notre pré-traitement est de

filtrer les utilisateurs et objets ayant moins de 10 revues. Nous séparons alors les revues restantes en trois ensembles : entraînement, validation et test. Nous nous assurons que chaque utilisateur à 70% de ces revues dans l'ensemble d'entraînement et 15% en validation et test. Les caractéristiques des ensembles obtenus sont présentées dans la table 1.

Nom	Utilisateurs	Objets	Entraînement	Validation	Test
Yelp	38665	25384	411735	88229	88229
Amazon	135599	186009	1260457	270098	270098
RateBeer	11732	46935	931226	199549	199549
Movies	258356	129512	4017660	860927	860928

Tableau 1. *Description des ensembles de données utilisés dans nos expériences. Chaque ligne reporte les mesures pour un ensemble particulier. Les colonnes sont, de gauche à droite, le nombre total d'utilisateurs puis d'objets et le nombre de revues pour les ensembles d'entraînement, de validation et de test.*

4.2. *Représentation des textes*

Nous utilisons une procédure simple pour la représentation du texte, qui est connue depuis (Pang et Lee, 2008) pour son efficacité. Pour un jeu de données, nous extrayons un dictionnaire sur l'ensemble des textes d'entraînement en considérons les mots qui apparaissent dans plus de 10 textes différents pour éviter les mots trop spécifiques, les fautes d'orthographes et les erreurs dans la segmentation des chaînes de caractères. Chaque texte de revue utilisateur, est alors représenté comme un sac de mots présentiel (binaire, présence/absence) sur le dictionnaire précédent. Les représentations obtenues sont de très grande dimension et creuses. Les tailles des dictionnaires de chaque jeu de données sont présentées dans la table 2.

Nom	Documents d'entraînement	Mots uniques	Dictionnaire (filtré)
Yelp	411 735	161 394	48 983
Amazon	1 260 457	542 545	101 004
RateBeer	931 226	199 548	53 217
Movies	4 017 660	447 247	269 385

Tableau 2. *Chaque ligne correspond à un jeu de données. La colonne de gauche rappelle le nombre de documents dans l'ensemble d'entraînement, la colonne centrale le nombre total de chaînes de caractères distinctes, la colonne de droite donne le nombre de mots sélectionnés dans le dictionnaire (ie apparaissant dans plus de 10 documents).*

4.3. *Sélection des paramètres*

Le protocole de sélection des meilleurs paramètres pour nos modèles est le suivant : nous séparons nos données en trois ensembles (entraînement, validation et test).

Nous construisons une grille de valeurs pour les paramètres et apprenons les modèles sur les ensembles d'apprentissage. Les meilleurs paramètres sont identifiés en utilisant comme critère la performance sur l'ensemble de validation. Les modèles sont ensuite entraînés à nouveau, utilisant ces paramètres, sur l'union des ensembles d'entraînement et de validation. La seule exception concerne le régresseur qui est entraîné en sortie du modèle de recommandation et sur l'ensemble de validation directement, comme présenté en section 3.2, pour éviter le sur-apprentissage.

4.4. *Performance en classification de polarité*

Nous comparons les performances de nos modèles en utilisant le taux d'erreur en classification qui compte le pourcentage d'erreurs de classification sur les données de test. Ces performances sont présentées table 3 pour nos trois modèles :

LSVM pour *Linear SVM* est un SVM linéaire entraîné classiquement pour la classification de polarité. Comme l'indique (Pang et Lee, 2008), il s'agit d'une référence solide.

Surprise est notre chaîne composée d'un système de recommandation et de la correction associée au régresseur défini en section 3.2.

LSVM + Surprise les deux modèles présentés combinés linéairement *a posteriori*.

Nom	LSVM	Surprise	LSVM + Surprise
Yelp	6.07	8.51	5.97
Amazon	6.26	9.51	6.04
RateBeer	8.75	10.22	6.01
Movies	4.16	5.57	3.88

Tableau 3. *Taux d'erreur en classification sur la base de test pour chaque jeu de données. Chaque ligne est un jeu différent, chaque colonne un modèle.*

Les performances du modèle de surprise sont, sans surprise, moins bonnes que celles d'un SVM linéaire. Comme présenté dans (Pang et Lee, 2008), ce dernier est un très bon modèle pour la tâche de classification de polarité alors que notre modèle s'intéresse plus à la l'extraction d'un vocabulaire de surprise qu'aux performances pures en classification de polarité. Nous avons poursuivi cette campagne d'expériences en testant la combinaison *a posteriori* des deux modèles : le SVM et le modèle de Surprise exploitent bien une information différente et le modèle hybride permet un gain important en classification (les bases de test comptent entre 100k et 800k documents). Cette architecture rappelle celle des Time-Delay Neural Network (TDNN) où l'erreur commise dans le passée est souvent passée en argument du classifieur à l'instant t : il s'agit de combiner les informations présentes (LSVM) avec un profil issu de l'historique (Système de recommandation) associé à une mesure de correction (régresseur de surprise).

4.5. *Extraction du vocabulaire de surprise*

Nous proposons ici de comparer les poids extraits par le modèle de référence, LSVM, et notre modèle Surprise. Pour chacun de ces modèles, chaque poids est associé à un mot. Pour LSVM, un poids positif fort est associé à des mots comme "awesome" ou "great" utilisés généralement pour décrire des avis positifs. À l'inverse, un poids négatif fort est associé à des mots comme "worst" ou "horrible" utilisés dans des critiques négatives. Pour le modèle Surprise, le phénomène est différent puis qu'il ne s'agit pas, pour le modèle, de prédire directement la polarité mais de compenser l'erreur d'un système de recommandation en utilisant le texte. Notre hypothèse est qu'alors :

– un poids positif est attribué aux mots qui expliquent pourquoi l'utilisateur est agréablement surpris par l'objet ;

– un poids négatif, au contraire, est affecté à des mots qui présentent une déception de l'utilisateur ou une situation désagréable.

Les mots forts comme "disgusting" ou "amazing" vont généralement avoir des poids forts dans les deux modèles, mais certains mots spécifiques vont émerger dans le modèle surprise. Nous nous concentrons maintenant sur la comparaison des poids des mots les plus positifs et les plus négatifs pour le modèle de référence et le modèle Surprise. La table 4 donne les 10 mots les plus positifs pour chaque modèle (colonnes) et chaque jeu de données (lignes). La liste des mots, dans chaque cellule est triée par ordre décroissant de pondération, ainsi le premier est le mot le plus positif.

Cette liste est bien différente entre les deux modèles, ce qui révèle l'extraction de vocabulaires différents. Sur Yelp !, les utilisateurs décrivent des services disponibles dans une ville, comme des restaurants ou des bars et notre modèle de surprise détecte des mots utilisés pour décrire une amélioration dans le service "upgraded" et "cares". Sur Amazon, le vocabulaire extrait est révélateur de la guerre des commentaires qui prend place sur le portail et sélectionne les mots utilisés par les défenseurs des articles en question, majoritairement des livres, comme "misunderstood", "reviewers" et "skeptical". RateBeer est un site particulier où une communauté d'amateurs se rassemble pour évaluer des bières et notre modèle extrait une surprise quand à la dégustation d'une bière avec des termes comme "surprising", "refreshing" ou "underrated". Enfin, sur Movies, qui est également issu d'Amazon, un comportement similaire à celui pour Amazon est observé et révélateur d'une guerre des commentaires entre les utilisateurs ("haters") mais également d'un appréciation malgré les critiques ("complain", "rocked").

Analysons maintenant les termes les plus négatifs, toujours par rapport à notre modèle de base, qui est globalement similaire à celui pour les mots positifs. Ces mots sont rassemblés dans la table 5, utilisant la même convention que pour les mots positifs dans la table 4 : un modèle par colonne et un jeu de données par ligne. Pour Yelp !, les mots négatifs sont associés à un mauvais service ("rude"). Sur Amazon, chose particulière, beaucoup d'utilisateurs sont déçus, en particulier de la livraison plus que du produit, et utilisent les commentaires pour demander un remboursement ("refund").

	Mots les plus positifs	
Données	LSVM	Surprise
Yelp	delicious, excellent, perfection, pleasantly, downside, amazing, awesome, complaint, fantastic, perfect	**amazing**, **outstanding**, excellent, **cares**, best, blast, complaints, **upgraded**, incredible, thorough
Amazon	refreshing, complaint, pleasantly, excellent, hilarious, awesome, bravo, rocks, funniest, succeeds	**haters**, best, **complaining**, refreshing, awsome, **misunderstood**, rocks, **reviewers**, **skeptical**, great
RateBeer	delicious, excellent, superb, wonderful, awesome, yummy, rjt, fantastic, lovely, perfect	**underrated**, **refreshing**, favorite, **surprised**, favorites, **enjoyed**, marks, rocks, lagers, girls
Movies	pleasantly, bbii, coulardeau, balian, quotable, complaints, refreshing, complacency, bedford, unfairly	awesome, awsome, **haters**, **complaints**, **complaining**, **complain**, great, funniest, **rocked**, refreshing

Tableau 4. *Mots les plus positifs pour chaque modèle (colonnes) et pour chaque jeu de données (lignes). Il s'agit des 10 termes ayant les poids les plus positifs dans le vecteur de paramètre du modèle.*

Sur RateBeer, le comportement reflète celui pour les mots positifs avec la présence ici de terme associés à une mauvaise expérience de dégustation, en particulier sur des bières que l'utilisateur pensait apprécier ("overrated", "yuck"). Enfin sur Movie, le comportement est similaire à Amazon encore une fois, car les données proviennent toutes d'Amazon.com et montre des déceptions sur la réception du produit et des demandes de remboursement ("defective", "refund"). La demande de remboursement est un très bon marqueur de déception : il faut que l'acheteur ait été suffisamment tenté pour acheter le produit et suffisamment déçu pour le retourner.

5. Conclusion

Le filtrage collaboratif et, plus généralement, la recommandation de produits est aujourd'hui une des applications la plus étudiée pour personnaliser l'accès à l'information et construire des profils d'utilisateurs. L'enjeu principal est d'exploiter au mieux l'historique des utilisateurs (et des objets) pour proposer des recommandations pertinentes. Nous montrons que ces profils sont aussi très intéressants pour d'autres tâches. Cette capacité à construire et exploiter un contexte permet d'améliorer les techniques d'analyse de sentiments, en terme de taux d'erreur en classification. Différents aspects temporels sont alors mélangés : le passé pour extraire les profils et construire un modèle de correction d'erreur et le présent, lors de l'analyse de la revue courante. Cette

	Mots les plus négatifs	
Données	LSVM	Surprise
Yelp	worst, mediocre, meh, horrible, bland, disappointing, terrible, overrated, disappointment, poisoning	worst, horrible, **rude**, terrible, **awful**, mediocre, bland, **waste**, meh, poisoning
Amazon	disappointment, yawn, worst, waste, uninspired, overrated, stinks, poorly, boring	waste, disappointing, disappointment, ugh, worst, overrated, boring, stinks, horrible, **refund**
RateBeer	mode, drain, meh, infected, disappointing, mess, disappointment, boring, mediocre, undrinkable	drain, **yuck**, **overrated**, disappointment, disappointing, undrinkable, **mess**, awful, drainpour, **worst**
Movies	kgharris, stinks, noooo, overrated, disappointing, disappointment, stinker, waste, disgraceful, lucasfilm	waste, **worst**, **refund**, **boring**, disappointing, disappointment, stinks, horrible, overrated, **defective**

Tableau 5. *Mots les plus négatifs pour chaque modèle (colonnes) et pour chaque jeu de données (lignes). Il s'agit des 10 termes ayant les poids les plus négatifs dans le vecteur de paramètre du modèle.*

combinaison se révèle particulièrement efficace. Cependant, la principale contribution de cet article réside dans la constitution d'une nouvelle ressource lexicale : le vocabulaire de surprise. Traditionnellement, les mots marqueurs d'opinion forts sont de type "great", "amazing" ou "disgusting". En entraînant un modèle de régression opérant sur les textes pour corriger la sortie d'un système de recommandation, nous avons introduit un contexte et modifié les poids associés aux termes du dictionnaire. L'analyse des poids de notre modèle de surprise révèle effectivement la mise en valeur de termes différents, qui traduisent une différence entre l'attente de l'utilisateur et son expérience avec l'objet mais également la distinction entre ce que pense l'utilisateur et d'autres utilisateurs.

Remerciements

Ce travail a été réalisé avec le soutien des projets FUI AMMICO et ITER-RH (Investissement d'Avenir).

6. Bibliographie

Adomavicius G., Tuzhilin A., « Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions », *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, nº 6, p. 734-749, 2005.

Bennett J., Lanning S., « The netflix prize », *Proceedings of KDD cup and workshop*, vol. 2007, p. 35, 2007.

Blitzer J., Dredze M., Pereira F., « Biographies, bollywood, boom-boxes and blenders : Domain adaptation for sentiment classification », *ACL*, vol. 7, Citeseer, p. 440-447, 2007.

Breese J. S., Heckerman D., Kadie C., « Empirical analysis of predictive algorithms for collaborative filtering », *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., p. 43-52, 1998.

Burke R., « Knowledge-based recommender systems », *Encyclopedia of library and information systems*, vol. 69, p. 175-186, 2000.

Burke R., « Hybrid recommender systems : Survey and experiments », *UMUAI'02*, vol. 12, nº 4, p. 331-370, 2002.

Chakraborty T., Even-Dar E., Guha S., Mansour Y., Muthukrishnan S., « Selective call out and real time bidding », *Internet and Network Economics*, Springer, p. 145-157, 2010.

Diaz F., Metzler D., Amer-Yahia S., « Relevance and ranking in online dating systems », *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 66-73, 2010.

Dorffner G., « Neural networks for time series processing », *Neural Network World*, Citeseer, 1996.

Fürnkranz J., Hüllermeier E., *Preference learning*, Springer, 2010.

Ganu G., Elhadad N., Marian A., « Beyond the Stars : Improving Rating Predictions using Review Text Content. », *WebDB*, 2009.

Glorot X., Bordes A., Bengio Y., « Domain adaptation for large-scale sentiment classification : A deep learning approach », *ICML'11*, p. 513-520, 2011.

Graves A., *Supervised sequence labelling with recurrent neural networks*, vol. 385, Springer, 2012.

Jindal N., Liu B., « Opinion spam and analysis », *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ACM, p. 219-230, 2008.

Kantor P. B., Rokach L., Ricci F., Shapira B., *Recommender systems handbook*, Springer, 2011.

Koren Y., « Factorization meets the neighborhood : a multifaceted collaborative filtering model », *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, p. 426-434, 2008.

Koren Y., Bell R., « Advances in collaborative filtering », *Recommender Systems Handbook*, Springer, p. 145-186, 2011.

Koren Y., Bell R., Volinsky C., « Matrix factorization techniques for recommender systems », *Computer*, vol. 42, nº 8, p. 30-37, 2009.

Krizhevsky A., Sutskever I., Hinton G. E., « Imagenet classification with deep convolutional neural networks », *Advances in neural information processing systems*, p. 1097-1105, 2012.

Lawrence S., Giles C. L., Tsoi A. C., Back A. D., « Face recognition : A convolutional neural-network approach », *Neural Networks, IEEE Transactions on*, vol. 8, nº 1, p. 98-113, 1997.

Le Q. V., Mikolov T., « Distributed Representations of Sentences and Documents », *arXiv preprint arXiv :1405.4053*, 2014.

Low Y., Gonzalez J., Kyrola A., Bickson D., Guestrin C., Hellerstein J. M., « Graphlab : A new framework for parallel machine learning », *arXiv preprint arXiv :1006.4990*, 2010.

McAuley J. J., Leskovec J., « From amateurs to connoisseurs : modeling the evolution of user expertise through online reviews », *Proceedings of the 22nd international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, p. 897-908, 2013a.

McAuley J., Leskovec J., « Hidden factors and hidden topics : understanding rating dimensions with review text », *Proceedings of the 7th ACM conference on Recommender systems*, ACM, p. 165-172, 2013b.

McFee B., Lanckriet G. R., « Metric learning to rank », *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, p. 775-782, 2010.

McLaughlin M. R., Herlocker J. L., « A collaborative filtering algorithm and evaluation metric that accurately model the user experience », *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 329-336, 2004.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed representations of words and phrases and their compositionality », *Advances in Neural Information Processing Systems*, p. 3111-3119, 2013.

Pang B., Lee L., « Opinion mining and sentiment analysis », *Foundations and trends in information retrieval*, vol. 2, n° 1-2, p. 1-135, 2008.

Pazzani M. J., Billsus D., « Content-based recommendation systems », *The adaptive web*, Springer, p. 325-341, 2007.

Poussevin M., Guardia-Sebaoun E., Guigue V., Gallinari P., « Recommandation par combinaison de filtrage collaboratif et d'analyse de sentiments », *CORIA*, 2014.

Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., « GroupLens : an open architecture for collaborative filtering of netnews », *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, ACM, p. 175-186, 1994.

Schafer J. B., Konstan J., Riedl J., « Recommender systems in e-commerce », *Proceedings of the 1st ACM conference on Electronic commerce*, ACM, p. 158-166, 1999.

Teevan J., Dumais S. T., Horvitz E., « Personalizing search via automated analysis of interests and activities », *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 449-456, 2005.

Vedaldi A., « An open implementation of the SIFT detector and descriptor », *UCLA CSD*, 2007.