CLASSIFICATION NON SUPERVISÉE DES DONNÉES DE HAUTES DIMENSIONS ET EXTRACTION DES CONNAISSANCES DANS LES SERVICES WEB DE QUESTION-RÉPONSE

par

Mohamed Bouguessa

Thèse présentée au Département d'informatique
en vue de l'obtention du grade de philosophiæ doctor (Ph.D.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 16 mars 2009

Your file Votre référence
ISBN: 978-0-494-48534-7
Our file Notre référence
ISBN: 978-0-494-48534-7

Le 16 mars 2009

le jury a accepté la thèse de M. Mohamed Bouguessa dans sa version finale.

Membres du jury

M., Shengrui Wang
Directeur
Département d'informatique
Université de Sherbrooke

Monsieur Jean-Pierre Dussault
Président rapporteur
Département d'informatique
Université de Sherbrooke

Monsieur Ernest Monga
Membre interne au programme
Département de mathématiques
Université de Sherbrooke

Monsieur Benoît Dumoulin
Évaluateur externe
Yahoo! Inc.

Monsieur Pierre Hansen
Évaluateur externe
HEC Montréal

À la mémoire de mon père.

À la mémoire de ma grand-mère.

Sommaire

Cette thèse à publication propose d'étudier deux problématiques différentes : 1) la classification non supervisée (*clustering*) des données de hautes dimensions, et 2) l'extraction des connaissances dans les services Web de question-réponse. Nos contributions sont présentées à travers trois chapitres. Dans le premier chapitre, nous proposons un algorithme de *projected clustering* nommé PCKA (Projected Clustering based on the K-means Algorithm). Contrairement à la vaste majorité des approches existantes, PCKA est capable de découvrir des structures de *clusters* qui existent dans différents sous-espaces de faibles dimensionnalités et ce en utilisant une mesure de similarité bien adaptée aux caractéristiques particulières des données multidimensionnelles. La fiabilité de PCKA est illustrée à travers des tests et des comparaisons avec les approches existantes sur une variété de données synthétiques et réelles. Le deuxième chapitre aborde le problème de l'identification des utilisateurs experts dans les forums Internet de question-réponse. Notre contribution inclut le développement d'une approche probabiliste qui se base sur le modèle de mélange de distributions de la loi Gamma. Notre approche permet de séparer, de façon systématique, les utilisateurs experts des non-experts alors que les approches existantes fournissent une liste ordonnée d'utilisateurs seulement. Le troisième chapitre étudie le problème de l'identification des communautés dans les forums Internet de question-réponse. Notre contribution inclut l'introduction du nouveau concept de "communauté de partage des connaissances". Ces communautés sont définies par les interactions entre les utilisateurs experts et non-experts. Pour identifier ce type de communauté nous représentons notre environnement sous la forme des données transactionnelles et nous proposons un algorithme de *clustering* nommé TRANCLUS (TRAnsaction CLUStering). Les *clusters* identifiés par TRANCLUS représentent les communautés que nous cherchons à découvrir. Notre approche est validée sur des données extraites de plusieurs forums de Yahoo ! Answers.

Remerciements

Tout d'abord, je remercie Dieu de m'avoir aidé à mener cette thèse à terme.

Je tiens à remercier sincèrement mon directeur de thèse, le professeur Shengrui Wang, pour la pertinence de son encadrement, ses encouragements dans les moments difficiles et surtout pour ses qualités humaines. Sa rigueur scientifique, ses conseils judicieux, et sa disponibilité ont joué un rôle déterminant pour mener ce travail à terme. Avec lui, j'ai appris l'efficacité, la flexibilité et surtout la patience. Qu'il trouve ici, l'expression de ma profonde reconnaissance.

Je remercie également Benoît Dumoulin, mon superviseur de stage de recherche chez Yahoo ! Canada, de m'avoir accueilli dans son groupe et de la confiance qu'il m'a témoignée en me donnant accès aux données de Yahoo ! Answers pendant une année entière. Son dynamisme et sa disponibilité ont rendu mon stage chez Yahoo ! Canada très agréable.

Mes remerciements vont aussi au Conseil de Recherches en Sciences Naturelles et en Génie du Canada pour le soutien financier pendant trois ans entières. Je remercie également le Département d'Informatique de l'Université de Sherbrooke de m'avoir donné l'opportunité d'enseigner.

Merci à mon père et à ma mère. Toutes les bonnes choses que j'ai pu réalisés dans cette vie c'est grâce à vous. Je peux écrire des milliers de lignes pour exprimer ma reconnaissance envers tout ce que vous avez fait pour moi. Mais cela ne suffit pas. Je vous dis merci pour tout et je vous dédie ce travail.

Je tiens aussi à remercier mes sœurs et mes frères pour leur encouragement et leur soutien.

Mes vifs remerciements vont tout particulièrement à ma femme. Merci de m'avoir soutenu dans les moments difficiles. Merci d'être aussi patiente avec moi. Merci pour tout.

Hannine, ma petite fille, merci d'avoir rendu ma vie très agréable.

Table des matières

i

Sommaire iii

Remerciements iv

Table des matières v

Introduction 1

0.1 Clustering des données de hautes dimensions 2

0.1.1 Projected clusters 4

0.1.2 Projected clustering 5

0.2 Extraction des connaissances dans les services Web de question-réponse . . 7

0.2.1 Identification des experts 9

0.2.2 Identification des communautés 10

1 Classification non supervisée des données de hautes dimensions 13

2 Identification des experts dans les services Web de question-réponse 52

3 Identification des communautés dans les services Web de question-réponse 63

Conclusion 114

Introduction

De nos jours, l'informatique et les moyens numériques de communication associés permettent le stockage, la manipulation et le transfert de quantités importantes de données. Actuellement, certaines bases de données comportent des millions d'entrées pour des milliers de champs. Nous pouvons citer en exemple, l'analyse des fichiers .log de connections à des serveurs Web. Ces mégabases de données, qui ne cessent d'augmenter jour après jour, sont peu exploitées, alors qu'elles cachent des informations décisives face au marché et à la concurrence. Le besoin d'extraire de l'information pertinente de ces données est alors un enjeu d'actualité. La plupart des techniques d'analyse de données traditionnelles rencontrent des difficultés sur ce type de données [11], [17] (ex : les modèles habituellement utilisés pour des dimensions faibles ne sont généralement pas directement transposables dans des dimensions bien supérieures). Pour combler ce besoin, une nouvelle discipline émerge : le forage de données (*Data Mining*).

Le forage de données est l'ensemble des algorithmes et méthodes destinées à l'exploration et l'analyse de grandes bases de données informatiques en vue de détecter dans ces données des règles, des associations, des tendances inconnues, des structures particulières restituant de façon concise l'essentiel de l'information utile. Généralement, les techniques de forage de données se déclinent en deux grandes catégories [17] : 1) les techniques descriptives, et 2) les techniques prédictives. L'objectif de ces dernières est principalement l'inférence sur les données pour faire la prédiction alors que l'objectif des techniques descriptives est la découverte des patterns (associations, corrélations, structures homogènes) qui proposent une vue réductrice et simplifiée d'un ensemble de données et qui révèlent des relations utiles. Contrairement aux techniques prédictives, les techniques descriptives de forage de données sont souvent de nature exploratoire et nécessitent des méthodes de post-traitement pour valider et expliquer les résultats.

Dans notre travail, nous nous intéressons aux approches descriptives de forage de données. Spécifiquement, nous nous focalisons sur les techniques de classification non-supervisée (*clustering*) et les approches à modèle (*model-based approaches*). Cette thèse étudie deux problématiques majeures :

1. Le forage de données de hautes dimensions ;
2. L'extraction des connaissances dans les forums Internet de question-réponse.

Nos contributions incluent :

1. Le développement d'un algorithme de *projected clustering* pour les données de hautes dimensions ;
2. Le développement d'une approche automatique pour l'identification des experts dans les forums Internet de question-réponse ;
3. L'introduction d'un nouveau concept de "communauté de partage des connaissances" dans les forums Internet de question-réponse et le développement d'un algorithme de *clustering* pour identifier ce type de communautés.

Le but des deux sections suivantes est de préciser davantage le cadre général et les objectifs de notre travail ainsi que nos contributions. En premier lieu, nous discutons la problématique de *clustering* des données de hautes dimensions. Nous illustrons les principaux éléments qui ont motivé notre étude. En second lieu, motivé par la récente explosion des sites Web communautaires qui placent l'utilisateur au centre de l'action, nous nous intéressons au développement et à l'application des techniques de forage de données pour étudier les liens/connections qui existent entre les individus afin d'identifier leurs rôles et extraire les communautés virtuelles qui partagent les mêmes intérêts. Spécifiquement, nous nous focalisons sur l'analyse des interactions des utilisateurs dans les forums Internet de question-réponse dont le but d'identifier les experts et extraire les communautés qui se construisent autour d'eux.

0.1 Clustering des données de hautes dimensions

Le processus du *clustering* vise à construire des groupes (*clusters*) d'objets similaires à partir d'un ensemble hétérogène d'objets. Chaque *cluster* issu de ce processus doit vérifier

les deux propriétés suivantes : 1) La cohésion interne (les objets appartenant à ce *cluster* soient les plus similaires possibles) et 2) L'isolation externe (les objets appartenant aux autres *clusters* soient les plus distinctes possibles). Le *clustering* repose sur une mesure précise de la similarité / dissimilarité des objets que nous voulons regrouper. Cette mesure est appelée distance.

La distance euclidienne, communément utilisée, considère que deux objets sont similaires si et seulement si les valeurs de tout leurs attributs [1] sont proches les unes des autres. Autrement dit, la distance euclidienne (et beaucoup d'autres distances) traite toutes les dimensions de la même manière en leur accordant la même importance. Cependant, dans le cadre des données de hautes dimensions, certaines dimensions peuvent être discriminantes pour la formation d'un certain *cluster*, alors que ces mêmes dimensions peuvent s'avérer peu pertinentes pour la formation d'un autre *cluster*. En d'autres termes, les *clusters* peuvent exister dans différentes combinaisons des sous-espaces de dimensions et non dans tout l'ensemble des dimensions [2], [3].

La figure 1, illustre ce phénomène. Dans cet exemple [2], nous avons quatre objets $\{x_1, x_2, x_3, x_4\}$ à grouper. Si nous utilisons la distance euclidienne comme mesure de similarité, il est fort probable que x_2 et x_3 vont être placés dans le même *cluster*, puisque leur distance (c.-à-d. 41.23) est la plus petite comparativement aux distances entre n'importe quels paires d'objets. Cependant, une simple inspection visuelle suggère qu'il y a deux *clusters* : $C_1 = \{x_1, x_2\}$ et $C_2 = \{x_3, x_4\}$. Les dimensions $\{A_1, A_2, A_3\}$ forment le *cluster* C_1, alors que les dimensions $\{A_3, A_4, A_5\}$ forment le *cluster* C_2. Les sous-ensembles de dimensions $\{A_4, A_5\}$ et $\{A_1, A_2\}$ représentent les dimensions non pertinentes pour C_1 et pour C_2 respectivement.

	A_1	A_2	A_3	A_4	A_5
x_1	0	0	0	10	100
x_2	0	0	0	70	30
x_3	20	10	20	50	50
x_4	80	80	20	50	50

figure 1 – Un ensemble de données qui contient deux *projected clusters*.

[1] Dans cette thèse, les termes "attribut" et "dimension" vont êtres utilisés alternativement pour designer le même concept.

[2] Exemple tiré de [13].

Face à cette situation, il est évident qu'un algorithme de *clustering* traditionnel (ex : K-means) qui est basé sur une distance définie globalement entre les objets va échouer à identifier correctement les *clusters*. Pour pallier à ce problème, une nouvelle classe d'algorithme de *clustering* appelée *projected clustering* émerge. En général, les techniques de *projected clustering* exploitent le fait que dans les données de hautes dimensions, différents groupes d'objets peuvent êtres corrélés à travers différents sous-ensembles de dimensions. Les *clusters* identifiés par ce type d'algorithmes sont appelés *projected clusters*.

0.1.1 Projected clusters

Soit DB un ensemble d'objets représentés dans un espace vectoriel numérique continu de dimension d. Nous notons par $A = \{A_1, A_2, \ldots, A_d\}$ l'ensemble des attributs. Soit $X = \{x_1, x_2, \ldots, x_N\}$ un ensemble de N objets, tel que $x_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{id})$. Chaque x_{ij} $(i = 1, \ldots, N; j = 1, \ldots, d)$ correspond à la valeur de l'objet x_i dans l'attribut A_j. Dans le présent travail, nous supposons que x_i appartient soit à un et un seul cluster soit à l'ensemble des données qui représente le bruit (outliers) OUT. Nous désignons par nc le nombre de *clusters* fourni par l'utilisateur. Un *projected cluster* C_s $(s = 1, \ldots, nc)$ est défini par la paire (SP_s, SD_s), où SP_s est un sous-ensemble d'objets de DB et SD_s est un sous-ensemble de dimensions de A, tel que tous les points dans SP_s sont fortement corrélés lorsqu'ils sont projetés sur chaque dimension de SD_s. L'ensemble de dimensions dans SD_s sont appelées dimensions pertinentes pour le *cluster* C_s, alors que l'ensemble de dimensions restantes, c.-à-d. $A - SD_s$, sont les dimensions non-pertinentes pour le *cluster* C_s.

Dans notre travail, nous nous focalisons sur les *projected clusters* qui sont parallèles aux axes et qui possèdent les propriétés suivantes :

1. Les *projected clusters* doivent être denses. Spécifiquement, les valeurs des attributs des objets projetés sur chaque dimension de $\{SD_s\}_{s=1,\ldots,nc}$ forme des régions de haute densité comparativement la densité des objets projetés sur chaque dimension de $\{A - SD_s\}_{s=1,\ldots,nc}$.
2. Les sous-ensembles de dimensions $\{SD_s\}_{s=1,\ldots,nc}$ peuvent ne pas être disjoints et peuvent avoir des cardinalités différentes.

3. Pour chaque *projected cluster* C_s, les objets de SP_s projetés sur chaque dimension de SD_s sont les plus similaires possibles comparativement aux autres objets qui n'appartiennent pas à C_s.

La première propriété est essentiellement basée sur le fait que les dimensions pertinentes d'un *cluster* contiennent des régions de haute densité comparativement aux dimensions non-pertinentes de telle sorte que le concept de "densité" est relativement comparable à travers toutes les dimensions de DB. La raison de la deuxième et la troisième propriété est triviale. À titre d'exemple, l'ensemble de données illustrés dans la figure 1 contient deux *projected cluster* : $C_1 = (SP_1, SD_1)$ avec $SP_1 = \{x_1, x_2\}$ et $SD_1 = \{A_1, A_2, A_3\}$. $C_2 = (SP_2, SD_2)$ avec $SP_2 = \{x_3, x_4\}$ et $SD_2 = \{A_3, A_4, A_5\}$.

0.1.2 Projected clustering

Les algorithmes de *projected clustering* se distinguent des techniques de *clustering* traditionnelles par le fait qu'ils doivent identifier, à la fois, les *clusters* et leurs dimensions pertinentes. Face à ce problème, un nombre restreint d'algorithmes de *projected clustering* ont été proposés récemment. En dépit du fait que ces approches réussissent à identifier des *projected clusters*, elles rencontrent des difficultés majeures à détecter des *projected clusters* qui sont caractérisés par un nombre très faible de dimensions pertinentes. Par exemple, les données d'expression de gènes (gene expression data) représentent un cas concret des données de hautes dimensions qui des *projected clusters* de très faible dimensionnalité [5], [22]. Par conséquent, l'application des algorithmes existants de *projected clustering* sur ce type de données entraîne une perte d'information substantielle, car ces derniers ne sont pas capables d'identifier correctement les *clusters* ainsi que leurs dimensions pertinentes.

Nos études et expérimentations démontrent que les algorithmes existants de *projected clustering* sont seulement fiables lorsque le nombre de dimensions pertinentes des *clusters* n'est pas trop inférieur à la dimensionnalité de tout l'ensemble de données. Par exemple, certains algorithmes, comme PROCLUS [1] et ORCLUS [2], utilisent une fonction de similarité qui considère toutes les dimensions, au même degré d'importance, afin de trouver une approximation initiale des *clusters*. Après cela, les dimensions pertinentes de chaque *cluster* sont déterminées en utilisant un certain nombre d'heuristiques et le *clustering* est alors raffiné en se basant sur les dimensions précédemment sélectionnées. Ici, il est clair

qu'une mesure de similarité qui utilise toutes les dimensions pour calculer la similarité entre deux objets biaise le mécanisme de détection des dimensions pertinentes et par conséquent affecte grandement les résultats de ces algorithmes. Un autre exemple est HARP [20], un algorithme hiérarchique de *projected clustering* qui se base sur l'hypothèse que deux objets peuvent probablement appartenir au même *cluster* s'ils sont similaires l'un à l'autre à travers plusieurs dimensions. Cependant, lorsque le nombre de dimensions pertinentes pour chaque *cluster* est très petit comparativement à la dimensionnalité de tout l'ensemble de données, cette hypothèse devient complètement non-valide.

Outre le problème mentionné ci-dessus, certains algorithmes, comme PROCLUS [1], ORCLUS [2], DOC [16] et ses dérivées FASTDOC [16] et CFPC [13], nécessitent que le nombre moyen des dimensions pertinentes soit fourni par l'utilisateur. Dans plusieurs cas pratiques, cette tâche est difficile à réaliser. Contrairement à ces prédécesseurs, HARP [20] propose un mécanisme automatique pour identifier les dimensions pertinentes. Cependant, en plus du fait que HARP est fiable seulement si la dimensionnalité des *clusters* n'est pas trop petite par rapport à la dimensionnalité de tout l'ensemble de données, il nécessite que l'utilisateur lui fournisse le pourcentage maximal des points qui représentent le bruit. En pratique, ce paramètre n'est pas évident à fournir et une fausse estimation par l'utilisateur du pourcentage du bruit dans l'ensemble de données affecte la qualité des résultats. A fin d'améliorer HARP, ses auteurs ont proposé un autre algorithme nommé SSPC [21]. SSPC améliore grandement la performance de ces prédécesseurs. Cependant, SSPC est un algorithme de *clustering* semi-supervisé et son bon fonctionnement est conditionné par un minimum de connaissances sur les données à regrouper - disponibilités des labels qui indiquent : 1) l'appartenance d'un objet à un *cluster* spécifique, et/ou 2) la pertinence de certaines dimensions pour un *cluster* particulier. Ici, il est clair que dans plusieurs applications réelles, ces informations ne sont pas toujours disponibles.

Objectif et contribution

Notre objectif est le développement d'un algorithme de *projected clustering* capable de surmonter les difficultés des approches existantes en considérant certains facteurs qui constituent une source d'échec de la plupart de ces approches. Dans ce contexte, notre contribution consiste à proposer un nouvel algorithme de *projected clustering* que nous avons nommé PCKA (Projected Clustering based on the K-means Algorithm). Notre al-

gorithme est composé de trois phases : 1) Analyse de la pertinence des dimensions, 2) Extraction et élimination du bruit, et 3) Identification des *projected clusters*. PCKA fait partie de la famille des algorithmes de *clustering* par partition, et il est capable de réduire la complexité et d'extraire des informations pertinentes d'un ensemble de données de hautes dimensions. Cela est grâce à l'utilisation d'une formulation statistique, bien adaptée au contexte des données multidimensionnelles, qui prend en compte à la fois la masse de données, leur hétérogénéité et la performance algorithmique. Notre algorithme est capable aussi de découvrir les structures de *clusters* qui existent dans les différents sous-espaces de très faibles dimensionnalités en utilisant une mesure de similarité bien adaptée aux caractéristiques particulière des données multidimensionnelles. PCKA est présenté dans le chapitre 1 de cette thèse.

0.2 Extraction des connaissances dans les services Web de question-réponse

De nos jours, nous assistons à une évolution du World Wide Web suite à l'émergence de nouveaux services Web d'interactivité qui placent l'utilisateur et ses relations avec les autres au centre de l'Internet. Ces services sont communément appelés les services "Web 2.0". Les technologies Web 2.0 sont la source de nouvelles applications sur les sites Internet, d'une ergonomie améliorant le confort de l'utilisateur et surtout d'interactions plus fortes entre le site et les internautes et entre les internautes eux-mêmes. Elles favorisent alors la montée en puissance d'un Web plus collaboratif, replaçant ainsi l'utilisateur au cœur des services en ligne. À titre d'exemple, nous citons les sites Web communautaires et de réseautage social (social networking). En effet, les sites Web de réseau sociaux sont des services participatifs et collaboratifs qui visent à établir des connexions entre les utilisateurs du Web pour améliorer la créativité et relancer le partage des connaissances. Le challenge pour les entreprises qui possèdent ces sites Web consiste à bien comprendre comment les utilisateurs interagissent entre eux de manière à leur offrir des innovations qui assureront leur fidélité. Sans cela, les utilisateurs désertent rapidement ces sites et ces derniers deviennent moribonds. Motivés par ces nouveaux challenges, nous nous sommes intéressés à analyser les interactions des utilisateurs dans les forums Internet de question-réponse. Cette

problèmatique représente le deuxième volet que cette thèse propose d'étudier.

L'importance des forums Internet de question-réponse, vient du fait que ces services se voient de plus en plus comme des services complémentaires aux moteurs de recherche Internet. Parfois, il n'est pas évident de trouver une réponse à nos questions et préoccupations à partir d'une simple recherche du Web. À titre d'exemple, citons les cas d'un programmeur Java non expérimenté qui n'arrive pas à compléter son application suite à un bug dont il ne connaît pas la source. Ici, il n'est pas évident de trouver une solution à ce problème en cherchant le Web via un moteur de recherche à partir de certains mots clés. Dans ce cas, il est plus approprié de trouver et de poser le problème à un programmeur expert en Java. Les forums Internet de question-réponse, là où les utilisateurs viennent pour poser et répondre aux questions et par conséquent partagent leurs connaissances, nous offrent cette possibilité et proposent des outils qui aident les internautes dans leurs quêtes d'informations dans plusieurs domaines.

En général, dans les forums de question-réponse, le principal mode d'interaction des utilisateurs experts est de fournir des réponses informatives aux questions des autres participants. Dans ce contexte, les utilisateurs qui posent souvent des questions identifient les experts comme une source d'information fiable qui peut être complémentaire à d'autres sources de connaissances. Des études récentes [14], [19] suggèrent que les utilisateurs experts jouent un rôle critique dans la création, la promotion et le maintien des communautés virtuelles. Dans le contexte des forums de question-réponse, ces communautés peuvent être perçues comme des groupes formés par les utilisateurs qui posent souvent des questions et les experts ; de telle sorte que ces derniers sont le cœur de ces communautés puisqu'ils représentent la source de connaissances que les autres utilisateurs recherchent. Ce type de communautés sont souvent considérées dans le monde d'affaire comme un important moyen pour générer de la connaissance et motiver la créativité. Les compagnies qui possèdent ces forums de question-réponse veulent développer des outils afin de : 1) identifier automatiquement les experts, et 2) détecter et promouvoir des communautés virtuelles qui favorisent le partage des connaissances. Dans cette thèse, nous proposons d'étudier ces deux thématiques qui constituent deux préoccupations actuelles majeures. Comme une étude de cas pratique, nous nous focalisons sur Yahoo ! Answers, un service Internet de question-réponse qui est constitué de plusieurs forums divers en terme de contenus et très large en nombre de participants.

0.2.1 Identification des experts

Il existe plusieurs autres raisons qui motivent l'identification des experts dans les forums Internet de question-réponse. Par exemple, le routage des questions nouvellement posées aux experts appropriés aide de façon significative les utilisateurs qui ont posé ces questions en leur fournissant un service efficace qui vise à minimiser l'effort que ces derniers doivent effectuer pour trouver des réponses correctes à leurs questions. Les utilisateurs experts peuvent aussi jouer un important rôle dans l'amélioration du contenu du site et ce par l'évaluation des réponses des autres participants.

Récemment, le problème d'identification des experts a attiré l'intérêt de la communauté scientifique. La plupart des approches existantes dédiées à l'identification des experts représentent leurs environnements (c.-à-d. les interactions entre les utilisateurs) comme un graphe, dont les nœuds représentent les individus et les arcs représentent les interactions entres eux [9], [10], [23]. Cette représentation graphique permet l'utilisation des techniques d'analyse des liens (link analysis techniques), afin d'analyser la topologie du graphe et estimer un score qui relate l'importance d'un nœud dans le graphe en entier. Le score estimé pour chaque nœud représente le degré d'expertise relatif à chaque utilisateur.

Le résultat final des approches existantes d'identification d'experts est une liste ordonnée d'utilisateurs selon leur degré d'expertise. À partir de cette liste, les K premiers utilisateurs sont choisis comme experts. La faiblesse de telles approches réside principalement dans le choix de la valeur de K. Généralement, le choix du K est basé sur une connaissance *a priori* de l'application en question. Cependant, dans plusieurs cas pratiques, le choix de la valeur du K n'est pas évident. Par exemple, dans Yahoo ! Answers il y a plusieurs forums de dynamiques et de contenus très différents. Dans ce contexte, le choix d'une valeur appropriée de K s'avère alors comme une tâche complexe s'il est fait de façon manuelle, car nous devons inspecter le comportement des utilisateurs dans chaque forum ce qui est exorbitant. En plus de ce problème, dans notre étude nous avons constaté que, dans le contexte des forums Internet de question-réponse comme Yahoo ! Answers, la plupart des techniques d'analyse de liens, comme HITS [12] ou PageRank [15], ne sont pas fiables pour estimer correctement le degré d'expertise des participants.

Objectif et contribution

Le choix de la valeur de K est crucial vu qu'il donne plus de pouvoir aux utilisateurs sélectionnés comme experts. Il est clair qu'un choix inapproprié de K peut affecter de façon négative la qualité du service. Au meilleur de notre connaissance, aucune méthode formelle qui permet de choisir automatiquement les experts dans les forums Internet de question-réponse n'a encore été proposée. Notre objectif est de développer une approche automatique, qui ne requiert aucun paramètre, pour séparer les utilisateurs experts des non-experts plutôt que de fournir une liste ordonnée des utilisateurs seulement. Dans notre approche, nous illustrons, premièrement, comment estimer le degré d'expertise des participants. Deuxièmement, à partir des degrés d'expertise estimés, nous proposons une approche efficace et systématique pour identifier les experts. Notre contribution consiste à proposer un modèle probabiliste qui s'appuie sur l'utilisation des mélanges de distribution de la loi Gamma pour estimer la fonction de densité de probabilité des degrés d'expertise des utilisateurs. Une fois que la fonction de densité de probabilité est estimée, nous démontrons comment discriminer automatiquement entre les utilisateurs experts et non-experts. Cette approche est présentée dans le chapitre 2 de cette thèse.

0.2.2 Identification des communautés

Le savoir est un atout essentiel qui doit être géré stratégiquement [18]. De nos jours, de plus en plus des services Web de *social networking* investissent dans les solutions de gestion des connaissances et du savoir. La localisation des experts et l'identification des communautés sont devenues des aspects importants dans un système d'extraction et de gestion des connaissances. Dans le contexte des forums Internet de question-réponse, l'extraction et la gestion des connaissances sont étroitement liées à l'identification des experts et à l'exploration de leurs interactions dans le site avec les autres participants afin d'identifier des communautés d'utilisateurs qui partagent les mêmes intérêts. Cette stratégie conduit à la création d'un précieux service Web de partage des connaissances. En effet, l'identification des communautés en se basant exclusivement sur les interactions entre les experts et les utilisateurs qui posent des questions permet de promouvoir des interactions entre les participants de telle sorte que les membres de la même communauté apprennent les uns des autres, résolvent les problèmes ensemble et contribuent ainsi à la création de nouvelles

connaissances. Il est clair que l'identification de ce type de communautés améliore de façon substantielle le contenu ainsi que le service offert aux utilisateurs de n'importe quel forum Internet de question-réponse.

Une propriété intéressante de l'interaction entre les utilisateurs experts et ceux qui posent souvent des questions, est que tous ces utilisateurs sont généralement liés par des intérêts communs. Les utilisateurs qui posent des questions préfèrent recevoir des réponses de personnes ayant suffisamment d'expertise sur le sujet qui fait l'objet de la question. En contre partie, les experts fournissent des réponses à : 1) des questions pour lesquelles ils sont intéressés par leurs contenus, plutôt que par qui a posé la question ; 2) des questions reliées à leurs domaine(s) d'expertise. Dans ce contexte, il est raisonnable de supposer que les utilisateurs qui posent beaucoup de questions et qui interagissent souvent avec le même ensemble d'experts sont aussi plus liés les uns aux autres que ceux qui ne le font pas. Par conséquent, les interactions entre les participants conduisent à la formation de communautés d'utilisateurs qui partagent les mêmes intérêts dans un large éventail de sujets. L'apprentissage et le partage des connaissances seront donc les motivations principales qui encouragent les utilisateurs des forums Internet de question-réponse de se réunir et de former des communautés. Dans chaque communauté, les experts sont perçues comme les piliers de cette dernière et la source du savoir que le reste des participants recherchent.

Objectif et contribution

Notre objectif principal consiste à développer un algorithme capable d'identifier le type de communauté mentionné ci-dessus. Nos contributions consistent premièrement à, l'introduction du nouveau concept de "communauté de partage des connaissances" (*knowledge-sharing communities*). Chaque communauté est définie par un ensemble d'experts, et d'utilisateurs qui posent des questions de telle sorte que ces derniers montrent un comportement plus homogène, en termes d'interactions avec les experts, que partout ailleurs. Deuxièmement et afin d'identifier les communautés de partage de connaissances, nous représentons les interactions entres les participants sous la forme des données transactionnelles, de telle sorte que chaque transaction résume toutes les interactions d'un utilisateur avec les experts qui ont répondu à ses questions. Par la suite nous développons un algorithme de *clustering* transactionnel, que nous nommons TRANCLUS (TRANsaction CLUStering), capable d'identifier ces communautés. À notre connaissance, le problème dans cette forme n'a ja-

mais été étudié dans la littérature courante. Il convient aussi de noter que, contrairement à la vaste majorité des algorithmes de *clustering* des données transactionnelles, TRANCLUS effectue le *clustering* de façon systématique et n'exige aucun paramètre à fournir par l'utilisateur. En pratique, cette caractéristique de TRANCLUS représente un avantage majeur. Comme une démonstration pratique de la fiabilité de notre approche d'identification des communautés de partage des connaissances, nous analysons des données réelles extraites de Yahoo ! Answers qui représentent l'activité des utilisateurs pendant une année complète. Ce travail est présenté en détail dans le chapitre 3 de cette thèse.

Chapitre 1

Classification non supervisée des données de hautes dimensions

Dans le contexte des données multidimensionnelles, les *clusters* existent dans de différentes combinaisons des sous-espaces de dimensions. Ces *clusters* sont communément appelés *projected clusters*. Afin d'identifier ce type de clusters, un certain nombre d'algorithmes de *projected clustering* ont été récemment proposés. Cependant, la vaste majorité de ces algorithmes se heurtent à des difficultés majeures lorsque les *clusters* existent dans des sous-espaces de faible dimensionnalité. Face à ce problème, nous proposons un nouvel algorithme de *projected clustering* que nous avons nommé PCKA (Projected Clustering based on the K-means Algorithm). Contrairement aux approches existantes, l'algorithme que nous proposons ne présume aucun modèle de distribution statistique sur les données à regrouper. En Outre, PCKA n'impose aucune restriction sur la taille des *clusters* ou le nombre de dimensions pertinentes pour chaque *cluster*. Un *projected cluster* doit avoir un nombre significatif de dimensions pertinentes, avec un degré de pertinence élevé, dans lesquelles un grand nombre de points sont proches les uns des autres. Pour atteindre cet objectif, PCKA procède en trois phases :

1. **Analyse de la pertinence des attributs** : le but de cette première phase est de découvrir toutes les dimensions qui exhibent une structure de *cluster*, et ce en identifiant les régions denses ainsi que leur emplacement dans chaque dimension. Les dimensions identifiées représentent des candidats potentiels pour les dimensions pertinentes des

clusters.

2. **Extraction et élimination du bruit** : en se basant sur les résultats de la première phase, le but de cette deuxième phase est d'identifier et éliminer le bruit (outlier). Comme la majorité des algorithmes de *clustering*, PCKA considère les points qui représentent le bruit comme des points qui n'exhibent aucune similarité ni entre eux-mêmes, ni avec le reste des points dans l'ensemble de données.
3. **Identification des projected clusters** : le but de cette dernière phase est d'identifier les *clusters* ainsi que leurs dimensions pertinentes. Le processus de *clustering* est basé sur une version modifiée de l'algorithme K-means dans laquelle le calcul de la distance entre un point et le centre de *cluster* est limité seulement aux dimensions qui contiennent des régions denses. A partir des *clusters* identifiés, la dernière étape de PCKA consiste à raffiner les résultats de la phase 1 en sélectionnant les dimensions pertinentes de chaque *cluster*.

Dans les pages suivantes, PCKA est présenté de façon détaillée dans un article intitulé **Mining Projected Clusters in High-Dimensional Spaces**. Cet article est accepté pour publication par le journal international **IEEE Transactions on Knowledge and Data Engineering** [6]. Une version très préliminaire de PCKA (un court article de 4 pages) intitulé **A K-means-based Algorithm for Projective Clustering** est apparue dans les actes de **IEEE International Conference on Pattern Recognition (ICPR 2006)** [8]. Une application de notre approche sur les données génétiques intitulée **PCGEN : A Practical Approach to Projected Clustering and its Application to Gene Expression Data** est publiée dans les actes de **IEEE International Symposium on Computational Intelligence and Data Mining (CIDM 2007)** [5].

Note : ma contribution au chapitre 1 de cette thèse inclut le développement de l'algorithme PCKA, l'élaboration des tests de validation et la rédaction de l'article. Mon directeur de recherche, le professeur Shengrui Wang, a supervisé et validé toutes les étapes de développement de l'algorithme ainsi que la rédaction de l'article.

Mining Projected Clusters in High-Dimensional Spaces

Mohamed Bouguessa and Shergrui Wang

Abstract

Clustering high-dimensional data has been a major challenge due to the inherent sparsity of the points. Most existing clustering algorithms become substantially inefficient if the required similarity measure is computed between data points in the full-dimensional space. To address this problem, a number of projected clustering algorithms have been proposed. However, most of them encounter difficulties when clusters hide in subspaces with very low dimensionality. These challenges motivate our effort to propose a robust partitional distance-based projected clustering algorithm. The algorithm consists of three phases. The first phase performs attribute relevance analysis by detecting dense and sparse regions and their location in each attribute. Starting from the results of the first phase, the goal of the second phase is to eliminate outliers, while the third phase aims to discover clusters in different subspaces. The clustering process is based on the K-means algorithm, with the computation of distance restricted to subsets of attributes where object values are dense. Our algorithm is capable of detecting projected clusters of low dimensionality embedded in a high-dimensional space and avoids the computation of the distance in the full-dimensional space. The suitability of our proposal has been demonstrated through an empirical study using synthetic and real datasets.

Index Terms

Data mining, clustering, high dimensions.

I. Introduction

Data mining is the process of extracting potentially useful information from a dataset [1]. Clustering is a popular data mining technique which is intended to help the user discover and understand the structure or grouping of the data in the set according to a certain similarity measure [2]. Clustering algorithms usually employ a distance metric (e.g., Euclidean) or a similarity

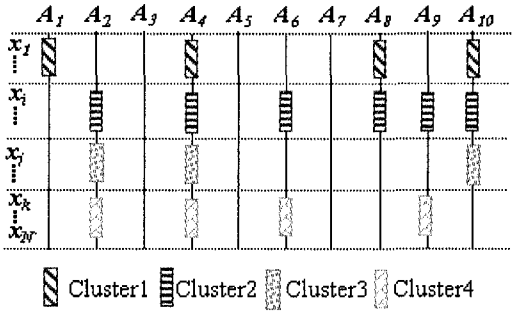


Fig. 1. Example of dataset containing projected clusters.

measure in order to partition the database so that the data points in each partition are more similar than points in different partitions.

The commonly used Euclidean distance, while computationally simple, requires similar objects to have close values in all dimensions. However, with the high-dimensional data commonly encountered nowadays, the concept of similarity between objects in the full-dimensional space is often invalid and generally not helpful. Recent theoretical results [3] reveal that data points in a set tend to be more equally spaced as the dimension of the space increases, as long as the components of the data point are i.i.d. (independently and identically distributed). Although the i.i.d. condition is rarely satisfied in real applications, it still becomes less meaningful to differentiate data points based on a distance or a similarity measure computed using all the dimensions. These results explain the poor performance of conventional distance-based clustering algorithms on such data sets.

Feature selection techniques are commonly utilized as a preprocessing stage for clustering, in order to overcome the curse of dimensionality. The most informative dimensions are selected by eliminating irrelevant and redundant ones. Such techniques speed up clustering algorithms and improve their performance [4]. Nevertheless, in some applications, different clusters may exist in different subspaces spanned by different dimensions. In such cases, dimension reduction using a conventional feature selection technique may lead to substantial information loss [5].

The following example provides an idea of the difficulties encountered by conventional clustering algorithms and feature selection techniques. Fig. 1 illustrates a generated dataset set composed of 1000 data points in 10-dimensional space. Note that this dataset is generated based on the data

generator model described in [5]. As we can see from Fig. 1, there are four clusters that have their own relevant dimensions (e.g., cluster 1 exists in dimensions A_1, A_4, A_8, A_{10}). By relevant dimensions, we mean dimensions that exhibit cluster structure. In our example, there are also three irrelevant dimensions A_3, A_5 and A_7 in which all the data points are sparsely distributed, i.e. no cluster structure exist in these dimensions. Note that the rows in this figure indicate the boundaries of each cluster.

For such an example, a traditional clustering algorithm is likely to fail to find the four clusters. This is because the distance function used by these algorithms gives equal treatment to all dimensions, which are, however, not of equal importance. While feature selection techniques can reduce the dimensionality of the data by eliminating irrelevant attributes such as A_3, A_5 and A_7, there is an enormous risk that they will also eliminate relevant attributes such as A_1. This is due to the presence of many sparse data points in A_1, where a cluster is in fact present. To cope with this problem, new classes of projected clustering have emerged.

Projected clustering exploits the fact that in high-dimensional datasets, different groups of data points may be correlated along different sets of dimensions. The clusters produced by such algorithms are called "projected clusters". A projected cluster is a subset SP of data points, together with a subspace SD of dimensions, such that the points in SP are closely clustered in SD [5]. For instance, the fourth cluster in the dataset presented in Fig. 1 is $(SP_4, SD_4) = (\{x_k, \ldots, x_n\}, \{A_2, A_4, A_6, A_9\})$. Recent research has suggested the presence of projected clusters in many real-life datasets [6].

A number of projected clustering algorithms have been proposed in recent years. Although these previous algorithms have been successful in discovering clusters in different subspaces, they encounter difficulties in identifying very low-dimensional projected clusters embedded in high-dimensional space. Yip et al. [7] observed that current projected clustering algorithms provide meaningful results only when the dimensionalities of the clusters are not much lower than that of the dataset. For instance, some partitional projected clustering algorithms, such as PROCLUS [5] and ORCLUS [8], make use of a similarity function that involves all dimensions in order to find an initial approximation of the clusters. After that, relevant dimensions of each cluster are determined using some heuristics and the clustering is refined based on the relevant dimensions previously selected. Here, it is clear that a similarity function that uses all dimensions misleads the relevant dimensions detection mechanism and adversely affect the performance of these

algorithms. Another example is HARP [9], a hierarchical projected clustering algorithm based on the assumption that two data points are likely to belong to the same cluster if they are very similar to each other along many dimensions. However, when the number of relevant dimensions per cluster is much lower than the dataset dimensionality, such an assumption may not be valid. In addition, some existing projected clustering algorithms, such as PROCLUS [5] and ORCLUS [8], require the user to provide the average dimensionality of the subspaces, which is very difficult to establish in real-life applications.

These observations motivate our effort to propose a novel projected clustering algorithm, called PCKA (Projected Clustering based on the K-means Algorithm). PCKA is composed of three phases: attribute relevance analysis, outlier handling and discovery of projected clusters. Our algorithm is partitional in nature and able to automatically detect projected clusters of very low dimensionality embedded in high-dimensional space, thereby avoiding computation of the distance in the full-dimensional space.

The remainder of this paper is organized as follows. In Section 2, we provide a brief overview of recent projected clustering algorithms and discuss their strengths and weaknesses. Section 3 describes our projected clustering algorithm in detail. Section 4 presents experiments and performance results on a number of synthetic and real datasets. Our conclusion is given in Section 5.

II. Related Work

The problem of finding projected clusters has been addressed in [5]. The partitional algorithm PROCLUS, which is a variant of the K-medoid method, iteratively computes a good medoid for each cluster. With the set of medoids, PROCLUS finds the subspace dimensions for each cluster by examining the neighboring locality of the space near it. After the subspace has been determined, each data point is assigned to the cluster of the nearest medoid. The algorithm is run until the sum of intracluster distances ceases to change. ORCLUS [8] is an extended version of PROCLUS that looks for non-axis-parallel clusters, by using Singular Value Decomposition (SVD) to transform the data to a new coordinate system and select principal components. PROCLUS and ORCLUS were the first to successfully introduce a methodology for discovering projected clusters in high-dimensional spaces, and they continue to inspire novel approaches.

A limitation of these two approaches is that the process of forming the locality is based on

the full dimensionality of the space. However, it is not useful to look for neighbors in datasets with very low-dimensional projected clusters. In addition, PROCLUS and ORCLUS require the user to provide the average dimensionality of the subspace, which also is very difficult to do in real life applications.

In [10], Procopiuc et al. propose an approach called DOC (Density-based Optimal projective Clustering) in order to identify projected clusters. DOC proceeds by discovering clusters one after another, defining a projected cluster as a hypercube with width $2w$, where w is a user-supplied parameter. In order to identify relevant dimensions for each cluster, the algorithm randomly selects a seed point and a small set, Y, of neighboring data points from the dataset. A dimension is considered as relevant to the cluster if and only if the distance between the projected value of the seed point and the data point in Y on the dimension is no more than w. All data points that belong to the defined hypercube form a candidate cluster. The suitability of the resulting cluster is evaluated by a quality function which is based on a user-provided parameter β that controls the trade-off between the number of objects and the number of relevant dimensions. DOC tries different seeds and neighboring data points, in order to find the cluster that optimizes the quality function. The entire process is repeated to find other projected clusters. It is clear that since DOC scans the entire dataset repetitively, its execution time is very high. To alleviate this problem, an improved version of DOC called FastDOC is also proposed in [10].

DOC is based on an interesting theoretical foundation and has been successfully applied to image processing applications [10]. In contrast to previous approaches, (i.e. PROCLUS and ORCLUS), DOC is able to automatically discover the number of clusters in the dataset. However, the input parameters of DOC are difficult to determine and an inappropriate choice by the user can greatly diminish its accuracy. Furthermore, DOC looks for clusters with equal width along all relevant dimensions. In some types of data, however, clusters with different widths are more realistic.

Another hypercube approach called FPC (Frequent-Pattern-based Clustering) is proposed in [11] to improve the efficiency of DOC. FPC replaces the randomized module of DOC with systematic search for the best cluster defined by a random medoid point p. In order to discover relevant dimensions for the medoid p, an optimized adaptation of the frequent pattern tree growth method used for mining itemsets is proposed. In this context, the authors of FPC illustrate the analogy between mining frequent itemsets and discovering dense projected clusters around

random points. The adapted mining technique is combined with FastDOC to discover clusters. However, the fact that FPC returns only one cluster at a time adversely affects its computational efficiency. In order to speed up FPC, an extended version named CFPC (Concurrent Frequent-Pattern-based Clustering) is also proposed in [11]. CFPC can discover multiple clusters simultaneously, which improves the efficiency of the clustering process.

It is shown in [11] that FPC significantly improves the efficiency of DOC/FastDOC and can be much faster than the previous approaches. However, since FPC is built on DOC/FastDOC it inherits some of their drawbacks. FPC performs well only when each cluster is in the form of a hypercube and the parameter values are specified correctly.

A recent paper [9] proposes a hierarchical projected clustering algorithm called HARP (a Hierarchical approach with Automatic Relevant dimension selection for Projected clustering). The basic assumption of HARP is that if two data points are similar in high-dimensional space, they have a high probability of belonging to the same cluster in lower-dimensional space. Based on this assumption, two clusters are allowed to merge only if they are similar enough in a number of dimensions. The minimum similarity and minimum number of similar dimensions are dynamically controlled by two thresholds, without the assistance of user parameters. The advantage of HARP is that it provides a mechanism to automatically determine relevant dimensions for each cluster and avoid the use of input parameters, whose values are difficult to set. In addition to this, the study in [6] illustrates that HARP provides interesting results on gene expression data.

On the other hand, as mentioned in Section 1, it has been shown in [3] that, for a number of common data distribution, as dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point. Based on these results, the basic assumption of HARP will be less valid when projected clusters have few relevant dimensions. In such situations the accuracy of HARP deteriorates severely. This effect on HARP's performance was also observed by Yip et al. in [7].

In order to overcome the limitation encountered by HARP and other projected clustering algorithms, the authors of HARP propose in [7] a semi-supervised approach named SSPC (Semi-Supervised Projected Clustering). This algorithm is partitional in nature and similar in structure to PROCLUS. As in semi-supervised clustering, SSPC makes use of domain knowledge (labeled data points and/or labeled dimensions) in order to improve the quality of a clustering. As reported in [7], the clustering accuracy can be greatly improved by inputting only a small amount of

domain knowledge. However, in some applications, domain knowledge in the form of labeled data points and/or labeled dimensions is very limited and not usually available.

A density-based algorithm named EPCH (Efficient Projective Clustering by Histograms) is proposed in [12] for projected clustering. EPCH performs projected clustering by histogram construction. By iteratively lowering a threshold, dense regions are identified in each histogram. A "signature" is generated for each data point corresponding to some region in some subspace. Projected clusters are uncovered by identifying signatures with a large number of data points [12]. EPCH has an interesting property in that no assumption is made about the number of clusters or the dimensionality of subspaces. In addition to this, it has been shown in [12] that EPCH can be fast and is able to handle clusters of irregular shape. On the other hand, while EPCH avoids the computation of distance between data points in the full-dimensional space, it suffers from the curse of dimensionality. In our experiments [13], we have observed that when the dimensionality of the data space increases and the number of relevant dimensions for clusters decreases, the accuracy of EPCH is affected.

A field that is closely related to projected clustering is subspace clustering. CLIQUE [1] was the pioneering approach to subspace clustering, followed by a number of algorithms in the same field such as ENCLUS [14], MAFIA [15] and SUBCLU [16]. The idea behind subspace clustering is to identify all dense regions in all subspaces, whereas in projected clustering, as the name implies, the main focus is on discovering clusters that are projected onto particular spaces [5]. The outputs of subspace clustering algorithms differ significantly from those of projected clustering [5]. Subspace clustering techniques tend to produce a partition of the dataset with overlapping clusters [5], [9]. The output of such algorithms is very large, because data points may be assigned to multiple clusters. In contrast, projected clustering algorithms produce disjoint clusters with a single partitioning of points [5], [9], [10], [11], [12]. Depending on the application domain, both subspace clustering and projected clustering can be powerful tools for mining high-dimensional data. Since the major concern of this paper is projected clustering, we will focus only on such techniques. Further details and a survey on subspace clustering algorithms and projected clustering algorithms can be found in [17] and [18].

III. THE ALGORITHM PCKA

A. Problem Statement

To describe our algorithm, we will introduce some notation and definitions. Let DB be a dataset of d-dimensional points, where the set of attributes is denoted by $A = \{A_1, A_2, \ldots, A_d\}$. Let $X = \{x_1, x_2, \ldots, x_N\}$ be the set of N data points, where $x_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{id})$. Each x_{ij} $(i = 1, \ldots, N; j = 1, \ldots, d)$ corresponds to the value of data point x_i on attribute A_j. In what follows, we will call x_{ij} a 1-d point. In this paper, we assume that each data point x_i belongs either to one projected cluster or to the set of outliers OUT. Given the number of clusters nc, which is an input parameter, a projected cluster C_s, $s = 1, \ldots, nc$ is defined as a pair (SP_s, SD_s), where SP_s is a subset of data points of DB and SD_s is a subset of dimensions of A, such that the projections of the points in SP_s along each dimension in SD_s are closely clustered. The dimensions in SP_s are called relevant dimensions for the cluster C_s. The remaining dimensions, i.e., $A - SD_s$, are called irrelevant dimensions for the cluster C_s. The cardinality of the set SD_s is denoted by d_s, where $d_s \leq d$ and n_s denotes the cardinality of the set SP_s, where $n_s < N$.

PCKA is focused on discovering axis-parallel projected clusters which satisfy the following properties:

1) Projected clusters must be dense. Specifically, the projected values of the data points along each dimension of $\{SD_s\}_{s=1,\ldots,nc}$ form regions of high density in comparison to those in each dimension of $\{A - SD_s\}_{s=,\ldots,nc}$.
2) The subset of dimensions $\{SD_s\}_{s=1,\ldots,nc}$ may not be disjoint and they may have different cardinalities.
3) For each projected cluster C_s, the projections of the data points in SP_s along each dimension in SD_s are similar to each other according to a similarity function, but dissimilar to other data points not in C_s.

The first property is based on the fact that relevant dimensions of the clusters contain dense regions in comparison to irrelevant ones and such a concept of "density" is comparatively relative across all the dimensions in the dataset. The reason for the second and third properties is trivial. In our clustering process, which is K-means-based, we will use the Euclidean distance in order to measure the similarity between a data point and a cluster center such that only dimensions that contain dense regions are involved in the distance calculation. Hence, the discovered clusters

should have, in general, a concave (near spherical) shape [1].

Note that the algorithm that we propose does not presume any distribution on each individual dimension for the input data. Furthermore, there is no restriction imposed on the size of the clusters or the number of relevant dimensions of each cluster. A projected cluster should have a significant number of selected (i.e., relevant) dimensions with high relevance in which a large number of points are close to each other [1]. To achieve this, PCKA proceeds in three phases:

1) **Attribute relevance analysis**: The goal is to identify all dimensions in a dataset which exhibit some cluster structure by discovering dense regions and their location in each dimension. The underlying assumption for this phase is that, in the context of projected clustering, a cluster should have relevant dimensions in which the projection of each point of the cluster is close to a sufficient number of other projected points (from the whole data set), and this concept of "closeness" is relative across all the dimensions. The identified dimensions represent potential candidates for relevant dimensions of the clusters.
2) **Outlier handling**: Based on the results of the first phase, the aim is to identify and eliminate outlier points from the dataset. Like the majority of clustering algorithms, PCKA considers outliers as points that do not cluster well [5].
3) **Discovery of projected clusters**: The goal of this phase is to identify clusters and their relevant dimensions. The clustering process is based on a modified version of the K-means algorithm in which the computation of distance is restricted to subsets where the data point values are dense. Based on the identified clusters, in the last step of our algorithm we refine the results of phase 1 by selecting the appropriate dimensions of each cluster.

Looking to the design of our algorithm, it is clear that our strategy to identify clusters and their relevant dimensions cannot be viewed as globally optimizing an objective function. In other words, our technique is not similar to "full" iterative clustering algorithms, such as PROCLUS [5] and SSPC [7], which require an objective function to be optimized. The proposed algorithm belongs to the broad category of techniques, such as CLIQUE [1] and EPCH [12], that do not treat projected/subspace clustering as an optimization problem and thus do not admit an explicit

[1]This stipulation ensures that the high relevance of the selected dimensions of each cluster is not due to a random chance. Hence, some degenerative situations in which we can get a large cluster with a low number of relevant dimensions are successfully avoided.

objective function. On the other hand, PCKA follows the general principle of projected clustering and uses various techniques pertaining to minimizing the inter-cluster similarity and maximizing the intra-cluster similarity in the relevant dimensions. To identify projected clusters, PCKA proceeds phase-by-phase and avoids the difficulty of attempting to solve a hard combinatorial optimization problem. The algorithms of these phases are described in the following subsections.

B. Attribute Relevance Analysis

In the context of projected clustering, irrelevant attributes contain noise/outliers and sparse data points, while relevant ones may exhibit some cluster structure [5]. By cluster structure we mean a region that has a higher density of points than its surrounding regions. Such dense region represents the 1-d projection of some cluster. Hence, it is clear that by detecting dense regions in each dimension we are able to discriminate between dimensions that are relevant to clusters and irrelevant ones.

In order to detect densely populated regions in each attribute we compute a sparseness degree y_{ij} for each 1-d point x_{ij} by measuring the variance of its k nearest (1-d point) neighbors $(k - nn)$.

Definition 1: The sparseness degree of x_{ij} is defined as $y_{ij} = \frac{\sum_{r \in p_i^j(x_{ij})} (r - c_i^j)^2}{k+1}$, where $p_i^j(x_{ij}) = \{nn_k^j(x_{ij}) \cup x_{ij}\}$ and $|p_i^j(x_{ij})| = k + 1$. nn_k^j denotes the set of $k - nn$ of x_{ij} in dimension A_j and c_i^j is the center of the set $p_i^j(x_{ij})$, i.e., $c_i^j = \frac{\sum_{r \in p_i^j(x_{ij})} r}{k+1}$.

Intuitively, a large value of y_{ij} means that x_{ij} belongs to a sparse region, while a small one indicates that x_{ij} belongs to a dense region.

Calculation of the k nearest neighbors is, in general, an expensive task, especially when the number of data points N is very large. However, since we are searching for the k nearest neighbors in a one-dimensional space, we can perform the task in an efficient way by pre-sorting the values in each attribute and limiting the number of distance comparisons to a maximum of $2k$ values.

The major advantage of using the sparseness degree is that it provides a relative measure on which the dense regions are more easily distinguishable from sparse regions. On the other hand, when a dimension contains only sparse regions, all the estimated y_{ij} for the same dimension tend to be very large. Our objective now is to determine whether or not dense regions are present in a dimension.

In order to identify dense regions in each dimension, we are interested in all sets of x_{ij} having a small sparseness degree. In the preliminary version of PCKA described in [13], dense regions are distinguished from sparse regions using a user pre-defined density threshold. However, in such an approach an inappropriate choice of the value of the density threshold by the user may affect the clustering accuracy. In this paper, we develop a systematic and efficient way to discriminate between dense and sparse regions. For this purpose we propose to model the sparseness degree y_{ij} of all the 1-d points x_{ij} in a dimension as a mixture distribution. The probability density function (*PDF*) is therefore estimated and the characteristics of each dimension are identified.

Note that, in order to identify dense regions, it might be possible in some cases to fit the original data points in each dimension as a mixture of Gaussian distribution (or another more complex mixture distribution). However, this would unnecessarily limit the generality and thus the applicability of our algorithm. For instance, when a dimension contains only noise/outliers (i.e., data points with sparse distribution), the use of a Gaussian distribution is not the best choice due to its symmetric shape restriction. On the other hand, the sparseness degree is a natural choice for representing the local density of a point in a dimension. We will show that the sparseness degree tends to have a smooth and multi-modal distribution, which more naturally suggests a mixture distribution. These features make our approach suitable to deal with various possible distributions in each dimension. Below, we propose an effective way to identify dense regions and their location in each dimension.

1) PDF *estimation:* Since we are dealing with one-dimensional spaces, estimating the histogram is a flexible tool to describe some statistical properties of the data. For the purpose of clarification, consider the dataset presented in Fig. 1. The histograms of the sparseness degrees calculated for dimensions A_1, A_2, A_3 and A_4 are illustrated in Fig. 2. Here k is chosen to be $\sqrt{N}$ and the calculated y_{ij} are normalized in the interval $]0, 1]$. The histograms presented in Fig. 2 suggest the existence of components with different shape and/or heavy tails, which inspired us to use the gamma mixture model.

Formally, we expect that the sparseness degrees of a dimension d follows a mixture density of the form:

$$G(y) = \sum_{l=1}^{m} \gamma_l G_l(y, \alpha_l, \beta_l) \tag{1}$$

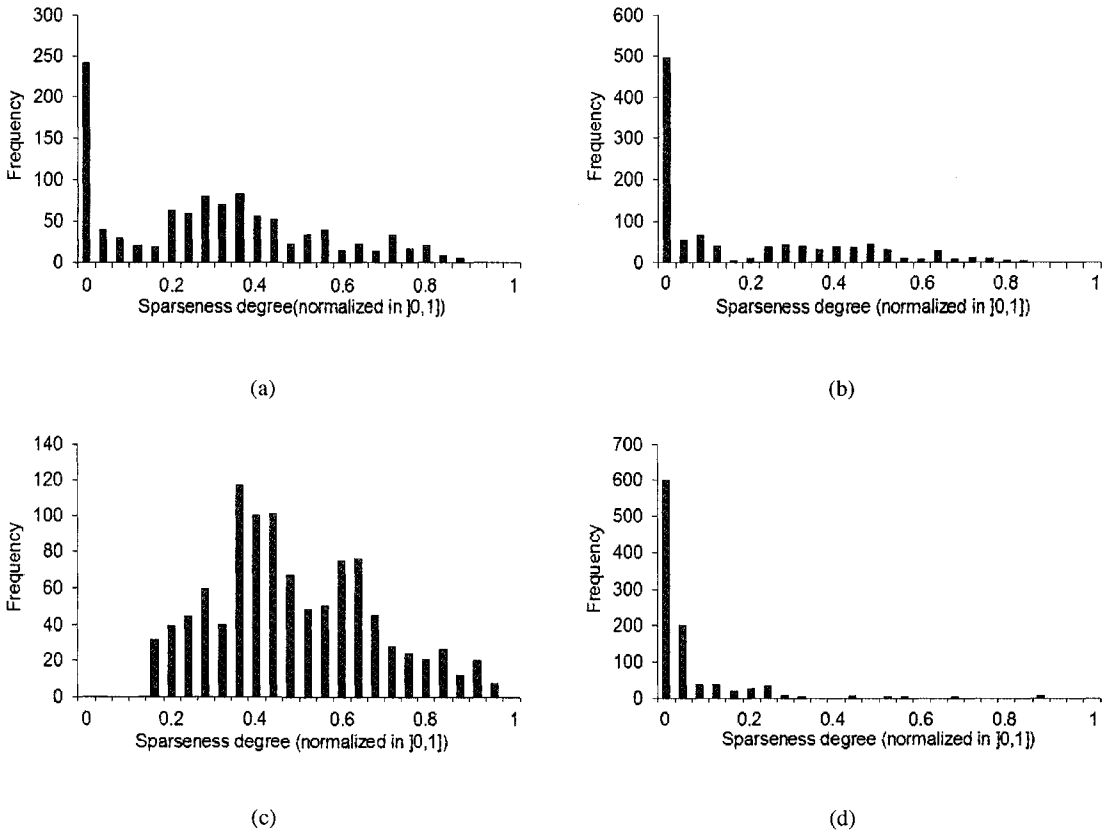


Fig. 2. Histograms of the sparseness degree of: (a) dimension A_1, (b) dimension A_2, (c) dimension A_3 and (d) dimension A_4.

where $G_l(.)$ is the lth gamma distribution with parameters α_l and β_l which represent, respectively, the shape and the scale parameters of the lth component; and $\gamma_l(l = 1, \ldots, m)$ are the mixing coefficients, with the restriction that $\gamma_l > 0$ for $l = 1, \ldots, m$ and $\sum_{l=1}^{m} \gamma_l = 1$. The density function of the lth component is given by

$$G_l(y, \alpha_l, \beta_l) = \frac{\beta_l^{\alpha_l}}{\Gamma(\alpha_l)} y^{\alpha_l - 1} \exp(-\beta_l y) \tag{2}$$

where $\Gamma(\alpha_l)$ is the gamma function given by $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} \exp(-t) dt; \;\; t > 0$.

As mentioned above, each gamma component G_l in equation (2) has two parameters: the shape parameter α_l and the scale parameter β_l. The shape parameter allows the distribution to take on a variety of shapes, depending on its value [19], [20]. When $\alpha_l < 1$, the distribution is highly skewed and is L-shaped. When $\alpha_l = 1$, we get the exponential distribution. In the case

of $\alpha_l > 1$, the distribution has a peak (mode) in $(\alpha_l - 1)/\beta_l$ and skewed shape. The skewness decreases as the value of α_l increases. This flexibility of the gamma distribution and its positive sample space make it particularly suitable to model the distribution of the sparseness degrees.

A standard approach for estimating the parameters of the gamma components G_l is the maximum likelihood technique [21]. The likelihood function is defined as

$$\begin{aligned} L_{G_l}(\alpha_l, \beta_l) &= \prod_{y \in G_l} G_l(y, \alpha_l, \beta_l) \\ &= \frac{\beta_l^{\alpha_l N_l}}{\Gamma^{N_l}(\alpha_l)} \prod_{y \in G_l} y^{\alpha_l - 1} \exp(-\beta_l \sum_{y \in G_l} y) \end{aligned} \tag{3}$$

where N_l is the size of the lth component. The logarithm of the likelihood function is given by

$$\log(L_{G_l}(\alpha_l, \beta_l)) = N_l \alpha_l \log(\beta_l) - N_l \log(\Gamma(\alpha_l)) + (\alpha_l - 1) \sum_{y \in G_l} \log(y) - \beta_l \sum_{y \in G_l} y \tag{4}$$

To find the values of α_l and β_l that maximize the likelihood function, we differentiate $\log(L_{G_l}(\alpha_l, \beta_l))$ with respect to each of these parameters and set the result equal to zero:

$$\frac{\partial}{\partial \alpha_l} \log(L_{G_l}(\alpha_l, \beta_l)) = N_l \log(\beta_l) - N_l \frac{\Gamma'(\alpha_l)}{\Gamma(\alpha_l)} + \sum_{y \in G_l} \log(y) = 0$$

$$\Rightarrow -\log(\beta_l) + \frac{\Gamma'(\alpha_l)}{\Gamma(\alpha_l)} = \frac{1}{N_l} \sum_{y \in G_l} \log(y) \tag{5}$$

and

$$\frac{\partial}{\partial \beta_l} \log(L_{G_l}(\alpha_l, \beta_l)) = \frac{N_l \alpha_l}{\beta_l} - \sum_{y \in G_l} y = 0$$

$$\Rightarrow \beta_l = \frac{N_l \alpha_l}{\sum_{y \in G_l} y} \tag{6}$$

This yields the equation

$$\log(\alpha_l) - \Psi(\alpha_l) = \log(\frac{1}{N_l} \sum_{y \in G_l} y) - \frac{1}{N_l} \sum_{y \in G_l} \log(y) \tag{7}$$

where $\Psi(.)$ is the digamma function given by $\Psi(\alpha) = \frac{\Gamma_l(\alpha)}{\Gamma(\alpha)}$.
The digamma function can be approximated very accurately using the following equation [22]:

$$\Psi(\alpha) = \log(\alpha) - \frac{1}{2\alpha} - \frac{1}{12\alpha^2} + \frac{1}{120\alpha^4} - \frac{1}{252\alpha^6} + \dots \tag{8}$$

The parameter $\hat{\alpha}_l$ can be estimated by solving equation (7) using the Newton-Raphson method. $\hat{\alpha}_l$ is then substituted into equation (6) to determine $\hat{\beta}_l$.

The use of a mixture of gamma distributions allows us to propose a flexible model to describe the distribution of the sparseness degree. To form such a model, we need to estimate m, the number of components, and the parameters for each component. One popular approach to estimating the number of components m is to increase m from 1 to m_max and to compute some particular performance measures on each run, until partition into an optimal number of components is obtained. For this purpose, we implement a standard two-step process. In the first step, we calculate the maximum likelihood of the parameters of the mixture for a range of values of m (from 1 to m_max). The second step involves calculating an associated criterion and selecting the value of m which optimizes the criterion. A variety of approaches have been proposed to estimate the number of components in a dataset [23], [24]. In our method, we use a penalized likelihood criterion, called the Bayesian Information Criterion (BIC). BIC was first introduced by Schwartz [25] and is given by

$$BIC(m) = -2L_m + N_p \log(N) \tag{9}$$

where L is the logarithm of the likelihood at the maximum likelihood solution for the mixture model under investigation and N_p is the number of parameters estimated. The number of components that minimizes $BIC(m)$ is considered to be the optimal value for m.

Typically, the maximum likelihood of the parameters of the distribution is estimated using the EM algorithm [26]. This algorithm requires the initial parameters of each component. Since EM is highly dependent on initialization [27], it will be helpful to perform initialization by mean of a clustering algorithm [27], [28]. For this purpose we implement the FCM algorithm [29] to partition the set of sparseness degrees into m components. Based on such a partition we can estimate the parameters of each component and set them as initial parameters to the EM algorithm. Once the EM algorithm converges we can derive a classification decision about the

Algorithm 1 PDF estimation of the sparseness degrees of each dimension

Input: A_j, m_max, k
Output: $m, \alpha_l, \beta_l, \gamma_l$
Based on Definition 1, compute the sparseness degree y_{ij};
Normalize y_{ij} in the interval]0, 1];
for $m = 1$ to m_max **do**
  if m==1 **then**
    Estimate the parameters of the gamma distribution based on the likelihood formula using equations (6), (7) and (8);
    Compute the value of $BIC(m)$ using equation (9);
  else
    Apply the FCM algorithm as an initialization of the EM algorithm;
    Apply the EM algorithm to estimate the parameters of the mixture using equations (6), (7) and (8);
    Compute the value of $BIC(m)$ using equation(9);
  end if
end for
Select the number of components $\hat{m}$, such that
$\hat{m} = \arg_{min} BIC(m)$;

membership of each sparseness degree in each component. The procedure for estimating the PDF of the sparseness degrees of each dimension is summarized in Algorithm 1.

Let us focus now on the choice of the value of m_max. In our experiments on different datasets, we observed that when a dimension contains only sparse regions the sparseness degrees are well fitted, in general, by one gamma component. When a dimension contains a mix of dense and sparse regions the sparseness degrees are well fitted, in general, by two gamma components. This can be explained by the fact that the sparseness degree provides a relative measure in which sparse regions are easily distinguishable from dense regions. Such a concept of relativity between the values of the sparseness degree makes the choice of m_max fairly simple. Based on this, we believe that setting $m\_max = 3$ is, in general, a practical choice. The reader should be aware, however, that the choice of m_max is not limited to 3 and the user can set other values. However, setting values of $m\_max > 3$ can unnecessarily increase the execution time. Hence, we suggest using $m\_max = 3$ as a default value. The estimated $PDFs$ of the sparseness degrees of A_1, A_2, A_3 and A_4 are illustrated in Fig. 3.

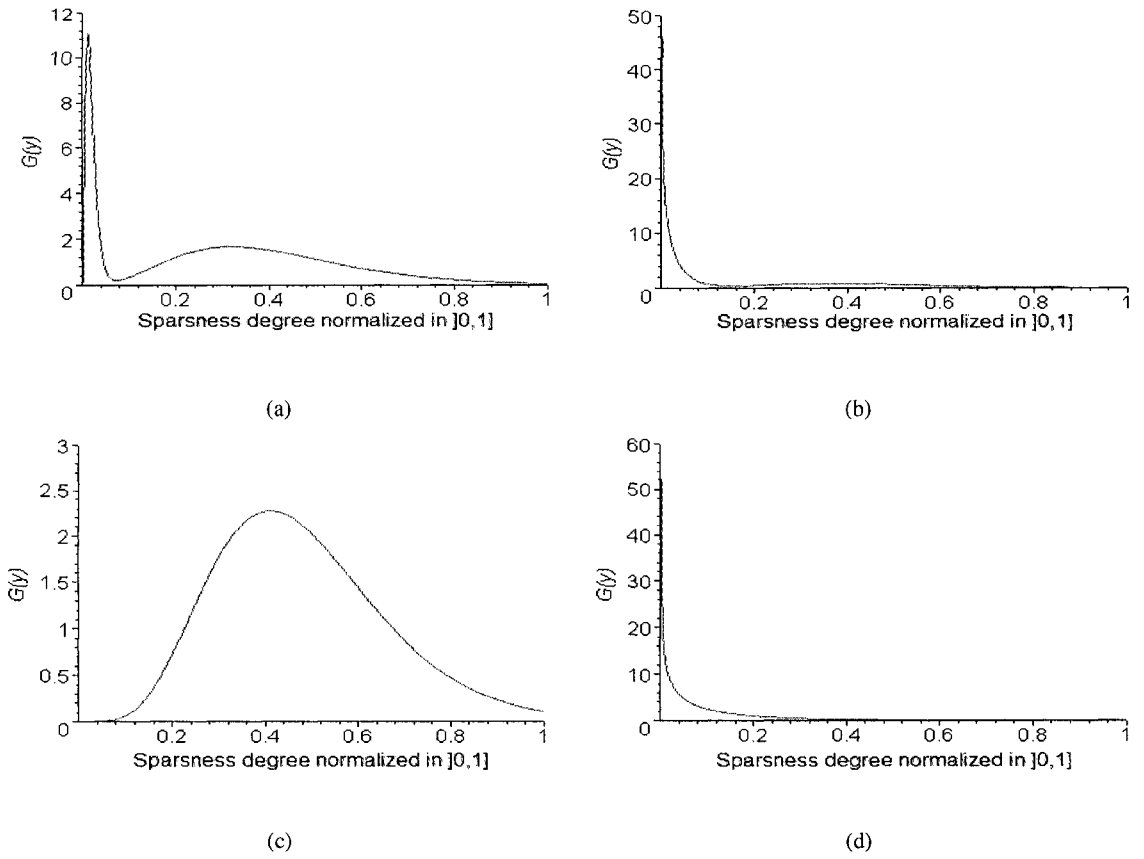


Fig. 3. $PDFs$ of the sparseness degrees of: (a) dimension A_1, (b) dimension A_2, (c) dimension A_3 and (d) dimension A_4. The sparseness degrees of A_1 and A_2 are well fitted by two gamma components, where the first component represents a dense regions while the second one represents a sparse regions. In the case of A_3 and A_4 the estimated $PDFs$ contain only one gamma component, which represents a sparse regions for A_3 and a dense region for A_4.

2) Detection of dense regions: Once the PDF of the sparseness degrees y_{ij} of each dimension is estimated, we turn to the problem of how to detect dense regions and their location in a dimension. For this purpose, we make an efficient use of the properties of the estimated PDF. As illustrated in Fig. 3, the locations of the components which represent dense regions are close to zero in comparison to those that represent sparse regions. Based on this observation, we propose a method to examine the location of each of the components in all the dimensions in order to find a typical value that best describes the sparseness degrees of each component. Let loc_q denote the location of the component q, where $q = 1, \ldots, m_{total}$ and m_{total} is the total number of all the components over all the dimensions in the dataset. Intuitively, a large value

of loc_q means that the component q corresponds to a sparse regions, while a small one indicates that this component corresponds to a dense regions.

The most commonly used measures of location for univariate data are the mean, the mode and the median. The most appropriate measure in our case is the median, due to the variable shape of the gamma distribution. For instance, when the shape parameter $\alpha_l < 1$, the distribution is L-shaped with a heavy tail. In this case, the distribution has no mode and the extreme value present in the tail affects the computation of the mean. When $\alpha_l > 1$, the distribution is skewed, which implies that the mean will be pulled in the direction of the skewness. In all these situations, the median provides a better estimate of location than does the mean or the mode, because skewed distribution and extreme values do not distort the median, whose computation is based on ranks.

In order to identify dense regions in each dimension, we are interested in all components with small values of loc_q. We therefore propose to use the MDL principle [30] to separate small and large values of loc_q. The MDL-selection technique that we use in our approach is similar to the MDL-pruning technique described in [1]. The authors in [1] use this technique to select subspaces with large coverage values and discard the rest. We want to use the MDL principle in similar way but in our case we want to select small values of loc_q and their corresponding components. The fundamental idea behind the MDL principle is to encode the input data under a given model and select the encoding that minimizes the code length [30].

Let $LOC = \{loc_1, \ldots, loc_q, \ldots, loc_{m_{total}}\}$ be the set of all loc_q values for each dimension in the entire dataset. Our objective is to divide LOC into two groups E and F, where E contains the highest values of loc_q and F, the low values. To this end, we implement the model described in Algorithm 2. Based on the result of this partitioning process, we selected the components corresponding to each of the loc_q values in F. In Fig. 4, we use the dataset described in Section 1 to provide an illustration of this approach. As shown in this figure, there is a clear cutoff point which allows us to select the components with the smallest loc_q values.

Based on the components selected by means of Algorithm 2, we can now definitively discriminate between dense and sparse regions in all the dimensions in the dataset.

Definition 2: Let $z_{ij} \in \{0, 1\}$, where z_{ij} is a binary weight.
If y_{ij} belongs to one of the selected components then $z_{ij} = 1$ and x_{ij} belong to a dense region; else z_{ij}= 0 and x_{ij} belongs to a sparse region.

From Definition 2, we obtain a binary matrix $Z_{(N*d)}$ which contains the information on whether

Algorithm 2 MDL-based selection technique

1: **Input**: LOC

2: **Output**: E, F

3: Sort the values in LOC in descending order;

4: **for** $q = 2$ to m_{total} **do**

5: $E = \{loc_i,\ i = 1, \ldots, q\}$ and the mean of E is μ_E;

6: $F = \{loc_j,\ j = q, \ldots, m_{total}\}$ and the mean of F is μ_F

7: Calculate the code length

$CL(q) = \log_2(\mu_E) + \sum_{loc_q \in E} \log_2(|loc_q - \mu_E|) + \log_2(\mu_F) + \sum_{loc_q \in F}(|loc_q - \mu_F|)$

8: **end for**

9: Select the best partitioning given by the pair (E, F), i.e., the one for which the corresponding $CL(q)$ is the smallest;

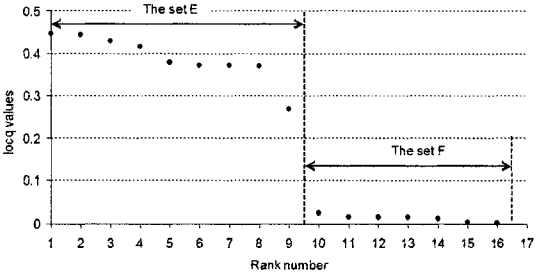


Fig. 4. Partitioning of the set LOC into two sets E and F.

each data point falls into a dense region of an attribute. For example, Fig. 5 illustrates the matrix Z for the data used in the example in Section 1.

It is clear that the computation of z_{ij} depends on the input parameter k (the number of nearest neighbors of 1-d point). Although it is difficult to formulate and obtain optimal values for this parameter, it is possible for us to propose guidelines for its estimation. In fact, the role of the parameter k is intuitively easy to understand and it can be set by the user based on specific knowledge of the application. In general, if k is too small, the sparseness degrees y_{ij} are not meaningful, since a 1-d point in a dense region might have a similar sparseness degree value to a 1-d point in a sparse region. Obviously, the parameter k is related to the expected minimum cluster size and should be much smaller than the number of objects N in the data. To gain a

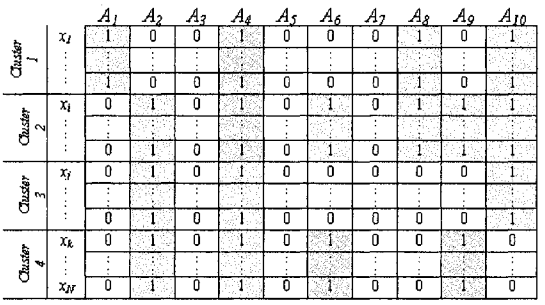


Fig. 5. The matrix $Z_{(N*d)}$.

Algorithm 3 Phase 1 of PCKA

1: **Input**: DB, k, m_max
2: **Output**: Z
3: $LOC \leftarrow \emptyset$;
4: Choose m_max;
5: **for** $j = 1$ to d **do**
6: Apply Algorithm 1(A_j, m_max, k, m, α_l, β_l, γ_l);
7: Apply EM to partition the sparseness degree in dimension j into m components;
8: **for** $i = 1$ to m **do**
9: $LOC \leftarrow LOC \cup median(component_i)$;
10: **end for**
11: **end for**
12: Apply Algorithm 2(LOC, E, F);
13: Based on the loc_q values in F, select the components which represent dense regions;
14: Based on Definition 2 compute the matrix Z;

clear idea of the sparseness of the neighborhood of a point we have chosen to set $k = \sqrt{N}$ in our implementation. Phase 1 of PCKA is summarized in Algorithm 3.

In summary, phase 1 of PKCA allows attribute relevance analysis to be performed in a systematic way without imposing any restriction on the distribution of the original data points, which is actually an advantage. Indeed, the sparseness degree is an effective feature indicating whether a point is more likely situated in a sparse or a dense region in a dimension. The gamma mixture model is used due to its shape flexibility, which allows it to provide valuable information about the distribution of points and the location of dense and sparse regions,

when considering each individual dimension. The MDL pruning technique is employed to automatically discriminate between dense and sparse regions over all the dimensions considered together. We believe that this combination of the three techniques makes our attribute relevance analysis approach particularly suitable for performing projected clustering, as our experiments will illustrate.

Finally, we should point out that our approach to detecting gamma components that correspond to dense regions is based on the assumption that the medians of all the gamma components of all dimensions are comparable. We have made this assumption since we are performing our task in a systematic way. However, such an assumption could have some negative influence on the results if the dataset contains clusters with very different densities. Specifically, clusters with very low density can be confused with sparse regions that do not contain any cluster structure. As a result, dimensions in which such clusters exist will probably not be identified as relevant. This limitation arises from the fact that we attempt to separate regions in all individual dimensions into two classes: "dense" and "sparse". To cope with this limitation, one possible solution is to extend our approach to deal with more general cases by adopting the local-density-based strategy developed in [31]. We will consider this issue in our future work.

C. Outlier Handling

In addition to the presence of irrelevant dimensions, high-dimensional data are also characterized by the presence of outliers. Outliers can be defined as a set of data points that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data [32]. Most of the clustering algorithms, including PCKA, consider outliers as points that are not located in clusters and should be captured and eliminated because they hinder the clustering process.

A common approach to identify outliers is to analyze the relationship of each data point with the rest of the data, based on the concept of proximity [33], [34]. However, in high-dimensional spaces, the notion of proximity is not straightforward [3]. To overcome this problem, our outlier handling mechanism makes an efficient use of the properties of the binary matrix Z. In fact, the matrix Z contains useful information about dense regions and their locations in the dataset DB. It is obvious that outliers do not belong to any of the identified dense regions; they are located in sparse regions in DB. Such points are highly dissimilar from the rest of the dataset and can be identified by using the binary weights z_{ij}. For this purpose, we use binary similarity

coefficients to measure the similarity between binary data points z_i for $(i = 1, \ldots, N)$ in the matrix Z.

Given two binary data points z_1 and z_2, there are four fundamental quantities that can be used to define similarity between the two [35]: $a = |z_{1j} = z_{2j} = 1|, b = |z_{1j} = 1 \wedge z_{2j} = 0|, c = |z_{1j} = 0 \wedge z_{2j} = 1|$ and $d = |z_{1j} = z_{2j} = 0|$, where $j = 1, \ldots, d$. One commonly used similarity measure for binary data is the Jaccard coefficient. This measure is defined as the number of variables that are coded as 1 for both states divided by the number of variables that are coded as 1 for either or both states. In our work, we require a similarity measure that can reflect the degree of overlap between the binary data points z_i in the identified dense regions in the matrix Z. Since dense regions are encoded by 1 in the matrix Z, we believe that the Jaccard coefficient is suitable for our task because it considers only matches on 1's to be important. The Jaccard coefficient is given as:

$$JC(z_1, z_2) = \frac{a}{a + b + c} \tag{10}$$

The Jaccard coefficient has values between 0 (not at all similar) and 1 (completely similar). A pair of points is considered similar if the estimated Jaccard coefficient between them exceeds a certain threshold ε. In all our experiments on a number of datasets, we observed that setting $\varepsilon = 0.70$, for the task of our study, represents an acceptable degree of similarity between two binary vectors. Our outlier handling mechanism is based on the following definition:

Definition 3: Let λ be a user-defined parameter, z_i a binary vector from Z, and $SV_i = \{z_j |\ JC(z_i, z_j) < \varepsilon \ and \ j = 1, \ldots, N\}$ the set of similar binary vectors of z_i. A data point x_i is an outlier with respect to parameters λ and ε if $|SV_i| < \lambda$.

The above definition has intuitive appeal since in essence it exploits the fact that in contrast to outliers, points which belong to dense regions (clusters) generally have a large number of similar points. Based on this, the value of the parameter λ should not be larger than the size of the cluster containing the ith data point x_i and should be much smaller than N, the size of the dataset DB. Hence, setting $\lambda \approx \sqrt{N}$ is, in general, a reasonable choice. On the other hand, it is clear that when all of the binary weights for a binary data point z_i in the matrix Z are equal to zero, the related data point x_i is systematically considered as an outlier because it does not belong to any of the discovered dense regions.

The identified outliers are discarded from DB and stored in the set OUT, while their corre-

Algorithm 4 Phase 2 of PCKA

Input: DB, Z, ε, λ
Output: RDB, T, OUT
$OUT \leftarrow \emptyset$;
Let *count* be a table of size N;
for $i = 1$ to N **do**
    $count[i] \leftarrow 0$;
end for
for $i = 1$ to N **do**
    if $\sum_{j=1}^{d} z_{ij} == 0$ **then**
        $OUT \leftarrow OUT \cup \{x_i\}$;
    else
        for $j = i + 1$ to N **do**
            Estimate the number of similar binary vector of z_i and z_j:
            if $JC(z_i, z_j) > \varepsilon$ **then**
                $count[i] \leftarrow count[i] + 1$;
                $count[j] \leftarrow count[j] + 1$;
            end if
        end for
        if $count[i] < \lambda$ **then**
            $OUT \leftarrow OUT \cup \{x_i\}$;
        end if
    end if
end for
$RDB \leftarrow DB - OUT$;
Based on RDB and OUT extract T from Z;

sponding rows are eliminated from the matrix Z. Thus Phase 2 of PCKA yields a reduced data set RDB with size $N_r = N - |OUT|$ and its new associated matrix of binary weights $T_{(N_r * d)}$. Our method for eliminating outliers is described in Algorithm 4.

D. Discovery of Projected Clusters

The main focus of Phase 3 of PCKA is to identify projected clusters. The problem of finding projected clusters is two-fold: we must discover the clusters and find the appropriate set of

dimensions in which each cluster exists [5]. To tackle this "chicken-and-egg" problem [9] we proceed in two steps:

1) In the first step, we cluster the data points based on the K-means algorithm, with the computation of distance restricted to subsets of dimensions where object values are dense.
2) Based on the clusters obtained in the first step, the second step proceeds to select the relevant dimensions of the identified clusters by making use of the properties of the binary matrix T.

In the following, we describe each step in detail.

As mentioned above, our goal in the first step is to identify cluster members. For this purpose, we propose a modification to the basic K-means algorithm. The standard K-means assumes that each cluster is composed of objects distributed closely around its centroid. The objective of the K-means is thus to minimize the squared distance between each object and the centroid of its cluster [9]. However, as can be expected from the discussion in Section 1, this is not an effective approach with high-dimensional data, because all the dimensions are involved in computing the distance between a point and the cluster center.

The problem described above can be addressed by modifying the distance measure, making use of the dimension relevance information identified in phase 1 of PCKA. More precisely, in order to cluster the data points in a more effective way, we modify the basic K-means by using a distance function that considers contributions only from relevant dimensions when computing the distance between a data point and the cluster center. In concrete terms, we associate the binary weights t_{ij} $(i = 1, \ldots, N_r; j = 1, \ldots, d)$ in matrix T to the Euclidian distance. This makes the distance measure more effective because the computation of distance is restricted to subsets (i.e., projections) where the object values are dense. Formally, the distance between a point x_i and the cluster center v_s $(s = 1, \ldots, nc)$ is defined as

$$dist(x_i, v_s) = \sqrt{\sum_{j=1}^{d} t_{ij} \times (x_{ij} - v_{sj})^2} \tag{11}$$

In this particular setting, our modified K-means algorithm will allow two data points with different relevant dimensions to be grouped into one cluster if the two points are close to each other in their common relevant dimension(s). Each cluster obtained will have dimensions in which there are a significant number of cluster members which are close to each other. These

clusters are actually already projected clusters and we have adopted a straightforward strategy to detect relevant dimensions for each cluster.

Specifically, we make use of the density information stored in matrix T to determine how well a dimension contributes to the formation of the obtained clusters. In fact, the sum of the binary weights of the data points belonging to the same cluster over each dimension gives us a meaningful measure of the relevance of each dimension to the cluster. Based on this observation, we propose a relevance index W_{sj} for each dimension in cluster C_s. The index W_{sj} for the dimension j ($j = 1, \ldots, d$) in cluster C_s is defined as follows:

$$W_{sj} = \frac{\sum_{t_i \in C_s} t_{ij}}{|C_s|} \tag{12}$$

W_{sj} represent the percentage of the points in the cluster C_s who have the dimension j as their relevant dimension.

Definition 4: Let $\delta \in]0, 1]$. A dimension A_j is considered δ-relevant for the cluster C_s if $W_{sj} > \delta$.

In the above definition, δ is a user-defined parameter that controls the degree of relevance of the dimension A_j to the cluster C_s. It is clear that the larger the value of the relevance index, the more relevant the dimension to the cluster. Based on this property, it is possible to perform a more general analysis on $\{W_{sj} | \ j = 1, \ldots, d\}$ or even on $\{W_{sj}\}_{s=1,\ldots,nc;\ j=1,\ldots,d}$ to automatically determine relevant dimensions. Currently, it is still not clear whether this advantage is significant. On the other hand, since W_{sj} is a relative measure, it is not difficult to choose an appropriate value for δ. In our experiments, we set $\delta = 0.8$ since it is a good practical choice to ensure a high degree of relevance. Adopting the simple thresholding approach also better illustrates the contribution of attribute relevance analysis (i.e., phase 1) to PCKA. Phase 3 of our algorithm is summarized in Algorithm 5.

IV. Empirical Evaluation

In this section we devise a series of experiments designed to evaluate the suitability of our algorithm in terms of:

1) **Accuracy**: the aim is to test whether our algorithm, in comparison with other existing approaches, is able to correctly identify projected clusters.

Algorithm 5 Phase 3 of PCKA

Input: RDB, T, nc, δ
Output: $v_s, U_{(N_r*nc)}, SD_s$
{v_s: cluster centers where $s = 1, \ldots, nc$.
$U_{(N_r*nc)}$: matrix of the membership degrees of each data point in each cluster.
SD_s: set of the relevant dimensions of each cluster.}
Choose the cluster centers v_s^0 $(s = 1, \ldots, nc)$ randomly from RDB;
repeat
  Compute the membership matrix $U_{(N_r*nc)}$:
  for $i = 1$ to N_r **do**
    for $j = 1$ to nc **do**
      if $dist(x_i, v_s) < dist(x_i, v_j)$ **then**
        $u_{ij} = 0$;
      else
        $u_{ij} = 1$;
      end if
    end for
  end for
  Compute the cluster center:
  $v_s^1 = \frac{\sum_{i=1}^{N_r}(u_{is} \times t_i \times x_i)}{\sum_{i=1}^{N_r} u_{is}}$ $(s = 1, \ldots, nc)$;
until convergence, i.e., no change in centroid coordinates;
Based on Definition 4, detect the set SD_s of relevant dimensions for each cluster C_s;

2) **Efficiency**: the aim is to determine how the running time scales with 1) the size and 2) the dimensionality of the dataset.

We compare the performance of PCKA to that of SSPC [7], HARP [9] , PROCLUS [5] and FASTDOC [10]. The evaluation is performed on a number of generated data sets with different characteristics. Furthermore, experiments on real data sets are also presented. All the experiments reported in this section were run on a PC with Intel Core 2 Duo CPU of 2.4GHz and 4GB RAM.

A. Performance Measure

A number of new metrics for comparing projected clustering algorithms and subspace clustering algorithms were recently proposed in [36]. The performance measure used in our paper

is the Clustering Error (CE) distance for projected/subspace clustering. This metric performs comparisons in a more objective way since it takes into account the data point groups and the associated subspace simultaneously. The CE distance has been shown to be the most desirable metric for measuring agreement between partitions in the context of projected/subspace clustering [36]. A deeper investigation of the properties of the CE distance and other performance measures for projected/subspace clustering can be found in [36].

Assume that GP is a collection $\{C_1, \ldots, C_s, \ldots, C_{nc}\}$ of nc generated projected clusters and RP is a collection $\{C'_1, \ldots, C'_s, \ldots, C'_{nc}\}$ of nc real projected clusters. To describe the CE distance, we need to define the union of projected clusters. Let U denote the union of the projected clusters in GP and RP. Formally, $U = U(GP, RP) = supp(GP) \bigcup supp(RP)$, where $supp(GP)$ and $supp(RP)$ are the support of clustering GP and RP, respectively. The support of clustering RP is defined as $supp(GP) = \bigcup_{s=1,\ldots,nc} supp(C_s)$, where $supp(C_s)$ is the support of projected cluster C_s, given by $supp(C_s) = \{x_{ij} | x_i \in SP_s \wedge A_j \in SD_s\}$. Let $M = (m_{ij})$ denote the confusion matrix, in which m_{ij} represents the number of 1-d points shared by the projected clusters C_s and C'_s, i.e., $m_{ij} = |supp(C_i) \bigcap supp(C'_j)|$. The matrix M is transformed, by using of the Hungarian method [37], in order to find a permutation of cluster labels such that the sum of the diagonal elements of M is maximized. D_{max} denotes this maximized sum. The generalized CE distance for projected/subspace clustering is given by

$$CE(RP, GP) = \frac{|U| - D_{max}}{|U|} \tag{13}$$

The value of CE is always between 0 and 1. The more similar the two partitions GP and RP, the smaller the CE value. When GP and RP are identical, the CE value will be zero.

B. Synthetic Data Generation Method

Since our focus is on axis-parallel clusters, we used the data generator model described in the PROCLUS paper [5] in order to simulate various situations. This data generation model has been used by most researchers in their studies [7], [9], [10], [11], [12] to evaluate the performance of projected clustering algorithms. The parameters used in synthetic data generation are the size of the dataset N; the number of clusters nc; the dataset dimensionality d; the average cluster dimensionality l_{real}; the domain of the values of each dimension $[min_j, max_j]$; the standard

deviation value range $[sdv_{min}, sdv_{max}]$, which is related to the distribution of points in each cluster; and the outlier percentage OP. Using these parameters, clusters and their associated subspaces are created randomly. Projected values of cluster points are generated according to the normal distribution in their relevant dimension, with the mean randomly chosen from $[min_j, max_j]$ and the standard deviation value from $[sdv_{min}, sdv_{max}]$. For irrelevant dimensions and outliers, the dimensional values are randomly selected from $[min_j, max_j]$

C. Parameters for Comparing Algorithms

As mentioned above, we compared the performance of PCKA to that of SSPC, HARP, PROCLUS and FASTDOC. In all the experiments on synthetic datasets, the number of clusters nc was set to the true number of projected clusters used to generate the datasets. PROCLUS requires the average cluster dimensionality as a parameter; it was set to the true average cluster dimensionality. Several values were tried for the parameters of FastDOC and SSPC, following the suggestions in their respective papers, and we report results for the parameter settings that produced the best results. HARP requires the maximum percentage of outliers as a parameter; it was set to the true percentage of outliers present in the datasets. In order to obtain satisfactory results and avoid initialization bias, non-deterministic algorithms such as SSPC, PROCLUS and FastDOC were run more than 10 times on each dataset and we consider only the best result for each of them. In all of the experiments, SSPC was run without any semi-supervision. For PCKA, in all the experiments we set $k = \lambda = \sqrt{N}$, $\epsilon = 0.7$ and $\delta = 0.8$.

D. Quality of Results

The performance of projected clustering algorithms is primarily affected by the average cluster dimensionality and the amount of outliers in the dataset. The main goal of the experiments presented in this subsection was to evaluate the capability of projected clustering algorithms to correctly identify projected clusters in various situations.

1) Robustness to the average cluster dimensionality: The main concern of the first set of experiments was to analyze the impact of cluster dimensionality on the quality of clustering. For this purpose, we generated sixteen different datasets with $N = 3000$ data points, number of dimensions $d = 100$, $min_j = -100$ and $max_j = 100$. In each dataset there were 5 clusters with sizes varying between 10% of N to 30% of N. For each relevant dimension of a cluster, the

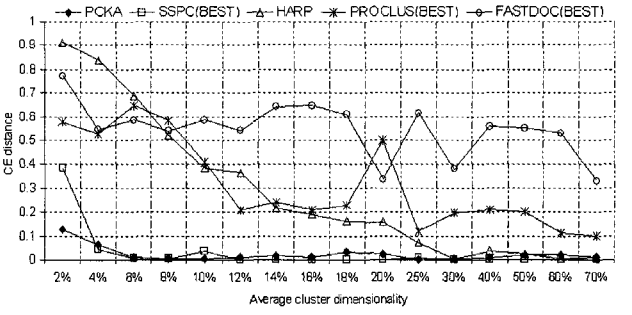


Fig. 6. CE distance between the output of each of the five algorithms and the true clustering.

values of sdv_{min} and sdv_{max} were set to 1% and 6% of the global standard deviation on that dimension, respectively. The average cluster dimensionality varied from 2% to 70% of the data dimensionality d. Since our goal in this first set of experiments was to analyze the impact of cluster dimensionality, no outliers were generated. Therefore, the outlier detection mechanism of PCKA, SSPC and PROCLUS was disabled. For FastDOC, since we could not disable its outlier elimination option, we chose the results with the highest accuracy after several runs. Fig. 6 illustrates the CE distance between the output of each of the five algorithms and the true clustering.

The values of the CE distance in Fig. 6 invite several comments.

PCKA is able to achieve highly accurate results and its performance is generally consistent. As we can see from Fig. 6, PCKA is more robust to variation of the average cluster dimensionality than the other algorithms. For instance, when the average cluster dimensionality is very low (l_{real} = 2% of d), which is a difficult case, only PCKA yields an acceptable results. The experiments show that our algorithm is able to detect clusters and their relevant dimensions accurately in various situations. PCKA successfully avoids the selection of irrelevant dimensions in all the datasets used in these experiments. This can be explained by the fact that PCKA starts by identifying dense regions and their locations in each dimension, enabling it to restrict the computation of the distance to subsets of dimensions where the projected values of the data points are dense.

SSPC encounters difficulties when the average cluster dimensionality is low as 2% of d. In other situations, the best results of SSPC are similar to those of PCKA. SSPC provides accurate

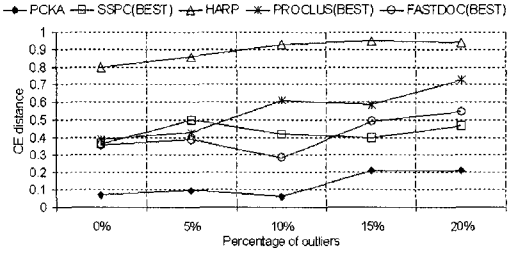
results and is able to correctly identify projected clusters. This may be due to the fact that SSPC makes use of an objective function that combines data point clustering and dimension selection into a single optimization problem [7].

HARP performs well when $l_{real} \geq 30\%$ of d, displaying performance comparable to that of PCKA and SSPC. On the other hand, when $l_{real} < 30\%$ of d, the performance of HARP is affected and its results are less competitive with those of PCKA and SSPC. We have observed that when $l_{real} < 20\%$ of d, HARP encounters difficulties in correctly identifying clusters and their relevant dimensions. When $l_{real} \in [20\% of\ d,\ 30\% of\ d]$, HARP clusters the data points well but tends to include many irrelevant dimensions, which explains the values of the CE distance in Fig. 6. This can be attributed to the fact that datasets with low-dimensional projected clusters mislead HARP's dimension selection procedures. In such situations, the basic assumption of HARP - i.e., that if two data points are similar in high-dimensional space, they have a high probability of belonging to the same cluster in lower-dimensional space - becomes less valid.

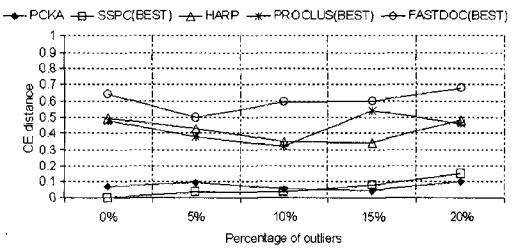
The results of PROCLUS are less accurate than those given by PCKA and SSPC. When datasets contain projected clusters of low dimensionality, PROCLUS performs poorly. When $l_{real} \geq 40\%$ of d, we have observed that PROCLUS is able to cluster the data points well but its dimension selection mechanism is not very accurate because it tends to include some irrelevant dimensions and discard some relevant ones. This behavior of PROCLUS can be explained by the fact that its dimension selection mechanism, which is based on a distance calculation that involves all dimensions by detecting a set of neighboring objects to a medoid, severely hampers its performance.

As we can see from Fig. 6, FastDOC encounters difficulties in correctly identifying projected clusters. In our experiments, we have observed that although the dimensionality of projected clusters is high, (i.e., $l_{real} = 70\%$ of d), FastDOC tends to select a small subset of relevant dimensions for some clusters. All the experiments reported here seem to suggest that in the presence of relevant attributes with different standard deviation values, FastDOC's assumption that projected clusters take the form of a hypercube appears to be less valid. In such situations, it is difficult to set correct values for the parameters of FastDOC.

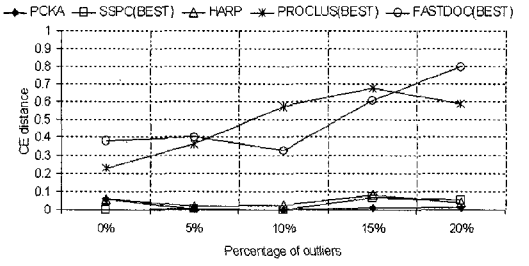
2) Outlier immunity: The aim of this set of experiments was to test the effect of the presence of outliers on the performance of PCKA in comparison to SSPC, HARP, PROCLUS and FastDOC. For this purpose, we generated three groups of datasets, each containing five datasets with



(a) Datasets with $l_{real} = 2\%$ of d.



(b) Datasets with $l_{real} = 15\%$ of d.



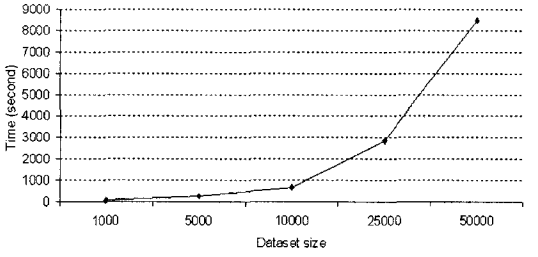
(c) Datasets with $l_{real} = 30\%$ of d.

Fig. 7. Immunity to outliers.

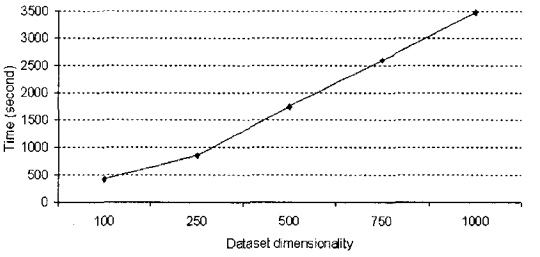
$N = 1000$, $d = 100$ and $nc = 3$. In each dataset in a group, the percentage of outliers varied from 0% to 20% of N. We set $l_{real} = 2\%$ of d for the first group, $l_{real} = 15\%$ of d for the second and $l_{real} = 30\%$ of d for the third. In all the datasets in the three groups, sdv_{min} and sdv_{max} were set to 1% and 6% of the global standard deviation, respectively. Fig. 7 illustrates the CE distance for the five algorithms.

As we can see from the figure, PCKA display consistent performance, as was observed for the first set of experiments on datasets without outliers. In difficult cases, (i.e., when $l_{real} = 2\%$ of d), PCKA provides much better results than all the other algorithms. All the results reported in Fig. 7 suggest that PCKA is less sensitive to the percentage of outliers in datasets. In addition to this variations in the average cluster dimensionality in the presence of different percentages of outliers have no major impact on the performance of PCKA. This can be explained by the fact that the outlier handling mechanism of PCKA makes an efficient use of the density information stored in the matrix Z, giving it high outlier immunity.

The performance of SSPC is greatly affected when $l_{real} = 2\%$ of d, with different values of OP. When $l_{real} \geq 4\%$ of d, the performance of SSPC is greatly improved and is not much



(a) Scalability of PCKA w.r.t. the dataset size.



(b) Scalability of PCKA w.r.t. the dataset dimensionality.

Fig. 8. Scalability experiments.

affected by the percentage of outliers in the datasets. HARP is less sensitive to the percentage of outliers in datasets when $l_{real} \geq 30\%$ of d. Otherwise, (i.e., when $l_{real} < 30\%$ of d, as in the case illustrated in Fig. 7), the performance of HARP is severely affected.This behavior of HARP is consistent with the analysis given in the previous subsection. On the other hand, we have observed in our experiments that HARP is sensitive to its maximum outlier percentage parameter. An incorrect choice of the value of this parameter may affect the accuracy of HARP.

Fig. 7 illustrate that PROCLUS and FastDOC encounter difficulties in correctly identifying projected clusters. PROCLUS tends to classify a large number of data points as outliers. Similar behavior of PROCLUS was also observed in [8] and [9]. The same phenomenon occurs for FastDOC. We found that FastDOC performs well on datasets that contain dense projected clusters with the form of a hypercube.

E. Scalability

In this subsection, we study the scalability of PCKA with increasing data set size and dimensionality. In all of the following experiments, the quality of the results returned by PCKA was similar to that presented in the previous subsection.

Scalability with the dataset size: Fig. 8(a) shows the results for scalability with the size of the dataset. In this experiment we used 50-dimensional data and varied the number of data points from 1000 to 100000. There were 4 clusters with $l_{real} = 12\%$ of d and 5% of N were generated as outliers. As we can see from Fig. 8(a), PCKA scales quadratically with increase in dataset size. In all our experiments, we observed that PCKA is faster than HARP. For $N \leq 25000$, the

execution time of PCKA is comparable to that of SSPC and PROCLUS when the time used for repeated runs is also included.

Scalability with dimensionality of the dataset: In Fig. 8(b) we see that PCKA scales linearly with the increase in the data dimension. The results presented are for data sets with 5000 data points grouped in 4 clusters, with $l_{real} = 12\%$ of d and 5% of N generated as outliers. The dimensionality of the data sets varies from 100 to 1000. As in the scalability experiments w.r.t. the data set size, the execution time of PCKA is usually better than that of HARP and comparable to those of PROCLUS and SSPC when the time used for repeated runs is also included.

F. Experiments on Real-World Data

In addition to our experiments on synthetic data, the suitability of our algorithm was also tested on three real-world data sets. A simple illustration of each of these is given bellow.

Wisconsin Diagnostic Breast Cancer Data (WDBC): This data can be obtained from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). The set contains 569 samples, each with 30 features. The samples are grouped into two clusters: 357 samples for benign and 212 for malignant.

Saccharomyces Cerevisiae Gene Expression Data (SCGE): This data set, available from http://cs.wellesley.edu/~btjaden/CORE, represents the supplementary material used in Brian Tjaden's paper [38]. The Saccharomyces Cerevisiae data contains the expression level of 205 genes under 80 experiments. The data set is presented as a matrix. Each row corresponds to a gene and each column to an experiment. The genes are grouped into four clusters.

Multiple Features Data (MF): This data set is available from the UCI Machine Learning Repository. The set consists of features of handwritten numerals ("0" – "9") extracted from a collection of Dutch utility maps. 200 patterns per cluster (for a total of 2,000 patterns) have been digitized in binary images. For our experiments, we used five feature sets (files):

1) mfeat-fou: 76 Fourier coefficients of the character shapes;
2) mfeat-fac: 216 profile correlations;
3) mfeat-kar: 64 Karhunen-Love coefficients;
4) mfeat-zer: 47 Zernike moments;
5) mfeat-mor: 6 morphological features.

TABLE I

ACCURACY OF CLUSTERING

Data Set	PCKA	SSPC (Best)	HARP	PROCLUS (Best)	FastDOC (Best)
WDBC	91.56	93.84	62.91	71.00	83.59
SCGE	98.53	96.09	88.29	86.34	45.85
MF	90.45	–	58.35	83.50	10.00

In summary, we have a data set with 2000 patterns, 409 features, and 10 clusters. All the values in each feature were standardized to mean 0 and variance 1.

We compared the performance of our algorithm, in terms of clustering quality, with that of SSPC, HARP, PROCLUS and FastDOC. For this purpose, we used the class labels as ground truth and measured the accuracy of clustering by matching the points in input and output clusters. We adopted the classical definition of accuracy as the percentage of correctly partitioned data points. We want to mention that the value of the average cluster dimensionality parameter in PROCLUS was estimated based on the number of relevant dimensions identified by PCKA. In addition to this, since there are no data points labeled as outliers in any of the data described above, the outlier detection option of PCKA, SSPC, HARP and PROCLUS was disabled. Table I illustrates the accuracy of clustering for the four algorithms.

As can be seen from Table I, PCKA is able to achieve highly accurate results in different situations involving data sets with different characteristics. These results confirm the suitability of PCKA previously observed on the generated data. The behavior of SSPC is also characterized by high clustering accuracy. However, with multiple feature data, the clustering process is stopped and SSPC returns an error. HARP yields moderate accuracy on WDBC and MF data, while its performance on SCGE data is acceptable. FastDOC yields acceptable accuracy on WDBC data, while it performs poorly on SCGE and MF data. The poor results may be due to inappropriate choice of parameters, although we did try different sets of parameters in our experiments. Finally, from the experiments on the three data sets, we can see that the accuracy achieved by PROCLUS is reasonable.

To provide more comparisons and to confirm the suitability of our approach, we also analyzed the qualitative behavior of non-projected clustering algorithms on the datasets considered in this

set of experiments. In [39] the standard K-means algorithms and three unsupervised competitive neural network algorithms – the neural gas network [40], the growing neural gas network [41] and the self-organizing feature map [42]– are used to cluster the WDBC data. The accuracy achieved by these algorithms on this data is between 90% and 92%, which is very close to the accuracy achieved by PCKA on the same data (WDBC). Such results suggest that the moderate number of dimensions of this data does not have a major impact on algorithms that consider all dimensions in the clustering process.

In [38], the algorithm CORE (Clustering Of Repeat Expression data), a non-projected clustering algorithm, is used to cluster SCGE data. CORE is a clustering approach akin to the K-means, designed to cluster gene expression datasets. Such datasets are expected to contain noisy dimensions [6], [38]. The accuracy achieved by the CORE algorithm on SCGE data is 72.68% [2], which is less than that achieved by PCKA and other projected clustering algorithms, as illustrated in Table 1. Similar to the study in [6], our result suggests that projected clustering algorithms are suitable for clustering gene expression data.

Finally, for MF data, contrary to the other two datasets, it seems there are few studies in the literature that use this set to evaluate clustering algorithms. Most of the work on this dataset is related to classification algorithms [43], [44]. Hence, in order to have an intuitive idea about the behavior of a method that considers all dimensions in the clustering process, we have chosen to run the standard K-means on this data. The accuracy achieved is 77.2%, which is less than that achieved by PCKA and PROCLUS on the same data, as can be seen from Table 1. The enhanced performance given by PCKA and PROCLUS in comparison to that of the K-means suggests that some dimensions are likely not relevant to some clusters in this data.

V. Conclusion

We have proposed a robust distance-based projected clustering algorithm for the challenging problem of high-dimensional clustering, and illustrated the suitability of our alogorithm in tests and comparisons with previous work. Experiments show that PCKA provides meaningful results and significantly improves the quality of clustering when the dimensionalities of the clusters

[2]We calculated the accuracy of CORE on SCGE data based on the clustering result available from: http://cs.wellesley.edu/~btjaden/CORE/yeast_clustering.txt

are much lower than that of the dataset. Moreover, our algorithm yields accurate results when handling data with outliers. The performance of PCKA on real data sets suggests that our approach could be an interesting tool in practice. The accuracy achieved by PCKA results from its restriction of the distance computation to subsets of attributes, and its procedure for the initial selection of these subsets. Using this approach, we believe that many distance-based clustering algorithms could be adapted to cluster high dimensional data sets.

There are still many obvious directions to be explored in the future. The interesting behavior of PCKA on generated datasets with low dimensionality suggests that our approach can be used to extract useful information from gene expression data that usually have a high level of background noise. From the algorithmic point of view, we believe that an improved scheme for PCKA is possible. One obvious direction for further study is to extend our approach to the case of arbitrarily oriented clusters. Another interesting direction to explore is to extend the scope of Phase 1 of PCKA from attribute relevance analysis to attribute relevance and redundancy analysis. This seems to have been ignored by all of the existing projected clustering algorithms.

ACKNOWLEDGMENT

We gratefully thank Kevin Yuk-Lap Yip of Yale University for providing us with the implementations of projected clustering algorithms used in our experiments. We also thank the reviewers for their valuable comments and important suggestions. This work is supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[2] A. K. Jain, M. N. Mutry, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[3] K. Beyer, J. Goldstein, R. Ramakrishan, and U. Shaft, "When Is Nearest Neighbor Meaningful?" *Proc. 7th Int'l Conf. Database Theory (ICDT'99)*, pp. 217–235, 1999.

[4] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 3, pp. 1–12, 2005.

[5] C.C. Aggarwal, C. Procopiuc, J. L. Wolf, P.S. Yu, and J.S. Park, "Fast Algorithm for Projected Clustering," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'99)*, pp. 61–72, 1999.

[6] K.Y.L. Yip, D. W. Cheung, M. K. Ng and K. Cheung, "Identifying Projected Clusters from Gene Expression Profiles," *Journal of Biomedical Informatics*, vol. 37, no. 5, pp. 345–357, 2004.

[7] K.Y.L. Yip, D.W. Cheng and M.K. Ng, "On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering," *Proc. Int'l Conf. Data Engineering (ICDE'05)*, pp. 329–340, 2005.

[8] C.C. Aggarwal and P.S. Yu, "Redefining Clustering for High Dimensional Applications," *IEEE Trans. Knowledge and Data Eng.*, vol. 14, no. 2, pp. 210–225, 2002.

[9] K.Y.L. Yip, D.W. Cheng and M.K. Ng, "HARP: A Practical Projected Clustering Algorithm," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 11, pp. 1387–1397, 2004.

[10] C.M. Procopiuc, M. Jones, P.K. Agarwal, and T.M. Murali, "Monte Carlo Algorithm for Fast Projective Clustering," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'02)*, pp. 418 – 427, 2002.

[11] M. Lung and N. Mamoulis, "Iterative Projected Clustering by Subspace Mining," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 2, pp. 176–189, 2005.

[12] E.K.K. Ng, A.W. Fu, and R.C. Wong, "Projective Clustering by Histograms," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 32, pp. 369–383, 2005.

[13] M. Bouguessa, S. Wang and Q. Jiang, "A K-means-based Algorithm for Projective Clustering," *Proc. 18th IEEE Int'l Conf. Pattern Recognition (ICPR'06)*, pp. 888–891, 2006.

[14] C.H. Cheng, A.W. Fu, Y. Zhang, "Entropy-based Subspace Clustering for Mining Numerical Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'99)*, pp. 84–93, 1999.

[15] S. Goil, H. Nagesh, and A. Choudhary, "MAFIA: Efficient and scalable subspace clustering for very large data sets," *Technical Report CPDC-TR-9906-010, Northwestern University*, 1999.

[16] K. Kailing, H.-P. Kriegel, and P. Kroger, "Density-Connected Subspace Clustering for High-Dimensional Data," *Proc. 4th SIAM Int'l Conf. Data Mining (SDM'04)*, pp. 246–257, 2004.

[17] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: A Review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[18] K.Y.L. Yip, "HARP: A Practical Projected Clustering Algorithm for Mining Gene Expression Data," Master's thesis, The Univ. of Hong Kong, Hong Kong, 2004.

[19] K. Bury, *Statistical Distributions in Engineering*. Cambridge University Press, 1998.

[20] N. Balakrishnan and V.B. Nevzorov, *A Primer on Statistical Distributions*. John Wiley and Sons, 2003.

[21] R.V. Hogg, J.W. McKean and A. T. Craig, *Introduction to Mathematical Statistics*, sixth ed. Pearson Prentice Hall, 2005.

[22] J.F. Lawless, *Statistical Models and Methods for Lifetime Data*. John Wiley and Sons, 1982.

[23] M. Bouguessa, S. Wang and H. Sun, "An Objective Approach to Cluster Validation," *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1419–1430, 2006.

[24] J.J. Oliver, R.A. Baxter and C.S. Wallace, "Unsupervised Learning Using MML," *Proc. of the 13th Int'l Conf. Machine Learning (ICML'96)*, pp. 364–372, 1996.

[25] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[26] A. Dempster, N. Laird and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of Royal Statistical Society, (Series B)*, vol. 39, pp. 1–37, 1977.

[27] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[28] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley and Sons, 1997.

[29] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[30] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. World Scientific Publ. Co., 1989.

[31] S. Breunig, H.-P. Kriegel , R. Ng and J. Sander, "LOF: Identifying Density-Based Local Outliers," *ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'00)*, pp. 93–104, 2000.

[32] J. Han and M. Kamber, *Data Mining, Concepts and Techniques*. Morgan Kaufman, 2001.

[33] F. Angiulli and C. Pizzuti, "Outlier Mining in Large High-Dimensional Data Sets," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 2, pp. 369–383, 2005.

[34] E.M. Knorr, R.T. Ng and V. Tucakov, "Distance-Based Outliers: Algorithms and Applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.

[35] T. Li, "A Unified View on Clustering Binary Data," *Machine Learning*, vol. 62, no. 3, pp. 199–215, 2006.

[36] A. Patrikainen and M. Meila, "Comparing Subspace Clusterings," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 7, pp. 902–916, 2006.

[37] C.H. Papadimitrio and K. Steiglizt, *Combinatorial Optmization, Algorithms and Complexity*. Prentice-Hall, 1982.

[38] Brian Tjaden, "An approach for clustering gene expression data with error information," *BMC Bioinformatics*, vol. 7, no. 17, 2006.

[39] K.A.J. Doherty, R.G. Adams and N. Davey, "Unsupervised learning with normalised data and non-Euclidean norms," *Applied Soft Computing*, vol. 7, no. 17, pp. 203–210, 2007.

[40] T.M. Martinetz, S.G. Berkovich and K.J. Schulten, "Neural gas network for vector quantization and its application to timeseries prediction," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[41] B. Fritzke, "A growing neural gas network learns topologies," *In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems 7*, pp. 625–632, 1995.

[42] T. Kohonen, *Self-Organizing Maps*. Springer, 1997.

[43] P. Mitra, C.A. Murthy and S.K. Pal , "Unsupervised Feature Selection Using Feature Similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[44] A.K. Jain, R.P.W. Duin and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

Chapitre 2

Identification des experts dans les services Web de question-réponse

Dans le deuxième chapitre de cette thèse nous considérons le problème d'identification automatique des experts dans les forums Internet de question-réponse. Comme étude de cas pratique, nous nous focalisons sur Yahoo ! Answers. Généralement, pour identifier les experts, une stratégie commune consiste à utiliser les techniques d'analyse des liens (link analysis techniques) afin de fournir une liste d'utilisateurs ordonnée selon leur degré d'expertise. À partir de cette liste, les K premiers utilisateurs sont sélectionnés comme experts. Le problème avec cette approche réside dans le choix de la valeur de K. La nature *ad hoc* de la sélection de la valeur de K de pratiquement toutes les approches existantes rend difficile leurs utilisations en pratique. Au meilleur de notre connaissance, dans le contexte des forums Internet de question-réponse, aucune approche formelle pour séparer systématiquement les utilisateurs experts des non-experts n'a encore été proposée.

Pour pallier à ce problème, nous proposons un modèle probabiliste qui permet d'identifier automatiquement les utilisateurs experts au lieu de fournir une liste ordonnée des utilisateurs seulement. Dans notre approche nous représentons notre environnement sous forme de graphe dirigé de telle sorte que les nœuds représentent les utilisateurs et les arcs modélisent les interactions entres eux. Il convient de noter que la direction des arcs est définie de l'utilisateur qui a posé une question vers l'utilisateur qui a fourni la meilleure réponse à cette question. Par la suite nous analysons le comportement des différentes tech-

niques d'analyse des liens sur notre graphe et ce pour déterminer les propriétés de chaque nœud et ainsi estimer le degré d'expertise de chaque utilisateur. Dans notre investigation, nous avons constaté que la technique de InDegree, qui considère seulement les liens entrant de chaque nœud, est la plus appropriée pour estimer le degré d'expertise de chaque utilisateur.

Une fois les degrés d'expertise des utilisateurs sont estimés, nous proposons d'analyser leurs propriétés statistiques. Dans ce contexte, l'estimation de l'histogramme est un outil flexible pour les décrire. Dans notre investigation, nous avons constaté que l'histogramme des degrés d'expertise que nous avons estimé contient des composantes qui exhibent des formes variables. Cette observation nous a inspirée de modéliser les degrés d'expertise des utilisateurs comme un mélange de loi Gamma. Notre choix de la distribution de Gamma repose sur le fait que cette dernière offre différentes formes symétriques et asymétrique ce qui la rend flexible. Le nombre de composantes du mélange est estimé par le critère BIC (Bayesian Information Criteria), alors que les paramètres de chaque composante sont estimés en utilisant l'algorithme EM (Expectation-Maximisation algorithm). Dans nos expérimentations, nous avons trouvé que les degrés d'expertise des utilisateurs sont bien représentés sous forme de mélange de deux composantes Gammas. Une de ces deux composantes, qui correspond aux degrés d'expertise les plus élevés, représente les utilisateurs experts. Notre approche est appliquée sur des données réelles qui représentent les interactions des utilisateurs dans plusieurs forums différents de Yahoo ! Answers. Afin de démontrer la fiabilité de notre approche, une analyse du contenu textuel généré par les experts identifiés par notre méthode est effectuée. Les résultats démontrent que les experts identifiés génèrent du contenu de haute qualité. Notre contribution est présentée de façon détaillée dans les pages suivantes dans un article intitulé **Identifying Authoritative Actors in Question-Answering Forums - The Case of Yahoo ! Answers**. Cet article est publié dans les actes de la conférence internationale **ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)** [4].

Note : ma contribution au chapitre 2 de cette thèse inclut le développement de l'approche dédiée à l'identification automatique des experts, l'élaboration des différents tests de validation et la rédaction de l'article. Mon superviseur de stage chez Yahoo ! Canada, M. Benoit Dumoulin et mon directeur de recherche, le professeur Shengrui Wang, ont conjointement supervisé et validé l'approche proposée ainsi que la rédaction de l'article.

Identifying Authoritative Actors in Question-Answering Forums - The Case of Yahoo! Answers

Mohamed Bouguessa
Department of Computer Science
University of Sherbrooke
Sherbrooke, Quebec, Canada
m.bouguessa@usherbrooke.ca

Benoît Dumoulin
Yahoo! Inc.
2821 Mission College Boulevard
Santa Clara, CA 95054, USA
benoitd@yahoo-inc.com

Shengrui Wang
Department of Computer Science
University of Sherbrooke
Sherbrooke, Quebec, Canada
s.wang@usherbrooke.ca

ABSTRACT

We consider the problem of identifying authoritative users in Yahoo! Answers. A common approach is to use link analysis techniques in order to provide a ranked list of users based on their degree of authority. A major problem for such an approach is determining how many users should be chosen as authoritative from a ranked list. To address this problem, we propose a method for automatic identification of authoritative actors. In our approach, we propose to model the authority scores of users as a mixture of gamma distributions. The number of components in the mixture is estimated by the Bayesian Information Criterion (BIC) while the parameters of each component are estimated using the Expectation-Maximization (EM) algorithm. This method allows us to automatically discriminate between authoritative and non-authoritative users. The suitability of our proposal is demonstrated in an empirical study using datasets from Yahoo! Answers.

Categories and Subject Descriptors

H.5.3 **[Information Interfaces and Presentation (e.g., HCI)]**: Group and Organization Interfaces – *Collaborative computing, Theory and models, Web-based interaction.* J.0 [Computer Applications] General.

General Terms

Algorithms, Design, Experimentation.

Keywords

Identification of authoritative actors, Yahoo! Answers, Link analysis, Mixture model.

1. INTRODUCTION

Internet surfers use the Web to find various information related to a wide range of topics. In this context, Internet search engines fulfill a very important role, as they are used to find a series of relevant documents relative to a specific topic. Internet search engines are extremely useful. Without access to a search engine, many of us would be paralyzed.

Web users are also constantly looking for new online services to complement search engines. Among these services are question-answering forums where users come to ask and answer questions and share their knowledge. In these online services, users also form communities around shared interests in a wide variety of topics.

There are generally three types of users in question-answering communities [1]: 1) users who only ask questions; 2) users who only answer questions and 3) users who ask and answer questions. It is obvious that a user who answers another user's question generally has more expertise on the subject than the asker [1]. Answerer people provide advice to askers without the promise of a return on their investment [2]. On the other hand, an asker prefers to receive answers from authoritative users with expertise on the specific subject. In this context, identifying prominent actors in question-answering communities is of great importance.

One typical example of a question-answering forum is Yahoo! Answers[1]. Yahoo! Answers is a very popular service and already reports millions of users. According to [3] it captures 96% of the question and answer market share. This success can be attributed to the significant number of participants with different skills and expertise. Hence, locating different sources of authoritative knowledge by estimating the authority of each user becomes a crucial issue to improve the content and the quality of the service that Yahoo! Answers offers its users.

Several other reasons motivate the identification of authoritative actors in Yahoo! Answers. For instance, routing each newly asked question to appropriate experts significantly helps in providing askers with efficient, helpful service, by minimizing the effort and time askers must invest to find or receive good answers to their questions. Moreover, enhancing the visibility of authoritative users on the site and connecting askers with experts can play a critical role in fostering communities around shared interests. The formation of such communities encourages collaboration and knowledge sharing between users. In addition, authoritative users can participate efficiently and saliently in improving the quality of the site content by selecting useful information from the huge mass of information available in Yahoo! Answers.

The problem of identifying authoritative users in a community of users has recently received growing attention from the research community [4, 5, 6]. Most of the existing approaches that attempt to discover authoritative actors represent the environment as a graph in which nodes represent users and arcs represent the interactions between them. The authority score is generally measured by means of a graph-based ranking algorithm such as

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA

[1] http://answers.yahoo.com/

HITS [7] or PageRank [8] or one of its variants, AuthorRank [6] and ExpertiseRank [1].

The output of graph-based ranking algorithms is a ranked list of users based on their degree of authority on subjects of interest. Based on this list, the top K users are considered as most authoritative. The weakness of such an approach resides in the unprincipled selection of the value of K. In general, the value of K is chosen solely on the basis of specific knowledge of an application. However, in many real-life applications, such as the ones suggested for Yahoo! Answers, it is very difficult to set the value of K. At the same time, this value is crucial, as it can give more power to the selected users. An inappropriate choice of the value of K can have a very negative impact on the quality of the service. Furthermore, since there are a large number of categories in Yahoo! Answers, setting appropriate values of K for each category is a very difficult task if performed manually, by inspecting users' behavior for each of them. In this context, automating the process of discovering authoritative actors in Yahoo! Answers becomes an absolute necessity.

To our knowledge, no principled method for choosing the value of K has yet been published. This motivated our effort to design a method for automatically discriminating between authoritative and non-authoritative users, rather than simply producing a ranked list. To the best of our knowledge, the method that we propose is the first attempt to systematically discover authoritative actors in question-answering forums. In our approach, we propose to model the authority scores of users as a mixture of gamma distributions. The number of components in each mixture is estimated by the Bayesian Information Criterion (BIC), while the parameters of each component are estimated by the EM algorithm. Based on extensive experiments on datasets extracted from Yahoo! Answers, we found that authority scores can be modeled as a mixture of two gamma distributions. One of these two components, we will show, corresponds to authoritative actors.

The remainder of this paper is organized as follows. Section 2 describes Yahoo! Answers. In Section 3, we provide a brief overview of recent related work. In Section 4, we analyze the suitability of several link analysis techniques and choose the most appropriate one for our problem. Section 5 describes in detail our approach for the identification of authoritative actors. Section 6 presents the application of our technique on Yahoo! Answers. Our conclusion is given in Section 7.

2. YAHOO! ANSWERS

Yahoo! Answers is an online community-based question-answering service organized according to a taxonomy of topics. In general, such Web-based communities become very popular places for people to ask and answer questions in order to help each other. An ever-growing number of users participate in Yahoo! Answers. As a result, it now hosts a very large number of questions and answers in a wide variety of domains. Participants can thus save time in their quest for information because they can get an answer relatively quickly or find what they are looking for among the existing questions and answers. In addition, users actively participate in regulating the whole system. A user can report abusive behavior by other users who are violating the community guidelines. A user can also mark interesting questions, evaluate answers and vote for the best answers.

As described in [9], the central elements of the Yahoo! Answers system are the questions. Each question has a life cycle. It starts in an "open" state where it receives several answers. Then at some point (determined by the asker, or by an automatic timeout in the system), the question is considered "closed," and can receive no further answers. At this stage, a "best answer" is selected either by the asker or by other users via a voting procedure; once a best answer is chosen, the question goes into the "resolved" state and becomes, in principle, a browsable piece of information.

One of the most important feature of the question life cycle described above is the interaction between the user who asks a question (the asker) and the user whose answer is selected as best answer (the best answerer). When an asker chooses a best answer, he provides a quality rating for the answer. Such an interaction between users allows the generation of a weighted directed graph G representing the flow of best answer selection among the users involved.

The graph G is denoted $G = (V, E, W)$, where V is a set of nodes representing users (asker or best answerer). A directed edge $e \in E$ *where* $e = (v_1, v_2)$ *and* $v_i \in V$, indicates that user v_1 has chosen user v_2 as the best answerer for his question. Finally, W is the set of weights w_{ij} associated with each edge connecting a pair of users (v_i, v_j). The magnitude of the link between two users is the frequency corresponding to how often one user who asks a question chooses the other user as best answerer. Figure 1 illustrates the graph G. As can be seen from this figure, there are three different types of users: 1) users who usually only ask questions; 2) users who usually only answer questions and 3) users who help each other and share their knowledge.

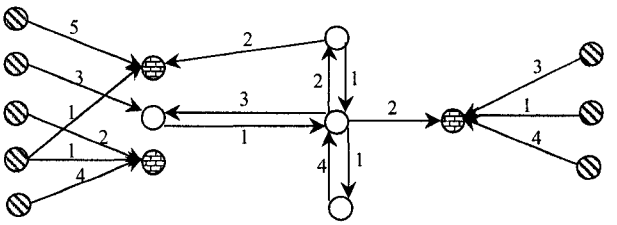


Figure 1. The graph G

An interesting property of the graph G is that users are usually bounded by shared interests and not social relationships. Users come to the site to ask and answer questions related to their interests. Often, askers take the time to choose the best answers. Therefore, the link structure depicted in Figure 1 reflects choices people make about what information is useful, interesting and authoritative. Our hypothesis is that the connection "asker – best answerer" implies a much stronger social bond than any other social relationship allowed in this service. In addition, as mentioned in [1], in a question and answer community, the direction of the links carries more information than just shared interests. It is obvious that the link between asker and best answerer usually indicates that the best answerer has a particular skill, experience, or expertise on a specific topic that the asker does not have.

The nature of the links between users (askers and best answerers) described above suggests that Yahoo! Answers is a valuable source of knowledge and seems particularly well suited

to the task of identifying authoritative actors, as participants usually communicate what they know. In this context, Yahoo! Answers offers an interesting case study since its content is highly multidisciplinary and therefore attracts users from a wide variety of fields; e.g., medicine, biology, mathematics, physics, philosophy, etc.

Since Yahoo! Answers is organized by categories (belonging to the taxonomy we mentioned previously), our goal consists in identifying authoritative users for each category. For this purpose, we exploit the structure of the graph representing the interaction between users (asker – best answerer) in each category.

3. RELATED WORK

The problem of identifying authoritative users in a community is mainly related to the problem of expert identification. Most of the recent work in this domain represents the environment by a graph or a network [1, 4, 5, 6, 10]. Such representations allow for the application of "link analysis" techniques and graph-based ranking algorithms in order to analyze the properties of each node (user status). The output of these algorithms is a list of all nodes (users) ranked according to their authoritative level.

Campbell at al. [4] describe a system that identifies expertise from email. For this purpose, a graph-based ranking algorithm and a content-based technique that takes into account only email texts are used. Based on the results in [4], the graph-based algorithm seems more suitable for the purpose of expert identification than a content-based approach. Dom et al. [5] compare various ranking algorithms, including HITS and PageRank, on both artificial and email networks. The experiments in [5] show that PageRank performs better than other ranking algorithms.

Liu et al. [6] explore the co-authorship network of past ACM, IEEE, and joint ACM/IEEE digital library conferences in order to measure the prestige of an author (prestige is closely related to the notion of expertise [1]). The co-authorship network is represented as a graph where each author is represented as a node and the collaboration relationship between actors (co-authorship) is represented as an edge. The authors in [6] propose three representations of their co-authorship network: 1) a non-weighted undirected graph; 2) a non-weighted directed graph and 3) a weighted directed graph. To investigate these graphs, several link analysis techniques, such as betweenness centrality and PageRank, are applied. A modified version of PageRank named AuthorRank is also proposed in order to analyze weighted directed graphs. The results in [6] show the clear advantage of the use of AuthorRank and PageRank in the co-authorship network.

In [10], an entropy model that combines information theory with statistical techniques is proposed to identify important nodes (authoritative users) from a graph. Important nodes are those which have the most effect on the graph's entropy when they are removed from the graph. The algorithm described in [10] comprises two phases. First the entropy of the whole graph is calculated. Second, the nodes in the graph are removed one by one and at each step the entropy of the remnant graph is estimated. The output of the algorithm is a ranked list of nodes based on the graph entropy. The authors in [10] illustrate the suitability of their approach for the identification of interesting influential members from the Enron email dataset (http://www.cs.cmu.edu/~enron/).

Expert identification from question-answering communities has been the object of study in recent work [1]. Zhang et al. [1] analyze data from the Java forum. The data is represented as a graph in which nodes correspond to users and arcs represent the interactions between users who ask a question and users who answer a question. The authors evaluate several graph-based ranking algorithms, including HITS and ExpertiseRank (a PageRank-like algorithm). The experiment in [1] reveals that a simple link-based metric could be a powerful tool for measuring the expertise level of users from question-answering communities. The problem of ranking users in question-answering forums was also studied in [11]. Instead of using graph-ranking algorithms, the authors propose a Bayesian-based approach to obtain a posterior estimate of user's credibility. Further details and a survey on the problem of identifying authoritative actors can be found in [12].

4. LINK ANALYSIS TECHNIQUES

Given the graph structure described in Section 2, we now turn to the problem of analyzing the properties of each node (user status). As mentioned in the previous section, link analysis techniques are widely used for this purpose since they provide a score of the relative authority of each node in the graph. A number of link analysis algorithms have been proposed [1, 7, 8]. The fundamental question now is what sort of model would work best for our application. In what follows, we discuss some important link analysis techniques which are close to the context of our study and then choose the most appropriate one for our application. We also argue why some techniques are not yet well suitable to the task of our study.

4.1 PageRank

PageRank and HITS were the pioneering approaches that introduced Link Analysis Ranking, in which hyperlink structures are used to determine the relative authority of a Web page. The PageRank assumption is that a node transfers its PageRank values evenly to all the nodes it connects to. A node has high rank if the sum of the ranks of its in-links is high. This covers both the case where a node has many in-links and that where a node has a few highly ranked in-links.

The idea of PageRank has an intuitive parallel for question-answering forums. To clarify this, let's take the example given in [1]. If *B* is able to answer *A*'s questions, and *C* is able to answer *B*'s questions, then *C* should receive a high authority score, since he is able to answer the questions of someone who himself has some expertise. However, this is not usually true in Yahoo! Answers. To clarify this point, let's take for instance the category "Programming & Design". In such a category, users ask questions related to different languages of programming such as Java, C++, PHP, etc. In this context, if user *B* answers user *A*'s questions, which are about Java, and user *C* answers *B*'s questions, which are about PHP, it is not possible to state that C is more expert than *B*, because *B* and *C* have different expertise. The former has expertise in Java while the latter has expertise in PHP.

The example described above is frequent in Yahoo! Answers since this latter is organized in categories corresponding to large subject areas. Indeed, in the same category, we can find users who have expertise on specific topics but ask questions about other topics. This stimulates the interaction between Yahoo! Answers participants and encourages them to help each other frequently. We claim that PageRank provides interesting results

when the interactions between users are around one specific subject only. The study in [1] illustrates this by using a PageRank-like algorithm called ExpertiseRank on data from the Java forum, in which the interactions between users are exclusively about Java programming.

4.2 HITS

Kleinberg, in his seminal paper [7], introduced the HITS algorithm. The fundamental assumption of HITS is that in a graph there are special nodes that act as hubs. Hubs contain collections of links to authorities [7]. A good hub is a node that points to good authorities, while a good authority is a node pointed to by good hubs. In the context of our study, askers can act as hubs and best answerers can act as authorities. HITS associates two scores to each node: a hub score and an authority score. However, based on experiments we performed on data from Yahoo! Answers, the algorithm HITS does not yield satisfactory results. The following two examples illustrate this point.

Example 1. Consider the graph depicted in Figure 2. Similar to the graph illustrated in Figure 1, the nodes in this graph correspond to users (askers or best answerers) and the directed edges represent the interactions between them. Since HITS is principally designed to work with directed non-weighted graphs, we did not consider the weight of edge in this example.

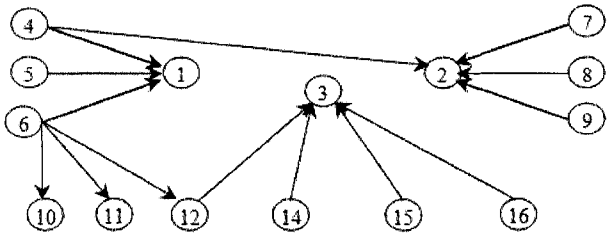


Figure 2. Example 1

In Yahoo! Answers, as described in Section 2, when an asker chooses the best answerer, he or she provides a "quality" rating. In other words, the interaction between the asker and the best answerer represents an endorsement of the quality of the answer of the best answerer. In this particular setting, as we can see from Figure 2, nodes 1, 2 and 3 should be ranked higher since they are frequently chosen as best answerers. However, the HITS algorithm assigns high authority scores to nodes 1, 2, 10, 11 and 12, but a near-zero authority score to node 3. This is because, the fact that node 6 points to node 1, which has a strong authority, increases its hub score, causes the authority scores of nodes 10, 11 and 12 to increase. On the other hand, nodes that point to node 3 have relatively small hub scores in comparison to the other nodes that point to nodes 1 and 2. Therefore, node 3 receives a very low authority score with the HITS algorithm. While this behavior of HITS could be very interesting in the context of Web-page ranking, it is not suitable in our study.

Example 2. Figure 3 represents an example taken from [14]. In this graph, there are two components. Nodes *N1–N7* correspond to the first component, while nodes *N8–N15* correspond to the second component. This example corresponds to some real situations in Yahoo! Answers. The HITS algorithm will allocate high authority scores to the nodes *N9–N15*, while giving zero authority score to node *N1* [14]. The reason for this is quite similar to example 1. Specifically, the fact that node *N8* points to many nodes contribute to increase its hub score. Hence, causing the nodes *N9–N15* to receive higher authority scores. However, intuition suggests that node *N1* is the most authoritative since it represents an answerer with a large number of best answers.

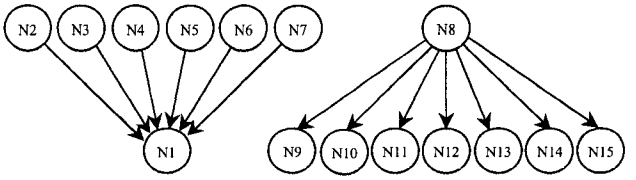


Figure 3. Example 2

This particular behavior of HITS is not surprising since its main idea is based on mutually reinforcing relationships between hubs and authorities. In some cases, such a strategy has a negative impact since it gives an inappropriate hub/authority score to a node. In other contexts, including the expert identification, similar behavior of HITS was also observed [1, 5, 13, 14].

We conclude that HITS is not an appropriate choice for our application. On the other hand, HITS is a powerful tool for ranking Web pages since its concept of hubs and authorities (which mutually reinforce each other during the ranking process) fits well with the hyperlink structure of the Web [7]. A deeper investigation on hubs and authorities framework defined by Kleinberg can be found in [14].

Note that HITS (like PageRank) was originally designed to rank nodes from directed non-weighted graphs. However, interactions between Yahoo! Answers participants are represented as directed weighted graphs. Here it is evident that using PageRank or HITS on our data will lead one to ignore a crucial type of information: the magnitude of interaction between users. Liu et al. [6] and Zhang et al. [1] propose AuthorRank and ExpertiseRank, respectively, which extend PageRank slightly by taking link weights into account. However, since these two algorithms are PageRank-based, they exhibit the same behavior as PageRank on our data. Therefore, in the context of our application, AuthorRank and ExpertiseRank could not be used to accurately identify authoritative users.

4.3 Z-score

Zhang et al. [1] also propose the Z-score measure in order to estimate the expertise level of participants in question-answering forums. The Z-score of a user is given by $z = (a - q)/\sqrt{(a - q)}$, where a is the number of answers and q is the number of questions posted by a user. The intuition underlying the Z-score is that it is highly probable that a user who answers many questions is an authority. On the other hand, if a user asks many questions, this implies that it is likely this user has no authority on a specific topic.

The experiments in [1] show that the Z-score provides accurate results, comparable to those of ExpertiseRank, and better than those of HITS. However, in our application, such a measure cannot be used. For instance, in the category "Programming & Design", we have a user A who is expert in C++ and interested in PHP. This user answers many questions related to C++ and usually his answers are chosen as best answers. On the other hand, the same user asks many questions about PHP programming since he has a lack of expertise on this specific topic. Here, it is possible that the Z-score of A is negative, which means that this user could not be considered as an authority.

Based on the example above, it is clear that the use of Z-scores to measure the authority of users in our application entails a severe loss of information. As with PageRank and ExpertiseRank, the Z-score provides useful results on data in which the interaction between users concerns only one topic [1].

4.4 InDegree

A simple technique that can be used to measure the authority of Yahoo! Answers participants is the InDegree. As the name implies, with the InDegree technique the authority of a node (user) is measured by the number of nodes that link to this node. A node with a high InDegree is likely to be a good authority. However, looking at our graph structure depicted in Figure 1, if we measure the InDegree of the nodes based only on the number of in-links we would be ignoring important information. It is clear that we need a weighted graph analysis. In the context of our study, the InDegree of a node will thus be measured based on the sum of the weights of edges that point to this node.

In [7], Kleinberg provides a solid argument against the InDegree technique, claiming that it is not sophisticated enough to capture the authoritativeness of a node in the context of a Web hyperlink environment. However, the graph representation that we adopt in our study is radically different from that studied by Kleinberg in [7]. As described in Section 2, our graph representation reflects choices people make about what information is useful, interesting and authoritative. A link from an asker A to the best answerer B denotes an endorsement for the quality of B's answer. In this context, the InDegree technique can be used to rate the authority of each user. In our case, the InDegree of a user is simply the number of best answers given by that user.

In Yahoo! Answers, it is reasonable to assume that a user with a high number of best answers in a specific category has some expertise on one or several topics of discussion in that category. For instance, a user who answers 100 questions with 50 best answers among them is better than a user who answers 200 questions with 0 best answers. This is because, in Yahoo! Answers we are not only interested in users who answer many questions, but also in authoritative users, i.e., trusted sources of correct information.

On the other hand, the reader should be aware that it is possible that some spammers create several accounts with different pseudonyms. Such users can ask and answer their own questions and choose their answers as best answers. Here, it is clear that by doing so, the value of the InDegree of theses users will increase. However, such behavior does not have a major impact on the estimation of the authority level because the community will usually identify theses users and report such so-called "abusive behaviors". As a consequence, the accounts of such users are deleted from the system by moderators.

For all of the above reasons, we surmise that the InDegree technique is more appropriate for our application. Below, we develop a technique to automatically identify authoritative users.

5. APPROACH

After analyzing the behavior of several link analysis techniques on Yahoo! Answers and clarifying why the InDegree technique (i.e., number of best answers) is the most appropriate one for our application, we now turn to solving the problem of identifying authoritative users for each category in the Yahoo! Answers taxonomy. A simple approach is to rank users based on their InDegree and select the top K users as authorities. However, as described in Section 1, in some applications such as Yahoo! Answers, it is difficult to set appropriate values for K. In an attempt to provide an automatic solution to this problem, we developed a systematic and efficient way to discriminate between authoritative and non-authoritative users.

Let $X = \{x_1, x_2, \ldots, x_N\}$ be the set of N users having at least one best answer in a specific category CAT. The InDegree of a user x_i is denoted by y_i, where y_i is the sum of the best answers of user x_i in category CAT. We normalize the InDegrees in such a way that $\sum_{i=1}^{N} (y_i)^2 = 1$. The normalized InDegree provides a relative score of the authority of each user. Intuitively, a large value of y_i means that x_i belongs to the set of authoritative users, while a small value indicates that x_i belongs to the set of non-authoritative users. In order to identify authorities in each category, we are interested in all sets of x_i having large values of y_i.

Estimating the histogram is a flexible tool to describe some statistical properties of the normalized InDegree y_i. For the purpose of clarification, we extract data from the category "Engineering". The histogram of the normalized InDegree of users in this category is given in Figure 4. As we can see, the histogram depicted in this figure suggests the existence of two components. One of these two components represents low values of y_i ($y_i < 0.04$), while the other one represents large values of y_i ($0.04 < y_i < 0.2$). We surmise that the first component represents non-authoritative users while the second one represents authoritative users. The fundamental question now is how to formally distinguish between authoritative and non-authoritative users. For this purpose we propose to model the normalized InDegree y_i of all the users in a specific category as a mixture distribution. The probability density function (*pdf*) is therefore estimated and the status of each user in a specific category is identified.

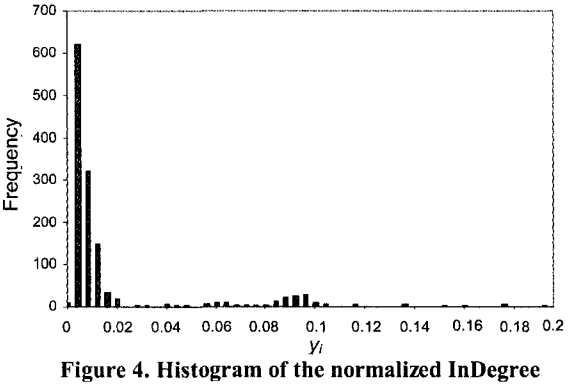


Figure 4. Histogram of the normalized InDegree

5.1 *pdf* Estimation

As mentioned above, the histogram depicted in Figure 4 suggests the existence of two components with different shapes and/or heavy tails. This interesting observation inspires us to use the gamma mixture model. Formerly, we expect that the normalized InDegree follows a mixture density of the form:

$$G(y) = \sum_{l=1}^{m} \gamma_l G_l(y, \alpha_l, \beta_l) \quad (1)$$

where $G_l(.)$ is the lth gamma distribution with parameters α_l and β_l representing, respectively, the shape and the scale parameters of the lth component; and γ_l ($l = 1, \ldots, m$) are the mixing

coefficients, with the restriction that $\gamma_l > 0$ for $l = 1, \ldots, m$ and $\sum_{l=1}^{m} \gamma_l = 1$.

The density function of the lth component is given by

$$G_l(y, \alpha_l, \beta_l) = \frac{\beta_l^{\alpha_l}}{\Gamma(\alpha_l)} y^{\alpha_l - 1} \exp(-\beta_l y) \quad (2)$$

where $\Gamma(\alpha_l)$ is the gamma function.

As mentioned above, each gamma component G_l in equation (2) has two parameters: the shape parameter α_l and the scale parameter β_l. The shape parameter allows the distribution to take on a variety of shapes, depending on its value [15]. When $\alpha_l < 1$, the distribution is highly skewed and is L-shaped. When $\alpha_l = 1$, we get the exponential distribution. In the case of $\alpha_l > 1$, the distribution has a peak (mode) in $(\alpha_l - 1) / \beta_l$ and skewed shape. The skewness decreases as the value of α_l increases. This flexibility of the gamma distribution and its positive sample space make it particularly suitable to model the distribution of the normalized InDegree y_i.

The use of a mixture of gamma distributions allows us to propose a flexible model to describe the distribution of the normalized InDegree y_i. To form such a model, we need to estimate m, the number of components, and the parameters for each component. A standard approach for estimating the parameters of the gamma components G_l is the maximum likelihood technique [16]. The likelihood function is defined as

$$\begin{aligned} L_{G_l}(\alpha_l, \beta_l) &= \prod_{y \in G_l} G_l(y, \alpha_l, \beta_l) \\ &= \frac{\beta_l^{\alpha_l N_l}}{\Gamma^{N_l}(\alpha_l)} \prod_{y \in G_l} y^{\alpha_l - 1} \exp(-\beta_l \sum_{y \in G_l} y) \end{aligned} \quad (3)$$

where N_l is the size of the lth component. The logarithm of the likelihood function is given by

$$\begin{aligned} \log(L_{G_l}(\alpha_l, \beta_l)) = N_l \alpha_l \log(\beta_l) - N_l \log(\Gamma(\alpha_l)) \\ + (\alpha_l - 1) \sum_{y \in G_l} \log(y) - \beta_l \sum_{y \in G_l} y \end{aligned} \quad (4)$$

To find the values of α_l and β_l that maximize the likelihood function, we differentiate $log(L_{Gl}(\alpha_l, \beta_l))$ with respect to each of these parameters and set the result equal to zero:

$$\frac{\partial}{\partial \alpha_l} \log(L_{G_l}(\alpha_l, \beta_l)) = N_l \log(\beta_l) - N_l \frac{\Gamma'(\alpha_l)}{\Gamma(\alpha_l)} + \sum_{y \in G_l} \log(y) = 0$$

$$\Rightarrow -\log(\beta_l) + \frac{\Gamma'(\alpha_l)}{\Gamma(\alpha_l)} = \frac{1}{N_l} \sum_{y \in G_l} \log(y) \quad (5)$$

and

$$\frac{\partial}{\partial \beta_l} \log(L_{G_l}(\alpha_l, \beta_l)) = \frac{N_l \alpha_l}{\beta_l} - \sum_{y \in G_l} y = 0$$

$$\Rightarrow \beta_l = \frac{N_l \alpha_l}{\sum_{y \in G_l} y} \quad (6)$$

This yields the equation

$$\log(\alpha_l) - \Psi(\alpha_l) = \log(\frac{1}{N_l} \sum_{y \in G_l} y) - \frac{1}{N_l} \sum_{y \in G_l} \log(y) \quad (7)$$

where $\Psi(.)$ is the digamma function given by $\Psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$.

The digamma function can be approximated very accurately using the following equation [17].

$$\Psi(\alpha) = \log(\alpha) - \frac{1}{2\alpha} - \frac{1}{12\alpha^2} + \frac{1}{120\alpha^4} - \frac{1}{252\alpha^6} + \cdots \quad (8)$$

The parameter α_l can be estimated by solving equation (7) using the Newton-Raphson method. Its estimated value $\hat{\alpha}_l$ is then substituted into equation (6) to determine $\hat{\beta}_l$.

Let us focus now on the problem of estimating m, the number of components in the distribution. Based on the histogram depicted in Figure 4, we can assume that the normalized InDegree could be modeled as a mixture of two gamma components. However, this is just a visual observation; there is no proof which guarantees that there are indeed two components. To address this issue, we will now describe a strategy we adopt for estimating the optimal number of components in a mixture.

One popular approach [18] to estimating the number of components m is to test values of m from 1 to *m_max* (*m_max* is an input parameter which represents the maximum number of components) against a performance measure and choose the number of components that optimizes the performance measure. For this purpose, we implement a standard two-step process. In the first step, we calculate the maximum likelihood of the parameters of the mixture for a range of values of m (from 1 to *m_max*). The second step involves calculating an associated criterion and selecting the value of m which optimizes the criterion. A variety of measures have been proposed to estimate the number of components in a dataset [18, 19]. In our method, we use a penalized likelihood criterion, called the Bayesian Information Criterion (*BIC*). *BIC* was first introduced by Schwartz [20] and is given by

$$BIC(m) = -2L_m + N_p \log(N) \quad (9)$$

where L is the logarithm of the likelihood at the maximum likelihood solution for the mixture model under investigation, and N_p is the number of parameters estimated. The number of components that minimizes *BIC(m)* is considered to be the optimal value for m.

Typically, the maximum likelihood of the parameters of the distribution is estimated using the EM algorithm [21]. This algorithm requires the initial parameters of each component. Since EM is highly dependent on initialization [22], it will be helpful to perform initialization by mean of a clustering algorithm [22]. For this purpose we implement the Fuzzy C-Means algorithm (FCM) [23] to partition the set of y_i into m components. Based on such a partition we can estimate the parameters of each component and set them as initial parameters for the EM algorithm. The procedure for estimating the number of components is summarized in Figure 5.

Based on extensive experiments on data extracted from a large numbers of categories from Yahoo! Answers, we found that the optimal number of components in a mixture is always two. This suggests that the normalized InDegrees are well fitted by two gamma components, the first representing non-authoritative users with small values of InDegree while the second represents authoritative users with large values of InDegree. The *pdf* of the

normalized InDegree represented in Figure 4 is illustrated in Figure 6.

Input: $\{y_i\}$, m_max
Output: the optimal number of components m;
begin
  for $m = 1$ *to* m_max **do**
    if $m == 1$ **then**
      Estimate the parameters of the gamma distribution based on the likelihood formula using equations (6), (7) and (8);
      Compute the value of *BIC(m)* using equation (9);
    else
      Apply FCM as an initialization of the EM algorithm;
      Apply the EM algorithm to estimate the parameters of the mixture using equations (6), (7) and (8);
      Compute the value of *BIC(m)* using equation (9);
    end;
  end;
      Select the number of components $\hat{m}$, such that
      $\hat{m} = arg_{min} BIC(m)$;
end

Figure 5. Estimation of the number of components *m*

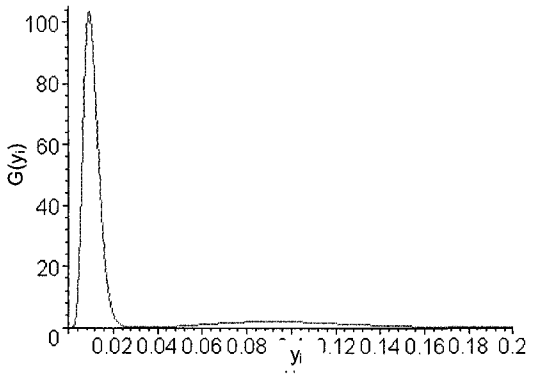


Figure 6. *pdf* of the normalized InDegree

5.2 Summary of Authoritative User Identification Procedure

Based on the analysis above, it is reasonable now to assume that the normalized InDegree could be modeled as a mixture of two gamma components. The following steps can then be implemented to efficiently discriminate authoritative from non-authoritative users.

1. For a given category, estimate the InDegree of each user;
2. Normalize y_i, where $\sum_{i=1}^{N} (y_i)^2 = 1$;
3. Estimate the *pdf* of the normalized InDegree with $m = 2$;
   3.1. Apply FCM as initialization of the EM algorithm;
   3.2. Apply EM to estimate the parameters of the mixture;
4. Use the results of the EM algorithm in order to derive a classification decision about the membership of y_i in each component.

Based on this algorithm, we can definitively discriminate between authoritative and non-authoritative users in a specific category by selecting the component which represents large values of InDegree y_i. In the following section we devise an empirical study designed to illustrate the suitability of our approach.

6. EXPERIMENTS

In this section, we put our approach at work using data from Yahoo! Answers. It worth to point out that there is no standard method in the literature that our technique could be compared to. To the best of our knowledge, the method that we propose is the first attempt to automatically discriminate authoritative and non authoritative users, while existing approaches provide a ranked list of users only. Furthermore, for the reasons mentioned in section 4, we do not consider techniques such as HITS [7] and PageRank [8] (and their derivatives such as ExpertiseRank [1]) for comparison since they are not well suited for the task of our study. On the other hand, there is also a shortage of standard benchmark data which could be used to evaluate approaches designed to identify expert/authoritative users. All these make the evaluation of the proposed method a challenging task. In this particular context, we have adopted a principled way of evaluating the technique presented in this paper. In the following, we saliently illustrate the suitability of our method. First, we describe the datasets used in our experiments, and then we report the results of our experiments and provide discussions.

6.1 Datasets

We report experiments on datasets which represent users' activities over one full year (from May 2006 to April 2007), for six categories: "Engineering", "Biology", "Programming & Design", "Mathematics", "Physics" and "Chemistry". Since we are attempting to identify authoritative actors in each category, our datasets contain interactions between askers and best answerers only. The dataset statistics are reported in Table 1. Note that, due to commercial-in-confidence, all the datasets statistics are reported in percentage. As illustrated in this table, more than half of the users in each category ask questions only. This indicates that Yahoo! Answers is really a place where users come to get answers to their questions by relying on other users' expertise in different topics. Also, it is interesting to see that a large fraction of users are only interested in sharing their knowledge by answering questions only. This repartition of users' activities is very common into many other categories not described here. It can however be less clear in some other categories where users are engaging into discussions more than into knowledge sharing (this is the case for example in the "Politics" category).

Table 1. Datasets statistics

Category	% users who ask only	% users who answer only	% users who ask and answer
Engineering	65%	31%	4%
Biology	60%	36%	4%
Programming	66%	29%	5%
Mathematics	64%	31%	5%
Physics	60%	34%	6%
Chemistry	63%	32%	5%

6.2 Identifying Authoritative Users

We use our approach to identify authoritative users in each category, as presented in Table 1. As mentioned in Section 5.1, in all cases we found that the normalized InDegrees are well fitted by two gamma components. The component that contains large InDegree values represents authoritative users. In general, we found that only a few hundreds users who are authoritative. Table 2 provides an idea of the percentage of authoritative users identified in each category.

Table 2. Percentage of authoritative users in each category

Category	% authoritative users
Engineering	0.74%
Biology	0.66%
Programming	0.72%
Mathematics	0.50%
Physics	0.70%
Chemistry	0.70%

In order to evaluate the effectiveness of our approach, we looked at the behavior/activity of a significant number of identified authoritative users in the site and manually evaluated their answers. Since there are millions of question and answers in Yahoo! Answers, it is impossible for us to investigate all of the users manually. We thus selected a few hundreds of authoritative users from the categories "Programming & Design" and "Mathematics". We specifically chose these two categories because they are close to our domains of expertise.

We performed a thorough analysis and observed that all of the selected users are very active, with a strong presence on the site. In most cases, they provide detailed answers of good quality to a large number of questions. Furthermore, we investigated each selected user's *profile page* (in Yahoo! Answers, profile pages allow individuals to provide information about themselves and their expertise). This yielded very interesting and encouraging results. For instance, we found that the selected users in the "Mathematics" category included math teachers and graduate students. In the category "Programming & Design", there were a number of software engineers, Web programmers and students. Such users are valuable sources of knowledge.

In our investigation we also observed that such users play a significant role in regulating the whole system on the site. In several cases, they provide an objective evaluation of the answers of other users through the voting mechanism available on the site. We believe such users are potential candidates to perform a given organizational role on the site.

Based on these encouraging results, we expect that other authoritative users identified by our approach in other categories display similar behavior to the users we analyzed from the categories "Programming & Design" and "Mathematics". To confirm our claim, we now investigate the quality of the content generated by all the identified authoritative users in a more systematic way.

6.3 Quality of Content

The aim of this set of experiments is to evaluate the quality of the content generated by all the identified authoritative users. We expect these users to generate high-quality content (i.e., questions and answers of high quality). Hence, evaluating the quality of the content generated by authoritative users can also be a validation of the suitability of our method. For this purpose, we use the quality metric described in [9] as the "gold standard" for evaluation. Due to space limit, we provide, in the following, a brief description of this approach. Further details can be found in [9].

The work in [9] addresses the problem of identifying high-quality content in community-driven question-answering sites. The approach combines the analysis of the textual content with the user feedback on the site in order to estimate a quality score for each question and answer. The quality score described in [9] is the confidence score of a binary classifier trained on high and low quality examples. The value of the quality score is always between 0 and 1. When the question or answer is of high quality, the value of the quality score is close to 1. On the other hand, a question or answer with low quality receives a very small quality score value (close to 0). The experiments in [9] illustrate that such an approach to identifying high-quality content achieves an accuracy close to that of humans. Figure 7 shows the average quality score of authoritative users in each category.

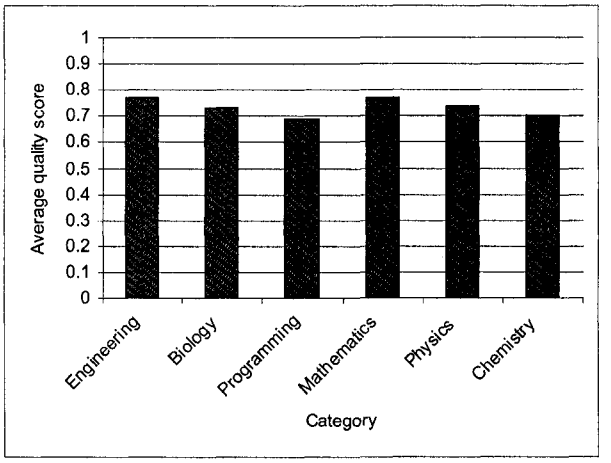


Figure 7. Average quality score of the answers of the identified authoritative users in each category

As we can see from this figure, the average quality score of authoritative users in each category is generally between 0.7 and 0.77, which is a relatively high quality score. This result represents another source of confirmation concerning the suitability of our approach for identifying users that contribute significantly to generate high-quality content in Yahoo! Answers. Moreover, such results also indicate that askers are very selective in choosing the best answerers. We can thus rely on them and on their judgment with a pretty high level of confidence.

7. CONCLUSION

In this paper we addressed the problem of the automatic identification of authoritative users in a Web-based question-answering community. Specifically, we analyzed data from Yahoo! Answers, a large-scale community question-answering site. We represented our environment as a weighted directed graph built from the interactions between askers and best answerers. The behavior of several link analysis techniques on our data was analyzed. We concluded that a simple technique such as the InDegree is the most appropriate for rating the authority level of each user.

In order to automatically identify authoritative users in each category, we proposed a probabilistic approach based on a mixture model. First, we estimated the normalized InDegree of each user in each category. Next, we analyzed their statistical properties. We found that the normalized InDegrees are well fitted by two gamma components. One of these two components, which contains large values of InDegree, represents authoritative users.

We illustrated the suitability of our proposal on datasets extracted from a number of different categories. Experiments showed that our approach is able to automatically identify authoritative users in each analyzed category. Moreover, we evaluated the content generated by the identified authoritative users. Our results clearly demonstrate that such users contribute significantly to the generation of high-quality content. Such results confirm the effectiveness of the proposed approach for identifying prominent users who are rich sources of knowledge.

8. ACKNOWLEDGMENTS

The work of the first author was done while internship at Yahoo!. The authors wish to thank Gilad Mishne for useful discussions and anonymous reviewers for their valuable suggestions.

9. REFERENCES

[1] J. Zhang, M.S. Ackerman and L. Adamic. Expertise Networks in Online Communities: Structure and Algorithms. Proceedings of the 16th ACM International World Wide Web Conference (WWW'07), pages 221-230, 2007.

[2] T.C. Turner, M.A. Smith, D. Fisher and H.T. Welser. Picturing Usenet: Mapping Computer-Mediated Collective Action. Journal of Computer-Mediated Communication, 10 (4), article 7, 2005.

[3] L. Prescott, Yahoo! Answers captures 96% of Q and A market share, 2006.

[4] C.S. Campbell, P.P. Maglio, A. Cozzi and B. Dom. Expertise Identification using Email Communication. Proceedings of the 12th ACM International Conference on Information and Knowledge Management (CIKM'03), pages 528-531, 2003.

[5] B. Dom, I. Eiron, A. Cozzi and Y. Zhang. Graph-Based Ranking Algorithms for E-mail Expertise. Proceedings of 8th ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'03), pages 42-48, 2003.

[6] X. Liu, J. Bollen, M. L. Nelson and H. V. Sompel. Co-authorship Network in the Digital Library Research Community. Information Processing and Management, 41 (6): 1462-1480, 2005.

[7] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, 46 (5): 604-632, 1999.

[8] L. Page, S. Brin, R. Motwani and T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web, Stanford Digital Library Technologies Project, 1998.

[9] E. Agichtein, C. Castillo, D. Donato, A. Gionis and G. Mishne. Finding High-Quality Content in Social Media. Proceedings of the 1st ACM International Conference on Web Search and Data Mining (WSDM'08), pages 183-194, 2008.

[10] J. Shetty and J. Adibi. Discovering Important Nodes through Graph Entropy: The Case of Enron Email Database. Proceedings of the 3rd International Workshop on Link Discovery, pages 74-81, 2005.

[11] B. Dom and D. Paranjpe. A Bayesian Technique for Estimating the Credibility of Question Answerers. Proceedings of SIAM Conference on Data Mining (SDM'08), pages 399-409, 2008.

[12] D. Yimam and A. Kobsa. Expert Finding Systems for Organisations: Problem and Domain Analysis and the DEMOIR Approach. Journal of Organizational Computing and Electronic Commerce, 13 (1): 1-24, 2003

[13] K. Bharat and M. Henzinger. Improved Algorithms for Topic Distillation in Hyperlinked Environments. Proceedings of the 21st Annual International ACM SIGIR Conference (SIGIR'98), pages 104-111, 1998.

[14] A. Brodin, G. O. Roberts, J. S. Rosenthal and P. Tsaparas. Link Analysis Ranking: Algorithms, Theory, and Experiments. ACM Transactions on Internet Technology 5 (1): 231-297, 2005.

[15] N. Balakrishnan and V.B. Nevzorov. A Primer on Statistical Distributions. John Wiley and Sons, 2003.

[16] R. V. Hogg, J.W. McKean and A.T. Craig. Introduction to Mathematical Statistics. Pearson Prentice Hall, sixth ed., 2005.

[17] J.F. Lawless. Statistical Models and Methods for Lifetime Data. John Wiley and Sons, 1982.

[18] M. Bouguessa, S. Wang and H. Sun. An Objective Approach to Cluster Validation. Pattern Recognition Letters 27 (13): 1419-1430, 2006.

[19] J.J. Oliver, R.A. Baxter and C.S Wallace. Unsupervised Learning Using MML. Proceedings of the 23rd International Conference on Machine Learning (ICML'06), pages 364-372, 2006.

[20] G. Schwarz. Estimating the Dimension of a Model. Annals of Statistics, 6 (2): 461-464, 1978.

[21] A. Dempster, N. Laird and D. Rubin. Maximum Likelihood from Mixture Models. Journal of Royal Statistical Society, (Series B): 1-37, 1977.

[22] M.A.T. Figueiredo and A.K. Jain. Unsupervised Learning of Finite Mixture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (3): 381-396, 2002.

[23] J.C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. New York Plenum 1981.

Chapitre 3

Identification des communautés de partage des connaissances dans les services Web de question-réponse

Le dernier chapitre de cette thèse considère le problème de l'identification des communautés dans les forums Internet de question-réponse. Dans notre investigation des travaux existants liés à l'identification des communautés dans le Web, nous avons constaté qu'il n'y a pas une définition quantitative unique et rigoureuse, qui est universellement acceptée, du concept de communauté. Cela est dû essentiellement au fait que la définition du terme communauté varie selon le contexte de chaque application. Il convient de noter que le travail présenté dans ce chapitre est différent de tous les travaux antérieurs liés à l'identification des communautés dans le Web, puisque nous étudions le problème d'identification des communautés dans le contexte particulier des forums Internet de question-réponse.

Dans notre travail, nous somme intéressés à identifier ce que nous appelons "les communautés de partage des connaissances" qui satisfont les propriétés suivantes :

1. Une communauté de partage des connaissances est définie par un ensemble d'experts et d'utilisateurs qui posent des questions.
2. À l'intérieur de chaque communauté, les utilisateurs qui posent des questions montrent un comportement plus homogène, en terme de leurs interactions avec les experts, que partout ailleurs.

3. Les utilisateurs experts peuvent appartenir à plus d'une communauté.

À partir de la description des communautés de partage des connaissances mentionnée ci-dessus, il est clair que l'identification de ce type de communautés repose en premier lieu sur l'identification des experts. Pour cette fin, nous avons utilisé nos résultats obtenus dans le chapitre 2 de cette thèse. Spécifiquement, pour identifier les communautés de partage des connaissances, nous avons exploré les interactions des participants avec tout les experts identifiés par notre approche présentée dans le chapitre précédant. Nous proposons de représenter les interactions entre les utilisateurs qui posent des questions et les experts sous la forme de données transactionnelles de telle sorte que chaque transaction résume toutes interactions d'un utilisateur avec tous les experts qui ont répondu à ces questions. Par la suite, pour regrouper les utilisateurs qui montrent un comportement homogène en terme de leurs interactions avec les experts, nous proposons un nouvel algorithme de *clustering* de données transactionnelles que nous avons nommé TRANCLUS (TRANsaction CLUStering). Les *clusters* identifiés par notre algorithme représentent les communautés que nous cherchons à découvrir.

Contrairement à la vaste majorité des algorithmes existants de *clustering* des données transactionnelles, TRANCLUS est libre de tout paramètre ce qui représente un avantage majeur en pratique. TRANCLUS est un algorithme itératif de nature partitionnelle qui tente d'optimiser une nouvelle fonction objectif. Spécifiquement, pour identifier les *clusters*, TRANCLUS commence premièrement par parcourir l'ensemble des transactions de manière séquentielle de telle sorte que la destination de la prochaine transaction est guidée par notre nouvelle fonction objectif. Une fois que la première passe à travers tout l'ensemble de données est effectuée, TRANCLUS effectue quelques autres passes pour raffiner le *clustering*. La fonction objectif que nous proposons permet d'effectuer un processus de *clustering* systématique en excluant toutes interventions de l'utilisateur. Notre algorithme est évalué sur des ensembles de données synthétiques et réelles de caractéristiques différentes. Comme une démonstration pratique, nous utilisons TRANCLUS pour identifier les communautés de partage des connaissances dans six forums différents de Yahoo ! Answers. Notre contribution est présentée dans les pages suivantes à travers un article intitulé **Discovering Knowledge-Sharing Communities in Question-Answering Forums** [7]. Cet article est soumis au journal international **ACM Transactions on Knowledge Discovery form Data, Special Issue on Knowledge Discovery for Web Intelligence**.

Note : ma contribution au chapitre 3 de cette thèse inclut l'introduction du concept de " communautés de partage de connaissances ", le développement de l'algorithme TRANCLUS, l'élaboration des tests expérimentaux et la rédaction de l'article. Mon directeur de recherche, le professeur Shengrui Wang et mon superviseur de stage chez Yahoo ! Canada, M. Benoit Dumoulin, ont conjointement supervisé et validé toutes les étapes de développement de l'approche proposée dans ce chapitre ainsi que la rédaction de l'article.

Discovering Knowledge-Sharing Communities in Question-Answering Forums

MOHAMED BOUGUESSA and SHENGRUI WANG
University of Sherbrooke
and
BENOIT DUMOULIN
Yahoo! Inc.

In this paper, we define what we call a knowledge-sharing community in a question-answering forum as a set of askers and authoritative users such that, within each community, askers exhibit more homogeneous behavior in terms of their interactions with authoritative users than elsewhere. A procedure for discovering members of such a community is devised. As a case study, we focus on Yahoo! Answers, a large and diverse online question-answering service. Our contribution is twofold. First, we develop a method for automatic identification of authoritative actors in Yahoo! Answers. To this end, we estimate and then model the authority scores of participants as a mixture of gamma distributions. The number of components in the mixture is determined using the Bayesian Information Criterion (BIC), while the parameters of each component are estimated using the Expectation-Maximization (EM) algorithm. This method allows us to automatically discriminate between authoritative and non-authoritative users. Second, we represent our environment as a type of transactional data such that each transaction summarizes the interaction of an asker with a specific set of authoritative users. Then, to group askers on the basis of their interactions with authoritative users, we develop a parameter-free transactional clustering algorithm which is based on a novel criterion function. The identified clusters correspond to the communities that we aim to discover. To evaluate the suitability of our clustering algorithm, we conduct a series of experiments on both synthetic and public, real-life data. Finally, we put our approach to work using data from Yahoo! Answers which represent users' activities over one full year.

Categories and Subject Descriptors: H.5.3 [**Information Interfaces and Presentation (e.g., HCI)**]: Group and Organization Interfaces—*Web-based interaction*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Design, Experimentation

1. INTRODUCTION

Internet surfers use the Web to find various information related to a wide range of topics. In this context, Internet search engines fulfill a very important role, as they are used to find a series of relevant documents relative to a specific topic. Internet search engines are extremely useful and become an indispensable tool for most Web users.

A part of this work was done while the first author was at Yahoo! as an intern.
Author's address: Mohamed Bouguessa and Shengrui Wang: Department of Computer Science, University of Sherbrooke, 1500 Boulvard de l'Université , Qc, J1K 2R1, Canada;
contact email:{mohamed.bouguessa, shengrui.wang}@usherbrooke.ca.
Benoit Dumoulin: Yahoo! Inc., 2821 Mission College Boulevard, Santa Clara, CA 95054, USA;
contact email: benoitd@yahoo-inc.com.

Web users are also constantly looking for new online services to complement search engines. Among these services are question-answering forums where users come to ask and answer questions and share their knowledge. There are generally three types of users in question-answering forums [Zhang et al. 2007]: 1) users who only ask questions; 2) users who only answer questions and 3) users who ask and answer questions. It is reasonable to expect that a user who correctly answers another user's question generally has more expertise on the subject than the asker; and an asker prefers to receive answers from authoritative people with expertise on the specific subject.

Summarizing the analysis of some previous studies [Zhang et al. 2007; Welser et al. 2007] in question-answering forums the primary mode of interaction of authoritative users is the provision of thorough help and informative responses to other group members' questions. In this context, askers may seek authoritative people as a source of information, to replace other sources such as documents and databases or complement them in various ways [Maybury 2006]. Such interaction between participants in these online services allows the formation of communities around shared interests in a wide variety of topics [Welser et al. 2007].

Recent research [Maybury 2006; Yimam and Kobsa 2003] suggests that authoritative users can play a critical role in fostering and sustaining Web communities. In this context, communities can be viewed as groups of askers and authoritative users, with the latter constituting the core of these communities since they are the source of the knowledge sought by the askers. Such communities are often seen in the business world as important means for generating value and motivating contributions [Welser et al. 2007; Lesser and Storck 2001]. The hosts and users of online question-answering services would like to be able to identify the providers of the most valuable information, and to discover and promote communities that support such sharing [Wenger et al. 2002; Lesser and Storck 2001]. In this paper, we effectively address this issue. As a case study, we focus on Yahoo! Answers [1], a large and diverse online question-answering service.

Yahoo! Answers is a very popular service and already reports millions of participants. A few months after it was launched, it attracted a large number of users and it continues to grow [Gyongyi et al. 2008]. According to Prescott [2006], Yahoo! Answers captures 96% of the question and answer market share. A recent study [Adamic et al. 2008] suggests that Yahoo! Answers is an active social world with tremendously diverse knowledge and opinions being exchanged. This success can be attributed to the wide variety of topics available at Yahoo! Answers and to the significant number of participants with different skills and expertise. This makes Yahoo! Answers an interesting case study, since it contains a rich store of information. In this context, locating different sources of authoritative knowledge, and discovering community structure become crucial issues to improve the content and the quality of the service that Yahoo! Answers offers its users.

[1] http://answers.yahoo.com/

1.1 Goal

Our goals are:

(1) To develop a systematic approach to automatically discriminate between authoritative and non-authoritative users.

(2) To develop a clustering algorithm targeted to discover "knowledge sharing communities" based on the interactions between askers and authoritative users. Specifically, we aim to cluster askers who exhibit homogenous behaviors in terms of their interaction with authoritative users.

In the following pages, we present a number of elements that motivate our study. First, we illustrate the benefit of discovering authoritative users. Second, we provide a definition of the knowledge-sharing communities we aim to discover, and illustrate why discovering communities based on the interactions between askers and authoritative users helps to improve the quality of the services that any question-answering forum offers to its participants. A brief description of the strategy that we have adopted to identify such communities is given. Finally, we conclude this section by describing our contributions and the plan of this paper.

1.2 Discovering Authorities

In addition to the discussion above about the benefit of discovering authoritative users, there are several other reasons for identifying such actors in question-answering forums. For instance, routing each newly asked question to appropriate experts significantly helps in providing askers with efficient, helpful service, by minimizing the effort and time askers must invest to find or receive good answers to their questions. In this context, authoritative people perform an important role because they collectively donate vast amounts of valuable advice to those who ask questions. Furthermore, authoritative users can participate effectively in improving the quality of the site content.

The problem of identifying authoritative users has received growing attention [Zhang et al. 2007; Campbell et al. 2003; Dom et al. 2003]. In an attempt to discover authoritative users, most of the existing approaches use link analysis techniques such as HITS [Kleinberg 1999], or PageRank [Page et al. 1998] or one of its variants such as AuthorRank [Liu et al. 2005] and ExpertiseRank [Zhang et al. 2007]. The output of such approaches is a list of users ranked according to their degree of authority on subjects of interest. Based on this list, the top K users are considered as most authoritative. The weakness of such an approach resides in the unprincipled selection of the value of K. In general, the value of K is chosen solely on the basis of specific knowledge of an application. However, in many real-life applications, it is very difficult to set the value of K. For instance, since there are a large number of different forums in Yahoo! Answers, setting an appropriate value of K for each forum is a very difficult task if performed manually, by inspecting users' behavior for each of them. Automating the process of discovering authoritative actors thus becomes an absolute necessity. To summarize, the value of K is crucial, as it can give more power to the selected users. An inappropriate choice of this value can have a very negative impact on the quality of the service.

To our knowledge, no principled method for choosing the value of K has yet been

proposed. This motivated our effort to design an approach for automatically discriminating between authoritative and non-authoritative users, rather than simply producing a ranked list. Once authoritative users are discovered, we turn to the problem of identifying communities that form around them.

1.3 Discovering Communities

Knowledge is a critical asset that needs to be managed strategically [Wenger et al. 2002]. Increasingly, organizations are investing in knowledge-management solutions to manage and leverage both implicit and explicit knowledge assets. Community detection and expertise location have become important aspects of knowledge-management systems [Salvetti and Srinivasan 2005]. Likewise, in question-answering forums, managing knowledge is principally based on discovering authoritative users who represent a potential source of knowledge, and exploring their interactions with askers in order to detect communities. Such a strategy leads to the creation of a valuable online knowledge service. This is why, in this paper, we focus exclusively on the interactions between askers and authoritative users rather than any other type of interactions between participants. Focusing on interactions between askers and authoritative users is an appropriate way to discover and promote communities in which seeking and sharing knowledge is the first priority.

The identification of such communities allows closer interaction and communication between participants in the group as people learn from one another, solve problems together and create new knowledge. Furthermore, in question-answering forums, enhancing the visibility of authoritative users on the site and connecting them with askers plays a critical role in fostering communities around shared interests. As a result, there is a considerable gain in terms of the time and effort spent by askers to search for relevant information, since another person (i.e., authoritative user) in the community could easily provide assistance. Furthermore, discovering and cultivating such communities results in a much greater degree of orderly high-level structure. Hence, groups (i.e., communities) can grow in a more structural way by attracting new members with the same focus of interest. Learning would be the reason that motivates the community to come together [Wenger et al. 2002]. As new members join the group they have access to existing members and learn from them. Members of a community may also take advantage of their connections to get to know others. In summary, discovering communities based on the interactions between askers and authoritative users is crucial to improve the quality of the content of question-answering forums.

In this paper, we use a transactional data model to represent the interactions between askers and authoritative users. In general, the term "transaction" refers to a set of "items" [Han and Kamber 2006]. A transactional dataset consists of a number of transactions, each of which contains a varying number of items. The most common example of transactional data is market basket data, which consists of the sets of items bought together by customers. One such set of items is called a transaction. Other typical examples of transactional data are Web usage data, customer interest profiles, patient symptoms records, and image features. Actually, transactional data occurs in many different applications such as bioinformatics, medical diagnosis, scientific data analysis and Web mining [Tan et al. 2006]. For instance, in information retrieval, a document can be summarized by a set of key-

$T_1 = \{e_1, e_2\}$ $T_5 = \{e_3, e_4, e_5, e_6\}$
$T_2 = \{e_1, e_2, e_3\}$ $T_6 = \{e_3, e_4, e_5\}$
$T_3 = \{e_1, e_2, e_3\}$ $T_7 = \{e_3, e_4, e_5, e_6\}$
$T_4 = \{e_2, e_3\}$ $T_8 = \{e_4, e_5, e_6\}$

Fig. 1. An example of a transactional dataset.

words associated with it. Thus, Web search engines could represent documents (and even user queries) as a type of transaction [Xiao and Dunham 2001].

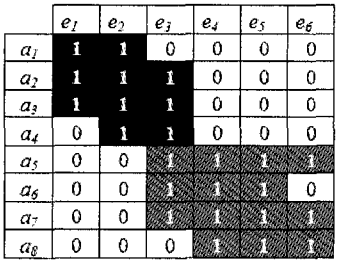
In the context of our study, we represent the interactions between an asker and authoritative users as a type of transaction. Specifically, we associate with each asker a a transaction T that contains a set of authoritative users who answered the questions of asker a. Figure 1 provides a simple example of interactions between askers and authoritative users. In this figure, each e_d $(d = 1, \ldots, 6)$ corresponds to an authoritative user and each set T_i $(i = 1, \ldots, 8)$ summarizes the interaction of asker a_i with a specific set of authoritative users. For instance, the transaction $T_1 = \{e_1, e_2\}$ summarizes the interactions of asker a_1 with the authoritative users e_1 and e_2. Note that in transactional data the size of the transactions varies. To summarize, the transactional data model offer a simple and flexible way to represent the interaction between askers and authoritative users. In Section 4, we will give a formal definition of a transaction and a formal description of the data model that we use.

An interesting property of the interaction between an asker and authoritative users is that users are usually linked by shared interests and not social relationships (i.e., friendship). Askers prefer to receive answers from authoritative people with expertise on the specific subject. An authoritative user provides answers to: 1) questions for which he/she is interested by the content rather than who posted the question; 2) questions related to his/her area(s) of expertise. Hence, we expect that askers who interact frequently with the same set of authoritative users are also more related to each other than those who do not. As a result, the interactions between askers and authoritative users leads to the formation of communities of shared interests in a wide variety of topics. In each community, we view authoritative users as the central/prominent elements that the askers seek.

The concept of a community is qualitatively intuitive [Radicchi et al. 2004]. As suggested in [Zhang et al. 2007; Balakrishnan and Deo 2006; Radicchi et al. 2004], there is no single quantitative, rigorous definition of a community that is commonly accepted. This is mainly due to the fact that the definition of a community varies according to the specific case under study [Caldarelli 2007]. In this paper, we are interested in identifying what we call "knowledge-sharing communities" in question-answering forums, which satisfy the following properties:

(1) A knowledge-sharing community is defined by a set of askers and authoritative users.
(2) Within each community, askers exhibit more homogeneous behavior in terms of their interactions with authoritative users than elsewhere.
(3) Authoritative users may belong to more than one community.

As discussed above, the reason for the first property is primarily that we aim



	e_1	e_2	e_3	e_4	e_5	e_6
a_1	1	1	0	0	0	0
a_2	1	1	1	0	0	0
a_3	1	1	1	0	0	0
a_4	0	1	1	0	0	0
a_5	0	0	1	1	1	1
a_6	0	0	1	1	1	0
a_7	0	0	1	1	1	1
a_8	0	0	0	1	1	1

Fig. 2. Boolean representations of the interaction between askers and authoritative users.

to create and promote a valuable online knowledge-sharing service. The second property is based on the fact that two askers who interact with a lot of common authoritative users are more likely to be associated than other askers who do not. For instance, consider the interactions of askers a_1, a_2 and a_5 which are summarized respectively by the transactions T_1, T_2 and T_5, as depicted in Figure 1. As we can see from this figure, T_1 and T_2 exhibit a significant overlap in comparison to T_5. This implies that askers a_1 and a_2 are more highly associated with each other than asker a_5 since their interactions with authoritative users are more similar than those of a_5. Finally, the third property is based on the observation that in question-answering forums there may be some authoritative users who answer a lot of questions related to their specialties and thus it is possible that we may find them in more than one group.

Based on the discussion above, a simple visual inspection of Figure 1 suggests the existence of two communities. The first community is defined by the set T_1, T_2, T_3, T_4 while the second one is defined by T_5, T_6, T_7, T_8. This means that in the first community, askers a_1, a_2, a_3 and a_4 are grouped together since they interact with a common set of authoritative users: e_1, e_2 and e_3; whereas askers a_5, a_6, a_7 and a_8 are grouped in the second community since they interact most often with e_3, e_4, e_5 and e_6. This can be better seen in the bitmapped representation depicted in Figure 2. In this figure, the interaction between askers and authoritative users can be viewed as a Boolean matrix in which a value 1 corresponds to the presence of the interaction between asker a_i and authoritative users e_d, while a value 0 corresponds to its absence. The black cells of this matrix indicate askers and authoritative users that form the first community while the gray cells indicate the second community. It is important to note that we have used a Boolean matrix here to summarize the interaction between askers and authoritative users for the purpose of visual clarification only. Modeling interactions between users as a type of transaction is more appropriate, since transactional data allow a compact representation, thereby avoiding the use of a sparse, high-dimensional matrix which poses significant challenge to any mining techniques in terms of time and space complexity.

As can be seen from Figure 2, askers grouped in the same community exhibit homogeneous behavior in terms of their interactions with authoritative users, in comparison to other askers in other groups. This implies that since in this paper

we model our environment as types of transactions, a community could be defined as a subset of transactions that exhibits a higher degree of overlap between them than elsewhere. This in turn corresponds to the definition of a cluster in the context of transactional data clustering [Cesario et al. 2007; Yan et al. 2006; Yang et al. 2002; Wang et al. 1999]. In this setting, transactional clustering algorithms are appropriate for discovering knowledge-sharing communities. In the remainder of this paper, the terms "community" and "cluster" will be used interchangeably. A formal description of a cluster is given in Section 4.

A number of transactional clustering algorithms have been proposed in the literature [Yan et al. 2006; Yang et al. 2002; Wang et al. 1999]. However, the majority of these techniques are dependent on multiple parameters which may be difficult to tune, especially in real-life applications. For instance, some algorithms [Yun et al. 2004; Giannotti et al. 2002; Xiao and Dunham 2001] require the number of clusters as an input parameter. However, with real datasets it is extremely difficult to set an appropriate value for the number of clusters. To alleviate this problem, the authors in [Yan et al. 2006] propose two new validity indices and suggest running the clustering algorithm with different numbers of clusters and choosing the number of clusters that optimizes the proposed validity index. However, such an approach penalizes the efficiency of the algorithm and further inflates its time requirements. Some existing algorithms [Yang et al. 2002; Wang et al. 1999] succeed in automatically identifying the number of clusters. However, they still require a number of parameters to be tuned by the user, which limits their applicability to real problems. In real applications, clustering algorithms should have as few parameters as possible - ideally, none at all [Keogh et al. 2004]. To our knowledge, only the algorithm proposed in [Cesario et al. 2007] is a parameter-free transactional clustering algorithm.

1.4 Contributions and Paper Layout

The major contributions [2] of this paper are:

(1) Proposing a systematic approach to identify authoritative users in question-answering forums. In our approach, we propose to model the authority scores of users as a mixture of gamma distributions. The number of components in each mixture is estimated by the Bayesian Information Criterion (BIC), while the parameters of each component are estimated by the Expectation-Maximization (EM) algorithm. Based on extensive experiments on datasets extracted from Yahoo! Answers, we found that authority scores can be modeled as a mixture of two gamma distributions. As we will show, one of these two components corresponds to authoritative actors.

(2) Proposing a fully automatic approach for discovering knowledge-sharing communities in question-answering forums based on the interactions between askers and authoritative users. Specifically, we represent such interactions between users as transactions and we develop a parameter-free transactional clustering algorithm which is based on a new criterion function. We call our algo-

[2] This paper is a substantially expanded journal version of the ACM Knowledge Discovery and Data Mining Conference paper [Bouguessa et al. 2008].

rithm TRANCLUS to denote the fact that it is a TRANsaction CLUStering algorithm. The algorithm aims to cluster askers that exhibit homogeneous behaviors based on their interaction with authoritative users. We illustrate the suitability of our proposal on real data from Yahoo! Answers.

The rest of this paper is organized as follows. Section 2 describes Yahoo! Answers. Section 3 gives a detailed description of our approach for the identification of authoritative users. In Section 4, we present the TRANCLUS algorithm. An empirical evaluation of TRANCLUS is given in Section 5. Section 6 presents the application of our approach to discover knowledge-sharing communities in Yahoo! Answers. In Section 7, we provide an overview of related work. Finally, our conclusion is given in Section 8.

2. YAHOO! ANSWERS

The purpose of this section is to provide the reader with the necessary background to understand the main characteristics and the question-answering mechanism of Yahoo! Answers [3]. Yahoo! Answers is an online question-answering service organized according to a taxonomy of topics. Specifically, questions and answers are posted within categories (or forums). There are 26 top-level and more than 1000 lower-level categories in Yahoo! Answers. The categories range from Mathematics to History to Medicine to Philosophy. The content of Yahoo! Answers is therefore highly multidisciplinary and attracts users from a wide variety of fields. As a result, it now hosts a very large number of questions and answers in a wide variety of domains. Yahoo! Answers participants can thus save time in their quest for information because they can get an answer relatively quickly or find what they are looking for among the existing questions and answers.

The central elements of the Yahoo! Answers system are the questions. Each question has a life cycle. It starts in an "open" state where it receives several answers. Then at some point (determined by the asker, or by an automatic timeout in the system), the question is considered "closed", and can receive no further answers. At this stage, a "best answer" is selected either by the asker or by other users via a voting procedure. Once a best answer is chosen, the question goes into the "resolved" state and becomes, in principle, a browsable piece of information. Note that Yahoo! Answers participants do not limit their activity to asking and answering questions, but also actively participate in regulating the whole system. A user can report abusive behavior by other users who are violating the guidelines. A user can also mark interesting questions, evaluate answers and vote for the best answers. Finally, it is important to note that during the question life cycle described here, a user is not allowed to answer the same question more than once, nor to answer his/her own question. Consequently, Yahoo! Answers does not support real-time interactions and discussions.

Recently, the characteristics and dynamics of Yahoo! Answers have been studied in [Adamic et al. 2008; Gyongyi et al. 2008]. The main focus of these studies is to analyze Yahoo! Answers' knowledge-sharing activity and to understand the

[3]Note that the description of the operating mechanism of Yahoo! Answers provided in this paper is based on current service policies that may change in the future.

behavior of its participants by investigating their activity levels, rules, interests, etc. For instance, Adamic et al. [2008] analyze the interactions between Yahoo! Answers participants across a number of different categories. The authors suggest partitioning Yahoo! Answers categories into three groups. The first group contains categories in which users are more engaged in expressing their opinions (e.g., the Politics category). The second group contains categories in which people both seek and provide advice and commonsense expertise on questions where there may be several legitimate answers or no single factual answer (e.g., the Marriage category). The third group contains categories in which users ask focused questions that require factual answers (e.g., the Physics category). Quite similar results were also found in [Gyongyi et al. 2008].

The considerable diversity of categories described above suggests that Yahoo! Answers participants have different interests and behaviors. In fact, as observed in [Adamic et al. 2008; Gyongyi et al. 2008], some users focus narrowly on specialized, technical categories (e.g., Science & Mathematics) while others participate in general categories (e.g., Family & Relationship, Politics & Government). Participants in general categories tend to ask and answer a large number of questions. This is mainly due to the fact that the nature of questions in such categories does not require factual answers; participants are more engaged in expressing their opinions and providing advice. On the other hand, in specialized technical categories (what are usually called question-answer forums, as suggested in [Adamic et al. 2008]), askers ask concrete questions and expect similarly factual answers, ideally given by an authoritative user who has a particular expertise in the domains related to the main focus of these categories. This suggests that such categories are a valuable source of knowledge and they seem particularly well suited to the task of our study, as participants usually communicate what they know. In this paper, we focus on categories in which knowledge sharing and factual expertise are sought (e.g, categories such as biology, programming, physics, etc.). For a given category, our goal is to identify authoritative users and then to discover communities that form around them.

3. AUTOMATIC IDENTIFICATION OF AUTHORITATIVE USERS

This section addresses the problem of automatic discrimination between authoritative and non-authoritative users. First, we illustrate how to estimate a user's level of authority. Second, based on the estimated authority score of each user, we develop a systematic and efficient way to discriminate between authoritative and non-authoritative users.

3.1 Estimating the Authority Score

One of the most important features of the question life cycle described in Section 2 is the selection of the best answer. In specialized technical forums, askers ask focused questions, triggered by concrete information needs [Gyongyi et al. 2008]. Often, askers take the time to choose the best answer, since they are more interested in the content and the quality of the answer than in who posted it. When an asker chooses a best answer, he provides a quality rating for the answer. In our manual investigation of questions and answers from the categories Mathematics and Programming & Design, we found that most of the best answers selected by

askers were indeed best answers. Similar results were also found by Adamic et al. [2008] in their investigation of the categories Programming & Design, Cancer and Celebrity. This indicates that the best-answer selection mechanism reflects choices people make about what information is useful, interesting and authoritative.

For instance, when an asker A chooses the answer of a user B as best answer, this denotes an endorsement of the quality of B's answer. Here, it is clear that user B has a particular skill, experience, or expertise on a specific subject that user A does not have. Intuitively, users who have some expertise in a specific topic tend to have their answers to questions related to their specialties selected as best frequently [Adamic et al. 2008]. In this setting, we expect that the number of best answers of a user in a specific category is a potential measure to estimate the authority level of each user in each category.

Indeed, in categories where requests for factual answers dominate, it is reasonable to assume that the fact that a user has a high number of best answers is a potential indicator of authority. A similar suggestion was also made in [Adamic et al. 2008]. On the other hand, in categories in which users are more engaged in expressing their opinions or providing advice, the best-answer criterion is more appropriate for mining some relationship (social connections) between participants rather than estimating the level of authority of each user. For instance, in categories in which users are more engaged in expressing their opinions (e.g., the Politics category) the best answer may be the one that agrees with the asker's opinions. Exploring such information can help, for example, to build a network of friends. We will consider this issue in our future work. As mentioned earlier, in this paper we focus on categories in which users are more engaged in seeking and sharing knowledge.

One should be aware that it is possible for spammers to create several accounts with different pseudonyms. Such users can ask and answer their own questions and choose their answers as best answers. Here, it is clear that this will result in an increase in the number of best answers for these users. However, such behavior does not have a major impact on the estimation of authority level. This is because 1) we have observed that such spamming behavior of users is less often found in technically focused categories; and 2) due to the small number of users displaying such behavior in these categories, Yahoo! Answers participants will usually identify them and report such so-called "abusive behaviors". As a consequence, the accounts of such users are deleted from the system by moderators.

Based on the discussion above, we surmise that the number of best answers is appropriate for estimating user's level of authority. We turn now to the problem of identifying authoritative users. A very simple approach is to rank users based on their number of best answers and select the top K users as authorities. However, as described in Section 1, in some applications such as Yahoo! Answers, it is difficult to set appropriate values for K. Setting an inappropriate value of K may greatly affect the result. In addition, since there are a significant number of different technical categories in Yahoo! Answers, manually setting the appropriate value of K for each category is not possible. In an attempt to provide an automatic solution to this problem, we will now develop a systematic and efficient way to discriminate between authoritative and non-authoritative users.

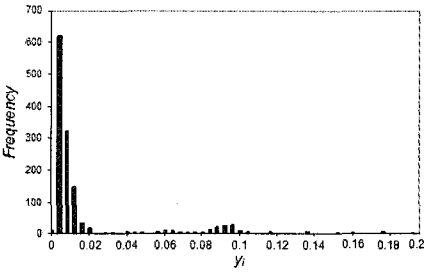


Fig. 3. Histogram of the authority scores.

3.2 Approach

Let $U = \{u_1, u_2, \ldots, u_N\}$ be a set of N users having at least one best answer in a specific category CAT. The authority score of a user u_i is denoted by y_i, where y_i is the number of the best answers for user u_i in category CAT. We normalize the authority score in such a way that their square sum to 1: $\sum_{i=1}^{N}(y_i)^2 = 1$. The normalized authority scores provide a relative measure of the authority of each user in each category. Intuitively, a large value of y_i means that u_i belongs to the set of authoritative users, while a small value indicates that u_i belongs to the set of non-authoritative users. In order to identify authorities in each category, we are interested in all sets of u_i having large values of y_i.

Estimating the histogram is a flexible tool to describe some statistical properties of the normalized authority score y_i. For purposes of clarification, we extract data from the category "Engineering". The histogram of the authority score of users in this category is given in Figure 3. As we can see, the histogram depicted in this figure suggests the existence of two components. One of these two components represents low values of y_i ($y_i < 0.04$), while the other one represents large values of y_i ($0.04 < y_i < 0.2$). We surmise that the first component represents non-authoritative users while the second one represents authoritative users. The fundamental question now is how to formally distinguish between authoritative and non-authoritative users. For this purpose we propose to model the authority score y_i of all the users in a specific category as a mixture distribution. The probability density function (*pdf*) is therefore estimated and the status of each user in a specific category is identified.

3.2.1 *pdf Estimation.* The histogram depicted in Figure 3 suggests the existence of components with different shapes and heavy tails. Based on this observation, and in order to fit the variation of the authority score well, it is more appropriate to use a statistical distribution model which is suitable for dealing with such shape variations. Among the existing models in the literature, the gamma mixture model is the most appropriate one. This is because the gamma distribution involves a shape parameter α (> 0) which allows the distribution to take on a variety of shapes, depending on its value [Balakrishnan and Nevzorov 2003]. For instance, when $\alpha < 1$, the distribution is highly skewed and L-shaped. When $\alpha = 1$, we get the exponential distribution. In the case of $\alpha > 1$, the distribution has a peak

(mode) at $(\alpha-1)/\beta$ and a skewed shape. The skewness decreases as the value of α increase. Note that β denotes the scale parameter which is the second parameter of the gamma distribution. In summary, as suggested in [Yang 1996], the gamma distribution is sufficiently general to fit several situations. This flexibility of the gamma distribution and its positive sample space make it particularly suitable for modeling the distribution of the normalized authority score y_i.

Formally, we expect that the authority score follows a mixture density of the form:

$$G(y) = \sum_{l=1}^{m} \gamma_l G_l(y, \alpha_l, \beta_l) \tag{1}$$

where $G_l(.)$ is the lth gamma distribution with parameters α_l and β_l, representing, respectively, the shape and the scale parameters of the lth component; and $\gamma_l (l = 1, \ldots, m)$ are the mixing coefficients, with the restriction that $\gamma_l > 0$ for $l = 1, \ldots, m$ and $\sum_{l=1}^{m} \gamma_l = 1$. The density function of the lth component is given by

$$G_l(y, \alpha_l, \beta_l) = \frac{\beta_l^{\alpha_l}}{\Gamma(\alpha_l)} y^{\alpha_l - 1} \exp(-\beta_l y) \tag{2}$$

where $\Gamma(\alpha_l)$ is the gamma function given by $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} \exp(-t)dt; \;\; t > 0$.

The use of a mixture of gamma distributions allows us to propose a flexible model to describe the distribution of the authority score y_i. To form such a model, we need to estimate m, the number of components, and the parameters for each component. A standard approach for estimating the parameters of the gamma components G_l is the maximum likelihood technique [Hogg et al. 2005]. The likelihood function is defined as

$$L_{G_l}(\alpha_l, \beta_l) = \prod_{y \in G_l} G_l(y, \alpha_l, \beta_l) = \frac{\beta_l^{\alpha_l N_l}}{\Gamma^{N_l}(\alpha_l)} \prod_{y \in G_l} y^{\alpha_l - 1} \exp(-\beta_l \sum_{y \in G_l} y) \tag{3}$$

where N_l is the size of the lth component. The logarithm of the likelihood function is given by

$$\log(L_{G_l}(\alpha_l, \beta_l)) = N_l \alpha_l \log(\beta_l) - N_l \log(\Gamma(\alpha_l)) + (\alpha_l - 1) \sum_{y \in G_l} \log(y) - \beta_l \sum_{y \in G_l} y \tag{4}$$

To find the values of α_l and β_l that maximize the likelihood function, we differentiate $\log(L_{G_l}(\alpha_l, \beta_l))$ with respect to each of these parameters and set the result equal to zero:

$$\frac{\partial}{\partial \alpha_l} \log(L_{G_l}(\alpha_l, \beta_l)) = N_l \log(\beta_l) - N_l \frac{\Gamma^{'}(\alpha_l)}{\Gamma(\alpha_l)} + \sum_{y \in G_l} \log(y) = 0$$

$$\Rightarrow -\log(\beta_l) + \frac{\Gamma^{'}(\alpha_l)}{\Gamma(\alpha_l)} = \frac{1}{N_l} \sum_{y \in G_l} \log(y) \tag{5}$$

and

$$\frac{\partial}{\partial \beta_l} \log(L_{G_l}(\alpha_l, \beta_l)) = \frac{N_l \alpha_l}{\beta_l} - \sum_{y \in G_l} y = 0$$

$$\Rightarrow \beta_l = \frac{N_l \alpha_l}{\sum_{y \in G_l} y} \tag{6}$$

This yields the equation

$$\log(\alpha_l) - \Psi(\alpha_l) = \log(\frac{1}{N_l} \sum_{y \in G_l} y) - \frac{1}{N_l} \sum_{y \in G_l} \log(y) \tag{7}$$

where $\Psi(.)$ is the digamma function given by $\Psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$. The digamma function can be approximated accurately using the following equation[Lawless 1982]:

$$\Psi(\alpha) = \log(\alpha) - \frac{1}{2\alpha} - \frac{1}{12\alpha^2} + \frac{1}{120\alpha^4} - \frac{1}{252\alpha^6} + \ldots \tag{8}$$

The parameter $\hat{\alpha_l}$ can be estimated by solving equation (7) using the Newton-Raphson method. $\hat{\alpha_l}$ is then substituted into equation (6) to determine $\hat{\beta_l}$.

Let us focus now on the problem of estimating m, the number of components in the distribution. Based on the histogram depicted in Figure 3, we can assume that the authority score could be modeled as a mixture of two gamma components. However, this is just a visual observation; there is no proof which guarantees that there are indeed two components. To address this issue, we have adopted the following strategy for estimating the optimal number of components in a mixture.

In fact, the popular approach for estimating the number of components m is to test values of m from 1 to m_max (m_max is an input parameter which represents the maximum number of components) against a performance measure and choose the number of components that optimizes the performance measure. For this purpose, we implement a standard two-step process. In the first step, we calculate the maximum likelihood of the parameters of the mixture for a range of values of m (from 1 to m_max). The second step involves calculating an associated criterion and selecting the value of m which optimizes the criterion. A variety of measures have been proposed to estimate the number of components in a dataset [Bouguessa et al. 2006; Oliver et al. 1996]. In our method, we use a penalized likelihood criterion, called the Bayesian Information Criterion (BIC). BIC was first introduced by Schwarz [1978] and is given by

$$\text{BIC}(m) = -2L_m + N_p \log(N) \tag{9}$$

where L_m is the logarithm of the likelihood at the maximum likelihood solution for the mixture model under investigation and N_p is the number of parameters estimated. The number of components that minimizes BIC(m) is considered to be the optimal value for m.

Typically, the maximum likelihood of the parameters of the distribution is estimated using the Expectation-Maximization (EM) algorithm [Dempster et al. 1977]. This algorithm requires the initial parameters of each component. Since EM is highly dependent on initialization [Jain et al. 2000], it will be helpful to perform initialization by mean of a clustering algorithm [Figueiredo and Jain 2002]. For this purpose we make use of the Fuzzy C-Means algorithm (FCM) [Bezdek 1981] to partition the set of y_i into m components. Based on such a partition we can estimate the parameters of each component and set them as initial parameters for

Algorithm 1: Estimation of the number of components m.

Input : $\{y_i\}$, m_max
Output: The optimal number of components m

```
1  begin
2      for m = 1 to m_max do
3          if m==1 then
4              Estimate the parameters α and β using equations (6), (7) and (8);
5              Compute the value of BIC(m) using equation (9);
6          else
7              Apply the FCM algorithm as an initialization of the EM algorithm;
8              Apply the EM algorithm to estimate the parameters of the mixture: α_l, β_l
               using equations (6), (7) and (8);
9              Compute the value of BIC(m) using equation (9);
10     Select the number of components m̂, such that m̂ = arg_min BIC(m);
11 end
```

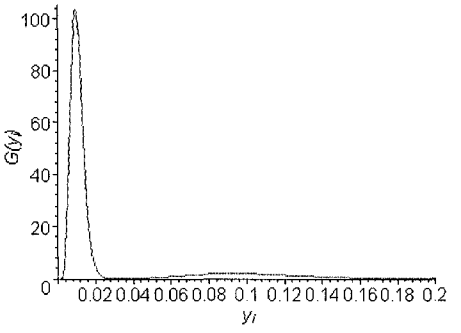


Fig. 4. *pdf* of the authority score.

the EM algorithm. The procedure for estimating the number of components is summarized in Algorithm1.

Based on extensive experiments on data extracted from a large number of categories from Yahoo! Answers, we found that the optimal number of components in a mixture is always two. This suggests that the authority score are well fitted by two gamma components, the first representing non-authoritative users with small values of y_i while the second represents authoritative users with large values of y_i. The *pdf* of the authority scores represented in Figure 3 is illustrated in Figure 4.

3.2.2 *Summary of Authoritative Users Identification Procedure.* Based on the above analysis, it is reasonable now to assume that the authority scores could be modeled as a mixture of two gamma components. The steps described in Algorithm 2 can then be implemented to efficiently discriminate authoritative from non-authoritative users. Based on this algorithm, we can definitively discriminate between authoritative and non-authoritative users in a specific category by selecting the component which represents large values of y_i.

Algorithm 2: Identifying authoritative users.

Input : A set $U = \{u_1, u_2, \ldots, u_N\}$ of N users
Output: A set $E = \{e_1, e_2, \ldots, e_d\}$ of d authoritative users

```
1 begin
2     For a given category, estimate the authority score y_i of each user;
3     Normalize y_i, where sum_{i=1}^{N} (y_i)^2 = 1;
      // Estimate the pdf of the authority scores with m = 2
4     Apply FCM as initialization of the EM algorithm;
5     Apply EM to estimate the parameters of the mixture;
6     Use the results of the EM algorithm in order to derive a classification decision about
      the membership of y_i in each component;
7 end
```

4. THE ALGORITHM TRANCLUS

Let us now focus on developing an algorithm capable of discovering communities that form around authoritative users. For this purpose, we introduce TRANCLUS, a transactional clustering algorithm that maps the problem of discovering knowledge-sharing communities in question-answering forums to the problem of clustering transactions. In a nutshell, TRANCLUS is a parameter-free iterative clustering algorithm. Our algorithm first scans the dataset in a sequential manner such that the destination of the next transaction is guided by a novel criterion function. Once the first scan of the dataset is completed, TRANCLUS performs a few other passes over the dataset in order to refine the clustering. In the following, first we provide a formal description of the clustering problem. Second, we develop new criterion functions that guide the clustering process. Finally, we present the general scheme of TRANCLUS.

4.1 Problem Statement

To describe our algorithm, we will introduce some notations and definitions. Let $A = \{a_1, a_2, \ldots, a_n\}$ denote the set of n askers, such that each asker a_i of A has received at least one answer from an authoritative user e_j $(j = 1, \ldots, d)$ of E. Recall that $E = \{e_1, e_2, \ldots, e_d\}$ is the set of d authoritative users identified in the previous section. In this paper, we propose to represent our environment as a type of transaction.

Definition 1. A transaction T_i $(i = 1, \ldots, n)$ where $T_i \subseteq E$ and $\mid T_i \mid \geq 1$ consists of all the authoritative users who answered the questions of the asker a_i.

The definition above allows us to associate, with each asker a_i of A, a transaction T_i that summarizes his/her interactions with authoritative users. Accordingly, we get a set $TD = \{T_1, T_2, \ldots, T_n\}$ which is a collection of transactions that summarizes the interactions of all the askers a_i with the identified authoritative users. In the current literature, TD is called transactional data and the elements of each transaction T_i are called items. In the context of our study, an item corresponds to an authoritative user $e \in E$. Note that the transactions contain varying numbers of items.

Based on the data model described above, we expect the problem of discovering knowledge-sharing communities in question-answering forums to be closely related

to the problem of discovering clusters in the set TD. The clustering problem consists of partitioning the original collection of transactions in TD into a set $C = \{C_1, C_2, \ldots, C_{nc}\}$ where $C_1 \cup C_2 \cup \cdots \cup C_{nc} = TD$, and $\forall(C_r, C_t) \in C$: $C_r \cap C_t = \emptyset$ where $r \neq t$. Let n_s denote the size of C_s and $E = \{e | \ e \in T, T \in C_s\}$ where $s = 1, \ldots, nc$ denotes the set of items in C_s. Summarizing the proposals of some previous studies [Cesario et al. 2007; Yan et al. 2006; Yang et al. 2002; Wang et al. 1999], in the context of transactional data clustering, a cluster should satisfy the following properties:

(1) A cluster C_s $(s = 1, \ldots, nc)$ is any non-empty subset of transactions of TD.
(2) Transactions in C_s exhibit a high degree of overlap in comparison to any transactions in $(TD - C_s)$.
(3) The subset of items $\{E_s\}_{s=1,\ldots,nc}$ may not be disjoint.

The reason for the first property is trivial. The second property is based on the assumption that a cluster should contain homogeneous transactions. By homogeneous transactions we mean that all the transactions grouped in the same cluster share at least a lot of items in common. This means that a significant proportion of the items in a transaction are also present in another transaction of the same cluster. This second property of a cluster satisfies the second property of a knowledge-sharing community as described in the Introduction, in the sense that askers from the same community exhibit homogeneous behavior in terms of their interactions with authoritative users. Finally, the third property of a cluster given above is based on the fact that transactions that belong to different clusters may share some (or, in some cases, no) items. This means that clusters may have overlapping items. This indicates that authoritative users may belong to more than one community, which in turn satisfies the third property of a knowledge-sharing community.

To summarize, the description of a cluster given above satisfies the properties of a knowledge-sharing community highlighted in Section 1. Accordingly, the problem of discovering such communities is defined as follows: *Given the set A of askers and the set E of authoritative users, construct TD based on Definition 1 and then partition it into a set of clusters* $C = \{C_1, C_2, \ldots, C_{nc}\}$. The identified clusters represent the communities we want to discover.

4.2 Developing a Criterion Function

The search for a clustering $C = \{C_1, C_2, \ldots, C_{nc}\}$ is guided by the criterion function $CF(C)$ which is defined as

$$CF(C) = \sum_{s=1}^{nc} \left[\frac{n_s}{n} CF(C_s) \right] \tag{10}$$

$$CF(C_s) = \frac{1}{n_s} \sum_{e \in C_s} F(e, C_s) \tag{11}$$

$$F(e, C_s) = occ(e, C_s) \cdot W(e, C_s) \tag{12}$$

$$W(e, C_s) = WLF(e, C_s) \cdot WGF(e, TD) \tag{13}$$

$T_1 = \{e_1, \underline{e_2}, \underline{e_3}, e_6, e_7, e_8\}$ $T_4 = \{e_1, \underline{e_4}, \underline{e_5}, e_3, e_{13}, e_{14}, e_{15}\}$
$T_2 = \{e_1, \underline{e_2}, \underline{e_3}, e_9, e_{10}\}$ $T_5 = \{e_1, \underline{e_4}, \underline{e_5}, e_2, e_{16}, e_{17}\}$
$T_3 = \{e_1, \underline{e_2}, \underline{e_3}, e_{11}, e_{12}\}$ $T_6 = \{e_1, \underline{e_4}, \underline{e_5}, e_6, e_{18}, e_{19}, e_{20}\}$

Fig. 5. A transactional dataset that contains two clusters.

where

—$occ(e, C_s)$ denotes the number of transactions in C_s that contains the item e;
—$WLF(e, C_s)$ denotes the Weighted Local Frequency of item e in C_s;
—$WGF(e, TD)$ denotes the Weighted Global Frequency of the item e in TD.

The criterion function $CF(C)$ is composed of the score component $CF(C_s)$ of each cluster, which in turn is the sum of the score components $F(e, C_s)$ of all items e in C_s. $F(e, C_s)$ is the occurrence of item e in cluster C_s weighted by $W(e, C_s)$ which is defined by two weights: $WLF(e, C_s)$ and $WGF(e, TD)$. The weight $WLF(e, Cs)$ measures the degree of participation of e in C_s, while the weight $WGF(e, TD)$ measures the importance of e in TD. Our goal is to find a clustering C that maximizes $CF(C)$. Note that the term "local" used in the description of the weight $WLF(e, C_s)$ is principally to denote the fact that this weight investigates item e based on its local occurrence frequency in cluster C_s. On the other hand, the term "global" used in the description of $WGF(e, TD)$ is simply to denote the fact that $WGF(e, TD)$ investigates item e based on its occurrence frequency in the whole dataset TD. The rational behind $CF(C)$ and the formulas of $WLF(e, C_s)$ and $WGF(e, TD)$ are given below.

In Equation (10), the term $\frac{n_s}{n}$ represents the relative contribution of cluster C_s to the partition C, and $CF(C_s)$ measures the quality of cluster C_s. In principle, a cluster is likely to be of good quality if it contains a sufficient number of transactions where certain items occur with higher frequency than elsewhere [Cesario et al. 2007]. Starting from this assumption, $CF(C_s)$ evaluates the quality of C_s by investigating the commonality of items within the transactions that it contains. Specifically, as can be seen from Equation (11), $CF(C_s)$ is defined as the sum of the score components $F(e, C_s)$ normalized by the size of C_s. In $F(e, C_s)$, Equation (12), we associate with each item e a weight $W(e, C_s)$ that measures the degree of participation of e in C_s. The reason for using $W(e, C_s)$ is principally to avoid equal treatment of the items in C_s. This is because giving all the items equal importance during the clustering process is not appropriate, since there are items which potentially contribute to the formation of clusters while others do not. $W(e, C_s)$ will give more discriminative power to frequent items that allow to us discriminate one cluster from another.

For purposes of clarification, consider the transactional dataset depicted in Figure 5. In the context of transactional clustering, one would expect two clusters to result from this dataset. They would look as follows: $C_1 = \{T_1, T_2, T_3\}$ and $C_2 = \{T_4, T_5, T_6\}$. Transactions in each cluster exhibit a significant overlap since they share a number of common items. For instance, items e_1, e_2 and e_3 are common to all the transactions in C_1, while items e_1, e_4 and e_5 are common to all the transactions grouped in C_2. From Figure 5, we observe that there are items with low occurrence frequency in each cluster (e.g., $e_7, e_8, e_9, e_{10}, e_{11}$ and e_{12} in C_1).

Such rare items are considered as less important in the clustering process since they do not help to distinguish between C_1 and C_2. In addition, item e_1 is common to all the transactions in the dataset. Although e_1 is of high occurrence frequency in each cluster, such an item is also considered as less important in the clustering process since e_1 cannot allow us to discriminate between C_1 and C_2. Among all the items, only e_2, e_3, e_4 and e_5 allow us to discriminate between C_1 and C_2. Such items should receive a high discriminating power in comparison to other items. This is accomplished by $W(e, C_s)$.

Equation (13) describes $W(e, C_s)$. As can be seen from this equation, $W(e, C_s)$ is defined by two weights: $WLF(e, C_s)$ and $WGF(e, TD)$. The reason for using $WLF(e, C_s)$ is to measure the degree of participation of item e in cluster C_s.

Definition 2. The Weighted Local Frequency of item e in C_s is defined as

$$WLF(e, C_s) = \frac{occ(e, C_s)}{n_s} \cdot \frac{occ(e, C_s)}{occ(e, TD)}$$

In the above definition $occ(e, TD)$ corresponds to the number of all transactions in TD that contain item e. As we can see from Definition 2, the weighted local frequency of an item is defined by two terms. The first term, $\frac{occ(e,C_s)}{n_s}$, represents the percentage of transactions in C_s that contain e. The larger the value of $\frac{occ(e,C_s)}{n_s}$, the greater the proportion of transactions in C_s sharing e. This means that the first term of $WLF(e, C_s)$ reflects the compactness of cluster C_s. The second term, $\frac{occ(e,C_s)}{occ(e,TD)}$, measures the proportion of transactions that are not in C_s but contain e. The range value of this term is always]0, 1]. The larger the value of $\frac{occ(e,C_s)}{occ(e,TD)}$, the smaller the proportion of transactions outside C_s containing e. This means that the second term of $WLF(e, C_s)$ reflects the degree of separation of C_s from the other clusters in C. A combination of these two terms, as described in Definition 2, represents a tradeoff between compactness and separation. In this way, we preserve as many frequent items as possible in a cluster and control the overlapping of items between clusters. A large value of $WLF(e, C_s)$ means that e is of higher occurrence frequency in C_s than elsewhere, which provides a meaningful measure of the degree of participation of e in C_s.

However, $WLF(e, C_s)$ is not sufficient for the purpose of clustering if it is used alone in the definition of $W(e, C_s)$. As we will show in the next subsection, this is mainly due to the iterative clustering strategy that we adopt to group transactions, in which the values of $occ(e, C_s)$ and n_s are updated during the clustering process. This implies that the weight $WLF(e, C_s)$ is not fixed during the clustering procedure: its value is determined by the current item distribution of clustering C. In this context, if $WLF(e, C_s)$ is used as the only metric in the estimation of $W(e, C_s)$ rare items will receive a high value of $WLF(e, C_s)$ if the transactions that contain such items are placed in singleton clusters. This may affect the accuracy of the clustering and favorite singleton clusters. To address this problem, we introduce a second weight, $WGF(e, TD)$, in the estimation of $W(e, C_s)$.

Definition 3. The Weighted Global Frequency of an item e in TD is defined as

$$WGF(e, TD) = \frac{occ(e, TD)}{n} \cdot (n - occ(e, TD) + 1)$$

In contrast to $WLF(e, C_s)$, the value of $WGF(e, TD)$ is fixed since it investigates item e in the unclustered data (i.e., the original set TD). $WGF(e, TD)$ is principally designed to evaluate how important an item e is in the whole dataset TD to the clustering problem. The intuition behind the definition of $WGF(e, TD)$ is that, while it seems that more items are relevant for the purpose of clustering, a lot isn't proportionally relevant than a few. Specifically, items that are too common across different transactions or rare items across the whole dataset will receive a low $WGF(e, TD)$ score, rendering them essentially less important during the clustering process. On the other hand, items between rare items and extremely frequent items in TD, will receive a relatively large $WGF(e, TD)$ score.

It is worth pointing out that the principle of WGF, as described above, draws its inspiration from the TF-IDF (Term Frequency - Inverse Term Frequency) principle which is often used in information retrieval and text mining. As the term implies, TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words that are extremely frequent in a corpus and rare words will receive a small TF-IDF score, while words that occur many times within a small number of documents will receive a large TF-IDF score. There are many different formulas to calculate TF-IDF. The formula for WGF is simpler and quite different from those for TF-IDF, since the data model considered in this paper (transactional data) is different from document data. More details about TF-IDF and its application can be obtained from any good book on text mining and information retrieval [Manning et al. 2008].

For purposes of clarification, recall the example in Figure 5. By Definition 2, the WGF of items $e_1, e_7, e_8, \ldots, e_{20}$ is 1 while the WGF of e_2, e_3, e_4 and e_5 is 2 and $WGF(e_6, TD) = 1.66$. Here, it is clear that we favor items e_2, e_3, e_4 and e_5 over the others. On the other hand, from Figure 5 we observe that item e_6 does not allow us to distinguish C_1 from C_2 and at the same time e_6 has a relatively large WGF score in comparison to the WGF score of rare items. This does not necessarily mean that item e_6 will be favored during the clustering process, because the relative strength of e_6 in C_1 and C_2, which is measured by WLF, is small. So, by multiplying WLF with WGF, as defined in Equation (13), the global weight of e_6 in both clusters, i.e., $W(e_6, C_1)$ and $W(e_6, C_2)$, will also be small, leaving item e_6 with no major impact during the clustering. This means that multiplying $WGF(e, TD)$ by $WLF(e, C_s)$ tends to regulate the weight of items during the clustering process such that high values of $WGF(e, TD)$ from low-frequency items are less relevant than those from high-frequency items.

In other words, by multiplying $WLF(e, C_s)$ by $WGF(e, TD)$, as in Equation (13), we boost the weight of items that contribute to the formation of clusters, lending them a high discriminating power. On the other hand, we pull down the weight of items that do not help to discriminate between clusters, rendering them negligible during the clustering process. So, the higher the value of $W(e, C_s)$, the more important the item e to the formation of the clusters C_s. This means that the maximum of $F(e, C_s)$ indicates that transactions in C_s exhibit a higher homogeneity than elsewhere which, in turn, means that the maximum of $CF(C)$ indicates that the clustering C is of good quality.

Algorithm 3: The TRANCLUS scheme.

```
   Input  : A set TD = {T_1, T_2, ..., T_n} of n transactions
   Output: A partition C = {C_1, C_2, ..., C_nc} of nc clusters
 1 begin
 2   for each item e in TD compute the component Z(e) = (n - occ(e, TD) + 1) ;
     // Initialization phase
 3   while not end of the dataset file TD do
 4     Read the next transaction < T_i, unknown >;
 5     Assign T_i to an existing or new cluster C_l to maximize CF(C);
 6     Write < T_i, C_l > back to TD;
     // Refinement phase
 7   while move == true do
 8     move = false ;
 9     while not end of the dataset file TD do
10       Read the next transaction < T_i, C_l >;
11       move T_i to an existing or new cluster C_t to maximize CF(C);
12       if C_l ≠ C_t then
13         Write < T_i, C_t > back to TD;
14         move = true;
15 end
```

Based on definitions 2 and 3, the criterion function $CF(C)$ described in Equation (10) can be defined as follows:

$$CF(C) = \sum_{s=1}^{nc} \left[\frac{n_s}{n} \cdot \frac{1}{n_s} \sum_{e \in C_s} \left(occ(e, C_s) \cdot \frac{occ(e, C_s)}{n_s} \cdot \frac{occ(e, C_s)}{occ(e, TD)} \cdot \frac{occ(e, TD)}{n} \cdot \left(n - occ(e, TD) + 1\right) \right) \right]$$

$$CF(C) = \frac{1}{n^2} \sum_{s=1}^{nc} \left[\frac{1}{n_s} \sum_{e \in C_s} \left(Z(e) \cdot \left(occ(e, C_s)\right)^3 \right) \right] \quad (14)$$

where

$$Z(e) = \left(n - occ(e, TD) + 1\right) \quad (15)$$

The transactional clustering problem is formally defined as follows: *Given a collection of transactions TD, find the optimal clustering C that maximizes $CF(C)$ described in Equation (14).* To find such an optimal clustering, in the following section we present TRANCLUS a $CF(C)$-based iterative clustering algorithm.

4.3 The TRANCLUS Scheme

The general scheme of TRANCLUS is specified in Algorithm 3. As can be seen, TRANCLUS performs a first scan of the dataset (line 2 of Algorithm 3) in order to estimate the component $Z(e)$ described in Equation (15). Next, the algorithm implements a partition-based clustering strategy which consists of two phases: 1) Initialization and 2) Refinement.

Initialization phase: The goal of this phase is to build an initial clustering C based on the criterion function $CF(C)$ described in Equation (14). Specifically, the algorithm reads each transaction T_i sequentially and either assigns T_i to an existing cluster (initially none) or creates T_i as a new cluster that maximizes $CF(C)$.

Refinement phase: As the name implies, the goal of this phase is to improve the result of the first phase in order to find a clustering C that optimizes $CF(C)$. To this end, the cluster assignment is refined in an iterative manner until no more improvement can be made with respect to $CF(C)$ in the clustering result. Specifically, the algorithm moves T_i to an existing or new cluster (or it may be left where it is) to maximize $CF(C)$. Note that any empty cluster generated is eliminated after a move. The iterative process is stopped if no transaction is moved from one cluster to another in a pass for all transactions in the clustering result. Otherwise, a new pass begins.

We would point out that the iterative process described above is not new; it is widely adopted by iterative-based clustering algorithms such as LargeItem [Wang et al. 1999] and CLOPE [Yang et al. 2002]. Such an iterative process is akin to K-means clustering in the sense that, as mentioned in [Wang et al. 1999], it scans the dataset iteratively and assigns the next transaction to the cluster that optimizes a criterion function. In this context, the key step for all of these algorithms is to find the destination cluster for each transaction, based on a criterion function. This feature is what distinguishes our approach from existing ones, because we have developed a new criterion function. Our criterion function, which is actually the cornerstone of TRANCLUS, allows us to perform transactional data clustering in a systematic way. In contrast to the vast majority of existing transactional data clustering algorithms (except AC-TD [Cesario et al. 2007]), our approach is free of any tunable parameter, which is a concrete advantage. Further, TRANCLUS is able to identify the optimal number of clusters without the extra computational cost required by repeated estimation and evaluation of a predefined number of clusters.

The most frequent operation invoked by TRANCLUS is the computation of $CF(C)$ in order to determine the destination of the next transaction (lines 5 and 11 of Algorithm 3). Specifically, we need to update the values of $\frac{1}{n_s}$ and $occ(e, C_s)$ after adding/removing T_i to/from C_s and then summing up the values from all the clusters. To avoid scanning all the transactions, which entail a high computational cost, the values of $\frac{1}{n_s}$ and $occ(e, C_s)$ are incrementally maintained in our implementation by using hash tables. This makes the computation of $CF(C)$ quite efficient, since in adding or removing a transaction to/from a cluster we consider only the value change of the current cluster being tested.

Finally, note that TRANCLUS converges to a local maximum of $CF(C)$ in a finite number of iterations. Our claim is based on the following two properties.

(1) To divide the dataset into nc clusters, it is clear that there are only a finite number of possible partitions C.

(2) From iteration I to iteration $I+1$, a change in the partitions yields an increase in $CF(C)$.

The first property is trivial. To show the second property, we recall that at each pass over the dataset the algorithm performs several tests by repeatedly adding/removing a particular transaction T_i to/from C_s in order to decide in which

cluster C_s the transaction T_i should be placed. The new partition generated by the algorithm at the end of the successive iteration $I + 1$ represents the best possible reassignment of the transaction T_i to a cluster C_s with respect to $CF(.)$. In addition, the new partition will not be retained if it does not result in an increase in $CF(.)$. This means that the sequence $CF(.)$ generated by the algorithm is strictly increasing. Therefore, TRANCLUS converges in a finite number of iterations.

5. EMPIRICAL EVALUATION OF TRANCLUS

Before using TRANCLUS to discover knowledge-sharing communities, we should first demonstrate its suitability. For this purpose, we devise in this section a series of experiments designed to evaluate the effectiveness of our algorithm. The evaluation is performed on a number of generated datasets with different characteristics. Experiments on benchmark real datasets that are widely used to evaluate transactional clustering algorithms are also presented.

5.1 Experiments on Synthetic Data

To better understand the properties of TRANCLUS and get an objective idea of its practical performance and applicability, synthetic datasets with controlled cluster structure were first used. As an important advantage of synthetic data, let us note that it allows experiments to be conducted in a controlled way, making it possible to answer specific questions concerning the performance of an algorithm and its behavior under particular conditions. In this setting, we have investigated the following two major aspects:

(1) Quality: the aim is to evaluate the performance of TRANCLUS in terms of clustering accuracy.

(2) Efficiency: the aim is to analyze the scalability of our algorithm.

5.1.1 *Evaluation Criteria.* Clustering evaluation criteria can be based on internal or external measures [Cesario et al. 2007]. An internal measure is often the same as the objective function that a clustering algorithm explicitly optimizes - in our case, the criterion function $CF(C)$. However, the goodness of each cluster should be judged not only by the clustering algorithm that generated it, but also by external assessment criteria, especially when objects have already been categorized by an external source; i.e., when class labels are available. Since in this subsection we investigate the behavior of TRANCLUS on a number of synthetic datasets where the labels of the input/original clusters are known (but of course not used in the clustering process), we used external criteria to evaluate the clustering results by calculating the correspondence between the clusters generated by a clustering algorithm and the original partitioning. In this setting, it is clear that the use of such criteria is appropriate since they help in understanding clustering results and hence in evaluating the adequacy of a clustering algorithm. Among the existing external measures available in the literature, we have chosen to use the *F-measure* and the *error rate of clustering.* Our choice is based on the fact that these two indices measure the quality of the results of a clustering algorithm in different ways. Hence, a comparison of the values of these indices gives us a clear idea about the qualitative behavior of TRANCLUS. In the following sections, we refer to clusters

in the original partition (i.e., the partition generated by the data generator) as input clusters and the clusters identified by the algorithm as output clusters.

F-measure: This measure combines the notions of Precision and Recall from the information retrieval literature [Manning et al. 2008]; it is defined as

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{16}$$

The Precision and Recall are defined as

$$Precision = \frac{a}{a+c}, \quad Recall = \frac{a}{a+b},$$

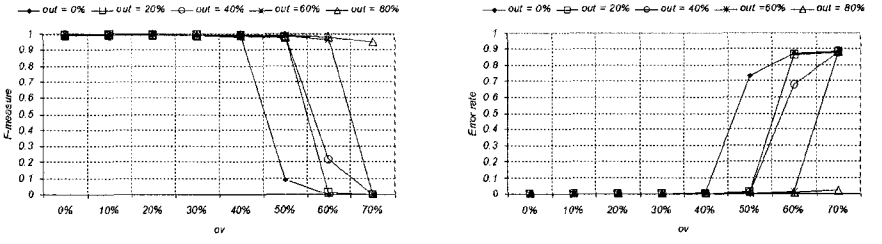
In the above equations, a correspond to the number of transaction pairs that are in the same cluster in both O and G. O here denotes the original partition and G denotes the partition generated by a clustering algorithm. b is the number of pairs in the same cluster in O but not in G, c is the number of pairs in the same cluster in G but not in O, and finally d is the number of pairs that are in different clusters in both O and G. The F-measure achieves its maximum, which is 1, when the clustering results perfectly match the external class labels.

Error Rate: The error rate (ER) of the clustering scheme is the proportion of transactions that are misclassified in the *confusion matrix*. The entry n_{ij} of the confusion matrix indicates the number of transactions belonging to the resulting cluster C_i, which were generated as a part of the input cluster c_j. The error rate is defined as

$$ER = \frac{\sum_i \sum_{j \neq h} n_{ij}}{n} \tag{17}$$

where h is the index of the input cluster c_j with maximal n_{ij}. The values of ER are always between 0 and 1 such that smaller ones (close to 0) indicate good clustering.

5.1.2 *Synthetic Data Generation.* We used the data generator model described in the AT-DC paper [Cesario et al. 2007], which was kindly provided by its authors. The parameters used in synthetic data generation are the size of the dataset n; the number of items d; the average number of item in each transaction t; the number of input clusters c; the percentage *out* of outlier items or the proportion of items in E that do not contribute to the formation of any clusters, where E corresponds to the set of all items in the dataset; and, finally, the percentage *ov* of overlapping among transactions of different clusters. As described in [Cesario et al. 2007], the synthetic data generation process works as follows: Initially, E is populated with d items. Then, a subset of items, with size proportional to *out*, is extracted from E, and c random subsets are generated from the remaining elements in E. Each subset S_i defines a cluster, and transactions for each cluster are generated starting from such a subset. In particular, a transaction in cluster c_i is generated by picking its size l from a normal distribution with mean t and fixed variance. Then, the transaction is populated with l items, *ov* percent of which are picked from S_i, and the remainder from the whole of E. In principle, the parameter *ov* has an influence on the separability of clusters. The larger the value of *ov*, the more pronounced the overlap among clusters. Such a data model allows us to simulate various situations, which in turn makes it possible to perform an objective experimental validation of a transactional clustering algorithm.



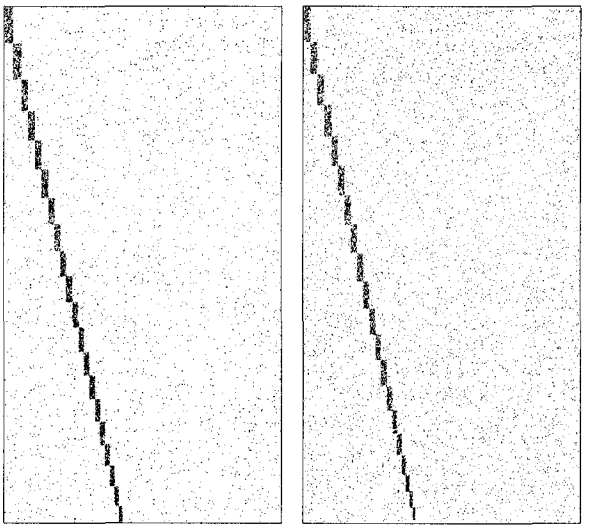
(a) (b)

Fig. 6. Quality of clustering on synthetic datasets. (a) F-measure. (b) Error rate.

5.1.3 *Quality of Clustering.* The main goal of the experiments presented in this subsection was to evaluate the capability of our algorithm to correctly identify clusters in various situations. Specifically, since the parameters *ov* and *out* have a direct influence on the quality of the results of a transactional clustering algorithm [Cesario et al. 2007], our aim is to analyze the impact of these two parameters on TRANCLUS. For this purpose, we generated a number of different datasets with $n = 10000$ transactions, number of item $d = 1000$, average transaction length $t = 20$ and number of clusters $c = 20$. The degree *ov* of overlapping among transactions of different clusters varies from 0% to 70%, while the percentage of outlier items varies from 0% to 80%. Figure 6 illustrates the clustering results of TRANCLUS on these datasets, evaluated with the F-measure index and the clustering error rate.

As we can see from Figure 6, when $ov \leq 40\%$, TRANCLUS achieves highly accurate results for different values of *out*. When the overlap between clusters is more pronounced, i.e., $50\% \leq ov \leq 70\%$, the performance of TRANCLUS is generally consistent as the value of *out* increases. Specifically, when $out > 60\%$ the algorithm performs well for different values of $ov < 80\%$. For instance, as can be seen from Figure 6, when $out = 80\%$ the algorithm withstands the increasing values of *ov* and maintains high clustering quality. On the other hand, when $ov \geq 80\%$, the values of the F-measure and the error rate are close to 0 and 1, respectively, for different values of *out*. This is due to the high overlap between clusters. In such "extreme" situations (i.e., $ov \geq 80\%$), clusters may have a large number of overlapping items, which makes it difficult to discriminate between them. To summarize, in general, TRANCLUS stands up well to increasing values of *ov* and *out*. Specifically, as *out* increases, the algorithm achieves high quality results for $ov < 80\%$. This behavior can be explained by the fact that, as discussed in Section 4, the criterion function $CF(C)$ that guides the clustering procedure preserves as many frequent items as possible in a cluster and controls overlapping of items between clusters. Furthermore, $CF(C)$ reduces the effect of outlier items by attributing them a low weight, thereby reducing their impact during the clustering process.

To provide a visual illustration of the qualitative behavior of TRANCLUS, Figure 7 shows the clustering results of our algorithm on some selected synthetic datasets. To this end, we estimate the transaction/item Boolean incidence matrix of the partition discovered by the algorithm, such that the rows of each matrix correspond to



(a) $ov = 20\%$. (b) $ov = 40\%$.

(c) $ov = 60\%$. (d) $ov = 70\%$.

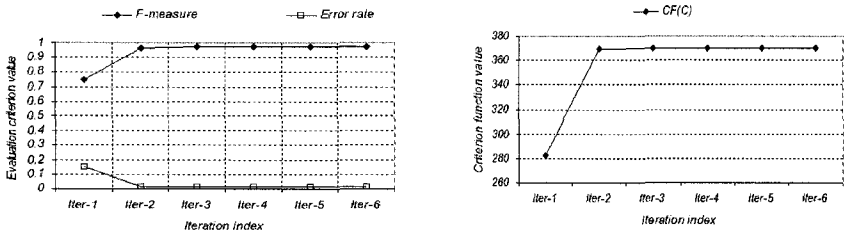
Fig. 7. Clustering results on four synthetic datasets with out = 60% and different values of ov.

transactions, while columns represent items. In each row, 1 indicates the presence of the corresponding item in a transaction, while 0 indicates its absence. Since a cluster may contain transactions that share a lot of common items, such a representation provides a visualization of the cluster structure in the datasets. Specifically,

we arrange the rows and columns of the transaction/item incidence matrix based on the cluster assignments: the transactions and items in the first cluster appear first, the transactions and items in the second cluster appear next, ..., and the transaction and items in the last cluster appear at the end. Note that, in our representation, clusters are sorted by their size. Such a representation was used in previous studies [Cesario et al. 2007; Li 2005] in order to assess the quality of results of a transactional/binary data clustering algorithm. Ideally, a good clustering would produce a block diagonal structure [Cesario et al. 2007; Li 2005]. For the sake of discussion and to avoid encumbering the paper, we will show the transaction/item Boolean incidence matrix that illustrates the results of the algorithm only for four selected synthetic datasets in which $out = 60\%$ and ov varies from 20% to 70%. Note that the analysis provided below is typical of the qualitative behavior of TRANCLUS.

In all of the matrices depicted in Figure 7, the shaded region represents non-zero entries, while in Figures 7(a), 7(b) and 7(c) the front of the right-hand side of each matrix, in which no block exists, corresponds to the set of items that do not participate in any clusters (outlier items), while the twenty block diagonals correspond to the twenty clusters discovered by the algorithm. It is stated in [Cesario et al. 2007] that when the degree of overlap is relatively low, the transaction/item incidence matrix of a good clustering should contain blocks that exhibit an internal density which is higher than that of surrounding regions. This is clearly illustrated in Figures 7(a), 7(b), and 7(c) which testify to the good quality of clustering. As can be seen from these figures, each discovered block is characterized by different subsets of items. This indicates that the clusters identified by our algorithm contain a set of items of high occurrence frequency that allow us to discriminate between clusters. On the other hand, as depicted in Figure 7(d), we observe that when there is a high degree of transaction overlap, TRANCLUS still discovers some meaningful structure but not the input clusters. Specifically, due to the presence of a large number of overlapping items within the originally generated clusters, the algorithm tends to discovers a large number of small clusters characterized by different subsets of items. This is apparent in Figure 7(d) as a large number of very small block diagonals. This illustrates the capacity of TRANCLUS to uncover meaningful relationships among transactions in some "difficult" situations.

To provide more insight into the qualitative behavior of TRANCLUS, Figure 8(a) illustrates the quality of the partitions generated by the algorithm during the clustering process, while Figure 8(b) depicts a typical convergence curve for our algorithm. In both figures, each point on the curves represents a partition generated by one iteration of the TRANCLUS clustering process and the results refer to one selected synthetic dataset in which $out = ov = 60\%$. Note that the first iteration, i.e., $Iter - 1$, corresponds to the initialization phase of TRANCLUS, while the remaining iterations correspond to passes made by the algorithm over the whole dataset during the refinement phase. As we can see from Figure 8, the algorithm converges to its local optimum after six iterations, and it achieves highly accurate results after only two iterations. We note that, as depicted in Figure 8(b), the partitions generated by the algorithm after the second iteration represent a mild improvement to the criterion function.



(a) (b)

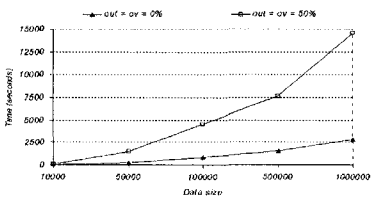
Fig. 8. (a) Quality of partitions. (b) Convergence curve.

5.1.4 *Scalability.* In this subsection, we evaluate the scalability of TRANCLUS with increasing data set size and number of items. We also study the impact of varying values of *out* and *ov* on the performance of the algorithms. For this purpose we generated a number of different synthetic datasets. In all of the following experiments, the quality of the results returned by TRANCLUS was similar to that presented in the previous subsection.

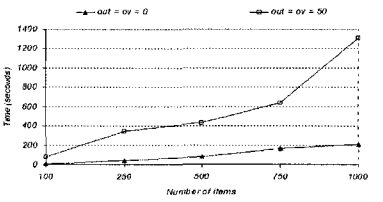
Scalability with respect to dataset size: Figure 9(a) shows the results for scalability with the size of the dataset. In this experiment we generated two groups of datasets. Each group contains five datasets such that $ov = out = 0\%$ in all of the data used in the first group, while the data in the second group was generated with $ov = out = 50\%$. In all the datasets (in both groups), the number of transactions n is varied from 10000 to 1000000, the number of items $d = 200$, average transaction length $t = 20$ and number of clusters $c = 10$. As we can see from Figure 9(a), TRANCLUS scales linearly with the increase in dataset size in both cases ($ov = out = 0\%$ and $ov = out = 50\%$). However, we observe from this figure that the performance of the algorithm is affected, since its running time increases as the value of *ov* increases.

Scalability with respect to the number of items: Figure 9(b) reports the results for scalability with the number of items in the dataset. The results in this figure refer to two groups of datasets such that data in the first group was generated with $out = ov = 0\%$ while data in second group was generated with $out = ov = 50\%$. In all of the datasets used in this set of experiments, the number of items d is varied from 100 to 1000, the number of transactions $n = 10000$, average transaction length $t = 20$ and number of clusters $c = 10$. As can be seen from Figure 9(b), TRANCLUS exhibits a linear behavior w.r.t. the number of items. On the other hand, as in the experiments on scalability w.r.t. dataset size, the running time of the algorithm increases as the value of *ov* increases. In this case (i.e., $out = ov = 50\%$), the running time curve of the algorithm depicted in Figure 9(b) does not exhibit "strict" linear behavior.

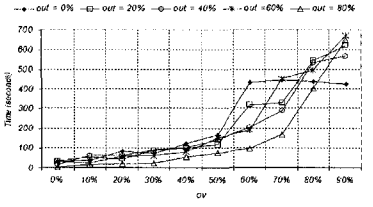
Scalability with respect to *out* and *ov*: To study the impact of varying values of *out* and *ov* on the performance of the algorithms, we generated a number of datasets in which the degree *ov* of overlap among transactions in different clusters varies from 0% to 90%, while the percentage of outlier items *out* varies from 0% to



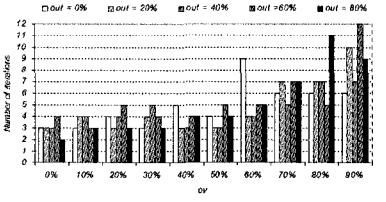
(a) Scalability w.r.t. dataset size.



(b) Scalability w.r.t. number of items.



(c) Runing time w.r.t. *out* and *ov*.



(d) Number of iterations w.r.t. *out* and *ov*.

Fig. 9. Scalability experiments.

80%. In all the datasets used in this set of experiments the number of transactions $n = 10000$, number of item $d = 200$, average transaction length $t = 20$ and number of clusters $c = 10$. As can be seen from Figure 9(c), the running time of the algorithm is mainly affected by increasing values of ov. This behavior of TRANCLUS is also apparent in Figure 9(d), where we observe that for larger values of ov, the algorithm performs more iterations to reach the optimal maximum of $CF(C)$. To summarize, the rate converge of the algorithm is affected by high values of ov.

5.2 Experiments on Real-Life Data

Our main goal in this section is to demonstrate the suitability of TRANCLUS on real-life datasets. For this purpose, we compare the performance of our algorithm to that of AT-DC, another transactional clustering algorithm recently proposed. It was shown in [Cesario et al. 2007] that AT-DC outperforms other existing algorithms. Furthermore, we have considered only AT-DC in the comparison primarily because it performs clustering in a fully systematic way, like TRANCLUS, which makes the comparison more objective. We do not consider existing parameter-laden transactional clustering algorithms in the comparison because it was shown in [Keogh et al. 2004] that parameter-laden algorithms are burdensome to use, and make it difficult to compare results across different methods. Some approaches require careful parameter tuning, since the quality of their results depends heavily on the initial parameter values set by the user. This means that an exhaustive search for the best parameter values is required. In other words, we would have to perform many experiments and present only the best results. This biases the comparison and presents the potential risk of fine-tuning the parameter values based on the observed performance while doing the experiments, which is impossible in a real situation. A parameter-free algorithm prevents us from imposing our prejudices

and presumptions on the problem at hand, and lets the data itself speak to us [Keogh et al. 2004].

We conducted experiments on real-life categorical datasets taken from the UCI Machine Learning Repository [4]. Specifically, we selected four categorical datasets: 1) Congressional Votes, 2) Mushrooms, 3) Zoo and 4) Internet Advertisements. The first three datasets are considered benchmark data and are widely used to evaluate categorical and transactional clustering algorithms [Cesario et al. 2007; Yang et al. 2002; Wang et al. 1999]. In all of these datasets, tuples have class labels defined based on some domain knowledge. As in our experiments on synthetic data, we ignore class labels during clustering but use them as ground truth in order to measure the accuracy of clustering. Below, we provide a description of each dataset used in this set of experiments. An evaluation of the performance of TRANCLUS on these datasets is reported together with a comparative analysis of the quality of its results relative to that of AT-DC. Note that all the results of AT-DC for the four categorical datasets considered in this set of experiments are reproduced from the original paper.

Congressional Votes. This data is a collection of 435 US Congressional Votes Records. Every record contains 16 attributes corresponding to one congressman's votes ("Yes" or "No" vote) on 16 key issues. The dataset contains 168 records labeled as "Republican" while the remaining 267 records are labeled as "Democrat". There are 288 missing values which are ignored during the clustering procedure. The 249^{th} record is deleted before clustering since all its values are missing. Each record within such data can be straightforwardly converted to a transaction by considering two scenarios. In the first scenario we perform the transformation ignoring the "No" value in the attributes. In this case each record is represented as a transaction in which items correspond to the index of the attribute with "Yes" value. In the second scenario we take into account both "Yes" and "No" votes. Here, each item is represented as a term "attribute-name = attribute-value".

In order to test the accuracy of the results, we determined the confusion matrix which indicated how well the output clusters matched with the input classes. It is clear that if the clustering algorithm performs well, each row and column is likely to have one entry which is significantly larger than the others. On the other hand, in the case where the clustering technique is so bad as to be completely random, the transactions are likely to be evenly distributed among different clusters [Aggarwal and Yu 2002]. As depicted in Figure 10(a) and Figure 11(a), TRANCLUS performs well on the vote data in both scenarios and the results are quite similar. One of the entries in each column of the confusion matrix illustrated in these two figures is indeed clearly larger than the rest of the entries. This indicates that each input class gets directed into one output cluster with the exception of some transactions which get distributed to other clusters. It is worth noting that the two singleton clusters, C_3 and C_4, identified by TRANCLUS corresponds to the 108^{th} and 184^{th} records. These two voting records are very different from the others, since they contain only one "Yes" vote over all the 16 issues, while the majority of the voters vote "Yes" for 6 to 10 issues. The two distinguished votes are picked up by TRANCLUS and are assigned to two separate singleton clusters. Here we believe that the behavior

[4]http://archive.ics.uci.edu/ml/

ClusterId.	*Democrat*	*Republican*
C_1	219	13
C_2	47	153
C_3	1	0
C_4	0	1

(a) Confusion matrix of TRANCLUS

ClusterId.	*Democrat*	*Republican*
C_1	202	25
C_2	46	0
C_3	7	64
C_4	12	78

(b) Confusion matrix of AT-DC

Fig. 10. Congressional votes, with ignoring the "No" value in the attributes.

ClusterId.	*Democrat*	*Republican*
C_1	224	9
C_2	42	157
C_3	1	0
C_4	0	1

(a) Confusion matrix of TRANCLUS

ClusterId.	*Democrat*	*Republican*
C_1	81	166
C_2	186	1

(b) Confusion matrix of AT-DC

Fig. 11. Congressional votes, with considering the "No" value in the attributes.

ClusterId.	*Edible*	*Poisonous*
C_1	17	3086
C_2	4191	830

(a) Confusion matrix of TRANCLUS

ClusterId.	*Edible*	*Poisonous*
C_1	3536	360
C_2	288	0
C_3	0	1734
C_4	192	454
C_5	0	1296
C_6	0	64
C_7	192	0
C_8	0	6

(b) Confusion matrix of AT-DC

Fig. 12. Clustering results for Mushroom.

of TRANCLUS is reasonable, since these two votes are not very representative of either of the two expected classes. On the other hand, in contrast to TRANCLUS, AT-DC produces a different partitioning of the data for each of the two scenarios. As can be seen from Figure 10(b) and Figure 11(b), the distribution of transactions over the discovered clusters suggests that the quality of the clusters produced by AT-DC is also good.

Mushrooms: This dataset contains 8124 tuples, each representing a mushroom characterized by 22 attributes, such as color, shape, odor, etc. Each mushroom is classified as either "Poisonous" or "Edible". There are 4208 edible and 3916 poisonous mushrooms in total. There are 2,480 missing values which are ignored during clustering. In order to convert this data to transactional data, we represent each tuple as a set of Attribute/Value pairs. Figure 12 illustrate the clustering results of TRANCLUS and AT-DC for this dataset. As can be seen from this figure, TRANCLUS identifies the two expected clusters with accuracy $\approx 90\%$ (only 10% of all the tuples are misclassified). This means that our algorithm discovers two output clusters in each of which the majority of points come from one input cluster. These results are generally indicative of a clean mapping from input to output clusters. On the other hand, in contrast to TRANCLUS, AT-DC achieves

ClusterId.	*Mammal*	*Bird*	*Invertebrate*	*Insect*	*Fish*	*Reptile*	*Amphibian*
C_1	41	0	0	0	0	2	0
C_2	0	20	0	0	0	0	0
C_3	0	0	10	0	0	0	0
C_4	0	0	0	8	0	0	0
C_5	0	0	0	0	13	3	4

(a) Confusion matrix of TRANCLUS

ClusterId.	*Mammal*	*Bird*	*Invertebrate*	*Insect*	*Fish*	*Reptile*	*Amphibian*
C_1	41	0	0	0	0	2	1
C_2	0	20	0	0	0	0	0
C_3	0	0	0	0	13	3	0
C_4	0	0	7	0	0	0	0
C_5	0	0	2	8	0	0	0
C_6	0	0	0	0	0	0	3
C_7	0	0	1	0	0	0	0

(b) Confusion matrix of AT-DC

Fig. 13. Clustering results for Zoo.

purity of classes by producing more clusters. Specifically, as depicted in Figure 12(b), the algorithm produces eight clusters, of which two (C_1 and C_4) contain misclassified tuples.

Zoo: This dataset contains 101 records with 16 Boolean-valued attributes such that each record corresponds to an animal. There are seven types of animals in this dataset. As can be seen from Figure 13(a), our algorithm identifies five clusters. As evident from the confusion matrix reported in this figure, the clustering quality is good. In fact, the Mammal, Bird, Invertebrate and Insect classes are accurately discovered in clusters C_1, C_2, C_3 and C_4, respectively. In cluster C_5, TRANCLUS confuses Reptile and Amphibian with Fish. This behavior of TRANCLUS is mainly due to the fact that in the Zoo data, animals in these three classes share a number of features in common. In addition to this, the Reptile class and the Amphibian class contain a limited number of animals: 5 and 4 animals respectively. The small number of samples, combined with the fact that animals in these two classes have some characteristics in common with animals in the Fish class, makes it difficult to distinguish between them. On the other hand, as depicted in Figure 13(b), the results of AT-DC are quite similar to those of TRANCLUS. The main source of error for AT-DC is that it confuses Fish with Reptile while, in contrast to TRANCLUS, the algorithm discovers three of the four amphibian animals in cluster C_6. Overall, the performance of TRANCLUS and AT-DC on the Zoo data is good since they succeed in identifying pure clusters (TRANCLUS misclassifies 9 records while AT-DC misclassifies 8 records).

Internet Advertisements: This dataset comprises a set of possible advertisements on Internet pages. It contains 3279 records such that each record represents a Web page. The features in this data encode phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. In total, there are 1557 features, of which 1554 are Boolean and the three remaining ones are "categorical" in nature (although they are numeric, several values occur frequently). Each record within this data is transformed to a transaction by con-

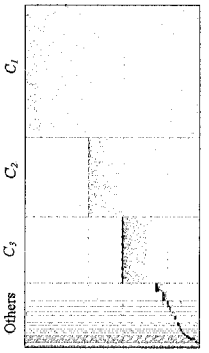


Fig. 14. Visualization of the clustering result of TRANCLUS on Internet Advertisements.

ClusterId.	*ad.*	*noad.*
C_1	157	1070
C_2	106	623
C_3	32	574
$C_4 - C_{79}$	0	494
$C_{80} - C_{92}$	77	0
C_{93}	78	2
C_{94}	2	28
C_{95}	3	21
C_{96}	1	4
C_{97}	1	3
C_{98}	1	2

(a) Confusion matrix of TRANCLUS.

ClusterId.	*ad.*	*noad.*
C_1	103	1263
C_2	3	163
C_3	27	450
C_4	52	525
C_5	78	2
C_6	0	75
C_7	170	3
C_8	0	74
C_9	2	75
C_{10}	0	53
C_{11}	0	50
C_{12}	0	23
C_{13}	0	30
C_{14}	24	19

(b) Confusion matrix of AT-DC.

Fig. 15. Clustering results for Internet Advertisements.

sidering non-zero entries only. This yields a transactional dataset with 2832 items. Finally, we note that the Internet Advertisements data is quite unbalanced, since 458 records are labeled as "advertisement" (ad) and 2821 records are labeled as "non-advertisement" (noad). Figure 14 illustrates the resulting transaction/item incidence matrix for the Internet Advertisements data using TRANCLUS. As is clearly visible from this figure, TRANCLUS produces a block-triangular matrix in which each block corresponds to a cluster (lines in the figure correspond to the frontier of each cluster). Such block structure indicates that clusters discovered by our algorithm are characterized by distinctive subsets of items, which in turn indicates the good quality of the clustering.

As depicted in Figure 15(a), TRANCLUS achieves reasonable class purity by producing 98 clusters. Clusters C_1, C_2 and C_3, which represent the class "noad", are the largest in size. Clusters $C_4 - C_{79}$ and $C_{94} - C_{98}$ also correspond to the class

"noad", while clusters $C_{80} - C_{93}$ represent the minority class (i.e., the class "ad"). As is visible from Figure 14, except for C_1, C_2 and C_3, the clusters are of relatively small size (the size of clusters varies from 6 to 80). The main reason TRANCLUS generates a large number of clusters is due to the criterion function $CF(C)$ which favors compact, separate structures. This is clearly apparent in Figure 14, in which we can see that the each of the small clusters contains a distinctive small subset of items. Recall that the Internet Advertisements data is of high dimensionality. In this situation, the results of TRANCLUS on this data reflect the general trend for high-dimensional data, in which clusters may hide in different subspaces [Bouguessa and Wang 2009; Agrawal et al. 2005; Aggarwal and Yu 2002]. On the other hand, from Figure 15(b) we can see that AT-DC also achieves a reasonable purity of classes by producing 14 clusters, among which C_5 and C_7 represent the class "ad". The error rate of AT-DC is 0.064 and that of TRANCLUS is 0.093. The main source of error for both algorithms is the minority class "ad", some of whose records are confused with the class "noad". In our investigation, we found that, in general, records in these two classes could be identical with respect to some attribute values. This is why both algorithms tend to generate several clusters, with respect to distinctive subsets of items, in each of which some records of "ad" are combined with records from "noad".

6. APPLICATION TO YAHOO! ANSWERS

In this section, we put our approach to work using data from Yahoo! Answers. First we identify authoritative users and then detect communities that form around them. The following steps summarize our approach to identify knowledge-sharing communities for a given category in Yahoo! Answers.

(1) Apply Algorithm 2 to identify the set E of authoritative users;
(2) Based on Definition 1, associate with each asker a_i $(i = 1, \ldots, n)$ a transaction T_i that summarizes his/her interactions with authoritative users.
(3) Apply TRANCLUS (Algorithm 3) to cluster the set of transactions $\{T_i\}$.

The resulting clusters from the above procedure correspond to the communities that we are attempting to discover. Below, we describe the datasets used in our experiments and then report the results, followed by a discussion.

6.1 Datasets

We conduct experiments on datasets which represent users' activities over one full year for six categories: "Biology", "Chemistry", "Engineering", "Mathematics", "Physics" and "Programming & Design". Some statistics on the datasets are reported in Figure 16. Note that due to commercial-in-confidence, all the dataset statistics are reported as percentages. As illustrated in this table, more than half the users in each category ask questions only. This indicates that Yahoo! Answers is really a place where users come to get answers to their questions by relying on other users' expertise on different topics. Also, it is interesting to see that a large fraction of users are only interested in sharing their knowledge by answering questions only. This distribution of users' activities is very common in many other categories not described here in which knowledge sharing and factual expertise are

Category	*%users who ask only*	*%users who answer only*	*%users who ask and answer*
Biology	60%	36%	4%
Chemistry	63%	32%	5%
Engineering	65%	31%	4%
Mathematics	64%	31%	5%
Physics	60%	34%	6%
Programming	66%	29%	5%

Fig. 16. Datasets statistics.

Category	*%authoritative users*
Biology	0.66%
Chemistry	0.70%
Engineering	0.74%
Mathematics	0.50%
Physics	0.70%
Programming	0.72%

Fig. 17. Percentage of authoritative users in each category.

sought. It can however be less clear in some other categories where users are more engaged in expressing their opinion than in knowledge sharing (this is the case, for example, in the "Politics" category).

6.2 Identifying Authoritative Users

We use our procedure described in Algorithm 2 to automatically identify authoritative users in each of the categories presented in Figure 16. We also provide an analysis to evaluate the suitability of the obtained results. It is worth pointing out, however, that there is no standard method in the literature to which our technique for identifying authoritative users could be compared. To the best of our knowledge, the method that we propose is the first attempt to automatically discriminate authoritative and non-authoritative users; existing approaches provide only a ranked list of users. Furthermore, there is also a shortage of standard benchmark data which could be used to evaluate approaches designed to identify experts/authoritative users. For all of these reasons, evaluating the proposed method is a challenging task. In view of this, we have adopted a principled way of evaluating the authoritative user identification technique presented in this paper. Below, we give a salient illustration of the suitability of our method.

We investigated all of the categories described in Figure 16 in order to identify authoritative users. As mentioned in Section 3.2, in all cases we found that the authority scores are well fitted by two gamma components. The component that contains large authority score values represents authoritative users. In general, we found only a few hundred users who are authoritative. Figure 17 provides an idea of the percentage of authoritative users identified in each category.

In order to evaluate the effectiveness of our approach, we looked at the behavior/activity of a significant number of identified authoritative users on the site and manually evaluated their answers. Since there are millions of questions and answers in Yahoo! Answers, it was impossible for us to investigate all of the users manually. We thus selected a few hundred of authoritative users from the cate-

gories "Programming & Design" and "Mathematics". We specifically chose these two categories because they are close to our domains of expertise.

We performed a thorough analysis and observed that all of the selected users are very active, with a strong presence on the site. In most cases, they provide detailed answers of good quality to a large number of questions. Furthermore, we examined each selected user's *profile page* (in Yahoo! Answers, profile pages allow individuals to provide information about themselves and their expertise). This yielded very interesting and encouraging results. For instance, we found that the selected users in the "Mathematics" category included math teachers and graduate students. In the category "Programming & Design", there were a number of software engineers, Web programmers and students. Such users are valuable sources of knowledge.

In our investigation we also observed that such users play a significant role in regulating the whole system on the site. In several cases, they provide an objective evaluation of the answers of other users through the voting mechanism available on the site. We believe such users are potential candidates to perform a given organizational role on the site. Based on these encouraging results, we expect other authoritative users identified by our approach in other categories to display similar behavior to the users we analyzed from the categories "Programming & Design" and "Mathematics". To confirm our claim, we will now investigate the quality of the content generated by all the identified authoritative users in a more systematic way.

6.2.1 *Quality of Content.* The aim of this set of experiments is to evaluate the quality of the content generated by all the identified authoritative users. We expect these users to generate high-quality content (i.e., questions and answers of high quality). Hence, evaluating the quality of the content generated by authoritative users can also be a validation of the suitability of our method. For this purpose, we use the quality metric described in [Agichtein et al. 2008] as the "gold standard" for evaluation. We provide, below, a brief description of this approach.

The work in [Agichtein et al. 2008] addresses the problem of identifying high-quality content in question-answering Web sites. The approach combines analysis of the textual content with user feedback on the site in order to estimate a quality score for each question and answer. The quality score described in [Agichtein et al. 2008] is the confidence score of a binary classifier trained on high and low quality examples. The value of the quality score is always between 0 and 1. When the question or answer is of high quality, the value of the quality score is close to 1. On the other hand, a question or answer with low quality receives a very small quality score value (close to 0). The experiments in [Agichtein et al. 2008] illustrate that such an approach to identifying high-quality content achieves an accuracy close to that of humans. Figure 18 shows the average quality score of authoritative users identified by our approach in each category.

As we can see from this figure, the average quality score of authoritative users in each category is generally between 0.7 and 0.77, which is a relatively high quality score. This result constitutes another source of confirmation concerning the suitability of our approach for identifying users that contribute significantly to the generation of high-quality content in Yahoo! Answers. Moreover, such results also indicate that askers are very selective in choosing the best answerers in categories

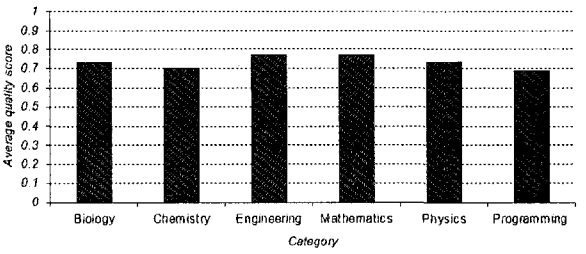


Fig. 18. Average quality score of the answers of the identified authoritative users in each category.

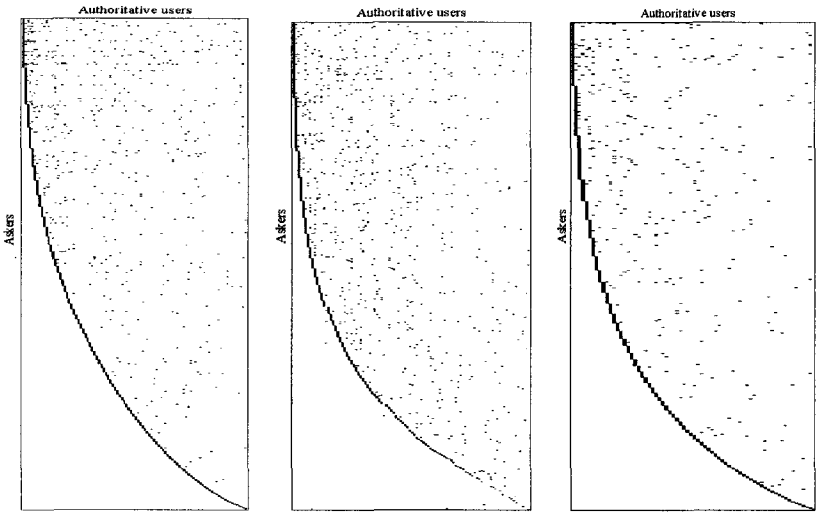
that require factual answers. We can thus rely on them and on their judgment with a pretty high level of confidence.

6.3 Identifying Communities

Now that authoritative users have been identified, let us focus on detecting knowledge-sharing communities. To this end, we first exploit the interactions between askers and authoritative users in Yahoo! Answers through the question-answering process and then represent such interactions as a type of transaction. Next, we use TRANCLUS to cluster askers on the basis of their interactions with authoritative users. In the previous section we tested our clustering algorithm on a number of synthetic and public, real-life datasets for which the cluster structure was known beforehand. The results demonstrate the suitability of our algorithm to discover meaningful structures. In contrast to the experiments devised in Section 5, in this set of experiments we analyze data for which the cluster structure is not known. Hence, we apply TRANCLUS in an exploratory fashion, and we report our findings.

Before describing our results, it should be noted that we did not consider existing graph-based community detection algorithms in the following experiments. Such approaches could not be used to identify the knowledge-sharing communities that we aim to discover because they are not primarily designed for that purpose. Most existing community detection methods represent their environment as a graph and define a community as a set of nodes which are more densely connected than elsewhere. Such a definition does not effectively reflect the interactions between askers and authoritative users in question-answering forums. It is consequently not straightforward, and indeed very difficult, to perform a fair and objective comparison between our approach and existing ones.

We used TRANCLUS to discover knowledge-sharing communities in all of the categories described in Figure 16. For purposes of illustration, Figure 19 shows the results when our algorithm is applied to the categories Biology, Chemistry and Engineering. The results depicted in this figure are very representative of the general trend of our algorithm for the remaining categories described in Figure 16. Note that, as with the experiments conducted in Section 5, the clustering results depicted in Figure 19 represent the transaction/item incidence matrix which summarizes the interaction between users in each category. The rows of each matrix



(a) Biology. (b) Chemistry. (c) Engineering.

Fig. 19. Clustering results for three selected categories.

correspond to askers, while the columns correspond to authoritative users. The shaded regions in each matrix represent non-zero entries, which in turn indicate the presence of an interaction between askers and authoritative users.

From a visual inspection of Figure 19 we can assess the results. The block-diagonal structure in each matrix testifies to the good quality of the clusters identified by TRANCLUS. The community structure is clearly visible since each of the discovered clusters is characterized by a distinctive subset of authoritative users. Interestingly, one particularly notable feature of the results is that in the majority of the clusters discovered by TRANCLUS, we found that askers interact most frequently with the same authoritative user. This is clearly apparent in Figure 19. In fact, the majority of the blocks (or dense regions) in each of the matrices depicted in this figure correspond to the interactions of askers (grouped in the same cluster) with one authoritative user. This means that within each of the discovered communities, askers revolve around one dominant authoritative user. The number of askers in each discovered community varies from 10 to more than 2500 for the categories considered in this set of experiments.

Our findings described above are a good reflection of human interactions in forums designed for the sole purpose of knowledge sharing. In fact, as illustrated in Figure 16, in technically focused forums the majority of participants are novices (more than half of the users ask questions only). Those who have expertise will primarily answer, while those who do not will be posting the majority of the questions. As stated earlier, authoritative users do not answer questions at random; they have a certain degree of focus. Thus, they answer questions for which they believe they can provide thorough help to askers, without keeping close track of

the help provided and ensuring that they receive equal benefits. Askers, for their part, post focused questions that require factual answers. In this setting, it is clear that the interaction between participants is not random. Askers tend to interact repeatedly with a relatively small set of authoritative users who are interested by their questions. One such mechanism directs askers to membership in clusters in which they closely interact with a few authoritative users. In our application, we found that the majority of the discovered communities are principally built around one dominant authoritative user.

Another distinguishing characteristic for the knowledge-sharing communities identified by our clustering algorithm is that the overlap among them is relatively low. That is, the number of authoritative users that appear in several clusters is small. We can observe this from Figure 19. In fact, in addition to the block diagonal structures, we observe in each matrix depicted in this figure that there are a number of points distributed across all the clusters. These points indicate the presence of interaction between an asker and authoritative user. The knowledgeable reader can observe in this rendering that the distribution of such points is relatively sparse. This indicates that the overlap among the clusters is relatively low. To quantitatively illustrate this point, we estimate the average overlap among the discovered clusters in each category as follows:

$$av\_overlap = \frac{\sum_{i=1}^{nc} \sum_{j=i+1}^{nc} overlap(C_i, C_j)}{nc(nc-1)/2} \tag{18}$$

Recall that nc is the number of clusters while $overlap(C_i, C_j)$ denotes the overlap between two clusters, defined as

$$overlap(C_i, C_j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|} \tag{19}$$

where, E_i and E_j correspond to the set of authoritative users in cluster C_i and C_j respectively. The component $overlap(C_i, C_j)$ is simply the *Jaccard coefficient* and takes values between 0 and 1. A value of $overlap(C_i, C_j)$ close to 0 indicates that the number of authoritative users that appears in both clusters C_i and C_j is intrinsically low. Accordingly, a value of *av_overlap* close to 0 indicates the same think for all the discovered clusters. Figure 20 summarize the average overlap between the communities discovered by TRANCLUS for the six categories considered in this paper. As can be seen from this figure, the overlap between the discovered clusters is relatively low. This means that authoritative users do not answers questions in an anarchic manner, but they are focused to provide help to a specific set of askers who share an interest with them. This testifies to effectiveness of our authoritative users' identification procedure in discovering the most knowledgeable users on the site that are willing to help other participants.

To summarize, the nature of the interactions between askers and authoritative users, in factual technically focused forums, tend to foster communities in which askers and authoritative users are strongly connected to each other in the sense that: 1) askers interact repeatedly with a limited number of authoritative users, and 2) the number of authoritative users that occur in different communities is relatively small.

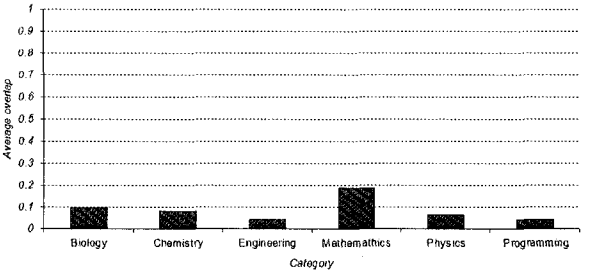


Fig. 20. Average overlap between the discovered clusters in each category.

Cluster1	{PHP, Website, HTML, JavaScript, Ajax, Java}
Cluster2	{C + +, net, games, Windows, Java, Microsoft}

(a) Programming

Cluster1	{electricity, circuit, transistor, capacitor, battery, resistor, signal, amplifier}
Cluster2	{mechanic, engine, motor, design, piping, fluid, machine}

(b) Engineering

Cluster1	{cell, dna, blood, human, chromosome, gene, virus}
Cluster2	{animal, mitosis, meiosis, cell, bacteria, chromosome, genetic}

(c) Biology

Fig. 21. The most frequent words identified within each selected cluster.

To provide a more qualitative analysis of our results, we also investigated the content generated by the clustered askers. Our goal was to verify whether askers who form a community did in fact post questions on the same topics. To this end, we performed an analysis on three categories: Programming, Engineering and Biology. Specifically, for each discovered cluster, we examined all the questions posted by askers in order to extract the most frequent words. Our investigation is based on the fact that questions under the same topic should share a set of common words. In this setting, it is reasonable to assume that the most frequent words represent the topic description of a cluster. Figure 21, illustrates the most frequent and relevant words for two selected communities for each of the three categories. Note that before extracting the most frequent words, we performed some preprocessing operations like removal of stop words and ignoring of some non-relevant frequent words such as "help", "question", "problem", etc.

From Figure 21 we can assess that the set of frequent words identified within each selected cluster indicates that, in Yahoo! Answers, the clustered askers tend to post questions on closely related topics. For instance, Figure 21(a) suggest that questions in the first cluster are about Web programming while questions in the second cluster are related to games programming in C++ and Java. Also, from Figure 21(b), we can surmise that questions posted by askers of Cluster 1 revolve around different subjects of electrical engineering while those in Cluster 2 are about mechanical engineering. Finally, it appears that the questions asked by the members

of the first cluster illustrated in Figure 21(c) are more related to human biology while questions in the second cluster are more related to animals.

The results depicted in Figure 21 suggest that questions posted by askers of the same community are closely related. Such results could be explained by the fact that, in Yahoo! Answers, we have found that each community of askers identified by TRANCLUS is built around one dominant authoritative user who does not appear frequently in other communities. Accordingly, askers of the same community are highly likely to ask questions on closely related topics since they interact with one authoritative user and this authoritative user provides answers to questions which are related to his specific domain of expertise.

7. LITERATURE REVIEW

7.1 Finding Community Structure

Depending on the application domain, a number of approaches for identifying community structure in different contexts have been proposed [Radicchi et al. 2004; Girvan and Newman 2002; Flake et al. 2000; Kumar et al. 1999; Gibson et al. 1998]. In these studies, the term community has been defined in more than one way [Zhang et al. 2007; Balakrishnan and Deo 2006; Radicchi et al. 2004]. For instance, Gibson et al. [1998] define a community in the Web as containing a core of central, "authoritative pages" (highly referenced pages) linked together by "hub pages" (pages that point to the authorities). In the same context of the WWW, Kumar et al. [1999] define a community as a dense directed bipartite subgraph. A dense bipartite graph is a graph whose node set can be partitioned into two sets L and R such that every node in L links to every node in R. Dourisboure et al. [2007] define a community in the Web as containing two sets of pages: the set of the Y centers of the community, i.e., pages sharing a common topic; and the set X of the fans, i.e., pages that are interested in the topic. Typically, every fan contains a link to most of the centers; at the same time, there are few links among centers and among fans. In contrast to all these approaches, Flake et al. [2000] define a community in the Web as a set of sites that have more links to members of the community than to non-members.

In the context of social and biological networks, Girvan and Newman [2002] define a communities in a network as subsets of vertices within which vertex-vertex connections are dense, but between which connections are sparse. Radicchi et al. [2004] discuss the case of two quantitative definitions of community. Specifically, they introduce the concept of "strong community" and "weak community". In a "strong community" each node has more connections within the community than with the rest of the graph. In a "weak community", the sum of all the edges connecting to nodes within a community is larger than the sum of all the connections toward nodes in the rest of the network. Zhou et al. [2006] introduce the concept of "semantic community" in social networks. A semantic community in a social network includes users with similar communications interests and topics that are associated with their communication. For more survey on existing approaches we refer the reader to [Danon et al. 2005; Newman 2004].

To summarize, the concept of community is general and its definition depends substantially on the context [Zhang et al. 2007; Radicchi et al. 2004]. Accordingly,

the approaches discussed above have been developed for different circumstances, depending on: 1) the context of the study; 2) the data representation model; and 3) the type of community that the algorithm is targeted to discover. The focus of this paper is, however, different from all of the existing work since we tackle with the problem of identifying knowledge-sharing communities in the particular context of question-answering forums. To the best of our knowledge, the problem in this form has not been addressed in the literature so far.

7.2 Expert-Finding Approaches

The problem of identifying authoritative users is mainly related to the problem of expert identification. Expertise-finder systems have been explored in a number of studies [Zhang et al. 2007; Maybury 2006; Yimam and Kobsa 2003; Ackerman et al. 2002]. In the past few years, a number of commercial systems have been developed that an enterprise can deploy to support finding its own experts or those of other organizations. These systems attempt to leverage the social network (relationships among people) within an organization to help find the appropriate expert [Zhang et al. 2007]. For instance, Referral Web [Kautz et al. 1997] from AT&T provides access to experts across an enterprise, aiming to make the basis for referral transparent to the user. It generates social networks based on bibliographic information and supporting context to deduce actual experts and associated referral paths. Autonomy's IDOL Server [5] is another commercial system that analyzes employees' search and publication histories, on the basis of the documents they access and submit on the intranet, to determine concepts that are indicative of their expertise.

Systems such as Tacit's KnowledgeMail [6] and Xpertfinder [Sihn and Heeren 2001] determine user expertise from email message traffic. Tacit's KnowledgeMail builds user interest profiles by scanning email and matching it to document taxonomies. Xpertfinder uses a pre-existing hierarchy of subject areas, characterized by word frequencies, to identify experts in specific areas by analyzing the word frequencies in email written by each individual. Mattox et al. [1999] describe a system named ExpertFinder which exploits technical papers, presentations, resumes, home pages etc. on MITRE's corporate intranet to enable the location of relevant experts. ExpertFinder considers someone as an expert on a particular topic if they are linked to a wide range and/or a large number of documents about that topic. Specifically, as mentioned in [Campbell et al. 2003], ExpertFinder uses the number of self-published documents containing topic keyword(s) and the frequency of person mentions near topic keyword(s) in non-self-published documents to produce expertise scores and ranks.

A common feature of the majority of the approaches discussed above is the creation of knowledge profiles via the frequency of encountered keywords. In this way, such approaches may reflect whether a person knows about a topic, but it is difficult to assess that person's relative expertise [Zhang et al. 2007]. Recently, Campbell et al. [2003] compared a content-based approach that looks only at email content and a graph-based ranking method that looks at social networks from email communications. They found that the graph-based algorithm extract more information

[5] www.autonomy.com
[6] www.tacit.com

than the content-based approach. Likewise, Dom et al. [2003] compare various ranking algorithms, including HITS and PageRank, on both artificial and email networks. The experiments in [Dom et al. 2003] show that PageRank performs better than other ranking algorithms.

Zhang et al. [2007] analyze data from the Java Forum seeking to identify users with high expertise. For this purpose, they evaluate several graph-based ranking algorithms, including HITS and ExpertiseRank (a PageRank-like algorithm). In addition, in order to perform more comparison between existing ranking methods, the authors also develop a synthetic model that simulates various network structures. The experiment in [Zhang et al. 2007] reveals that a simple link-based metric could be a powerful tool for measuring the expertise level of participants. Further details and survey on the problem of identifying authoritative actors can be found in [Maybury 2006; Yimam and Kobsa 2003; Ackerman et al. 2002].

The output of the vast majority of the approaches discussed above is a list of all users ranked according to their level of expertise/authority. A major problem for such approaches is to determine how many users should be chosen as authoritative from a ranked list. In Section 3, we effectively addressed this issue by describing an approach that allows automatic identification of authoritative users without any parameter setting.

7.3 Transactional Clustering Algorithms

Transactional data is a particular facet of categorical data in which records are made up of non-numerical values. Specifically, a transactional dataset can be transformed into a traditional categorical dataset (a row-by-column Boolean table) by treating each item as an attribute and each transaction as a row [Cesario et al. 2007; Yang et al. 2002]. Categorical clustering algorithms [Zaki et al. 2007] can then be used to cluster such transformed data. However, the transformation of data from transactional to Boolean significantly increases the dimensionality of the set and thus may affect the efficiency of categorical clustering algorithms [Yan et al. 2006]. To alleviate this problem, a number of transactional clustering algorithms have been proposed. Below, we discuss previous work on clustering transactional data.

The LargeItem algorithm introduced by Wang et al. [1999] cluster the transactions by iteratively optimizing a global criterion function which is based on the notion of large items (i.e., items in a cluster having occurrence rates greater than the user-defined parameter θ). The main assumption of this approach is that large items are "popular" items in a cluster and consequently contribute to similarity in a cluster, whereas small items contribute to dissimilarity in a cluster. The criterion function proposed in [Wang et al. 1999] attempts to minimize two components: 1) the inter-cluster similarity, which measures the overlap of large items between clusters, and 2) the intra-cluster similarity, which measures the union summation of small items. In order to put these two components together in a single criterion function, the authors introduce a weight w that controls their relative importance in the clustering process. A weight $w > 1$ gives more emphasis to the intra-cluster similarity and a weight $w < 1$ gives more emphasis to inter-cluster dissimilarity.

The CLOPE algorithm introduced by Yang et al. [2002] is similar in structure to LargeItem but uses a different criterion function. Unlike LargeItem, the authors of CLOPE do not consider the inter-cluster similarity. Specifically, CLOPE attempts

to maximize only the intra-cluster similarity, based on the fact that the goodness of a cluster is higher if the average frequency of an item is high compared to the number of items appearing within a transaction. In order to control the number of clusters, the authors introduce the repulsion parameter r in the criterion function. It is worth noting that both CLOPE and LargeItem are suitable for clustering large databases, since they attempt to optimize a global criterion function. Computing global criterion functions is much faster than local criterion functions based on pair-wise similarities [Yang et al. 2002]. On the other hand, LargeItem and CLOPE share a common drawback related to their dependence on a set of parameters (i.e., θ, w and r) that need to be properly tuned. Because there is no clear guideline to find appropriate settings of these parameters, proper tuning is difficult in real applications.

Giannotti et al. [2002] consider the problem of clustering Web log sessions. A session is defined in [Giannotti et al. 2002] as a set of Web pages visited by a user in a semantically homogeneous way and it is represented as a type of transaction. In order to cluster such user sessions, the authors propose a K-means-based transactional clustering algorithm named Transactional K-means (TrK-means). Similar to the standard K-means, TrK-means attempts to minimize the squared distance between each transaction and the cluster representative of its cluster. The distance function considered in [Giannotti et al. 2002] is the Jaccard coefficient while the cluster representative is the set of large items in the cluster. The user of TrK-means must set two parameters: the number of clusters and the cluster representative threshold. Like the standard K-means, the TrK-means is fast, but its use in real applications is very limited since, in practice, the number of clusters is usually unknown to the user. Further, the dependence of the algorithm on the cluster representative threshold could affect its accuracy, since setting different threshold values may lead to different clustering results. Other transactional clustering algorithms such as OAK [Xiao and Dunham 2001] and K-tode [Yun et al. 2004] also suffer from their dependence on the number of clusters, which needs to be set by the user.

Yang and Padmanabhan [2005] introduce a pattern-based approach named GHIC to deal with the specific problem of clustering customer Web transactions. The approach is based on the idea that there may be natural behavioral patterns among customers in different groups of transactions. To represent behavior patterns in Web transactions, the authors use the Apriori algorithm [Agrawal and Srikant 1994] to identify frequent itemsets. In contrast to all existing transactional clustering algorithms, GHIC allows a set of itemsets to describe a cluster instead of just a set of items. The authors in [Yang and Padmanabhan 2005] argue why such a strategy is appropriate for clustering customer transactions. GHIC is hierarchical in nature and attempts to maximize a criterion function which is based on two measures: 1) the difference between clusters and 2) the similarity of transactions within a cluster. The difference between clusters, which corresponds to the inter-cluster similarity, is based on the fact that the support of any pattern in one cluster should be different from the support in the other cluster. The similarity measure for a cluster, which corresponds to the intra-cluster similarity, is simply the number of frequent itemsets (i.e., patterns) in the cluster. In order to generate "balanced

clusters", the authors introduce another component to the criterion function and three user-specified weights to bring the difference and similarity measures to comparable values. The experiments in [Yang and Padmanabhan 2005] illustrate that GHIC is able to successfully cluster customer transactions. Unfortunately, however, this approach is parameter-laden. Further, since GHIC is designed to deal with a specific problem in which it adopts the new approach of associating itemsets with behavior patterns and using that concept to guide the clustering process, it cannot be applied to the general problem of clustering transactional data.

Yan et al. [2006] introduce an iterative algorithm named WCD similar in structure to LargeItem and CLOPE. The algorithm attempts to maximize a "density"-based criterion function named Expected Weighted Coverage Density (EWCD). The EWCD criterion function, which corresponds to the intra-cluster similarity measure, is based on the frequency of specific groups of items: the higher the frequency of such groups, the stronger the clustering. The WCD algorithm needs the number of clusters as an input parameter. To address this problem, the authors introduced the SCALE framework which is designed to perform the transactional data clustering in four steps: 1) Sampling, 2) Clustering structure assessment, 3) Clustering and 4) Evaluation. In the first and second step, SCALE determines candidates for critical clustering structures and generates the candidate for the best number Ks of clusters. In third step, SCALE runs the WCD algorithm with different numbers of clusters and choose, in the fourth step, the clustering that optimizes a specific validity index. To this end, the authors propose two validity indices. The first is the LISR measure, which aims to measure the preservation of frequent itemsets. The cluster number that maximizes LISR is considered to be the optimal number of clusters. The computation of the LISR index depends on the minimum support, which is a user-specified parameter. The second validity index is the AMI measure which aims to measure the inter-dissimilarity of clusters. The maximum of AMI, as a function of the number of clusters, is sought for a well-defined partition.

The WCD algorithm alleviates some of the drawbacks of its predecessors algorithms since it addresses the problem of identifying the "optimal" number of clusters in the data. However, the strategy adopted entails an extra computational cost due to a repeated running of the algorithm with different numbers of clusters. Also, computation of the LISR index depends on a user-defined parameter which, may introduce a usability problem in real applications. Further, the results of WCD depend heavily on the sampled data extracted in the first step of SCALE. In several practical cases, the sampled data do not necessarily reflect/preserve the clustering structure of the whole dataset.

Recently, Cesario et al. [2007] introduced the AT-DC algorithm, which is a divisive approach that resembles the general schema of a top-down decision tree learning algorithm. AT-DC starts from an initial partition containing a single cluster which represents the whole dataset and then iteratively splits a cluster within the partition into two sub-clusters. During this iterative process, based on a quality measure, the algorithm evaluates whether the split yields a new partition that exhibits better clustering quality than the original cluster. If the generated sub-clusters are of higher quality than the original cluster, it is removed and the sub-clusters are retained. Otherwise, the sub-clusters are discarded and a new candidate cluster

is considered for splitting. The quality function used to measure the quality of a partition is based on the fact that items that contribute to the formation of clusters exhibit higher local (within-cluster) occurrence frequency compared to their occurrence frequency in the whole dataset. TRANCLUS and AT-CD improve on the aforementioned approaches since they implement a parameter-free, fully automatic strategy to cluster transactions.

8. CONCLUSION

In this paper, we addressed the problem of identifying community structures in question-answering forums. As a case study, we focused on Yahoo! Answers, a large and diverse online question-answering site. Specifically, we analyzed data from Yahoo! Answers categories in which interactions between participants favor knowledge sharing and factual expertise. We began by discussing the benefits of identifying authoritative users that provide the most valuable information and we illustrated that such knowledgeable users can play critical roles in fostering and sustaining communities in question-answering forums. In fact, we showed that the interactions between askers and authoritative users lead to the formation of communities in which seeking and sharing knowledge is the first priority. In this context, we introduced the concept of a knowledge-sharing community. This type of community is defined by a set of askers and authoritative users such that, within each community, askers exhibit more homogeneous behavior in terms of their interactions with authoritative users than elsewhere. A procedure was devised to discover members of such a community. In our approach we proceeded in two phases: first, we identified authoritative users and then we discovered the communities that form around them.

In order to automatically identify authoritative users, we proposed a probabilistic approach based on a mixture model. First, we estimated the authority score of each user. Next, we analyzed their statistical properties. We found that the authority scores are well fitted by two gamma components. One of these two components, which contains large values of the authority score, corresponds to authoritative users. Once authoritative users have been detected, we focused on the problem of detecting communities. To this end, we represented our environment as a type of transactional data such that each transaction summarizes the interaction of one asker with authoritative users who answered his/her questions. Then, we developed a parameter-free transactional clustering algorithm named TRANCLUS to cluster askers who exhibit homogeneous behaviors in terms of their interaction with authoritative users. The suitability of TRANCLUS was demonstrated through an empirical study on both synthetic and public, real-life data.

Finally, we put our approach to work using data from six Yahoo! Answers categories which represent users' activities over one full year. First, we began by detecting authoritative users in each category. Experiments showed that our authoritative user identification procedure is able to effectively identify, in a fully systematic manner, the most prominent actors who are rich sources of knowledge. We also evaluated the content generated by the identified authoritative users. Our results clearly demonstrate that such users contribute significantly to the generation of high-quality content. Then, we used TRANCLUS to cluster askers on the basis

of their interaction with authoritative users. Interestingly, the algorithm discovered strong communities within each of which askers are closely clustered around one dominant authoritative user. This "dense" community structure supports trust, cooperation, and closer communication and thus facilitates knowledge transfers - which corresponds, in the end, to the goal of the approach devised in this paper.

ACKNOWLEDGMENTS

We gratefully thank Dr. Giuseppe Manco for providing the synthetic data program generator. We also thank the reviewers for their valuable comments and important suggestions.

REFERENCES

ACKERMAN, M. S., WULF, V., AND PIPEK, V. 2002. *Sharing Expertise: Beyond Knowledge Management.* MIT Press.

ADAMIC, L. A., ZHANG, J., BAKSHY, E., AND ACKERMAN, M. S. 2008. Knowledge sharing and Yahoo! Answers: Everyone knows something. In *Proc. 17th ACM Int'l Conf. World Wide Web (WWW'08).* 665–674.

AGGARWAL, C. AND YU, P. 2002. Redefining clustering for high dimensional applications. *IEEE Trans. Knowledge and Data Eng. 14,* 2, 210–225.

AGICHTEIN, E., CASTILLO, C., DONATO, D., GIONIS, A., AND MISHNE, G. 2008. Finding high-quality content in social media. In *Proc. First ACM Int'l Conf. Web Search and Web Data Mining (WSDM'08).* 183–194.

AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., AND RAGHAVAN, P. 2005. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery 11,* 1, 5–33.

AGRAWAL, R. AND SRIKANT, R. 1994. Fast algorithms for mining association rules. In *Proc. 20th ACM Int'l Conference Very Large Data Bases (VLBD'94).* 487– 499.

BALAKRISHNAN, H. AND DEO, N. 2006. Discovering communities in complex networks. In *Proc. 44th ACM Southeast Regional Conference.* 280–285.

BALAKRISHNAN, N. AND NEVZOROV, V. 2003. *A Primer on Statistical Distributions.* John Wiley and Sons.

BEZDEK, J. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York Plenum.

BOUGUESSA, M., DUMOULIN, B., AND WANG, S. 2008. Identifying authoritative actors in question-answering forums - the case of Yahoo! Answers. In *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'08).* 866–874.

BOUGUESSA, M. AND WANG, S. 2009. Mining projected clusters in high-dimensional spaces. *IEEE Trans. Knowledge and Data Eng. 21,* 4, 507–522.

BOUGUESSA, M., WANG, S., AND SUN, H. 2006. An objective approach to cluster validation. *Pattern Recognition Letters 27,* 13, 1419–1430.

CALDARELLI, G. 2007. Communities and clustering in some social networks - Tutorial. In *Int'l Workshop and Conf. on Network Science (NetSci'07).*

CAMPBELL, C. S., MAGLIO, P. P., COZZI, A., AND DOM, B. 2003. Expertise identification using email communication. In *Proc. 12th ACM Int'l Conf. Information and Knowledge Management (CIKM'03).* 528–531.

CESARIO, E., MANCO, G., AND ORTALE, R. 2007. Top-down parameter-free clustering of high-dimensional categorical data. *IEEE Trans. Knowledge and Data Eng. 12,* 12, 1607–1623.

DANON, L., DIAZ-GUILERA, A., DUCH, J., AND ARENAS, A. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment,* P09008.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society, (Series B) 39,* 1–37.

DOM, B., EIRON, I., COZZI, A., AND ZHANG, Y. 2003. Graph-based ranking algorithms for e-mail expertise. In *Proc. 8th ACM ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03).* 42–48.

DOURISBOURE, Y., GERACI, F., AND PELLEGRINI, M. 2007. Extraction and classification of dense communities in the web. In *Proc. 16th ACM Int'l Conf. World Wide Web (WWW'07).* 461–470.

FIGUEIREDO, M. AND JAIN, A. 2002. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Analysis and Machine Intelligence 24,* 3, 381–396.

FLAKE, G. W., LAWRENCE, S., AND GILES, C. L. 2000. Efficient identification of web communities. In *Proc. 6th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'00).* 150–160.

GIANNOTTI, F., GOZZI, C., AND MANCO, G. 2002. Characterizing web user accesses: a transactional approach to web log clustering. In *Proc. IEEE Int'l Conf. Information Technology: Coding and Computing (ITCC'02).* 312–317.

GIBSON, D., KLEINBERG, J. M., AND RAGHAVAN, P. 1998. Inferring Web communities from link topology. In *Proc. 9th ACM Int'l Conference Hypertext and Hypermedia (HYPERTEXT '98).* 225–234.

GIRVAN, M. AND NEWMAN, M. E. J. 2002. Community structure in social and biological networks. In *Proc. of the National Academy of Sciences of the United States of America (PNAS'02).* Vol. 99. 7821–7826.

GYONGYI, Z., KOUTRIKA, G., PEDERSEN, J., AND MOLINA, H. G. 2008. Questioning Yahoo! Answers. In *Proc. First WWW Workshop on Question Answering on the Web (QAWeb'08).*

HAN, J. AND KAMBER, M. 2006. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann.

HOGG, R., MCKEAN, J., AND CRAIG, A. T. 2005. *Introduction to Mathematical Statistics, sixth ed.* Pearson Prentice Hall.

JAIN, A., DUIN, R., AND MAO, J. 2000. Statistical pattern recognition: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence 22,* 1, 4–37.

KAUTZ, H., SELMAN, B., AND SHAH, M. 1997. Referral web: combining social networks and collaborative filtering. *Communications of the ACM 40,* 3, 63–65.

KEOGH, E., LONARDI, S., AND RATANAMAHATANA, C. A. 2004. Towards parameter-free data mining. In *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'04).* 206–215.

KLEINBERG, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46,* 5, 604–632.

KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. 1999. Trawling the web for emerging cyber-communities. *Computer Networks 31,* 11-16, 1481–1493.

LAWLESS, J. 1982. *Statistical Models and Methods for Lifetime Data.* John Wiley and Sons.

LESSER, E. L. AND STORCK, J. 2001. Communities of practice and organizational performance. *IBM Systems Journal 40,* 4, 831–841.

LI, T. 2005. A general model for clustering binary data. In *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'05).* 188–197.

LIU, X., BOLLEN, J., NELSON, M. L., AND SOMPEL, H. V. 2005. Co-authorship network in the digital library research community. *Information Processing and Management 41,* 6, 1462–1480.

MANNING, C. D., RAGHAVAN, P., AND SCHTZE, H. 2008. *Introduction to Information Retrieval.* Cambridge University Press.

MATTOX, D., MAYBURY, M., AND MOREY, D. 1999. Enterprise expert and knowledge discovery. In *Proc. 8th Int'l Conf. Human-Computer Interaction (HCI'99).* 303 –307.

MAYBURY, M. T. 2006. Expert finding systems. *MITRE Technical Report, MTR 06B000040.*

NEWMAN, M. 2004. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter 38,* 2, 321–330.

OLIVER, J., BAXTER, R., AND WALLACE, C. 1996. Unsupervised learning using MML. *Proc. of the 13th Int'l Conf. Machine Learning (ICML'96)*, 364–372.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The Pagerank citation ranking: Bringing order to the Web. *Stanford Digital Library Technologies Project.*

PRESCOTT, L. 2006. Yahoo! answers captures 96% of Q and A market share.

RADICCHI, F., CASTELLANO, C., CECCONI, F., LORETO, V., AND PARISI, D. 2004. Defining and identifying communities in networks. In *Proc. of the National Academy of Sciences of the United States of America (PNAS'04).* Vol. 101. 2658–2663.

SALVETTI, F. AND SRINIVASAN, S. 2005. Local flow betweenness centrality for clustering community graphs. In *Proc. First Int'l Workshop on Internet and Network Economics, Lecture notes in computer science.* Vol. 3828. 531–544.

SCHWARZ, G. 1978. Estimating the dimension of a model. *Annals of Statistics 6,* 2, 461–464.

SIHN, W. AND HEEREN, F. 2001. XPERTFINDER - expert finding within specified subject areas through analysis of e-mail communication. In *Proceedings of the Euromedia.* 279–283.

TAN, P.-N., STEINBACH, M., AND KUMAR, V. 2006. *Introduction to Data Mining.* Addison Wesley.

WANG, K., XU, C., AND LIU, B. 1999. Clustering transactions using large items. In *Proc. 8th ACM Int'l Conf. Information and Knowledge Management (CIKM'99).* 483–490.

WELSER, H. T., GLEAVE, E., FISHER, D., AND SMITH, M. 2007. Visualizing the signatures of social roles in online discussion groups. *Journal of Social Structure.*

WENGER, E., MCDERMOTT, R. A., AND SNYDER, W. 2002. *Cultivating Communities of Practice: A Guide to Managing Knowledge.* Harvard Business School Press.

XIAO, Y. AND DUNHAM, M. 2001. Interactive clustering for transaction data. In *Proc. 3rd ACM Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK'01), Lecture notes in computer science.* Vol. 2114. 121–130.

YAN, H., CHEN, K., LIU, L., BAE, J., AND YI, Z. 2006. Efficiently clustering transactional data with weighted coverage density. In *Proc. 15th ACM Int'l Conf. Information and Knowledge Management (CIKM'06).* 367–376.

YANG, Y., GUAN, X., AND YOU, J. 2002. CLOPE: A fast and effective clustering algorithm for transactional data. In *Proc. 8th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'02).* 682–687.

YANG, Y. AND PADMANABHAN, B. 2005. GHIC: A hierarchical pattern-based clustering algorithm for grouping web transactions. *IEEE Trans. Knowledge and Data Eng. 17,* 9, 1300–1304.

YANG, Z. 1996. Among-site rate variation and its impact on phylogenetic analysis. *Trends in Ecology and Evolution 11,* 9, 367–372.

YIMAM, D. AND KOBSA, A. 2003. Expert finding systems for organisations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce 13,* 1, 1–24.

YUN, C.-H., CHUANG, K.-T., AND CHEN, M.-S. 2004. Adherence clustering: an efficient method for mining market-basket clusters. *Information systems 31,* 3, 170–186.

ZAKI, M. J., PETERS, M., ASSENT, I., AND SEIDL, T. 2007. Clicks: An effective algorithm for mining subspace clusters in categorical datasets. *Data & Knowledge Engineering 60,* 1, 51–70.

ZHANG, H., GILES, C. L., FOLEY, H. C., AND YEN, J. 2007. Probabilistic community discovery using hierarchical latent gaussian mixture model. In *Proc. of the American Association for Artifcial Intelligence (AAAI'07).* 663–668.

ZHANG, J., ACKERMAN, M., AND ADAMIC, L. 2007. Expertise networks in online communities: Structure and algorithms. In *Proc. 16th ACM Int'l Conf. World Wide Web (WWW'07).* 221–230.

ZHOU, D., MANAVOGLU, E., LI, J., GILES, C. L., AND ZHA, H. 2006. Probabilistic models for discovering e-communities. In *Proc. 15th ACM Int'l Conf. World Wide Web (WWW'06).* 173–182.

Conclusion

Dans cette thèse, nous avons étudié deux problématiques différentes, à savoir le *clustering* des données de hautes dimensions et l'extraction des connaissances dans les services Web de question-réponse. À travers les trois chapitres de cette thèse nous avons présenté trois contributions liées à des thématiques différentes qui représentent des préoccupations actuelles majeures dans le domaine de forage de données.

Dans le premier chapitre de cette thèse, nous avons présenté PCKA, un algorithme de *projected clustering* pour les données de hautes dimensions. Nous avons illustré la fiabilité et la robustesse de notre algorithme à travers des tests et des comparaisons avec les approches existantes. Les résultats expérimentaux démontrent que PCKA est capable d'identifier des *projected clusters* de faible dimensionnalité. Nos tests démontrent aussi que la robustesse de PCKA est toujours maintenue même en présence d'une quantité considérable de bruits dans les données. La bonne performance de notre algorithme sur les données réelles illustre que PCKA est un outil pratique.

PCKA réussit à atteindre cette performance grâce à la restriction imposée dans le calcul de la distance qui considère seulement les attributs qui contiennent des régions denses, et à la stratégie de sélection de ces attributs qui est la phase 1 de PCKA. En effet, la phase 1 de PCKA, qui vise à analyser la pertinence des attributs, représente l'élément central de notre algorithme qui nous a permis de : 1) développer des outils appropriés pour traiter le problème complexe de l'extraction du bruit dans les données de hautes dimensions (pahse #2), et 2) développer une formulation de la distance bien adaptée au contexte particulier des données de hautes dimensions pour identifier correctement les *projected clusters* (pahse #3). Nous estimions que cette stratégie peut être utilisée par d'autres algorithmes de *clustering*, conçus pour des données de faible dimensionnalité, afin de les rendre directement transposable sur des données de dimensions bien supérieures.

PCKA vise à identifier des *projected clusters* qui sont parallèles aux axes. Une extension évidente de notre algorithme est de considérer le cas où les *clusters* sont orientés de façon arbitraire à travers les axes. Une autre direction à explorer dans le futur est d'étendre la portée de la phase 1 de PCKA de l'analyse de la pertinence des attributs à l'analyse de la pertinence et de la redondance des attributs. L'analyse de redondance des attributs paraît être ignorée par la totalité des algorithmes de *projected clustering*.

Dans le deuxième chapitre de cette thèse, nous avons étudié le problème de l'identification automatique des utilisateurs experts dans les forums Internet de question-réponse. Notre contribution inclut le développement d'une approche probabiliste qui se base sur le modèle de mélange de distributions de la loi Gamma. Premièrement, nous avons estimé le degré d'expertise de chaque utilisateur. Deuxièmement, nous avons analysé leurs propriétés statistiques. Dans nos expérimentations, nous avons trouvé que que les degrés d'expertise suivent un mélange de distribution de deux composantes Gamma. Une de ces deux composantes, qui corresponde aux degrés d'expertise les plus élevés, représente les utilisateurs experts.

Nous avons testé notre approche sur des données qui représentent les interactions des utilisateurs dans différentes forums de Yahoo ! Answers. Les résultats expérimentaux démontrent que notre méthode est capable d'identifier, de façon systématique, les utilisateurs les plus prometteurs qui représentent une source potentielle de savoir dans Yahoo ! Answers. De plus, nous avons évalué le contenu (la qualité des réponses) généré dans le site par les utilisateurs experts identifiés par notre approche. Nos résultats démontrent clairement que les experts que nous avons identifiés contribuent à générer un contenu de qualité.

Dans le troisième chapitre de cette thèse, nous nous sommes intéressés à identifier les communautés qui se construisent autour des experts. Nous avons démontré que les utilisateurs experts jouent un important rôle dans la création, la promotion et le maintien des communautés virtuelles dans les forums Internet de question-réponse. En effet, nous avons illustré que les interactions entre les experts et les autres participants conduisent à la formation des communautés dont lesquelles, l'apprentissage et le partage des connaissances sont la première priorité. Dans ce contexte, nous avons introduit le nouveau concept de "communautés de partage des connaissances". Ces communautés sont composées d'un ensemble d'experts et d'utilisateurs qui posent des questions de telle sorte que ces derniers montrent un comportement plus homogène, en terme d'interactions avec les experts, que

partout ailleurs.

Afin d'identifier les communautés de partage des connaissances, nous avons représenté notre environnement sous forme de données transactionnelles de telle sorte que chaque transaction résume les interactions d'un utilisateur spécifique avec tous les experts qui ont répondu à ces questions. Par la suite, nous avons développé un algorithme de *clustering* que nous avons nommé TRANCLUS et ce pour regrouper dans la même communauté/*cluster* les utilisateurs qui exhibent les mêmes comportements en terme d'interactions avec les experts. Contrairement à la vaste majorité des algorithmes transactionnels, TRANCLUS ne requiert aucun paramètre qui doit être fourni par l'utilisateur. La robustesse de TRANCLUS a été illustrée sur une variété de données synthétiques et réelles. Les résultats expérimentaux démontrent que TRANCLLUS est capable d'identifier correctement les *clusters* dans des situations complexes.

Nous avons appliqué notre approche sur des données extraites de six forums différents de Yahoo ! Answers. Les données que nous avons analysées représentent les activités des utilisateurs pendant une année complète. Un élément notable qui caractérise nos résultats expérimentaux, est que TRANCLUS identifie des communautés dans lesquelles, les utilisateurs qui posent souvent des questions gravitent autour d'un seul expert. Ces "denses" communautés que nous avons identifiées, encouragent la créativité, la collaboration, et facilitent ainsi le transfert des connaissances entre ses membres.

Bibliographie

[1] C.C. AGGARWAL, C. PROCOPIUC, J. L. WOLF, P.S. YU et J.S. PARK. « Fast Algorithm for Projected Clustering ». Dans ACM SIGMOD International Conference on Management of Data (SIGMOD'99), pages 61–72, 1999.

[2] C.C. AGGARWAL et P.S. YU. « Redefining Clustering for High Dimensional Applications ». IEEE Transactions on Knowledge and Data Engineering, 14(2) :210–225, 2002.

[3] R. AGRAWAL, J. GEHRKE, D. GUNOPULOS et P. RAGHAVAN. « Automatic Subspace Clustering of High Dimensional Data ». Data Mining and Knowledge Discovery, 11(1) :5–33, 2005.

[4] M. BOUGUESSA, B. DUMOULIN et S. WANG. « Identifying Authoritative Actors in Question-Answering Forums - The Case of Yahoo ! Answers ». Dans 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), pages 866–874, 2008.

[5] M. BOUGUESSA et S. WANG. « PCGEN : A Practical Approach to Projected Clustering and its Application to Gene Expression Data ». Dans IEEE International Symposium on Computational Intelligence and Data Mining (CIDM'07), pages 661–667, 2007.

[6] M. BOUGUESSA et S. WANG. « Mining Projected Clusters in High-Dimensional Spaces ». IEEE Transactions on Knowledge and Data Engineering, 21(4) :507–522, April 2009.

[7] M. BOUGUESSA, S. WANG et B. DUMOULIN. « Discovering Knowledge-Sharing Communities in Question-Answering Forums ». Submitted to ACM Transactions on

Knowledge Discovery from Data, Special Issue on Knowledge Discovery for Web Intelligence, September 2008.

[8] M. BOUGUESSA, S. WANG et Q. JIANG. « A K-means-based Algorithm for Projective Clustering ». Dans 18th IEEE International Conference on Pattern Recognition (ICPR'06), pages 888–891, 2006.

[9] C. S. CAMPBELL, P. P. MAGLIO, A. COZZI et B. DOM. « Expertise Identification using Email Communication ». Dans 12th ACM International Conference on Information and Knowledge Management (CIKM'03), pages 528–531, 2003.

[10] B. DOM, I. EIRON, A. COZZI et Y. ZHANG. « Graph-Based Ranking Algorithms for E-mail Expertise ». Dans 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03), pages 42–48, 2003.

[11] J. HAN et M. KAMBER. Data Mining : Concepts and Techniques, 2nd ed. Morgan Kaufmann, 2006.

[12] J. M. KLEINBERG. « Authoritative Sources in a Hyperlinked Environment ». Journal of the ACM, 46(5) :604–632, 1999.

[13] M. LUNG et N. MAMOULIS. « Iterative Projected Clustering by Subspace Mining ». IEEE Transactions on Knowledge and Data Engineering, 17(2) :176–189, 2005.

[14] M. T. MAYBURY. « Expert Finding Systems ». MITRE Technical Report, MTR 06B000040, 2006.

[15] L. PAGE, S. BRIN, R. MOTWANI et T. WINOGRAD. « The Pagerank Citation Ranking : Bringing Order to the Web ». Stanford Digital Library Technologies Project, 1998.

[16] C.M. PROCOPIUC, M. JONES, P.K. AGARWAL et T.M. MURALI. « Monte Carlo Algorithm for Fast Projective Clustering ». Dans ACM SIGMOD International Conference on Management of Data (SIGMOD'02), pages 418 – 427, 2002.

[17] P-N TAN, M. STEINBACH et V. KUMAR. Introduction to Data Mining. Addison Wesley, 2006.

[18] E. WENGER, R. A. MCDERMOTT et W. SNYDER. Cultivating Communities of Practice : A Guide to Managing Knowledge. Harvard Business School Press, 2002.

[19] D. YIMAM et A. KOBSA. « Expert Finding Systems for Organisations : Problem and Domain Analysis and the DEMOIR Approach ». Journal of Organizational Computing and Electronic Commerce, 13(1) :1–24, 2003.

[20] K.Y.L. YIP, D.W. CHENG et M.K. NG. « HARP : A Practical Projected Clustering Algorithm ». IEEE Transactions on Knowledge and Data Engineering, 16(11) :1387–1397, 2004.

[21] K.Y.L. YIP, D.W. CHENG et M.K. NG. « On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering ». Dans IEEE International Conference on Data Engineering (ICDE'05), pages 329–340, 2005.

[22] K.Y.L. YIP, D. W. CHEUNG, M. K. NG et K. CHEUNG. « Identifying Projected Clusters from Gene Expression Profiles ». Journal of Biomedical Informatics, 37(5) :345–357, 2004.

[23] J. ZHANG, M.S. ACKERMAN et L. ADAMIC. « Expertise Networks in Online Communities : Structure and Algorithms ». Dans 16th ACM International Conference on World Wide Web (WWW'07), pages 221–230, 2007.