Interactive User Group Analysis

Technical Report

Behrooz Omidvar-Tehrani
Univ. Grenoble Alps/CNRS
Grenoble, France
behrooz.omidvar-tehrani@imag.fr

Sihem Amer-Yahia
Univ. Grenoble Alps/CNRS
Grenoble, France
sihem.amer-yahia@imag.fr

Alexandre Termier
University of Rennes 1,
IRISA/INRIA
Rennes, France
alexandre.termier@imag.fr

ABSTRACT

User data is becoming increasingly available in multiple domains ranging from phone usage traces to data on the social Web. The analysis of user data is appealing to scientists who work on population studies, recommendations, and large-scale data analytics. We argue for the need for an interactive analysis to understand the multiple facets of user data and address different analytics scenarios. Since user data is often sparse and noisy, we propose to produce *labeled groups* that describe users with common properties and develop IUGA, an interactive framework based on *group discovery* primitives to explore the user space. At each step of IUGA, an analyst visualizes group members and may take an action on the group (add/remove members) and choose an operation (exploit/explore) to discover more groups and hence more users. Each discovery operation results in *k most relevant and diverse groups.* We formulate group exploitation and exploration as optimization problems and devise greedy algorithms to enable efficient group discovery. Finally, we design a principled validation methodology and run extensive experiments that validate the effectiveness of IUGA on large datasets for different user space analysis scenarios.

Categories and Subject Descriptors: H.2.8 [**Database management**]: Database Application; Data Mining.
General Terms: Algorithms.
Keywords: User Data; Interactive Analysis; Validation.

1. INTRODUCTION

One of the major applications of data-driven research is the analysis of *user data*. User data is the conjunction of a profile made of several attributes (e.g., age, occupation, location), and of user interests via activity data (e.g., application usage on smartphones, researchers' publications, movie ratings, exercise and eating habits). Building *labeled user groups* instead of focusing on individual data enables new findings and addresses issues raised by the peculiarities of user data such as noise and sparsity.[1] The discovery of user groups bears similarities to the *redescription mining* problem [20]: one wants to find groups of users that have identical values for some profile attributes (e.g. *gender* = `female` $\wedge$ *country* = `US`) and that exhibit common activity data (e.g. *keywords* = {`databases, social networks`} and *publish_in* = {`WWW, VLDB`}). The large number of possible groups hinders user data analysis. In this paper, we advocate an exploratory navigation of the space of groups and develop IUGA, an interactive user data analysis framework that provides analysts with the ability to incrementally discover user groups efficiently.

There exist numerous approaches that could be used for the discovery of user groups, many of which are based on *pattern mining* such as frequent itemset discovery [2] or subspace clustering [1]. The main limitation of those approaches is that, on real data, they can output millions of labeled groups inside which it is hard to know a priori which ones are of interest to the analyst. To alleviate that, a large body of work has been dedicated to providing knowledgeable analysts with the ability to specify *constraints* on groups of interest [6, 7]. However, that is not adapted to exploratory scenarios where only limited knowledge is available on the dataset and the analyst does not necessarily know which subset of the data is of interest. The same argument applies to expressing queries on raw data, e.g. using SQL [10]. SQL being declarative in nature, it is difficult to use it to express an exploration scenario which is iterative in nature: e.g., finding a set of groups with a SQL query then asking to find "related" groups. Other work proposed to reduce the output of a pattern mining algorithm to a *representative pattern set* of limited size (typically tens of labeled groups) [22]. However, resulting groups may be too coarse and miss groups that contain users of interest. Group granularity may be reduced with parameter relaxation but the number of resulting groups is bound to increase and quickly become hard to manage by the analyst.

When faced with the daunting task of analyzing large amounts of user data, an analyst may have different goals in mind. In this paper, we focus on helping analysts find one or several users of interest by exploring relevant groups until she reaches her target users. More specifically, an analyst may want to *discover and gather several users* who may be scattered in different groups of interest. An analyst may also be interested in *finding a specific group member*, i.e., a user, for whom she remembers some but not all information. We illustrate these variants in the following two realistic examples.

EXAMPLE 1 (A MULTI-TARGET EXAMPLE). *Martin is a PC Chair looking to build a program committee formed by geographically distributed male and female researchers with different seniority levels and different expertise. Figure 1 shows a simplified scenario for the* WEBDB *2014*[2] *PC.*

As is often the case, PC chairs think of a set of potential members first. In this case, G. Fletcher, M. Theobald, S. Michel and X.

[1] *http://www.wired.com/2014/04/forget-the-quantified-self-we-need-to-build-the-quantified-us/; https://jawbone.com/blog/napa-earthquake-effect-on-sleep/*

[2] *http://webdb2014.eecs.umich.edu*

CIKM'15 Melbourne, Australia

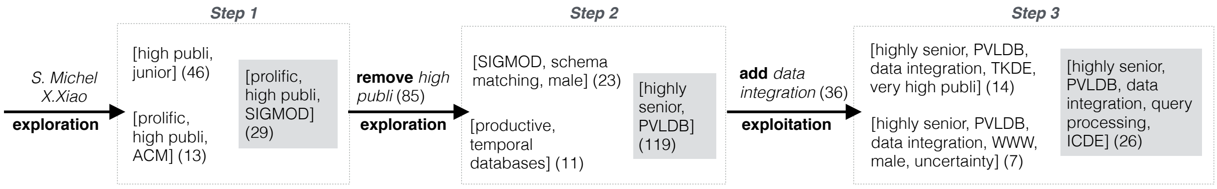
Step 1
Step 2
Step 3
S. Michel
X.Xiao
exploration
[high publi, junior] (46)
[prolific, high publi, ACM] (13)
[prolific, high publi, SIGMOD] (29)
remove high publi (85)
exploration
[SIGMOD, schema matching, male] (23)
[productive, temporal databases] (11)
[highly senior, PVLDB] (119)
add data integration (36)
exploitation
[highly senior, PVLDB, data integration, TKDE, very high publi] (14)
[highly senior, PVLDB, data integration, WWW, male, uncertainty] (7)
[highly senior, PVLDB, data integration, query processing, ICDE] (26)

Figure 1: Discovering Several Users (WEBDB 2014 Program Committee)

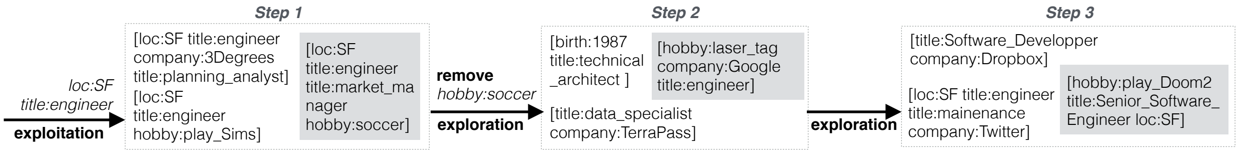
Step 1
Step 2
Step 3
loc:SF
title:engineer
exploitation
[loc:SF title:engineer company:3Degrees title:planning_analyst]
[loc:SF title:engineer hobby:play_Sims]
[loc:SF title:engineer title:market_manager hobby:soccer]
remove hobby:soccer
exploration
[birth:1987 title:technical_architect]
[title:data_specialist company:TerraPass]
[hobby:laser_tag company:Google title:engineer]
exploration
[title:Software_Developper company:Dropbox]
[loc:SF title:engineer title:mainenance company:Twitter]
[hobby:play_Doom2 title:Senior_Software_Engineer loc:SF]

Figure 2: Finding a Specific User

Xiao are 4 initial members. Martin decides to use S. Michel and X. Xiao as seeds because they are junior and prolific (high frequency of annual publications). The action of keeping those 2 researchers is followed by an exploration which delivers 3 groups each of which containing one of S. Michel or X. Xiao (Step1). He then decides to keep the highlighted one: `prolific`, `high publications` *and publishing at* `SIGMOD` *(with which* WEBDB *is associated.) The selected group contains 29 researchers out of which 4 geographically distributed (L. Popa, A. Doan, M. Benedikt, S. Amer-Yahia). In order to find more users related to that group, Martin decides to perform an action that removes the predicate* `high publi`*, because it has been investigated before.*

Step 2 is explores the resulting group and outputs 3 diverse groups. Martin ignores the first group because he has already seen enough male candidates. He notices that the highlighted group contains 119 highly senior researchers who published in PVLDB (which is related to WEBDB*). Therefore, he decides to get more information about that group by first adding the predicate* `data integration` *to specialize it and asking to split it into 3 groups. Step 3 shows the result of this exploitation operation. In particular, the group labeled with* `query processing`, `PVLDB` *and* `ICDE` *contains 26 senior researchers out of which 8 are of interest to the PC chairs (J. Wang, F. Bonchi, K. Chakrabarti, P. Fratenali, D. Barbosa, F. Naumann, Y. Velegrakis and X. Zhou). At this stage, Martin covered 80% of the* WEBDB *PC.* □

EXAMPLE 2 (A SINGLE TARGET EXAMPLE). *Nicole liked a person she met at last night's party in Cole Valley, San Francisco, but she doesn't remember his name and has lost his phone number. She only knows that he lived in the same neighborhood as Mike, the party host, and works as an engineer. Nicole asks Mike to have access to his social network which contains some of his friends' information: job title(s), company, location, birth year, and hobby(ies).*

Mike is an avid Facebook user and has over 800 friends, most of which are computer engineers and live in San Francisco (SF). Thus no querying mechanism could lead Nicole directly to the person she is looking for. Also, advanced search tools (e.g., Facebook Graph Search[3]) can only show similar people based on an input query. Nicole needs a tool for her navigational analysis of Mike's friends. She first uses the query `loc:SF` *and* `title:engineer` *which returns 3 different user groups among Mike's friends that are highly related to her query (Step 1 in Figure 2). Nicole remembers that the person was talking about "website design", thus he shouldn't be working for* `3Degrees` *which is a renewable energy certificate provider. She also remembers that he mentioned he only likes "shooting" computer games thus he should not belong to the group labeled with* `Sims`*, a life simulation computer game. So she prefers to select the group labeled* `loc:SF`, `title: engineer`, `title:market manager` *and* `hobby: soccer` *as a seed. As Nicole doesn't remember any discussion about sports, she prefers to remove the attribute* `hobby:soccer` *to widen her navigation scope. The tool then finds 3 other groups in Step 2.*

Nicole is sure the person she met is at least 30 years old. This eliminates the first group with `birth:1987`*. Also, she herself works for* `TerraPass` *and knows the person does not work there. Also being a fan of shooting games, the group with* `hobby:laser tag` *and* `company:Google` *would be the best choice. Now, the tool returns 3 other user groups in step 3. The title "Senior Software Engineer" captures her attention as she remembers he said he was a manager. This group contains 3 users among whom Kevin Systrom, co-founder of Instagram and Mike's friend, is the one she was searching for.* □

Acknowledging the limitations of previous solutions, a recent line of work based on *interactive data mining* is being developed.[4] Such work is based on providing operations on the result of pattern mining in order to help the analyst navigate in the space of labeled groups and find groups and group members she is most interested in [5]. Existing work in this area is mainly a description of systems that have been designed to show the potential of interactive approaches. In this paper, we propose two important contributions to further advance the state of the art of interactive data mining with a focus on interactive user data analysis:

1. IUGA, a formalization of interactive user data analysis based on simple yet powerful *group discovery* primitives that enable an exploratory navigation of user groups.
2. A principled validation methodology. To the best of our knowledge, there exists no such methodology.

IUGA is *an optimization-based interactive framework where analysts are free to select any group of interest at each step and use it as a seed for further optimization.* It is based on 3 key principles:

[3] *https://www.facebook.com/graphsearcher*

[4] *http://poloclub.gatech.edu/idea2013/; http://poloclub.gatech.edu/idea2014/*

P1: The analyst must be able to explore different groups and not be overwhelmed with analysis options. The analysis process is broken into successive steps during which an analyst chooses a seed group, examines the users it contains, takes actions such as remove/add users, and continues with a group discovery operation.

P2: Groups offered to the analyst must be of high quality. The analysis process must help the analyst cover the space of groups of interest. We propose a "holistic" measure that finds k groups that are relevant to the seed group and are as diverse as possible.

P3: The train of thought of the analyst must not be lost. Each interactive group discovery step must be fast.

Our examples show that with simple group discovery operations, an analyst can navigate a good proportion of the space of users of interest. In this work, we formalize two such operations: *opExplore()* that finds groups *outside of the seed group*, and *opExploit()* that finds groups *inside.* The examples also show that before applying group discovery operations to a seed group, an analyst may want to transform that group using *actions* that remove or add specific users (in our example by modifying group labels).

Devising an efficient multi-step group discovery approach is a challenge due to the large number of available groups. We hence propose to formulate group exploration and exploitation as optimization problems that find relevant and diverse groups at each step of the interaction. Both operations discover k diverse groups that have some relevance to the seed group, i.e., users in common. In the case of exploration, diversity aims to cover *as many different users as possible* outside of the seed group. For exploitation, diversity aims to *cover the seed group* while providing *distinct options* inside that group. We show that both problems are NP-complete by reductions from the MAXIMUM EDGE SUBGRAPH PROBLEM and the MAXIMUM COVERAGE PROBLEM respectively. We design GROUPDISCOVERY, a greedy algorithm to solve those problems.

Our last challenge is to devise a principled methodology that validates the need for an interactive multi-step group discovery approach. In particular, since our focus is to solve the multi-target and single-target search questions, we validated IUGA on two real use cases, namely, Program Committee (PC) formation by building a dataset from DBLP, and a single target scenario by building a synthetic dataset. Our results show that IUGA leads analysts to their target(s) in a small number of steps regardless of their starting points and their level of expertise.

The outline of the paper is as follows: in Section 2 we give necessary definitions and formalize the GROUPDISCOVERY operations as well as the Interactive User Group Analysis problem. In Section 3 we describe the IUGA algorithm. Section 4 presents detailed experiments. The related work is provided in Section 5. Last, we conclude and give some perspectives for future work in Section 6.

2. MODEL AND PROBLEM DEFINITION

We model user data as a set of users $\mathcal{U}$, a set of items I, and a database D of tuples $\langle u, i\rangle$ where $u \in \mathcal{U}$ and $i \in I$. A tuple $\langle u, i\rangle$ represents the action of u (such as *authored*, *recorded*, *rated*, *purchased*, *tagged*, *voted*, etc.) on i.

Each user u is also described with attributes drawn from a set of attributes A representing demographics information such as *gender* and *age*. We refer to each attribute in A as a_i and to its values as v_j^i. The domain of values of attribute a_i is D_{a_i} with $D_A = \cup D_{a_i}$. For example, if we use a_1 to refer to *gender*, it takes two values v_1^1 and v_2^1 representing *male* and *female* respectively.

A large number of datasets could be modeled in this manner. In the case of scientists, items are conferences they publish in and keywords they contribute on. For online collaborative rating sites, items are movies or restaurants they rate. The choice of what constitutes a user attribute or a user action depends on the application and does not affect our problem and solution.

Due to the sparsity of user data, we propose to analyze it based on forming user groups and providing a framework to the analyst to discover groups in a step-by-step fashion. We first define the notion of user group followed by the GROUPDISCOVERY Problem and finally, IUGA, our interactive user data analysis framework.

DEFINITION 1. **User Group**. *A user group g, is a subset of $\mathcal{U}$ to which is associated a label $l_g = [P_g, I_g]$ where P_g is a conjunction of predicates on user attribute $(a_1 = v_j^1) \wedge \ldots \wedge (a_n = v_j^n)$, and $I_g \subseteq I$ a set of items. Each user in g must satisfy P_g and have an action on each item in I_g. More formally, $\forall u \in g, P_g$ is true and $\forall u \in g, \forall i \in I_g, \langle u, i\rangle \in D$.*

In the case of a movie rating site, the group of reviewers defined by the label [(*age*=`25`) ∧ (*occupation*=`student`), {`Terminator 2`, `The Matrix`}] is the set of 25-year old students who rated both movies. Similarly, [(*seniority*=`junior`) ∧ (*gender*=`female`) ∧ (*pub rate*=`productive`), {`Algorithms`, `Data Mining`}] is the set of junior young female researchers who are productive and have expertise in algorithms and data mining.

To simplify readability, we will use a simplified representation for labels in the remainder of the paper. For example, [(*age*=`25`) ∧ (*occupation*=`student`), {`Terminator 2`, `The Matrix`}] becomes [`25`, `student`, `Terminator 2`, `The Matrix`].

We use $\mathcal{G}$ to refer to the set of all user groups formed by predicates on A and items in I. $\mathcal{G}$ is very large even with a small number of attribute values and items.

2.1 Group Discovery Operations

We now introduce our formalization of group discovery operations and actions that form building blocks of IUGA. We first define the *exploration* operation that allows to "navigate the group space in an outward way" starting from a set of users, it discovers groups containing new users.

DEFINITION 2. **Group Exploration**. *We define a function $gExplore(U, \mathcal{G}, \mu)$ that takes a set of users $U \subseteq \mathcal{U}$ and finds all groups in $\mathcal{G}$ that overlap with U with at least μ, a given threshold. More formally, $gExplore(U, \mathcal{G}, \mu) = \{(g, overlap(U, g)) | g \in \mathcal{G} \wedge g \neq U \wedge overlap(U, g) \geq \mu\}$ where $overlap(U, g) = \frac{|U \cap g|}{|U \cup g|}$ (i.e. Jaccard similarity coefficient).*

The overlap condition provides a progressive exploration of the space, which helps the analyst build an understanding of the underlying data. Figure 1 illustrates several steps in IUGA used to build the WEBDB 2014 PC. At the beginning, the analyst (here a PC chair) has two "seed members" in mind: S. Michel and X. Xiao. She then performs an exploration step over these two researchers, which produces (among other) three groups: a group of 46 researchers labeled as [`junior`, `high publi`], a group of 13 researchers labeled as [`prolific`, `high publi`, `ACM`], and a group of 29 researchers labeled as [`prolific`, `high publi`, `SIGMOD`]. All these groups lead to different research communities, the third one is most adapted to a database workshop.

When an interesting group is found, another important operation is *exploitation*, i.e., an operation that "drills down" into the most interesting subgroups contained in a seed group.

DEFINITION 3. **Group Exploitation**. *We define a function $gExploit(U, \mathcal{G})$ that takes a set of users $U \subseteq \mathcal{U}$ and finds all groups* *in $\mathcal{G}$ that are contained in U. More formally, $gExploit(U, \mathcal{G}) = \{g \in \mathcal{G} | g \subseteq U\}$.*

Figure 1 shows the result of applying *gExploit()* on the group labeled [`highly senior`, `PVLDB`, `data integration`] (Step 3). That results in 3 subgroups: one formed by 26 experts in query processing who publish in ICDE, the other by 14 prolific researchers who publish in TKDE, and the last one by 7 male researchers who work on uncertainty in databases. All 3 groups contain solely highly senior researchers who publish in PVLDB and work in the area of data integration. This example clearly illustrates that 32 out of 36 users of the selected group are covered.

2.2 Group Discovery Problem

The discovery of new groups relies on the two functions, *gExplore()* and *gExploit()* (Definitions 2 and 3 respectively), that are applied to a seed set of users. In order to comply with principles **P1** and **P2**, the number of groups returned to the analyst at each step must be limited, and output groups must exhibit diversity. Hence, we define the GROUPDISCOVERY Problem as follows: given a set of users $U \subseteq \mathcal{U}$, an overlap threshold μ, the GROUPDISCOVERY Problem returns k groups in $\mathcal{G}$, referred to as $\mathcal{G}_U$ and is expressed either as an exploration or an exploitation problem depending on an analyst's needs.

For exploration, we define $opExplore(U, \mathcal{G}, \mu, k)$ that must satisfy the following conditions:

1. $\mathcal{G}_U \subseteq gExplore(U, \mathcal{G}, \mu)$
2. $|\mathcal{G}_U| = k$
3. $diversity(\mathcal{G}_U)$ is maximized.

where $diversity(\mathcal{G}_U)$ is defined as follows:
$diversity(\mathcal{G}_U) = \Sigma_{\{g_i,g_j\} \subseteq \mathcal{G}_U | g_i \neq g_j}(1 - overlap(g_i, g_j))$

In exploration, the aim is to start from a seed set of users of interest U, and find k groups that have some relevance to U, using $gExplore(U, \mathcal{G}, \mu)$, and that have maximal diversity (as little overlap as possible with each other).

For exploitation, we define $opExploit(U, \mathcal{G}, k)$ that must satisfy the following conditions:

1. $\mathcal{G}_U \subseteq gExploit(U, \mathcal{G})$
2. $|\mathcal{G}_U| = k$
3. $divCoverage(\mathcal{G}_U)$ is maximized.

where $divCoverage(\mathcal{G}_U)$ is defined as follows:

$$divCoverage(\mathcal{G}_U) = diversity(\mathcal{G}_U) \times (|\bigcup_{g \in \mathcal{G}_U} g|/|U|) \quad (1)$$

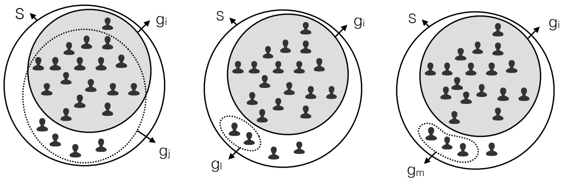

Figure 3: Illustrations of $diversity()$ and $divCoverage()$

In exploitation, the aim is to find k groups that maximize coverage of the seed set U. Choosing k groups that have the highest coverage may potentially cause high overlap between those groups. Figure 3 left illustrates that, with $k = 2$ and two highly overlapping groups g_i and g_j. Therefore, in the case of exploitation, we revisit the definition of diversity in a way that it prioritizes k diverse groups which cover as many users as possible in U. In [16], it is shown that there does not exist a unique optimal solution for both diversity maximization and coverage maximization. Therefore, the diversity formula is modified by adding $(|\bigcup_{g \in G_U} g|/|U|)$ (see Equation 1). For example, in Figure 3, $diversity(\{g_i, g_l\}) = diversity(\{g_i, g_m\}) = 1$. Thus for *opExplore()*, both g_l and g_m can be chosen with g_i. However, for *opExploit()*, g_m is preferred because $divCoverage(\{g_i, g_m\}) > divCoverage(\{g_i, g_l\})$.

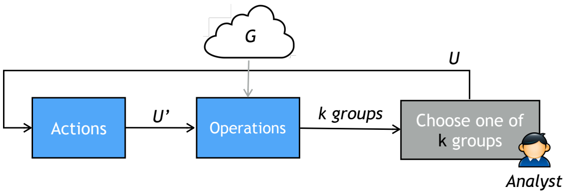

Figure 4: GROUPDISCOVERY within IUGA

2.3 Interactive User Group Analysis (IUGA)

IUGA builds on the GROUPDISCOVERY operations letting an analyst apply one of *opExplore()* or *opExploit()* on a set of users U and obtain k groups that constitute further analysis options. Figure 4 illustrates that process. In order to comply with principle **P3**, IUGA introduces a time limit parameter. Each step of IUGA solves the GROUPDISCOVERY Problem and returns the best possible k groups within a given time limit.

In addition to *opExplore()* and *opExploit()*, the analyst is provided with a set of actions that could be performed on a chosen group to transform it according to his/her needs. The analyst examines the set of k groups at each step and chooses a new seed group on which one of 3 actions could be performed: *actKeepUsers(U,U')*, *actModifyLabel(U,l)*, and *actUndo()* to undo the previous step. Table 1 summarizes each action. *actKeepUsers(U,U')* allows the analyst to mark which users to keep for the next step. *actModifyLabel(U,l)* is used to remove/add predicates or items to l_U, the label of U, resulting in new seed users.

Action	Description
actKeepUsers(U,U')	keeps U' users in U
actModifyLabel(U,l)	replaces l_U with a new label l
actUndo()	back-tracks to the previous step

Table 1: IUGA Actions

The ability to "manipulate group membership" using actions on a seed group, provides additional flexibility at each step. For example, as illustrated in Figure 1, in order to narrow down the set of 119 senior researchers who publish in PVLDB, the analyst adds the predicate `data integration` and obtains 36 researchers as the new seed set to analyze. Also in Figure 2, the analyst removes the predicate `hobby:soccer` to direct the navigation towards her preferences.

3. ALGORITHMS

Our GROUPDISCOVERY Problem requires to develop an efficient algorithm for dynamically finding and comparing user groups.

We first discuss the complexity of our problem, then we describe our algorithm.

3.1 Problem Complexity

We show that the GROUPDISCOVERY Problem is NP-complete by reductions from the MAXIMUM EDGE SUBGRAPH Problem for *opExplore()* and from the MAXIMUM COVERAGE Problem for *opExploit()*. We consider an infinite time limit in our proofs since that does not affect the complexity of our problem.

THEOREM 1. *The exploration version of the* GROUPDISCOVERY *Problem is NP-complete.*

PROOF. The decision version of the problem is as follows: For a given group g, a set of groups $\mathcal{G}$ and a positive integer k, an overlap threshold μ, is there a subset of groups $\mathcal{G}' \subseteq gExplore(g, \mathcal{G}, \mu)$ such that (i) $g' \in \mathcal{G}' \wedge g' \neq g \wedge overlap(g, g') \geq \mu$ and (ii) $\Sigma_{(g_1,g_2)\in\mathcal{G}'|g_1\neq g_2}(1 - overlap(g_1, g_2))$ is maximized. A verifier v which returns true if both conditions (i) and (ii) are satisfied runs in polynomial time in the length of its input.

To verify NP-completeness, we reduce the MAXIMUM EDGE SUBGRAPH (MES) [12] (also known as *Dense k-subgraph*) to the decision version of our problem. The problem of MES is defined as follows. Given an instance I consisting of a graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbb{N}$, and a positive integer k, find a subset $V' \subseteq V$, $|V'| = k$ such that the total weight of the edges induced by V', i.e., $\Sigma_{(v_i,v_j)}w(v_i, v_j)$ (where $(v_i, v_j) \in V' \times V'$) is maximized. This is an NP-complete problem [12] (originally reduced from the *Clique* problem).

Given I, we create an instance J of our problem as follows. J consists of a graph $G = (V, E)$ where the set of vertices $V = gExplore(g, \mathcal{G}, \mu)$ are groups that satisfy (i). Every pair of groups $(g_1, g_2) \in V \times V$ is also connected with a labeled edge $i.e.$ $w(g_1, g_2) = 1 - overlap(g_1, g_2)$. The subset $V' \subseteq V$ ($|V'| = k$) is then a subset of groups where the sum of the weights between each pair of groups in V' is maximized $i.e.$, $|E(V')| = \frac{k\times(k-1)}{2}$. The set V' is the most diverse subset of $\mathcal{G}$ that satisfies the overlap condition ($\forall g' \in \mathcal{G}, overlap(g, g') \geq \mu$). Therefore a set V' is a solution in instance I of MES *iff* it is a solution in instance J of our problem. Hence, the exploration problem is then NP-complete. □

THEOREM 2. *The exploitation version of the* GROUPDISCOVERY *Problem is NP-complete.*

PROOF. Similarly to the exploration version, a verifier v for exploitation runs in polynomial time in the length of its input. To verify NP-completeness, we reduce the MAXIMUM COVERAGE PROBLEM [17] to the decision version of our problem. The problem of MAXIMUM COVERAGE PROBLEM (MCP) is defined as follows. Given an instance I consisting of m sets $S = \{\ S_1 \ldots S_m\}$ where $S_i \in S_M$ (S_M being a reference set), and a positive integer k, find a subset $S' \subseteq S$, such that $|S'| = k$ and the number of covered elements in S_M, i.e., $|\cup_{S_i \in S'} S_i|/|S_M|$ is maximized. This is an NP-complete problem [17]. Given I, we can create an instance J of our problem which consists of m sets $S = gExploit(g, \mathcal{G}, \mu)$ and a reference group, i.e., $S_M = g_{in}$. In *opExploit()*, we are interested to have k groups $S' \subseteq S$ that cover maximum number of users in S_M, i.e., $|\cup_{S_i \in S'} S_i|/|S_M|$ is maximized. Therefore a set S' is a solution in instance I of MCP *iff* it is a solution in instance J of *opExploit()*. The exploitation version of the GROUPDISCOVERY Problem is then NP-complete. □

3.2 Group Discovery Algorithm

Our overall approach operates in two steps: an off-line process to produce initial user groups $\mathcal{G}$ and an online iterative process during which the analyst chooses a selected group for which k groups are discovered.

In the off-line process, a set of groups $\mathcal{G}$ are generated using the LCM closed frequent pattern mining algorithm [23] given a minimum support σ. Each frequent pattern corresponds to a user group, which has at least σ users. To feed LCM, we convert predicates in group labels into an item. For instance, the predicates (*gender* = `male`) and (*gender* = `female`) become two independent items. In addition, in order to speedup computing group relevance, we pre-compute an inverted index for each user group $g \in \mathcal{G}$ (as is commonly done in Web search). Each index $\mathcal{L}^g$ stores all other groups in $\mathcal{G}$ in decreasing order of their overlap with g. Thanks to the parameter μ, we only partially materialize the indices. In the case of datasets we used in our experiment, we only materialize in average 10% of the whole index size.

Algorithm 1 summarizes a single greedy procedure that solves the GROUPDISCOVERY Problem, be it exploration or exploitation. It is called at each step of IUGA (as described in Figure 4). The algorithm admits as input a user group g, an operation *op* (*gExplore()* or *gExploit()*), an overlap threshold μ, k, and a time limit $tlimit$, and returns the best k groups denoted $\mathcal{G}_g$. Line 1 selects the most overlapping groups with g by simply retrieving the k highest ranking groups in $\mathcal{L}^g$. Function $getNext(\mathcal{L}^g)$ (Line 2) returns the next group g_{in} in $\mathcal{L}^g$ in sequential order. Lines 3 to 11 iterate over the inverted indices to determine if other groups should be considered to increase diversity while staying within the time limit and not violating the overlap threshold with the selected group. Since groups in $\mathcal{L}^g$ are sorted on decreasing overlap with g, the algorithm can safely stop as soon as the overlap condition is violated (or if the time limit is exceeded).

Algorithm 1: GROUPDISCOVERY Algorithm

```
   Input: $g \in \mathcal{G}$, op, $\mu$, $k$, $tlimit$
   Output: $\mathcal{G}_g$
1  $\mathcal{G}_g \leftarrow$ top$k(\mathcal{L}^g)$
2  $g_{out} \leftarrow getNext(\mathcal{L}^g)$
3  while ($tlimit\ not\ exceeded \wedge overlap(g, g_{out}) \geq \mu$) do
4      for $g_{in} \in \mathcal{G}_g$ do
5          if $betterDiv(\mathcal{G}_g, g_{out}, g_{in}, op)$ then
6              $\mathcal{G}_g \leftarrow replace(\mathcal{G}_g, g_{out}, g_{in})$
7              break
8          end
9      end
10     $g_{out} \leftarrow getNext(\mathcal{L}^g)$
11 end
12 return $\mathcal{G}_g$
```

The algorithm then looks for a candidate group $g_{out} \in \mathcal{G}_g$ to replace in order to increase diversity. The boolean function $betterDiv()$ (Line 5) checks if by replacing g_{out} by g_{in} in $\mathcal{G}_U$, the overall diversity of the new $\mathcal{G}_U$ increases. Obviously, the diversity of a group set $\mathcal{G}_k$ depends on the operation *op*.

The number of diversity improvement loops (lines 3 to 11) is $|\mathcal{L}^g|$ in worst case. For each group $g_{in} \in \mathcal{G}_g$, we verify if the diversity score is improved by $betterDiv()$, hence $\mathcal{O}(k^2)$. The time complexity of the algorithm is then $\mathcal{O}(k^2.max_{g\in\mathcal{G}}|\mathcal{L}^g|)$.

4. EXPERIMENTS

Our experiments aim to validate the usability and efficiency of interactive analysis and the quality of discovered groups at each step. All experiments are implemented in C on a 2.4GHz Intel Core

Attribute	Description & Values
Seniority	Number of years since the author's first publication in DBLP with values "starting" (1 to 8), "junior" (9 to 12), "senior" (13 to 15), "highly senior" (16 to 21) and "confirmed" (22 and higher)
# publications	With values "very few" (3 to 14), "few" (15 to 28), "fair" (29 to 53), "high" (54 to 107) and "very high" (108 and higher)
Publication rate	Average number of publications per year with values "active" (0.18 to 1.47), "very active" (1.48 to 2.48), "productive" (2.49 to 3.71), "very productive" (3.72 to 6.0) and "prolific" (6.1 and higher)
Venues	set of all conferences and journals where the author published in, at least once
Topics	Set of topics extracted from the author's publications using the LDA topic model [4]
Gender	Based on matching the author's first name to an NLP resource with 54,915 names.[7]

Table 2: Researcher Attributes

i5 machine with an 8GB main memory, running OS X 10.9.2.

Summary of Results: In our first experiments, we observe that IUGA leads a knowledgable analyst to cover most PCs of major data management conferences in 12 steps (multi-target scenario). We also show that IUGA arrives sooner to target than its competitors (single-target scenario). Our second experiment is a user study of the quality of groups found by GROUPDISCOVERY in each step of IUGA. We find that most participants prefer IUGA to other options mainly because it helps them better understand the landscape of user groups.

4.1 Datasets

We use 2 real datasets for our experiments: DM-AUTHORS and MOVIELENS and one synthetic dataset with the same characteristics as MOVIELENS. DM-AUTHORS and the synthetic dataset are used to validate the interactive analysis and MOVIELENS is used to validate group quality.

The MOVIELENS 1M dataset[5] contains 1,000,209 ratings of 3,952 movies by 6,040 users. For each user, gender, age-group, occupation and zip-code are also provided. DM-AUTHORS contains 4907 researchers who have at least 3 publications in WWW, KDD, SIGMOD, CIKM, ICWSM, EDBT, ICDM, ICDE, RecSys, SIGIR or VLDB. Authors were crawled in October 2014 from DBLP[6] for years between 2000 and 2014. For each researcher, we compute the attributes summarized in Table 2. Values of the first 3 attributes are discretized using equal-frequency binning [13].

Table 3 summarizes the real datasets. It contains the number of user groups ($|\mathcal{G}|$) with at least σ users. For DM-AUTHORS, σ is set to a very low value because smaller groups are of interest (e.g., 976 researchers are associated to `high publi` predicate, but only 28 researchers have published both in `WWW` and in `CIKM`). For MOVIELENS, we set σ to 7% of all users in order to obtain an adequate number of groups for our group quality validation. In both datasets, we set μ in a way that each group overlaps with 10% of groups in $\mathcal{G}$, hence pruning around 90% of inverted indices.

	MOVIELENS	DM-AUTHORS
# users	6,040	4,907
# items	3,952	11,890
# attributes	4	4
# predicates	80	18
avg index size $|\mathcal{L}|$	485	79,127
# groups $|\mathcal{G}|$	4,918	790,017
σ	450	3
μ	0.1	0.01

Table 3: Real Datasets

Our synthetic dataset is generated by initializing a binary matrix of users and items M with *0* and then randomly adding some initial groups $\mathcal{G}_{initial}$, i.e., rectangles in M that are completely filled with *1*. For each group in $\mathcal{G}_{initial}$, we randomize its number of users (between 10 and 2000) and items (between 5 and 50 items). Then we mark $|\mathcal{G}_{target}|$ groups as target. We mine M with a minimum support threshold σ to obtain the group set $\mathcal{G}$. More details are presented in Table 4. All parameters are chosen in a way to mimic MOVIELENS with a larger number of users.

# users	10,000	$maxlenght$	50
# items	3000	$tlimit$	20 ms
$|\mathcal{G}_{target}|$	50	σ	10
$|\mathcal{G}_{initial}|$	500	μ	0.06

Table 4: Synthetic Dataset

4.2 Interactive Analysis Validation

We validate the effectiveness of our interactive analysis IUGA by addressing the two motivating scenarios described in Section 1. We first verify the utility of IUGA for building a program committee on DM-AUTHORS (multi-target). Then, we describe a thorough validation of IUGA using our synthetic dataset that mimics MOVIELENS (single-target). In both cases, $tlimit$ is set to $20ms$.

4.2.1 Multi-Target Scenario

We study the effectiveness of IUGA with a realistic task of interactively building the PC of major conferences/workshops in data management. We first start with an experiment with many PCs then we delve into the details of WEBDB 2014.

Summary of Many PCs.

Figure 6 illustrates the results of interactively building the PCs of the following conferences in 2014: SIGMOD, VLDB[8], WebDB and CIKM[9]. For a given PC, we start from 5% of its members and use IUGA to find the remaining ones. Target PC members should be discovered in user groups proposed in different steps of IUGA. The figure reports the number of steps to discover 50% and 80% of PC members as the average of 50 runs of IUGA for each PC.

We can observe that any PC selection can be done in 12.04 steps on average. CIKM's PC is the hardest to discover and WebDB's the easiest. Our conjecture is two key factors influence that: PC *size* and PC *diversity*. Indeed, the PCs of VLDB, CIKM and SIGMOD contain over 100 members while WebDB is smaller. This is why

[5] *http://www.grouplens.org/node/73*

[6] *http://dblp.uni-trier.de/db/*

[8] *We only considered* Review Board *members for VLDB*

[9] *We only considered the* knowledge management *track for CIKM*

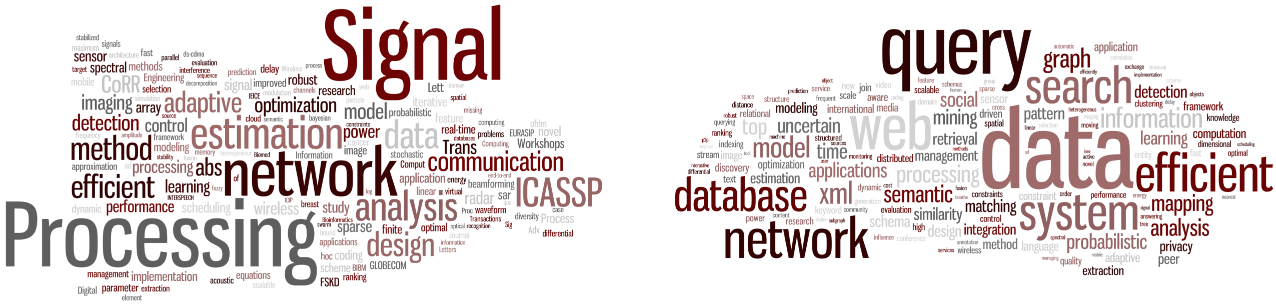

Figure 5: Word Frequency Cloud for Jian Li (left) and the Whole WEBDB 2014 PC (right)

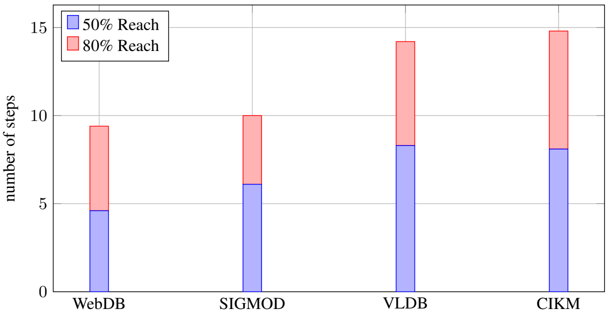

Figure 6: Number of Steps in IUGA for PC Selection

the former require a higher number of steps to cover 50% of their members (6.7, 6.5 and 5.9 steps respectively). In addition, the average pairwise Jaccard similarity[10] between PC members of CIKM is 7.35 . This high diversity results in more steps to reach 80% of their PCs (8.3 and 8.1 steps respectively). SIGMOD has the least heterogeneous PC which leads to 4.8 steps to reach 80% of its PC.

Focus on one PC.

We now have a closer look to the WEBDB 2014 PC selection. Our detailed illustration will show the following facts: **F1**: how the analysis of user groups is more useful than analyzing individual users; **F2**: how our actions and operations (defined in Section 2) are *adequate* and *necessary* in interactive analysis; **F3**: which users are reachable or not, depending on their similarity to other users; **F4**: how *relevance* and *diversity* contribute to the analysis.

We characterize different scenarios based on the *analyst's expertise* and the analysis *starting points*. We assume 5 virtual analyst summarized in Table 5. To measure the effect of expertise, we consider two cases where the analyst is *knowledgeable* about PC selection and the case where she is *a novice PC chair*. We also examine different starting points to build the PC: a subset of the final PC, a subset of the previous year's PC (i.e., WEBDB 2013), or a set of arbitrary researchers outside the PC. Figure 6 shows that the average number of steps to cover 80% of the PC is 9.4. At each analysis step, $k = 3$. Figures 7 and 8 illustrate the results. Notation is simplified by replacing *actKeepUsers()* with **keep** and *actModifyLabel()* with **add/remove**.

In the KNOWIN scenario (Figure 7), a knowledgeable analyst starts with a subset of the final PC, i.e., G. Fletcher, M. Theobald, S. Michel, and X. Xiao, and selects the last two as a seed group (because they are prolific young researchers with a high number of publications.) Exploring this group results in 3 groups out of which the one labeled with `SIGMOD` (the conference that hosts WEBDB) contains 4 researchers of interest (L. Popa, A. Doan, Benedikt and S. Amer-Yahia). This already shows the advantage of user group analysis (fact **F1**) where in one single step, 4 PC members are retrieved. The analyst then uses *actModifyLabel()* to replace the predicate `high publi` with `data integration` (i.e., the WEBDB main theme in 2014) and decides to exploit the resulting group. In step D, the analyst keeps only P. Fraternali and F. Naumann among 12 group members using *actKeepUsers()* action. This action makes it easier to reach groups containing items like `SIGMOD` (P. Fraternali and F. Naumann have 9 and 6 SIGMOD publications respectively) and `ICDE` (e.g., F. Naumann has 14 ICDE publications). This shows the necessity of actions, confirming fact **F2**. Up to step E, the analyst is able to find 14 out of 15 PC members. The missing PC member is Jian Li. We compare Li's word cloud in Figure 5 containing all his publication title words, conferences and journals, with the cloud for all WEBDB 2014 PC members. This shows that Li's research areas differ significantly. This is an observation of the fact **F3** that shows the limitation of interactive analysis.

Scenario	Analyst	Starting Point
KNOWIN	Knowledgeable	Inside WEBDB 2014
KNOWOUT	Knowledgeable	Outside WEBDB
KNOW13	Knowledgeable	Inside WEBDB 2013
NONEXPERTIN	Non-expert	Inside WEBDB 2014
NONEXPERTOUT	Non-expert	Outside WEBDB

Table 5: Validation Scenarios

In KNOWOUT (Figure 7), the knowledgeable analyst starts with J. Leskovec and A. Siebes, two researchers outside the final WEBDB PC. The *opExplore()* operation first finds k related groups that expand possible candidates. In step H, the analyst encounters the same group as in step A of scenario KNOWIN. This shows that in this case, a knowledgeable analyst only needs 2 more steps to reach relevant groups from a random departure point. Step H is also an illustration of fact **F4** and shows that all 3 returned groups are *relevant* and *diverse* leading the analyst to pick the group labeled with `SIGMOD`.

We verified another scenario KNOW13 where the analyst starts from a subset of the WEBDB 2013[11] PC which has 10 researchers in common with 2014. Due to lack of space, we do not illustrate it. We observed that KNOW13 is very similar to KNOWIN.

In NONEXPERTIN (Figure 8), we consider a junior PC. The aim is to observe the effect of *analyst expertise* by comparing this sce-

[10] *Computed based on the profile of researchers containing attributes defined in Section 4.1.*

[11] *http://webdb2013.lille.inria.fr*

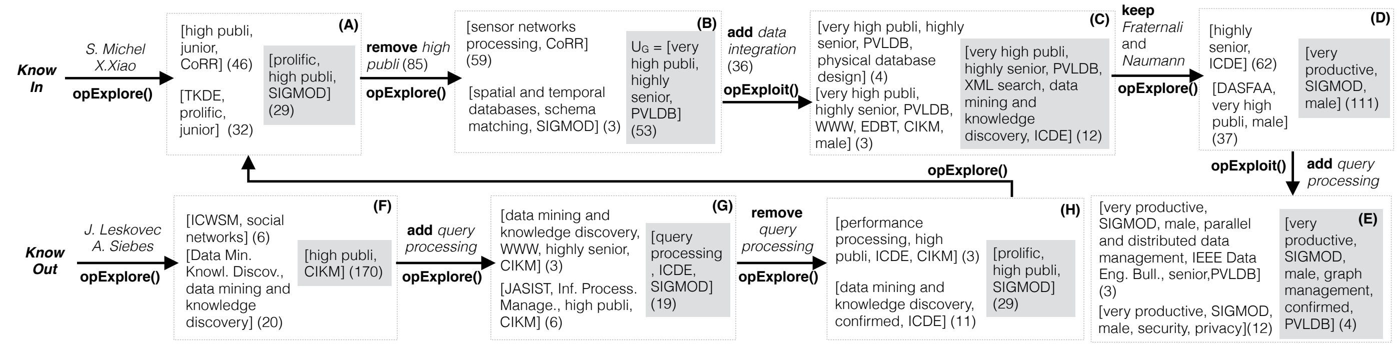
Know In
S. Michel X.Xiao
opExplore()
(A)
[high publi, junior, CoRR] (46)
[TKDE, prolific, junior] (32)
[prolific, high publi, SIGMOD] (29)
remove high publi (85)
opExplore()
(B)
[sensor networks processing, CoRR] (59)
[spatial and temporal databases, schema matching, SIGMOD] (3)
U_G = [very high publi, highly senior, PVLDB] (53)
add data integration (36)
opExploit()
(C)
[very high publi, highly senior, PVLDB, physical database design] (4)
[very high publi, highly senior, PVLDB, WWW, EDBT, CIKM, male] (3)
[very high publi, highly senior, PVLDB, XML search, data mining and knowledge discovery, ICDE] (12)
keep Fraternali and Naumann
opExplore()
(D)
[highly senior, ICDE] (62)
[DASFAA, very high publi, male] (37)
[very productive, SIGMOD, male] (111)
opExploit()
add query processing
opExplore()
(E)
[very productive, SIGMOD, male, parallel and distributed data management, IEEE Data Eng. Bull., senior,PVLDB] (3)
[very productive, SIGMOD, male, security, privacy](12)
[very productive, SIGMOD, male, graph management, confirmed, PVLDB] (4)
Know Out
J. Leskovec A. Siebes
opExplore()
(F)
[ICWSM, social networks] (6)
[Data Min. Knowl. Discov., data mining and knowledge discovery] (20)
[high publi, CIKM] (170)
add query processing
opExplore()
(G)
[data mining and knowledge discovery, WWW, highly senior, CIKM] (3)
[JASIST, Inf. Process. Manage., high publi, CIKM] (6)
[query processing , ICDE, SIGMOD] (19)
remove query processing
opExplore()
(H)
[performance processing, high publi, ICDE, CIKM] (3)
[data mining and knowledge discovery, confirmed, ICDE] (11)
[prolific, high publi, SIGMOD] (29)

Figure 7: Scenarios KnowIn (top) and KnowOut (bottom)

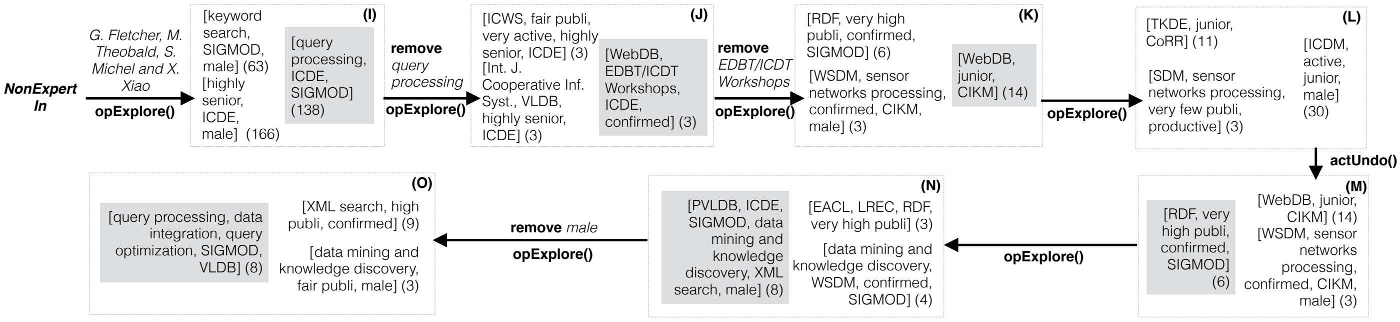
NonExpert In
G. Fletcher, M. Theobald, S. Michel and X. Xiao
opExplore()
(I)
[keyword search, SIGMOD, male] (63)
[highly senior, ICDE, male] (166)
[query processing, ICDE, SIGMOD] (138)
remove query processing
opExplore()
(J)
[ICWS, fair publi, very active, highly senior, ICDE] (3)
[Int. J. Cooperative Inf. Syst., VLDB, highly senior, ICDE] (3)
[WebDB, EDBT/ICDT Workshops, ICDE, confirmed] (3)
remove EDBT/ICDT Workshops
opExplore()
(K)
[RDF, very high publi, confirmed, SIGMOD] (6)
[WSDM, sensor networks processing, confirmed, CIKM, male] (3)
[WebDB, junior, CIKM] (14)
opExplore()
(L)
[TKDE, junior, CoRR] (11)
[SDM, sensor networks processing, very few publi, productive] (3)
[ICDM, active, junior, male] (30)
actUndo()
(M)
[WebDB, junior, CIKM] (14)
[WSDM, sensor networks processing, confirmed, CIKM, male] (3)
[RDF, very high publi, confirmed, SIGMOD] (6)
opExplore()
(N)
[EACL, LREC, RDF, very high publi] (3)
[data mining and knowledge discovery, WSDM, confirmed, SIGMOD] (4)
[PVLDB, ICDE, SIGMOD, data mining and knowledge discovery, XML search, male] (8)
remove male
opExplore()
(O)
[XML search, high publi, confirmed] (9)
[data mining and knowledge discovery, fair publi, male] (3)
[query processing, data integration, query optimization, SIGMOD, VLDB] (8)

Figure 8: Scenario NonExpertIn

nario with KnowIn. We will see that in the case of poor expertise, the analysis is done mostly by exploration. We also observe a tendency to manipulate group labels rather than group membership (specific researchers in groups). The analyst starts with 4 given researchers and applies *opExplore()* to expand the analysis scope. The analyst finds a group of 138 researchers labeled with `query processing`, `SIGMOD` and `ICDE`, and decides to expand that group by removing `query processing`. The analyst then navigates up to step L, where she does not find any helpful group. Thus she commands an *actUndo()* action. Up to step O, she finds 12 out of 15 PC members. Compared to KnowIn, the number of useless steps (without any PC member discovery) has increased.

Finally, in NonExepertOut, we examine the case where a non-expert analyst starts with researchers outside the final PC. In this scenario, the analyst may abandon a path and start again with different groups. She may need to repeat that until a satisfying starting point is found. In our experiment, a non-expert analyst jumps 4 times to land at step K, the first step of NonExepertIn.

4.2.2 Single-Target Scenario

The previous experiment showed how effective our interactive analysis is in building a program committee by "gathering" members of interest along the way during the analysis. In this experiment, we focus on validating the effectiveness of IUGA in finding a single target as described in Example 2 in Section 1. We use our synthetic dataset that was generated to scale up MovieLens. Our dataset is a matrix M with 3×10^7 cells, where squares with at least 10 users (i.e., minimum support σ) filled with 1, represent user groups. We propose a measure called Average Target Arrival (ATA), i.e., the average number of iterations to reach a group containing a target group starting from a non-target group. We randomly mark 50 groups as targets and compute ATA for those groups (we refer to target groups as $\mathcal{G}_{target}$). We compare IUGA with two different baselines: *unsupervised* and *interactive*. Briefly, if m_1 and m_2 are two different methods and ATA(m_1) < ATA(m_2), then m_1 is considered faster. Note that the concept of ATA differs significantly from finding the shortest path. For the latter, we assume the starting and target points are known, while this is not the case in the interactive analysis.

Algorithm 2: Experimental ATA Protocol $ATAalg$

```
Input: $\mathcal{G}, \mathcal{G}_{target}, k, \mu, g, len, method, maxlen$
Output: length of navigation path
if $g \in \mathcal{G}_{target}$ then return $len$
if $len > maxlen$ then return -1 // lost path
$\mathcal{G}_k \leftarrow choose(opExplore(g, \mathcal{G}, \mu, k, method),$
    $opExploit(g, \mathcal{G}, k, method))$
foreach $g \in \mathcal{G}_k$ do
    $ATAalg(\mathcal{G}, \mathcal{G}_{target}, k, \mu, g, len + 1, method, maxlen)$
end
```

Algorithm 2 illustrates how ATA is computed. We designed 200 different sessions each of which has a different synthetic dataset and is repeated 100 times for each method. Hence, we compute 20,000 ATA values for each one of the interactive analysis methods we defined. For a random group g_{rnd}, k groups are returned using *method*, and a random choice between *opExplore()* and *opExploit()* (the algorithm starts always with an *opExplore()*). *Each of* the k groups becomes the new seed. This depth-first recursive call terminates either when one group in $\mathcal{G}_{target}$ is found or when a path of length 50 has been built (*maxlength* in Table 4). These recursive calls form paths inside the group space. A path is called *valid* if its last group belongs to $\mathcal{G}_{target}$. The ATA is computed as

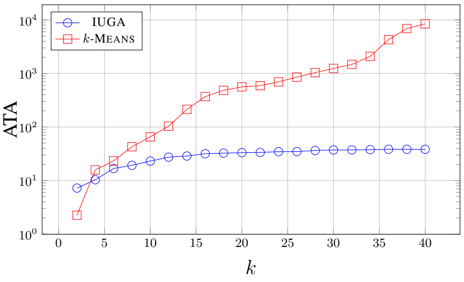

Figure 9: IUGA **Comparison with Clustering Algorithm**

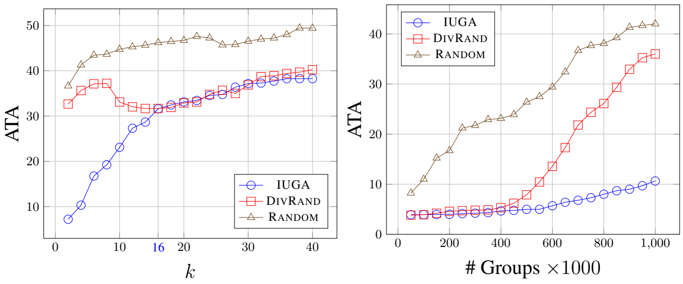

Figure 10: ATA as a Function of k and # Groups

the average of valid path lengths for each method.

Unsupervised Baseline Comparison.

This experiment compares IUGA to a variant of k-MEANS with *Jaccard* as the distance measure. k-MEANS is the representative of offline clustering algorithms with a constraint on the number of clusters (k). At each step, both IUGA and k-MEANS return k groups while respecting $timelimit$. Any number of iterations is allowed for k-MEANS within $timelimit$. We then report ATA for both methods. For k-MEANS, we randomly add/remove attributes at each step i so that a new set of k clusters is obtained in step $i+1$. Presence or absence of an attribute changes the clusters' membership, as the *Jaccard* distance between users varies. For instance, adding a specific value of `age` reduces the distance between two users having the same age.

Figure 9 illustrates ATAs for IUGA and k-MEANS in log scale. We vary k from 2 to 40 and observe how ATA for both algorithms evolves. While k-MEANS performs better for very small values of k, IUGA outperforms it by two orders of magnitude for higher values of k. When k is very small, clusters are huge. Thus most of the time, there exists a cluster that contains all users of a target group. For larger values of k, more clusters with smaller size are generated and more steps are needed to finally reach the target. We can conclude that the superiority of IUGA over unsupervised methods comes from the use of diversity at each step in order to cover as many users as possible.

Interactive Analysis Baseline Comparison.

We compare IUGA with some interactive analysis baselines: DIVRAND, RANDOM, EXHAUSTIVE and ILP. At each step, DIVRAND randomly generates as many sets of k groups as possible within $tlimit$ and returns the one with the highest diversity. RANDOM navigates randomly in the space of groups and does not respect $tlimit$. EXHAUSTIVE generates all possible k among n groups in $\mathcal{G}$, i.e., $C\binom{n}{r}$ and chooses the one with the highest diversity. ILP returns k groups with maximal diversity using an integer linear programming formulation (using CHOCO 3.0 solver[12]).

Table 6 illustrates ATA and execution times for IUGA and optimal methods. Since both EXHAUSTIVE and ILP generate optimal paths, their ATA are very close. However, their execution times are different. This experiment shows that IUGA is faster than EXHAUSTIVE and ILP (3.49 minutes faster than ILP) while maintaining a comparable ATA.

Figure 10 illustrates ATA for all heuristic-based methods: IUGA, DIVRAND and RANDOM by varying k from 2 to 40, and varying # groups from 50,000 to 1,000,000. Optimal methods (EXHAUSTIVE and ILP) do not terminate for this experiment. In general, we observe that IUGA has much lower ATA for $k \leq 16$ and $k \geq 30$. This simply shows that considering relevance and diversity at each step reduces ATA by an average of 15.91 steps.

	EXHAUSTIVE	ILP	IUGA
ATA	9.90	9.91	10.13
Time (sec)	862.47	213.12	3.35

Table 6: IUGA **vs Optimal Methods**

For $k \in [16, 30]$, DIVRAND and IUGA have close results. This shows that although the *relevance* component, i.e., the difference between DIVRAND and IUGA, is shown to be very useful in general, it is less effective for large values of k. In [5, 24], it is shown that in a context with too many options and no hint for further navigation, *long* jumps are preferred to *short* jumps. In our case, *relevance* tends to favor *short* directed jumps in the space of groups while DIVRAND does not. This is why when few options are available, IUGA performs better and DIVRAND performs as well as IUGA for larger values of k. In another research[13], it has been shown that people faced with numerous choices, whether good or bad, find it difficult to stay focused on a task. Choosing a small value of k is hence better both for performance and effectiveness.

We observe that increasing the number of groups has a huge effect on DIVRAND. When the number of groups increases, the target groups are more likely to be diverse. Thus, precision (ratio of valid paths over all navigated paths) decreases for all methods, while thanks to relevance, the decrease is negligible for IUGA.

4.3 Quality of Discovered Groups

Our last experiment focuses on a single step of IUGA by evaluating the quality of obtained groups at each iteration of GROUPDISCOVERY. We ask participants in a user study to compare the top 5 groups obtained by GROUPDISCOVERY with some competitive methods. We use MOVIELENS because people are usually familiar with movies and their attributes. We setup a questionnaire which was answered anonymously by 35 participants. The evaluation consists of a *comparative evaluation* where results of competitive methods are evaluated together, and an *individual evaluation* where each set of top 5 groups is evaluated separately.

In the comparative evaluation, we compare the top-5 groups obtained by GROUPDISCOVERY with those returned by 4 baselines: *Largest Groups*, *Most Overlapping*, *Least Overlapping* and *Most Concise* (groups which have the shortest description). Those baselines were designed using interestingness measures commonly used in pattern evaluation [14]. Each question contains an input group (e.g. [`Total Recall`, `Star Wars IV`] in MovieLens dataset)

[12] *http://choco-solver.org/?q=Choco3*

[13] *Too many choices (good or bad) can be mentally exhausting: http://phys.org/news127404469.html*

and sets of top-5 groups corresponding to each method. Participants chose the method that offers the most satisfying top-5 groups. Also, participants were instructed to select a justification for their preferred method: *it helps better understand who does what*, *it helps to discover new users*, *it helps to discover new group labels*.

Figure 11 illustrates the average percentages of responses for each analysis option. In this part, participants have mostly preferred the results of GroupDiscovery followed by *Most Concise* groups. Also, they have mostly justified their responses as *it helps better understand who does what* (52.75%). The choice of *Most Concise* groups reveals that people prefer groups with short labels.

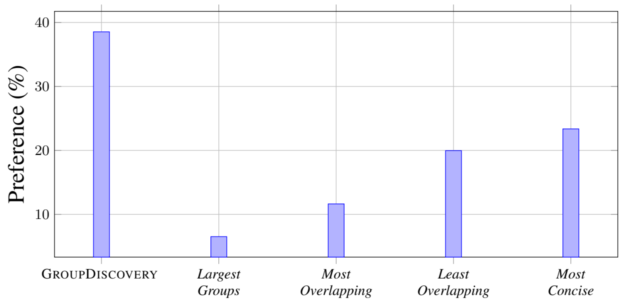

Figure 11: User Preference Results for Set Evaluation

In the *individual* evaluation, we compare each of top 5 groups of GroupDiscovery with *Most Overlapping* and *Least Overlapping* groups, i.e. two extremes. Participants have preferred groups of GroupDiscovery in 51.79% of cases, *Most Overlapping* in 33.12% and *Least Overlapping* in 15.09%. They justified their responses as follows: *Justification 1*: *understand the selected group*, *Justification 2*: *discover new users* or *Justification 3*: *understand the whole data*. Whenever our solution was selected, *Justification 1* was chosen by 56% of participants on average, followed by *Justification 2* (34.22%). In general, 63% of participants mentioned that their preferred group helps better understand the selected group (*Justification 1*), followed by 28% who believe the preferred group helps discover new users (*Justification 2*).

5. RELATED WORK

To the best of our knowledge, no approach has proposed and formalized an interactive group analysis framework. Recent studies [14] have shown an interest in reporting statistics about pre-defined groups, as opposed to our work where we look to discover users. Our work does relate to a number of others in functionality, interactivity/visualization and diversity.

Constrained-based Mining: IUGA bears similarity to constrained mining [6, 7], but the latter heavily relies on the analyst having some knowledge about the underlying data to be able to formalize constraints, as opposed to our data-driven work where relevant and diverse options are suggested to the analyst.

Interactive and Visual Analysis: Interactive pattern analysis approaches [3, 5] focus on learning the subjective measure in the mind of the analyst to guide pattern exploration. For example, OneClick [5] is a personalized interactive navigation approach that learns an interestingness function based on patterns that were *liked* or *unliked* by the analyst in previous steps. In IUGA, we adopt an approach based on exploration or exploitation and let the analyst choose which operation to apply at each step. The ability to *personalize* the navigation as in OneClick is an interesting direction for future work. Semi-automatic PC construction has been addressed in [8]. The proposed approach requires the definition of *quality* and *cost* values for each researcher. This is a subjective and challenging task that assumes full knowledge of researchers' profiles which is not always the case in out analysis tasks.

While visualization [21, 18] and interactive exploration differ in focus, there exist a few efforts that combine both. Few examples are data mining suites like RapidMiner, Knime, and MIME [15]. These approaches develop a toolbox to manipulate and visualize patterns according to preferences specified by the analyst at each step. In MIME, analysts can refine discovered patterns on the fly by selecting additional items. These methods do not provide semantics for exploration or exploitation nor do they rely on an optimization framework to cover the space of patterns.

Diversity: Diversity is a widely studied subject that finds its roots in Web search with a goal similar to ours. In [9], the concept of diversity in text retrieval and summarization is introduced to balance document relevance and novelty. Most approaches fall into two cases: *content-based* (e.g. [9]) and *intent-based* [11]. Our algorithm, GroupDiscovery, is based on a greedy approach similar to content-based diversification.

6. CONCLUSION

We introduced IUGA, the first interactive user data analysis framework that is based on a simple and intuitive optimization formulation: the GroupDiscovery Problem that finds the k most diverse and relevant user groups a seed group. IUGA relies on two group discovery operations: exploration and exploitation. We prove the hardness of our problem and devise greedy algorithms to help analysts navigate in the space of groups and reach one or several target users. Our extensive experiments on real and synthetic datasets show the utility of relevance and diversity in group discovery and in finding users of interest in different scenarios. We are currently pursing two improvements: (i) expressing group discovery as a multi-objective optimization problem and (ii) incorporating the *abstraction* operator defined in [19] to better summarize found groups.

[14] *http://blog.testmunk.com/how-teens-really-use-apps/*

7. REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[3] M. Bhuiyan, S. Mukhopadhyay, and M. A. Hasan. Interactive pattern mining on hidden data: a sampling-based solution. In *CIKM*, pages 95–104, 2012.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] M. Boley, B. Kang, P. Tokmakov, M. Mampaey, and S. Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. *IDEAS (ACM SIGKDD Workshop)*, 2013.

[6] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. Exante: Anticipated data reduction in constrained pattern mining. In *PKDD*, pages 59–70, 2003.

[7] C. Bucila, J. Gehrke, D. Kifer, and W. M. White. Dualminer: a dual-pruning algorithm for itemsets with constraints. In *Knowledge Discovery and Data Mining*, pages 42–51, 2002.

[8] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask?: jury selection for decision making tasks on micro-blog services. *VLDB*, 2012.

[9] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336, 1998.

[10] U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, O. Papaemmanouil, and S. B. Zdonik. Query steering for interactive data exploration. In *CIDR*, 2013.

[11] O. Chapelle, S. Ji, C. Liao, E. Velipasaoglu, L. Lai, and S.-L. Wu. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592, 2011.

[12] U. Feige, G. Kortsarz, and D. Peleg. The dense *k*-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[13] N. Friedman, M. Goldszmidt, et al. Discretizing continuous attributes while learning bayesian networks. In *ICML*, pages 157–165, 1996.

[14] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9, 2006.

[15] B. Goethals, S. Moens, and J. Vreeken. Mime: A framework for interactive visual pattern mining. In *PKDD*, volume 6913, pages 634–637, 2011.

[16] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *ACM SIGMOD SIGART*, pages 100–108. ACM, 2014.

[17] D. S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49. ACM, 1973.

[18] A. Leuski and J. Allan. Strategy-based interactive cluster visualization for information retrieval. *International Journal on Digital Libraries*, 3:170–184, 2000.

[19] B. Omidvar-Tehrani, S. Amer-Yahia, A. Termier, A. Bertaux, E. Gaussier, and M.-C. Rousset. Towards a framework for semantic exploration of frequent patterns. *IMMoA*, 2013.

[20] L. Parida. Redescription mining: Structure theory and algorithms. In *In Proc. AAAI'05*, pages 837–844, 2005.

[21] C. K. sang Leung, P. P. Irani, and C. L. Carmichael. WiFIsViz: Effective Visualization of Frequent Itemsets. In *ICDM*, 2008.

[22] A. Siebes, J. Vreeken, and M. van Leeuwen. Item sets that compress. In *SDM*, volume 6, pages 393–404. SIAM, 2006.

[23] T. Uno, M. Kiyomi, and H. Arimura. Lcm ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI*, 2004.

[24] R. West and J. Leskovec. Automatic versus human navigation in information networks. In *ICWSM*, 2012.