Flash points: Discovering exceptional pairwise behaviors in vote or rating data

Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre, Marc Plantevit

Flash points: Discovering exceptional pairwise behaviors in vote or rating data

Adnene Belfodil[1], Sylvie Cazalens[1], Philippe Lamarre[1], and Marc Plantevit[2]

[1] INSA Lyon, CNRS, LIRIS UMR5205, F-69621 France
[2] Université Lyon 1, CNRS, LIRIS UMR5205, F-69622 France

Abstract. We address the problem of discovering contexts that lead well-distinguished collections of individuals to change their pairwise agreement w.r.t. to their usual one. For instance, in the European parliament, while in overall, a strong disagreement is witnessed between deputies of the far-right French party *Front National* and deputies of the left party *Front de Gauche*, a strong agreement is observed between these deputies in votes related to the thematic: *External relations with the union*. We devise the method *DSC* (*Discovering Similarities Changes*) which relies on exceptional model mining to uncover three-set patterns that identify contexts and two collections of individuals where an unexpected strengthening or weakening of pairwise agreement is observed. To efficiently explore the search space, we define some closure operators and pruning techniques using upper bounds on the quality measure. In addition of handling usual attributes (e.g. numerical, nominal), we propose a novel pattern domain which involves hierarchical multi-tag attributes that are present in many datasets. A thorough empirical study on two real-world datasets (i.e., European parliament votes and collaborative movie reviews) demonstrates the efficiency and the effectiveness of our approach as well as the interest and the actionability of the patterns.

Keywords: Exceptional model mining; Subgroup discovery

1 Introduction

The last decade has witnessed a huge growth in the collection of rating (e.g., Amazon, IMDb, Yelp, Foursquare) or vote (e.g., Parltrack, Voteview) data. Such data depict the opinion (i.e., review, or vote) of people (e.g., IMDb users, European parliament member) on an item (e.g., movie, restaurant, ballot) and need to be analyzed by leveraging contextual information to discover new actionable insights that cannot be obtained otherwise. There has been a rapid rise in the analysis of such data in many applications such as fact checking or lead finding in political journalism, and collaborative rating analysis.

Fact checking has become increasingly common in political journalism. It contributes to the quality of news provided by media[3]. For instance, *Truth-O-Meter*[4] was extensively used during the 2016 US presidential debate. Delving

[3] *"increasing quality of journalism will lead to better decisions by citizens..."*[16]
[4] http://www.politifact.com/truth-o-meter/

deeply into the votes sessions makes it possible to enlighten some claims about consensus between politicians or finding some flashpoints (i.e., contexts that lead to strong disagreement). Average rating is not enough for an item. While some individuals are in agreement on many items, they can be in strong disagreement for certain types of items. Such information can directly be used for recommendation. For example, in Movielens dataset, while usually *middle-aged women* users are in agreement with *middle-aged men* users w.r.t. their overall ratings, these collections are in disagreement for *Comedy* movies released in 1998.

The discovery of descriptions that distinguish a group of objects given a target (class) has been widely studied in data mining and machine learning community under several vocables (subgroup discovery, emerging patterns, contrast sets) [14]. We consider here the well-established framework of subgroup discovery (SD)[22]. Given a set of objects taking a vector of attributes (of Boolean, nominal, or numerical type) as description, and a class label as a target, the goal is to efficiently discover subgroups of objects for which there is a high difference between the label distribution within the group compared to the distribution within the whole dataset. SD has been extended to a richer framework that handles more complicated target concepts, the so-called Exceptional Model Mining (EMM)[17]. A model is built over the labels from the objects in the subgroup and is compared to the model of the whole dataset using a quality measure. The more different is the model, the more exceptional is the subgroup. Many models have been investigated in the last decade [21, 8, 7, 13, 6]. However, no model in the EMM framework makes it possible to characterize collection of individuals whose pairwise agreement exceptionally deviates according to a subset of objects.

In this paper, we introduce the problem of discovering collections of individuals and particular contexts where their pairwise agreement exceptionally differs from their usual one as an instance of EMM. Fig. 1 gives an overview of our approach. Based on an *aggregation level* set a priori, the method begins by constituting collections of individuals (1). Bi-sets of individuals are identified by a description (2) and their global pairwise behavior is computed (3). The method eventually aims to identify subset of reviewed items (4) for which the related pairwise behavior (5) substantially differs from the global one (6). To discover such patterns, we have to simultaneously explore the search space associated to the reviewed items and the search space associated to the reviewers. To this end, we devise the method *DSC* (*Discovering Similarities Changes*) to discover three-set patterns $(context, collection_1, collection_2)$ that identify a context and two collections of individuals where an unexpected strengthening or weakening of pairwise agreement is observed. We define some closure operators and some effective pruning techniques based on the computation of tight upper bounds on the quality measure to efficiently explore the search space. *DSC* is able to handle numerical, nominal attributes and also hierarchical multi-tag attributes. The main contributions of this paper are manifold:

Problem formulation. We define the novel problem of exceptional pairwise behavior discovery in the EMM framework. This formulation makes it possible to consider several similarity measures to assess the pairwise agreement.

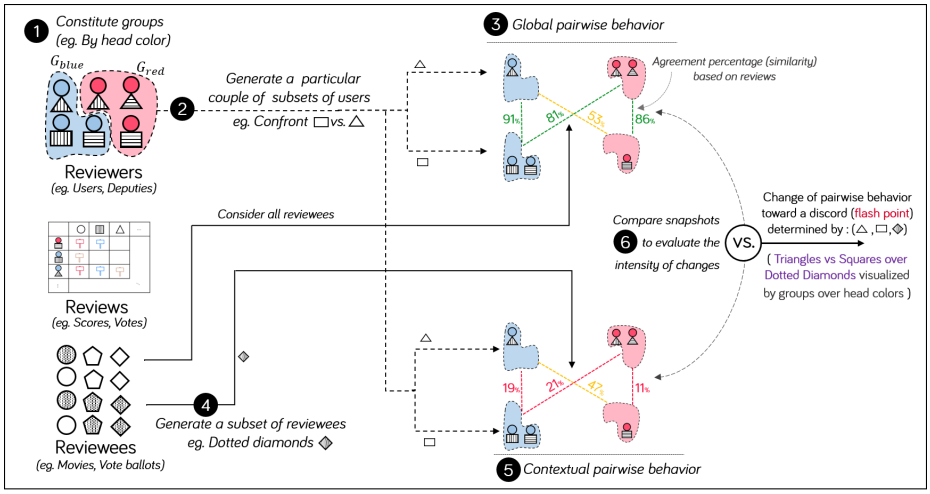


Fig. 1: Overview of *DSC*

Algorithm and analysis. We propose a branch-and-bound algorithm that efficiently exploits tight upper bounds and closure operators.
Evaluation. We report a thorough empirical study on real-world datasets that demonstrates the efficiency and the effectiveness of *DSC*.

The rest of the paper is organized as follows. Section 2 gives the formal definition of the exceptional pairwise behavior discovery problem. Section 3 presents the algorithms. Section 4 provides experimental results. Section 5 reviews the related work. Section 6 concludes and provides future directions.

2 Problem definition

Data describing individuals outcomes about items are numerous, ranging from vote data to collaborative ratings through social-media platforms. We model such data as a triple $\langle E, U, R\rangle$ where E is a collection of objects (e.g., ballots, items, restaurants) and $\mathcal{A}_E = \{e_1, ..., e_n\}$ depicts the schema of the studied objects described by n attributes. U identifies the individuals (e.g., social network users, parliament members) described by m attributes over the schema $\mathcal{A}_U = \{u_1, ..., u_m\}$. Eventually, R represents the reviews (e.g., opinions, votes, ratings) of individuals over the objects. Each element of R is a triple $r = (e, u, o)$ where $o \in O$ is the outcome of a user $u \in U$ over an item $e \in E$. The function $o(e, u)$ returns the outcome o of u over an item e.

A description c over E defines a set of restrictions over the domains of the attributes $\mathcal{A}_E$. Such a description gives a context and identifies a subgroup of E denoted E_c which is a collection of objects that fulfill the restrictions of c. We use the symbol $*$ to refer to the context that covers all the objects, therefore $E_* = E$. Similarly, a description g over U, which is a set of restrictions over the domains of the attributes $\mathcal{A}_U$, identifies a collection of individuals denoted U_g.

We aim to discover a context c and collections of individuals $U_{g'} \subseteq U$, $U_{g''} \subseteq U$ (*labeled respectively by their descriptions* g', g'' *over the attributes of* U) such that their pairwise agreement (similarities) differs exceptionally from the observed pairwise agreement over the whole objects. In other terms, we want to identify patterns (c, g', g'') that suggest an important change in pairwise behavior between $U_{g'}$ and $U_{g''}$ within a context c. To this end, the outcomes of $U_{g'}$ and $U_{g''}$ have to be compared. Therefore, we need to define a similarity function between individuals over a given subgroup of objects. However, ratings data are generally sparse which limits the set of objects that have been rated by a pair of individuals. To overcome this issue, we have to consider aggregates of individuals and their *aggregated outcome*. The operator γ_L builds a partition of U according to their values on the attributes $L \subseteq \mathcal{A}_U$. For instance, if U represents deputies affiliated to national parties depicted by the attribute np, $\gamma_{\{np\}}(U) = \{G_1, G_2, ...\}$ is a partition of U where each G_i represents a set of individuals affiliated to the same party.

We define an aggregated outcome operator $\theta : E \times 2^U \rightarrow O$ which maps an aggregate of individuals $G \subseteq U$ to its aggregated outcome w.r.t. an object e. For example, when dealing with movie ratings, aggregated outcome $\theta(e, G)$ can be defined as the mean of ratings given by some individuals of G to a movie e. We can compare the similarity between two sets of individuals based on their aggregated outcomes. The similarity measure is thus defined as: $sim : 2^E \times 2^U \times 2^U \longrightarrow [0, 1]$.

Our method relies on an EMM vision. Thus, we first need to determine a *model class* and a *quality measure* φ over this *model class*. We use a *similarity matrix* as a model to capture the pairwise agreement between pairs of user collections ($U_{g'}$, $U_{g''}$). Note that, contrary to common EMM approaches, there is *no unique base model* on the whole data but a model is *related to* a pair of descriptions (g', g'') identifying collections of individuals. The *base model* denoted $M_*^{g',g''}$, which represents the usual observed pairwise agreement over the whole objects between the candidate subgroups $U_{g'}, U_{g''}$, is defined as: $M_*^{g',g''} = \big(sim\,(E_*, i, j)\big)_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})}$. The model built for a context c depicting a subgroup of objects E_c is: $M_c^{g',g''} = \big(sim\,(E_c, i, j)\big)_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})}$.

The quality measure φ aims to quantify how much the model induced by the subgroup is different from the base model, i.e., how much the pairwise agreement observed over the whole objects differs from the one observed in a particular context between $U_{g'}$ and $U_{g''}$. Several quality measures can be defined according to the use case. For example, if we are interested in finding controversial contexts, we define $\varphi_{dissent}$ that captures the average similarity weakening between pairs of $\gamma_L(U_{g'}) \times \gamma_L(U_{g''})$:

$$\varphi_{dissent}(c, g', g'') = \frac{\sum_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})} max(sim(E_*, i, j) - sim(E_c, i, j), 0)}{|\gamma_L(U_{g'})|.|\gamma_L(U_{g''})|}$$

To find patterns (c, g', g'') that suggest an unexpected change of pairwise agreement, we rely on a well-known task, i.e, the discovery of *Top-k patterns* that fulfill a *minimum quality threshold* constraint σ_φ. Additional constraints can be taken into account (e.g. $\langle |E_c| \geq \sigma_E,\ |U_{g'}| \geq \sigma_U,\ |U_{g''}| \geq \sigma_U \rangle$).

3 Discovery of exceptional pairwise behaviors

In this section, we describe the enumeration principle based on closure operators, especially in the case of attributes whose domain is defined as a hierarchy. We then present different aggregates and similarities as well as the quality measures and their related tight upper and lower bounds. We eventually describe the algorithms to discover exceptional pairwise behaviors.

3.1 Candidate descriptions enumeration

Description language. Let $\mathcal{G}$ be a generic collection of tuples which can be either E or U, and $\mathcal{A}_\mathcal{G} = (a_1, a_2, ..., a_n)$ its schema defined over n attributes. We denote by $dom(a_i)$ the domain of an attribute a_i. A *description* $d = \langle r_1, r_2, ..., r_n \rangle$ is a conjunction of restrictions over the attributes domains, where each restriction r_i corresponds to the attribute a_i. The restriction definition depends on the attribute type. If an attribute a_i is *nominal* then the corresponding restriction r_i is assimilated to a membership into a subset of $dom(a_i)$. Otherwise, if an a_i is *numeric* then the corresponding restriction r_i refers to a membership into an interval. The set of all possible descriptions is denoted $\mathcal{D}$. A description $d \in \mathcal{D}$ defines by intent a *subgroup* (*extent*) $\mathcal{G}_d \subseteq \mathcal{G}$ which contains the tuples of $\mathcal{G}$ verifying the restrictions of d. In order to bind the descriptions of $\mathcal{D}$ to subgroups in $\mathcal{G}$, we define a mapping function $\delta : \mathcal{G} \to \mathcal{D}$ that maps each tuple $g \in \mathcal{G}$ to its description in $\mathcal{D}$. To define this mapping function, we rely on the corresponding mappings $\delta_{a_i} : dom(a_i) \to \mathcal{D}_{a_i}$ that maps the values of an attribute a_i to its corresponding restriction $r_i \in \mathcal{D}_{a_i}$. Given a tuple g, an attribute a_i and its value a_i^g in g, if a_i is numeric, the restriction is an interval $\delta_{a_i}(a_i^g) = [a_i^g, a_i^g]$. Otherwise, if a_i is nominal, the restriction is a singleton $\delta_{a_i}(a_i^g) = \{a_i^g\}$. Finally, with the former definitions, for a tuple $g = (a_1^g, ..., a_n^g)$ we have $\delta(g) = \langle \delta_{a_1}(a_1^g), ..., \delta_{a_n}(a_n^g) \rangle$.

Description space structure. To enumerate candidate descriptions (*or candidate subgroups by extent*), we traverse the search space $\mathcal{D}$ in a bottom-up fashion. This search space is commonly depicted as a *meet-semi lattice* structured by an *infimum operator* denoted by $\sqcap$ [10] which simply allows to get the lowest common description of two given descriptions. The infimum operator definition relies on the n infimum operator $\sqcap_{a_i}$ corresponding each to the type of the attribute a_i. Let a be a *numeric* attribute, the corresponding infimum operator $\sqcap_a$ computes the minimum interval enclosing two intervals. In the other hand, if a is *nominal*, the corresponding infimum operator $\sqcap_a$ is represented by a set union operator. Thus the meet-semi lattice $(D, \sqcap)$ is the result of the cartesian product of the meet-semi lattices $(\mathcal{D}_a, \sqcap_a)$ each corresponding to an attribute $a \in \mathcal{A}_\mathcal{G}$. The infimum operator allows us to define a *partial order* denoted by $\sqsubseteq$ between descriptions. Given two descriptions c and d, we have $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$.

Specialization and neighborhood relations. Let $c = \langle q_1, q_2, ..., q_n \rangle$ and $d = \langle r_1, r_2, ..., r_n \rangle$ be two descriptions of $\mathcal{D}$, r_i and q_i are two restrictions on the attribute a_i. r_i is a *specialization* of q_i iff $r_i \Rightarrow q_i$ which is equivalent to $q_i \sqsubseteq r_i \Leftrightarrow q_i \sqcap_{a_i} r_i = q_i$. A description d is a specialization of c (denoted $c \sqsubseteq d$) iff

$\forall i \in [1..n] : q_i \sqsubseteq r_i$. Obviously, $c \sqsubseteq d \iff \mathcal{G}_d \subseteq \mathcal{G}_c$ with $\mathcal{G}_d$ (*resp.* $\mathcal{G}_c$) the subgroup covered by d (*resp.* c). When traversing the search space we extend a description to more complex descriptions by atomic refinements. Thus, we define the *neighborhood relationship* $\prec$. We have $c \prec d$ *iff* $c \sqsubset d \wedge \nexists e \in \mathcal{D} : c \sqsubset e \sqsubset d$ and d is said to be an upper neighbor of c. To get the neighbors of a candidate description $c = \langle q_1, q_2, ..., q_n \rangle$, we rely on a similar neighborhood concept between restrictions. If a restriction q is over a nominal attribute a which is materialized by a subset $s_q \subseteq dom(a)$ membership, neighbors of q are candidates r which correspond to singletons of s_q. Similarly for a numeric attribute, candidate neighbors of a restriction r are the intervals q resulting from a left-minimal change or a right-minimal change on the interval bounds corresponding to r [12]. With these tools, we can easily define a refinement operator $\eta : \mathcal{D} \to 2^{\mathcal{D}}$ which maps to each description d its neighbors in $\mathcal{D}$ and we have:

$$\begin{aligned} \eta(c) &= \{d \in \mathcal{D} : d = \langle r_1, ..., r_n \rangle \succ c = \langle q_1, ..., q_n \rangle\} \\ &= \{d \in \mathcal{D} : \exists j \in [1..n] \mid r_j \succ q_j \; and \; \forall i \in [1..n] \mid i \neq j \Rightarrow r_i = q_i\} \end{aligned} \tag{1}$$

Additionally we define η_f that computes the neighbors of a given description c by refining the f^{th} restriction corresponding to the f^{th} attribute as follows:

$$\eta_f(c) = \{d \in \mathcal{D} : r_f \succ q_f \; and \; \forall i \in [1..n] \mid i \neq f \Rightarrow r_i = q_i\} \tag{2}$$

Closed descriptions. We rely on the concept of *closed descriptions* to significantly decrease the number of explored descriptions by avoiding redundancy. A description c is said to be *closed iff* for every specialization d (*i.e.* $c \sqsubset d$) there is at least one object in $\mathcal{G}$ covered by c but not by d. More formally, $\forall d \in \mathcal{D} : c \sqsubset d \Rightarrow \mathcal{G}_d \subsetneq \mathcal{G}_c$. Two descriptions c and d are considered as equivalent (denoted $c \equiv d$) *iff* $\mathcal{G}_c = \mathcal{G}_d$. We can adapt the *CbO* (*Close-by-One*) algorithm [15] for our use in *DSC*.

To define the *closure operator* of a description d of $\mathcal{D}$, we need to introduce two derivation operators that create a Galois connection between $2^{\mathcal{G}}$ and $\mathcal{D}$: Given $S \subseteq \mathcal{G}$, the description $S^{\square} \in \mathcal{D}$ covering the subgroup S is:

$$S^{\square} := \sqcap_{g \in S} \delta(g) = \langle \sqcap_{g \in S} \delta_{a_1}(a_1^g), ..., \sqcap_{g \in S} \delta_{a_n}(a_n^g) \rangle$$

Given a description d, the subgroup $d^{\square}$ covered by d is:

$$G_d = d^{\square} = \{g \in \mathcal{G} \mid d \sqsubseteq \delta(g)\}$$

$(.)^{\square\square}$ is a closure operator and for every $d \in \mathcal{D}$ $d^{\square\square}$ is a closed description.

Canonicity test. An important aspect in *CbO* enumeration is the *canonicity test*, which allows to determine if a description after closure was already generated and discard it, if appropriate. The canonicity test relies on a linear order $\lessdot$ between descriptions of $\mathcal{D}$. Given an arbitrary order between attributes $\mathcal{A}_{\mathcal{G}} = \{a_1, a_2, ..., a_n\}$, if $d = \langle r_1, ..., r_n \rangle$ comes from a closure after a refinement of the f^{th} restriction of $c = \langle q_1, ..., q_n \rangle$ then we have: $c \lessdot_f d \iff \forall i \in [1..f-1] \mid q_i = r_i \wedge q_f \lessdot_{a_f} r_f$. Note that, in our case, the test part $q_f \lessdot_{a_f} r_f$ is always valid when the f^{th} attribute is numeric or nominal. Although, the latter need to be assessed when the attribute is rather complex, such as for *HMT attributes* introduced in the next section.

3.2 Hierarchical multi-tag attribute (HMT)

Several votes and reviews datasets contain multi-tagged objects where each tag is a part of a hierarchical structure. For instance, the ballots in the European parliament can have multiple tags (e.g., the ballot *Gender mainstreaming in the work of the European Parliament* is tagged by *4.10.04-Gender equality* and *8.40.01-European Parliament*. Tag *4.10.04* itself identifies a hierarchy where tag *4.10* depicts *Social policy* which is a specialization of tag *4* that covers the ballots related to *Economic, social and territorial cohesion*). Let $\mathcal{G}$ be a set of tagged objects. For the sake of simplicity, each object g is described by a unique attribute *tags* which is a set of tags. Tags form a tree noted T.

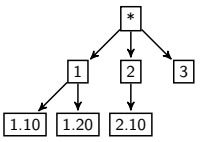
We can define the partial order $\leq$ between tags as the same usual partial order in a tree structure where the tree root is the minimum (e.g. $* < 1 < 1.20$). This allows us to define the ascendants (resp. descendants) operator $\uparrow$ (resp. $\downarrow$) of a tag $t \in T$. We have $\uparrow t = \{u \in T | u \leq t\}$ and $\downarrow t = \{u \in T | u \geq t\}$. Let t and u be two tags, t is a lower neighbor of u denoted $t \prec u$ iff $\nexists e \in T \mid t < e < u$. Thus t is a parent of u denoted as $t = p(u)$.

A restriction over an *HMT* attribute is assimilated as a membership in a set of tags $\{t_1, ..., t_n\}$. We denote the description domain by $\mathcal{D}$ which is a subset of 2^T. Each object $g \in \mathcal{G}$ is mapped by $\delta(g)$ to its corresponding description in $\mathcal{D}$. Obviously if $\delta(g) = \{t_1, t_2\}$, the object g is tagged *explicitly* by the tags t_1 and t_2 but also *implicitly* by all their generalization $\uparrow t_1$ and $\uparrow t_2$ as shown in Fig. 2.

To handle this attribute among the other attributes in the complex search space defined previously, we need to define the infimum operator $\sqcap_{HMT}$ between two descriptions of $\mathcal{D}$. Let $c = \{t_1, ..., t_n\}$ and $d = \{u_1, ..., u_m\}$ be two descriptions of $\mathcal{D}$, we define $\sqcap_{HMT}$ as: $c \sqcap_{HMT} d = max(\cup_{t \in c} \uparrow t \cap \cup_{u \in d} \uparrow u)$ where $max : 2^T \rightarrow 2^T$ is a function that maps each subset of tags $s \subseteq T$ to the leafs of the sub-tree compound of the tags of s: $max(s) = \{t \in s \mid (\downarrow t \setminus \{t\}) \cap s = \emptyset\}$

Intuitively $c \sqcap_{HMT} d$ depicts the set of the maximum explicit or implicit tags shared by the two descriptions. For instance, if $c = \{1.10, 2\}$ and $d = \{1.20, 2.10\}$, $c \sqcap_{HMT} d = \{1, 2\}$. A description d is said to be a specialization of c denoted $c \sqsubseteq d$ iff $c \sqcap_{HMT} d = c$ which means $\forall t \in c \;\; \exists u \in d \;\mid\; u \in \downarrow t$. A description c is considered as a lower neighbor of d denoted $c \prec d$ iff:

$$\begin{cases} \exists!\ (t,u) \in c \times d \ :\ t \prec u \ \wedge\ \forall t' \in (c \setminus t)\ \exists u' \in d \ :\ t' = u' & \text{if} \quad |d| = |c| \\ \forall t \in c\ \exists u \in d : t = u \wedge \exists!(t,u) \in c \times d \ \ \exists t' \in \uparrow t : p(u) = p(t') & \qquad\quad |d| = |c| + 1 \end{cases}$$



	tags
g_1	{1.20, 2.10}
g_2	{1, 3}
g_3	{1.10, 2.10, 3}
g_4	{2.10}
g_5	{1.20}

	*	1	1.10	1.20	2	2.10	3
g_1	×	×		×	×	×	
g_2	×	×					×
g_3	×	×	×		×		×
g_4	×				×	×	
g_5	×	×		×			

Fig. 2: A tags tree (left), a collection of tagged items (middle) and a vector representation (right)

Basically d is an upper neighbor of c, if either only one tag of d is refined in c by the neighborhood relation between tags or by adding a new tag in d that share parent with a tag in c or with one of its ascendants. The linear order between two conjunctions of tags $c = \{t_1, ..., t_n\}$ and $d = \{u_1, ..., u_n, ..., u_m\}$ given that d comes from a closure after refinement of the f^{th} tag of c is defined as: $c \lessdot_f d \iff \forall i \in [1..f-1] \ : \ t_i = u_i \ \wedge \ t_f \leq u_f$. The linear order between tags can be provided by the depth first search on T.

Based on the definitions of $\sqcap_{HMT}$, neighborhood relation between two sets of tags and the linear order between them, the attribute HMT can be easily handled with the aforementioned attributes (numeric and nominal) in the complex search space dealing with n attributes.

3.3 Aggregations, Similarities and Quality measures

An important aspect in *DSC* is the similarity measure between aggregates of individuals. Given $L \subseteq \mathcal{A}_U$ a set of individuals attributes on which we compute aggregates of individuals, a collection of individuals $U_g \subseteq U$ labeled by a description g, $\gamma_L(U_g) = \{G_1, G_2, ..., G_k\}$ is a partition of U_g. The aggregate outcome θ is defined according to the application domain. For example, the outcome of an aggregate of reviewers who give scores is defined as such: $\theta_{review}(e, G) = \frac{1}{|G|} \sum_{u \in G} o(e, u)$. The outcome of an aggregate G of European parliament members w.r.t a ballot is given by the vote of the majority[5] as $\theta_{votes}(e, G) = argmax_{v \in \mathcal{O}} \{count\,(v, \{o(e, u) \mid u \in G\})\,\}$.

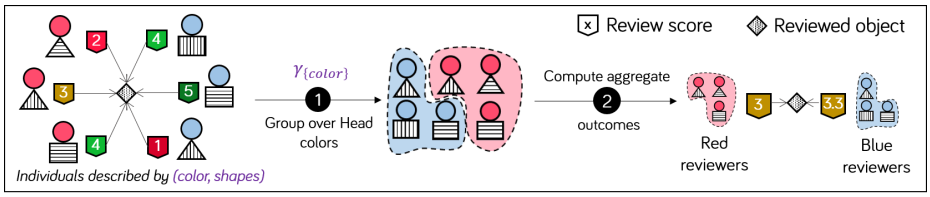


Fig. 3: Aggregates outcomes over one reviewed object

In this paper, we consider similarities that convey the average agreement proportion between two aggregates G_i, G_j based on their pairwise similarity *simobj* over each object. We define such similarities over $2^E \times 2^U \times 2^U$:

$$sim(E, G_i, G_j) = \frac{1}{|E|} \sum_{e \in E} simobj(E, G_i, G_j) \tag{3}$$

Indeed, the measure *simobj* which is defined over $E \times 2^U \times 2^U$ is adapted on the application domain. For example, if we want to compare deputies where vote decision can be either a *for*, *against* or *abstain*. we define:

[5]The same measure is used by ***votewatch*** to observe the voting behavior of deputies

$$simobj_{votes}(e, G_i, G_j) = \begin{cases} 1 & \text{if } \theta(e, G_i) = \theta(e, G_j) \\ 0 & \text{else} \end{cases} \quad (4)$$

For ratings ranging from 1 to 5, the similarity *simobj* is defined by how much the scores given by the two aggregates are close:

$$simobj_{review}(e, G_i, G_j) = 1 - \frac{1}{4}|\theta(e, G_i) - \theta(e, G_j)| \quad (5)$$

To discover interpretable patterns (c, g', g''), we define the two following quality measures $\varphi_{consent}$, $\varphi_{dissent}$ by relying on the defined similarities. $\varphi_{consent}$ makes it possible to consider a pattern as *"interesting"* if there is an important strengthening of similarities between individuals corresponding to g' and individuals corresponding to g'' for the context c. $\varphi_{dissent}$ aims to assess the weakening of similarities between individuals. We assume that the attributes $L \subseteq \mathcal{A}_U$ used to build partitions of individuals are given:

- $\varphi_{consent}(c, g', g'') = \frac{\sum_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})} max(sim(E_c,i,j) - sim(E_*,i,j), 0)}{|\gamma_L(U_{g'})|.|\gamma_L(U_{g''})|}$
- $\varphi_{dissent}(c, g', g'') = \frac{\sum_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})} max(sim(E_*,i,j) - sim(E_c,i,j), 0)}{|\gamma_L(U_{g'})|.|\gamma_L(U_{g''})|}$

3.4 Upper bounds on quality measures

To early discard unpromising descriptions, we follow a branch-and-bound approach in which an upper bound on the quality measure φ is computed for a candidate description. We first define a generic upper bound UB_{sim} and a lower bound LB_{sim} on *sim*. Given a threshold σ_E that fix the minimum threshold on objects subgroup size, G_i and G_j two aggregates of individuals, we have:

- $LB^1_{sim}(E, G_i, G_j) = max\left(\frac{\sigma_E - |E|(1 - sim(E,G_i,G_j))}{\sigma_E}, 0\right)$
- $LB^2_{sim}(E, G_i, G_j) = \frac{1}{\sigma_E} smallest(\{simobj(e, G_i, G_j) \mid e \in E\}, \sigma_E)$
- $UB^1_{sim}(E, G_i, G_2) = min\left(\frac{|E| * sim(E,G_i,G_j)}{\sigma_E}, 1\right)$
- $UB^2_{sim}(E, G_i, G_j) = \frac{1}{\sigma_E} largest(\{simobj(e, G_i, G_j) \mid e \in E\}, \sigma_E)$

where $smallest(S, n)$ (resp. $largest(S, n)$) computes the sum of the n *minimum* (resp. *maximum*) of given set S of real values. LB^1_{sim} (resp. UB^1_{sim}) is equivalent to LB^2_{sim} (resp. UB^2_{sim}) when *simobj* gives binary results such as $simobj_{votes}$.

Given a description (c, g', g''), we define the following upper bounds[6] on the quality measure of every specialization d of c $(\forall d \mid c \sqsubseteq d)$:

$$\varphi_{consent}(d, g', g'') \leq UB_{consent}(c, g', g'') \wedge \varphi_{dissent}(d, g', g'') \leq UB_{dissent}(c, g', g'')$$

- $UB_{consent}(c, g', g'') = \frac{\sum_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})} max(UB_{sim}(E_c,i,j) - sim(E_*,i,j), 0)}{|\gamma_L(U_{g'})|.|\gamma_L(U_{g''})|}$
- $UB_{dissent}(c, g', g'') = \frac{\sum_{(i,j) \in \gamma_L(U_{g'}) \times \gamma_L(U_{g''})} max(sim(E_*,i,j) - LB_{sim}(E_c,i,j), 0)}{|\gamma_L(U_{g'})|.|\gamma_L(U_{g''})|}$

where $UB_{consent}$ (resp. $UB_{dissent}$) corresponds to $\varphi_{consent}$ (resp. $\varphi_{dissent}$).

[6] Proofs of upperbounds are available in goo.gl/viQxhi

3.5 Algorithms

Algorithm 1 called *EnumCC* (*Enumerate Closed Candidates*) describes the exploration of the search space over a collection of objects $\mathcal{G}$ defined by the attributes $\mathcal{A}_\mathcal{G} = \{a_1, ..., a_n\}$. *EnumCC* enumerates the *closed descriptions* c that verify the constraint $\sigma_\mathcal{G}$ on the size of its corresponding subgroup starting from a description d. Given a description d, *EnumCC* computes its corresponding subgroup S_c, if its size exceeds the threshold, the closure c of d is computed and the linear order between them is verified. If so c is returned as a valid candidate. The algorithm then generates the neighbors by refining the attributes $\{a_f, ..., a_n\}$. The flag f determines the attribute that was refined to generate the description d. Finally, a recursive call is done to explore the lattice structure formed by d in a DFS fashion. The parameter *cnt* is a Boolean that allows to prune the search space based on the computation of the upper bound on the quality of a candidate description. *EnumCC* is depicted as a generator.

Algorithm 1: $EnumCC(\mathcal{G}, d, \sigma_\mathcal{G}, f, cnt)$

$S_c \leftarrow d^\square$
if $|S_c| \geq \sigma_\mathcal{G}$ **then**
  $c \leftarrow S_c^\square$
  if $d \preccurlyeq_f c$ **then**
    $cnt\_c \leftarrow copy(cnt)$
    yield (c, S_c, cnt_c) ; `// yield the results and wait for the next call`
    if cnt_c **then**
      foreach $j \in [f, n]$ **do**
        foreach $ngh \in \eta_j(c)$ **do**
          foreach $(c_{ngh}, S_{ngh}, cnt_{ngh}) \in EnumCC(S_c, ngh, \sigma_\mathcal{G}, j, cnt\_c)$ **do**
            yield $(c_{ngh}, S_{ngh}, cnt_{ngh})$

Algorithm 2: $DSC(E, U_1, U_2, L, \sigma_\mathcal{E}, \sigma_U, \sigma_\varphi, k)$

$\sigma_\varphi^{current} \leftarrow \sigma_\varphi$
$topk \leftarrow [\,]$
foreach $(g', U_{g'}, cont_{g'}) \in EnumCC(U_1, *, \sigma_{U_1}, 0, True)$ **do**
  foreach $(g'', U_{g''}, cont_{g''}) \in EnumCC(U_2, *, \sigma_{U_2}, 0, True)$ **do**
    foreach $(c, E_c, cont_c) \in EnumCC(\mathcal{E}, *, \sigma_\mathcal{E}, 0, True)$ **do**
      $UB \leftarrow UB_{dissent}(c, g', g'')$; `// resp.` $UB_{consent}$
      if $UB < \sigma_\varphi^{current}$ **then**
        $cont_c \leftarrow False$
      else
        $quality \leftarrow \varphi_{dissent}(c, g', g'')$; `// resp.` $\varphi_{consent}$
        if $quality \geq \sigma_\varphi^{current}$ **then**
          $pattern \leftarrow (c, g', g'')$
          update $topk$ ***by*** $\langle pattern, quality \rangle$ ***limits by*** k
          if $|topk| = k$ **then**
            $\sigma_\varphi^{current} \leftarrow min\_quality(topk)$
output $topk$

Algorithm 2 depicts *DSC* method based on the use of the closure operator and a branch and bound exploration. It is related to the task of finding topk patterns with a minimum quality threshold σ_φ. The algorithm first generates the candidate pattern (c, g', g''), subsequently the upper bound of the candidate pattern is computed. If it does not exceed the threshold σ_φ, the search space is pruned. Otherwise the quality measure of the candidate is computed. If its quality exceeds the same threshold σ_φ then the *topk* set is updated. Subsequently, if the size of *topk* exceeds k, the worst pattern found w.r.t. φ is discarded and σ_φ is dynamically updated with the minimal quality of the current *topk* set. Note that E defines the objects on which individuals U_1 and U_2 (two subsets of U) gives outcomes. L determines the attributes on which the individuals are aggregated. Finally, σ_E, σ_U determines the thresholds of subgroups sizes of respectively E and U.

4 Empirical study

In this section we report on both quantitative and qualitative experiments over the implemented algorithms. The algorithms were implemented in Python. The experiments were carried on an Intel Core i7-6700HQ 2.60 GHz machine with 16 GB RAM and were run by PyPy 5.4.1 For reproducibility purpose, the source code and the data are made available in our companion page[7]. These experiments aim to answer the following questions: ***Q1*** - Is the closure over an HMT attribute more effective than mining closed itemsets? ***Q2*** - Are the closing operator and the tights upper bounds effective and efficient? ***Q3*** - Does our algorithm scale w.r.t. different parameters? ***Q4*** - Does *DSC* provide actionable patterns?

Experiments were carried out on two real-world datasets: a movie review dataset *Movielens*[8] and the European parliament dataset *EPD*[9]. The main characteristics of these datasets are reported in Table 1. In Movielens, 18 movie genres are organized through a flat hierarchy.

4.1 Performance study

Q1 **-** We aim to study the performance of the closure operator in the presence of an HMT attribute. To this end, we compare it against the closure over item-

Characteristics	Movielens	EPD
#objects	1.681 movies	2.471 Ballots
#individuals	943	778
#outcomes	99.991	1.639.199
$\mathcal{A}_E$	1 HMT (18 tags), 1 Numeric	1 HMT (311 tags), 1 Numeric, 1 Nominal
$\mathcal{A}_U$	3 Nominal	3 Nominal

Table 1: Characteristics of the datasets

[7]https://github.com/Adnene93/DiscoveringSimilarityChanges
[8]https://grouplens.org/datasets/movielens/100k/
[9]http://parltrack.euwiki.org/

sets (i.e., scaling) as illustrated in Fig. 2. A tree of tags is characterized by its *height* and its branching factor (*k-ary tree*). A dataset of multi-tagged object is described by the maximum number of tags (*maxtags*) that an object can have and also its size. Fig. 4 reports the runtime and the number of explored candidates of the two closure operators when varying the branching factor, the tree height, the number of tags and the dataset size. For these experiments, we set the default values of these characteristics respectively to: 5, 3, 3 (*hierarchy of 125 tags*) and 5000 objects. HMTClosure exploits the structure of the tree and avoids exploring semantically equivalent descriptions (i.e : $\{3, 3.10.05\}$ *is semantically equivalent to* $\{3.10.05\}$) whereas *ISClosure* explores them. In all configurations, HMTClosure outperforms ISClosure on both the execution time and the number of explored candidates. These experiments demonstrate that taking into account hierarchical relations makes the closure operator more efficient and effective.

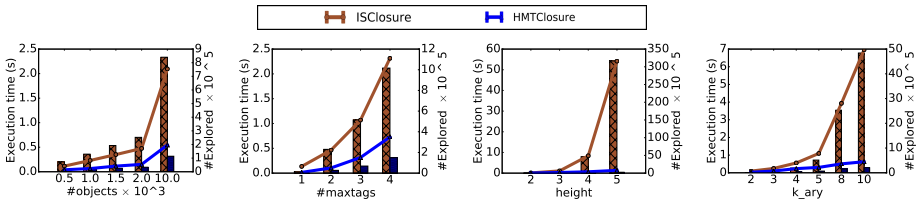


Fig. 4: Behavior of enumeration algorithms considering two closure operators for HMT attributes w.r.t. the number of objects, the height of the hierarchy, the number of tags and the branching factor which are set by default to respectively $5, 3, 3, 5000$.

Q2 **-** A baseline algorithm is obtained by deactivating the pruning techniques based on upper-bound and the closure operators . Thus, the baseline only pushes monotonic constraints. We compare *DSC* with the baseline and also with *closed* which is *DSC* without an upper bound computation on both Movielens and *EPD*. Notice that in *EPD* $UB^1_{dissent}$ and $UB^2_{dissent}$ are equivalent for the considered similarity. Therefore, we only report $UB^1_{dissent}$. We interrupt a method if its execution time exceeds one hour.

Figures 5 and 6 report the behavior (i.e., execution time and number of explored candidates) of the different methods when varying the characteristics of the datasets Movielens and *EPD*. Obviously, these experiments give evidence that each of the different optimizations of *DSC* are effective. For Movielens dataset, *DSC* is the most efficient when using $UB^2_{dissent}$ instead of $UB^1_{dissent}$. Indeed, $UB^2_{dissent}$ is more costly to compute than $UB^1_{dissent}$ but much tighter. The differences between the baseline and *DSC* are much more important on *EPD* because the HMT attribute is more complex than in Movielens. The experiments also demonstrate that the number of attributes used in a description of an object or a user heavily impacts the performance of the method as it increases the size of the search space.

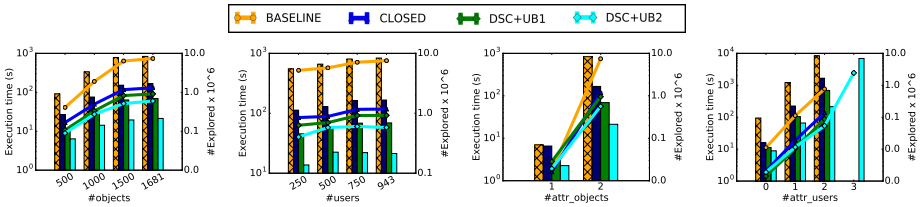


Fig. 5: Effectivness of *DSC* (*Top-5*) according to **Movielens** dataset characteristics which are set by default to $|E| = 1681$, $|U| = 943$, $\#attr_{objects} = 2$, $\#attr_{individuals} = 2$. The default thresholds are $\sigma_E = \sigma_U = 5$, $\sigma_\varphi = 0$, $|L| = 1$.

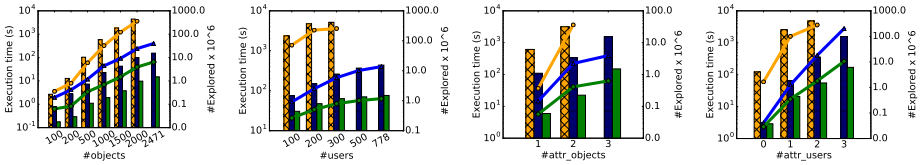


Fig. 6: Effectivness of *DSC* (*Top-5*) according to **EPD** dataset characteristics which are set by default to $|E| = 2471$, $|U| = 778$, $\#attr_{objects} = 3$, $\#attr_{individuals} = 2$. The default thresholds are $\sigma_E = \sigma_U = 15$, $\sigma_\varphi = 0$, $|L| = 1$.

Q3 **-** Fig. 7 reports the behavior of *DSC* on *EPD* when varying the input parameters (i.e., the minimum thresholds σ_E and σ_U and the quality measure). Obviously, when the thresholds increase (i.e. become more stringent) the number of explored patterns and thus the execution time decrease. Nevertheless, we observe that when decreasing σ_E, *DSC* remains efficient thanks to its pruning abilities based on upper-bound computations and closure operators. The execution time increases in line with the number of dimensions $|L|$ on which are computed the group of individuals while the number of explored descriptions remains roughly the same. Indeed, the computation of the model is more costly. Finally, a greater σ_φ leads to an important reduction of the number of explored candidates and therefore a better execution time. This demonstrates the effectiveness of the pruning properties implemented in *DSC*. Even if the two quality measures behave similarly, $\varphi_{consent}$ performs slightly better than $\varphi_{dissent}$ as by default the relation between the parliament's deputies w.r.t. their voting behavior is rather consensual.

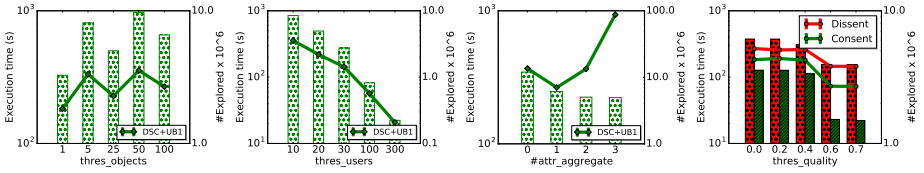


Fig. 7: Effectivness of *DSC* (top-5) over **EPD** according to constraints thresholds and quality measures. The default thresholds are $\sigma_E = \sigma_U = 15$, $\sigma_\varphi = 0$, $|L| = 1$

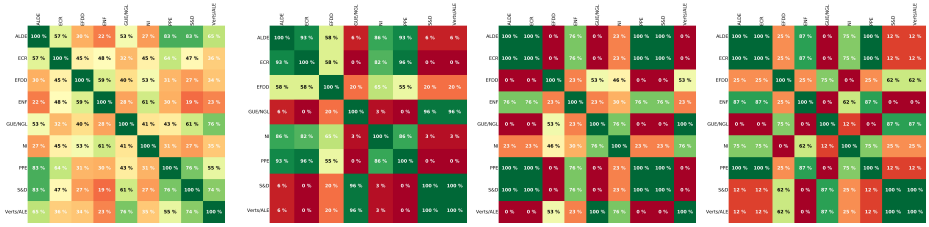
4.2 Qualitative results (*Q4*)

Table 2 describes some patterns found by *DSC* when looking for contexts that weaken the pairwise agreement between collections of reviewers identified by gender and age group in Movielens. For instance, middle-aged females tend to be in discord with their peer males for 1998 comedy movies (*13 movies, e.g.: The Wedding Singer*) in the best pattern. This can be observed by a significant decrease of similarity (of 35%) between the two aggregates from 86% to 51%. The diversification is done over the top-100 patterns. Two patterns are considered similar if one cover more than 50% of the reviewed objects contained in the second.

i	context (c)	g'	g''	$\lvert E_c \rvert$	$\lvert U_{g'} \rvert$	$\lvert U_{g''} \rvert$	φ
1	[['Comedy'], [1998, 1998]]	[['middle-age'], ['F']]	[['middle-age'], ['M']]	13	119	228	0.35
2	[['Horror', 'Comedy'], [1992, 1996]]	[['middle-age'], ['F']]	[['old'], ['M']]	9	119	45	0.31
3	[['Drama'], [1949, 1950]]	[['middle-age'], ['F', 'M']]	[['young'], ['F', 'M']]	5	347	508	0.3
4	[['Romance'], [1998, 1998]]	[['middle-age'], ['M']]	[['young'], ['F']]	11	228	134	0.3

Table 2: Diversified *Top-4* patterns discovered over Movielens by grouping on *agegroups*

i	context (c)	g'	g''	$\lvert E_c \rvert$	$\lvert U_{g'} \rvert$	$\lvert U_{g''} \rvert$	φ
1	[['3.40.16', '6.10.05', '6.20.02', '6.30'], [Feb. 2015, Feb. 2015]]	[['PPE']]	[['S&D']]	29	227	191	0.76
2	[['2.40', '3.30.03.04', '3.30.05', '3.30.06', '3.30.20', '3.30.25', '4.60.06'], [July. 2015, July. 2015]]	[['S&D']]	[['Verts/ALE']]	13	191	51	0.75
3	[['2.50.08', '2.80', '3.45.04'], [Feb.2016, Feb.2016]]	[['ALDE']]	[['S&D']]	8	75	191	0.71
4	[['6.40.04.02'], [Mar.2015, Mar.2015]]	[['GUE/NGL']]	[['Verts/ALE']]	10	59	51	0.67



(*1*) (*2*) (*3*) (*4*)

Fig. 8: Diversified *Top-4* patterns over EPD by grouping over political groups. (1) determine the usual pairwise observed between political groups and (2,3 & 4) illutstrate the heatmaps corresponding to the best 3 pattern found in top-k table

Fig. 8. reports the patterns discovered suggesting flash points (particular contexts that lead European groups to important similarities weakening). These patterns allow us to explicit the differences between groups that usually share the same political line. For example, while PPE and S&D vote mostly the same (76% of the cases), the top pattern (1) uncovers the ballots (contextualized by their themes - *such as 3.40.16 Raw materials and 6.10.05 Peace preservation* - and their time period - Feb. 2015) on where the two groups strongly diverge. This is witnessed by a decrease of pairwise agreement from 76% to 0%. The heatmaps illustrated in Fig. 8. depict the overall pairwise agreement changes observed for the pattern (1). Such results can provide insights for both political analysts and journalists, where the analytic tool provided by *DSC* allows to help discover ideological idiosyncrasies when comparing deputies against their peers, determining red lines between political groups or exhibiting contexts where nations deputies coalesce against others in critical subjects.

5 Related work

The problem of discovering exceptional subgroups based on the definition of a complex target model has been widely investigated in the recent years [17, 21, 8, 7, 18, 13]. Interestingly, de Sá et al. [6] use a similar matrix model to support the discovery of subgroups of individuals whose preference relation between ranked objects deviates from the norm. However, in the so-called exceptional preference mining, the dimensions of the model are fixed, i.e., the quality measure takes into account all objects and not dynamically a subset as in DSC. Dynamic EMM (i.e., EMM with a non-fixed model) has been recently investigated for different aims. Bosc et al. [4] propose a method to handle multi-label data where the number of labels per objects is much lower than the total number of labels which prevent the use of usual EMM model. Other dynamic EMM approaches aim to discover exceptional attributed sub-graphs [13, 3].

Thanks to open data policy, the analysis of political data has received much attention in the past decade. Most of them use basic data mining techniques. For instance, [11] uses clustering and PCA to identify cohesion blocs and dissimilarity blocs of voters within the US senate. Similar work was done on the Finnish [20] and the Italian [1] parliaments. An extensive tool was provided by [9] and applied to Swiss government datasets to detect opinion change of parliamentarians based on their expressed opinions before elections and votes cast afterwards.

Rating analysis has also received a wide interest in the last decade. In [5], the authors tackle the problem of rating interpretation by providing two methods (DEM, DIM). While the first one aims to discover groups of users that substantially agree for a given set of items, the second addresses the discovery of groups with an apparent inner discord. These two methods can be formalized as EMM instances with either a quality measure that assesses the average ratings of the identified subgroups or the average balance between positive and negative rating. While these methods consider a mono-objective measure (*rating average*), a similar work has been done to tackle multi-objective groups identification in [19]. It addresses a more complex statistical measure (rating distribution) and additionally coverage and diversity issues. In [2], the authors aim at using rating maps to identify subsets of reviews such that the distribution of rates observed is similar to the desired distributions.

6 Conclusion

In this paper, we introduced the novel problem of subjective exceptional pairwise behavior discovery in rating or vote data, rooted in the SD/EMM framework. We defined a branch-and-bound algorithm that exploits tight upper bounds and some closure operators to efficiently and effectively discover subgroups of interest. Experiments show that both quantitative and qualitative results are very satisfactory. We believe that this work opens new directions for future work. For example, the interactive discovery of exceptional pairwise behavior would make it possible to take into account prior knowledge. Such an exploration must be supported by instant mining algorithms.

Aknowledgement

This work has been partially supported by the project *ContentCheck* **ANR-15-CE23-0025** funded by the French National Research Agency.

References

1. A. Amelio and C. Pizzuti. Analyzing voting behavior in italian parliament: Group cohesion and evolution. In *ASONAM*, pages 140–146. IEEE, 2012.
2. S. Amer-Yahia, S. Kleisarchaki, N. K. Kolloju, L. V. Lakshmanan, and R. H. Zamar. Exploring rated datasets with rating maps. In *WWW*, 2017.
3. A. A. Bendimerad, M. Plantevit, and C. Robardet. Unsupervised exceptional attributed sub-graph mining in urban data. In *ICDM*, pages 21–30, 2016.
4. G. Bosc, J. Golebiowski, M. Bensafi, C. Robardet, M. Plantevit, J. Boulicaut, and M. Kaytoue. Local subgroup discovery for eliciting and understanding new structure-odor relationships. In *DS*, pages 19–34, 2016.
5. M. Das, S. Amer-Yahia, G. Das, and C. Yu. Mri: Meaningful interpretations of collaborative ratings. *PVLDB*, 4(11):1063–1074, 2011.
6. C. R. de Sá, W. Duivesteijn, C. Soares, and A. Knobbe. Exceptional preferences mining. In *DS*, pages 3–18. Springer, 2016.
7. W. Duivesteijn, A. J. Feelders, and A. Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016.
8. W. Duivesteijn, A. J. Knobbe, A. Feelders, and M. van Leeuwen. Subgroup discovery meets bayesian networks - an exceptional model mining approach. ICDM , 2010.
9. V. Etter, J. Herzen, M. Grossglauser, and P. Thiran. Mining democracy.ACM,2014.
10. B. Ganter and S. Kuznetsov. Pattern structures and their projections. ICCS,2001.
11. A. Jakulin and W. Buntine. Analyzing the us senate in 2003: Similarities, networks, clusters and blocs. 2004.
12. M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989–2001, 2011.
13. M. Kaytoue, M. Plantevit, A. Zimmermann, A. Bendimerad, and C. Robardet. Exceptional contextual subgraph mining. *Machine Learning*, pages 1–41, 2017.
14. P. Kralj Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, 10, 2009.
15. S. O. Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *PKDD*, pages 384–391. Springer, 1999.
16. S. Lacy and T. Rosenstiel. Defining and measuring quality journalism, 2015.
17. D. Leman, A. Feelders, and A. J. Knobbe. Exceptional model mining. In *ECML/PKDD*, 2008.
18. F. Lemmerich, M. Becker, and M. Atzmueller. Generic pattern trees for exhaustive exceptional model mining. In *ECML/PKDD*, pages 277–292, 2012.
19. B. Omidvar-Tehrani, S. Amer-Yahia, P.-F. Dutot, and D. Trystram. Multi-objective group discovery on the social web. In *ECMLPKDD*, 2016.
20. A. Pajala, A. Jakulin, and W. Buntine. Parliamentary group and individual voting behaviour in the finnish parliament in year 2003: a group cohesion and voting similarity analysis. 2004.
21. M. van Leeuwen and A. J. Knobbe. Diverse subgroup set discovery. *Data Min. Knowl. Discov.*, 25(2):208–242, 2012.
22. S. Wrobel. An algorithm for multi-relational discovery of subgroups. PKDD, 1997.