HAL
open science

Identifying exceptional (dis)agreement between groups - Technical Report

Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre, Marc Plantevit

To cite this version:

Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre, Marc Plantevit. Identifying exceptional (dis)agreement between groups - Technical Report. [Research Report] LIRIS UMR CNRS 5205. 2019. hal-02018813v2

HAL Id: hal-02018813

https://hal.science/hal-02018813v2

Submitted on 1 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Identifying exceptional (dis)agreement between groups

Technical Report

Adnene Belfodil · Sylvie Cazalens · Philippe Lamarre · Marc Plantevit

Received: date / Accepted: date

Abstract Under the term behavioral data, we consider any type of data featuring individuals performing observable actions on entities. For instance, voting data depicts parliamentarians who express their votes w.r.t. procedures. In this work, we address the problem of discovering exceptional (dis)agreement patterns in such data, i.e., groups of individuals that exhibit an unexpected mutual agreement under specific contexts compared to what is observed in overall terms. To tackle this problem, we design a generic approach, rooted in the Subgroup Discovery/Exceptional Model Mining framework, which enables the discovery of such patterns in two different ways. A branch-and-bound algorithm ensures an efficient exhaustive search of the underlying search space by leveraging closure operators and optimistic estimates on the interestingness measures. A second algorithm abandons the completeness by using a direct sampling paradigm which is provides an alternative when an exhaustive search approach becomes unfeasible. To illustrate the usefulness of discovering exceptional (dis)agreement patterns, we report a comprehensive experimental study on four real-world datasets relevant to three different application domains: political analysis, rating data analysis and healthcare surveillance.

Keywords Supervised Pattern Mining, Subgroup Discovery, Exceptional Model Mining, Behavioral Data Analysis.

Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre
INSA Lyon, CNRS, LIRIS UMR5205, F-69621 France

Marc Plantevit
Université Lyon 1, CNRS, LIRIS UMR5205, F-69622 France

E-mail: firstname.surname@liris.cnrs.fr

1 Introduction

Consider data describing the organization and votes of a parliamentary institution (e.g., European Parliament[1], US Congress[2]). Such datasets record the activity of each member in voting sessions held in the parliament, as well as information on the parliamentarians and the sessions. Table 1 provides an example. It reports the outcomes of European parliament members (MEPs) on legislative procedures. These procedures are described by attributes such as themes and dates. MEPs are characterized by their country, parliamentary group and age. The general trends are well known, and easy to check on these data with basics queries on data. For instance, the Franco-German axis is reflected by consensual votes between parliamentarians of both countries as well as the usual opposition between right wing and left wing. An analyst (e.g., a data journalist) is aware of these political positions and expects deeper insights. To this end, it is of major interest to discover groups of individuals that exhibit an unexpected mutual agreement (or disagreement) under specific conditions (contexts). For example, from Table 1, an exceptional (dis)agreement pattern is $p = (c = \langle \text{themes} = 7.30 \text{ Judicial Coop} \rangle, u_1 = \langle \text{country} = \text{France} \rangle, u_2 = \langle \text{country} = \text{Germany} \rangle)$, which reads: "in overall terms, while German and French parliamentarians are in agreement (comparing majority votes leads to 66%[3] of equal votes), an unexpected strong disagreement between the two groups is observed for *Judicial Cooperation related* legislative procedures (the respective majorities voted the same way only 33% of the time in the corresponding voting sessions, i.e. $\{e_3, e_5, e_6\}$)".

ide	themes	date
e_1	1.20 Citizen's rights	20/04/16
e_2	2.10 Free Movement of goods	16/05/16
e_3	1.20 Citizen's rights; 7.30 Judicial Coop	04/06/16
e_4	7 Security and Justice	11/06/16
e_5	7.30 Judicial Coop	03/07/16
e_6	7.30 Judicial Coop	29/07/16

(a) Entities (Voting sessions)

idi	country	group	age
i_1	France	S&D	26
i_2	France	PPE	30
i_3	Germany	S&D	40
i_4	Germany	ALDE	45

(b) Individuals (Parliamentarians)

idi	ide	outcome
i_1	e_1	For
i_1	e_2	Against
i_1	e_5	For
i_1	e_6	Against
i_2	e_1	For
i_2	e_3	Against
i_2	e_4	For
i_2	e_5	For
i_3	e_1	For
i_3	e_2	Against
i_3	e_3	For
i_3	e_5	Against
i_4	e_1	For
i_4	e_4	For
i_4	e_6	Against

(c) Outcomes

Table 1: Example of behavioral dataset - European Parliament Voting dataset

[1] http://parltrack.euwiki.org/

[2] https://voteview.com/data

[3] Since the majorities of $\langle u_1, u_2 \rangle$ voted respectively on $\{e_1, e_2, \underline{e_3}, e_4, \underline{e_5}, \underline{e_6}\}$ as follows: $\langle \text{For}, \text{For} \rangle, \langle \text{Against}, \text{Against} \rangle, \underline{\langle \textit{Against}, \textit{For} \rangle}, \langle \text{For}, \text{For} \rangle, \underline{\langle \textit{For}, \textit{Against} \rangle}, \underline{\langle \text{Against}, \text{Against} \rangle}$.

In this work, we aim to discover such exceptional (dis)agreement patterns not only in voting data but also in more generic data which involves individuals, entities and outcomes. Indeed, voting data are particular instances of a more general type of data that we call *behavioral data*. It regroups any type of data featuring individuals (e.g., parliamentarians, users in social networks, patients) performing observable actions (e.g., votes, ratings, intakes) on entities (e.g., procedures, movies, restaurants, medicines). From such datasets, we aim to discover exceptional (dis)agreement between groups of individuals on specific contexts. That is to say, an important difference between the groups' behaviors is observed compared to the usual context (i.e., the whole data). This could answer a large variety of questions. For instance, considering political data, an analyst may ask: *what are the controversial subjects in the European parliament in which groups or parliamentarians have divergent points of view?* In collaborative rating analysis, one may ask *what are the controversial items? And which groups are opposed?* In Healthcare surveillance, the analyst may want to know if some medicines are prescribed much more often for one group of individuals than another one.

The discovery of regions within the data that stand out with respect to a given target has been widely studied in data mining and machine learning communities under several names [43] (subgroup discovery [42,67], emerging patterns [19], contrast sets [7]). Subgroup Discovery (SD) is known as the most generic one as it is agnostic of the data and the pattern domain. For instance, subgroups can be defined by a conjunction of conditions on symbolic [46] or numeric attributes [32,6] as well as sequences [31]. Furthermore, the single target can be discrete or numeric [49]. Exceptional Model Mining (EMM) [48] extends SD by offering the possibility to handle complex targets [20,21,41,63]. However, no model in the EMM/SD framework makes it possible to investigate exceptional contextual (dis)agreement between groups. We made a first attempt to discover exceptional (dis)agreement patterns in [8]. However, the model proposed in [8] requires the specification of many non-intuitive parameters that may be source of misleading interpretation. In this paper, we strive to provide a simpler and more generic framework to analyze behavioral data.

Figure 1 gives an overview of the approach we devise to discover exceptional (dis)agreement between groups. At a high level of description, five steps are necessary to discover interesting (dis)agreement patterns. First, two groups of individuals (u_1, u_2) are selected by intents (1) . Then, their usual agreement on all their expressed outcomes is computed in step (2). All characterizable subsets of entities (contexts (c)) are then enumerated (3) and for each selected subset, the agreement between the two groups is measured (4) and compared to their usual agreement (5) to evaluate to what extent the mutual agreement changes conveyed by a (dis)agreement pattern (c, u_1, u_2). Eventually, all pairs of groups (at least conceptually) are confronted. The discovery of exceptional (dis)agreement patterns requires to explore (simultaneously) both the search space associated to the individuals and the search space related to the entities. Moreover, behavioral datasets may contain several types of attributes (e.g., numerical, categorical attributes potentially organized by a hierarchy), and

outcomes. This requires efficient enumeration strategies. Last but not least, different measures to capture agreement may be considered depending on the application domain. Accordingly, the proposed method must be generic.

The main contributions of this paper are threefold:

Problem formulation. We define the novel problem of discovering exceptional (dis)agreement between groups of individuals when considering a particular subset of outcomes compared to the whole set of outcomes. Much effort has been done to ensure a generic framework for behavioral data analysis.

Algorithms. We propose two algorithms to tackle the problem of discovering exceptional (dis)agreement patterns. DEBuNk[4] is a branch-and-bound algorithm that efficiently returns the complete set of patterns. It takes benefit from both closure operators and optimistic estimates. Quick-DEBuNk is an algorithm that samples the space of (dis)agreement patterns in order to support instant discovery of patterns.

Evaluation. We report an extensive empirical study on both synthetic and real-world datasets. Synthetic datasets with controlled ground truth allows to make some qualitative comparisons with some existing methods. It gives evidence that existing methods fail to discover inter-agreement patterns. The four real-world datasets are then used to demonstrate the efficiency and the effectiveness of our algorithms as well as the interest of the discovered patterns. Especially, we report three case-studies from different application domains: political analysis, rating data analysis and healthcare surveillance to demonstrate that our approach is generic. The analysis of political votes, which is done in collaboration with journalists, is also available online[5] [45].

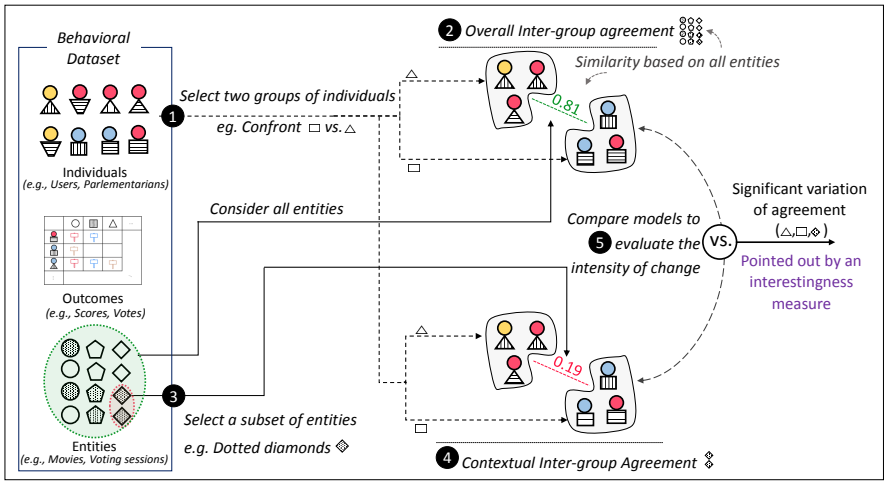


Fig. 1: Overview of the task of discovering exceptional (dis)agreement between groups

[4] DEBuNk stands for Discovering Exceptional inter-group Behavior patterNs

[5] http://contentcheck.liris.cnrs.fr

The rest of this paper is organized as follows. The problem formulation is given in Section 2. We present the *agreement* measure and how it is integrated into an interestingness measure to capture changes of inter-group agreement in Section 3. DEBuNk algorithm is presented in Section 4 while a pattern space sampling version, Quick-DEBuNk, is defined in Section 5. We report an empirical study in Section 6. Section 7 reviews the literature. Eventually, we conclude and discuss future directions of research in Section 8.

2 Problem Definition

We are interested in discovering exceptional (dis)agreement among groups in *Behavioral Datasets* defined as follows.

Definition 1 (*Behavioral Dataset*) A behavioral dataset $\langle G_I,\ G_E,\ O,\ o\rangle$ is defined by (i) a collection of Individuals G_I, (ii) a collection of Entities G_E, (iii) a domain of possible Outcomes O, and (iv) a function $o : G_I \times G_E \rightarrow O$ that gives the outcome of an individual i over an entity e, if applicable.

Elements from G_I (resp. G_E) have descriptive attributes, which set is denoted as $\mathcal{A}_I$ (resp. $\mathcal{A}_E$). Attributes $a \in \mathcal{A}_I$ (resp. $\mathcal{A}_E$) may be numerical or categorical. Furthermore, individuals or entities may be associated with a set of tags which are organized within a taxonomy. Such attributes are said to be *Hierarchical Multi-Tag* (HMT). For instance, in Table 1, deputies (i.e., individuals), described by their country (categorical), their political group (categorical) and their age (numerical), decide on some voting sessions outlined by a date (seen as a numerical attribute) and themes (an HMT attribute).

To describe sets of individuals and sets of entities, we define *group descriptions* and *contexts* respectively. Both are *descriptions* formalized by conjunctions of conditions on the values of the attributes, but we use these two different terms for ease of interpretation. For example, in Table 1, the context $c = \langle$ themes $\supseteq$ '7.30 Judicial Coop' $\rangle$ imposes the presence of '7.30 Judicial Coop' in the attribute 'themes' and identifies the set of entities $G_E^c = \{e_3, e_5, e_6\}$. Similarly, the group description $u = \langle$ group='S&D' $\rangle$ selects the set of individuals $G_I^u = \{i_1, i_3\}$. The set of all possible contexts (resp. group descriptions) is the *description space* denoted $\mathcal{D}_E$ (resp. $\mathcal{D}_I$).

Since we are interested in patterns highlighting exceptional (dis)agreement between two groups of individuals described by u_1 and u_2, in a context c compared to the overall context, the sought patterns are defined as follows:

Definition 2 ((Dis)Agreement Pattern) A (dis)agreement pattern is a triple $p = (c, u_1, u_2)$ where $c \in \mathcal{D}_E$ is a *context* and $(u_1, u_2) \in \mathcal{D}_I^2$ are two *group descriptions*. $\mathcal{P} = \mathcal{D}_E \times \mathcal{D}_I \times \mathcal{D}_I$ denotes the pattern space.

The extent fulfilling p is $ext(p) = (G_E^c, G_I^{u_1}, G_I^{u_2})$ with G_E^c the set of entities satisfying the conditions of context c and $G_I^{u_1}$ (resp. $G_I^{u_2}$), the set of individuals supporting the description u_1 (resp. u_2).

Several patterns may share the same extent. If $ext(p) = ext(p')$, patterns p and p' are said to be equivalent w.r.t their extension, denoted as $p \equiv_{G_E,G_I} p'$. Patterns are partially ordered in $\mathcal{P}$ by a *specialization* relationship. A description d_1 is a specialization of a description d_2, denoted $d_2 \sqsubseteq d_1$, iff $d_1 \Rightarrow d_2$. Consequently, a pattern p' is a specialization of a pattern p, denoted $p \sqsubseteq p'$, iff $c \sqsubseteq c'$, and $u_1 \sqsubseteq u'_1$, and $u_2 \sqsubseteq u'_2$. It follows that $ext(p') \subseteq ext(p)$ (i.e. $G_E^{c'} \subseteq G_E^c$ and $G_I^{u'_1} \subseteq G_I^{u_1}$ and $G_I^{u'_2} \subseteq G_I^{u_2}$).

A *quality measure* φ is required to assess the interestingness of a pattern. It assigns to each pattern p a positive real number. This value is computed considering exclusively the extent of the pattern. Therefore, two equivalent patterns have the same quality. Different quality measures are proposed in Sec. 3.

Our objective is to provide the user with a collection of patterns that captures exceptional (dis)agreement in a given behavioral dataset. A first intuitive idea is to provide all patterns of high quality, i.e. with a quality greater than a user-defined threshold σ_φ. This is of major importance, but considering the quality is not enough. Indeed, many different patterns may reach the same quality level just because they share the same extent. In such a case, we assume that the user would expect the system not to bother her with huge collections of patterns describing the same parts of the data. More interestingly, the system should provide her with the good generalizations, i.e., the patterns that are not a specialization of already found ones w.r.t. their extents. Additionally, some cardinality constraints can be added to avoid patterns of too small extent. Given two minimum support thresholds σ_E and σ_I, these constraints ensure, for a pattern $p = (c, u_1, u_2)$, that the size of the context extent (i.e. $|G_E^c| \geq \sigma_E$) and the size of both groups (i.e. $|G_I^{u_1}| \geq \sigma_I$ and $|G_I^{u_2}| \geq \sigma_I$) are large enough. Now, we introduce formally the core problem we tackle in this paper.

Problem Def. (*Discovering Exceptional (Dis)Agreement between Groups*).

Given a behavioral dataset $\langle G_I,\ G_E,\ O,\ o\rangle$, a quality measure φ, a quality threshold σ_φ and a set of cardinality constraints $\mathcal{C}$, the problem is to find the pattern set $P \subseteq \mathcal{P}$ such that the following conditions hold:

1. (*Validity*) $\forall p \in P\ :\ p$ *valid* that is p *satisfies* $\mathcal{C}$ and $\varphi(p) \geq \sigma_\varphi$.
2. (*Maximality*) $\forall p \in P\ \forall q \in \mathcal{P} : p \neq q$ and $q \equiv_{G_E,G_I} p \Rightarrow p \not\sqsubseteq q$
3. (*Completeness*) $\forall q \in \mathcal{P} \setminus P : q$ *valid* $\Rightarrow \exists p \in P$ s.t. $ext(q) \subseteq ext(p)$
4. (*Generality*) $\forall (p, q) \in P^2 : p \neq q \Rightarrow ext(p) \nsubseteq ext(q)$.

Condition (1) assures that the patterns in P are of high quality and satisfy the cardinality constraints. Condition (2) discard equivalent patterns by retaining only a unique representative which is the maximal one. Condition (3) ensures completeness and condition (4) ensures that only the most general patterns w.r.t. their extents are in P. In other words, the combination of conditions (3) and (4) guarantees that the solution P is minimal in terms of the number of patterns and that each valid pattern in $\mathcal{P}$ has a representative in P covering it. Considering the generic definition of the quality measure discussed here, this problem extends the former top-k problem addressed in [8]

by introducing conditions (3) and (4). That is, for a sufficiently large k, the method formerly provided in [8] solves this problem only limited to the two first conditions providing, hence, a solution with a much larger number of patterns.

3 Quality Measures and Inter-group Agreement Measurement

The previous section has already hinted at the fact that pattern interestingness is assessed with a quality measure φ whose generic definition is given. Here we show how such measure captures the deviation between the *contextual inter-group agreement* and the *usual inter-group agreement*.

3.1 Quality Measures

For any pattern $p = (c, u_1, u_2) \in \mathcal{P}$, we denote by p^* the pattern $(*, u_1, u_2)$ which involves all the entities. $\mathrm{IAS}(p^*)$ (resp. $\mathrm{IAS}(p)$) represents the usual (resp. contextual) inter-group agreement observed between the two groups u_1, u_2. In order to discover interpretable patterns, we define two quality measures that rely on $\mathrm{IAS}(p^*)$ and $\mathrm{IAS}(p)$.

- φ_{consent} assesses the strengthening of inter-group agreement in a context c:

$$\varphi_{\text{consent}}(p) = \max\left(\mathrm{IAS}(p) - \mathrm{IAS}(p^*), 0\right) \ .$$

- φ_{dissent} assesses the weakening of inter-group agreement in a context c:

$$\varphi_{\text{dissent}}(p) = \max\left(\mathrm{IAS}(p^*) - \mathrm{IAS}(p), 0\right) \ .$$

For instance, one can use φ_{consent} to answer: "*What are the contexts for which we observe more consensus between groups of individuals than usual?*".

3.2 Inter-group Agreement Similarity (IAS)

Several IAS measures can be designed according to the domain in which the data was measured (e.g., votes, ratings) and the user objectives. The evaluation of an IAS measure between two groups of individuals over a context requires the definition of two main operators: the *outcome aggregation operator* (θ) which computes an aggregated outcome of a group of individuals for a given entity, and a *similarity operator* (sim) which captures the similarity between two groups based on their aggregated outcomes over a single entity. These operators are defined in a generic way as following.

Definition 3 (Outcome Aggregation Operator θ) An aggregation operator is a function $\theta : 2^{G_I} \times G_E \to \mathbb{D}$ which transforms the outcomes of a group of individuals G_I^u over one entity $e \in G_E$ (i.e. $\{o^6(i, e) \mid i \in G_I^u\}$) into a value in a domain $\mathbb{D}$ (e.g. $\mathbb{R}$, *categorical values*).

[6] $o(i, e)$ returns the outcome expressed by an individual i to an entity e, if given.

Definition 4 (Similarity between aggregated outcomes sim**)** Function $\text{sim} : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$ assigns a real positive value $\text{sim}(x, y)$ to any couple of aggregated outcomes (x,y).

Based on these operators, we properly define IAS which assigns to each pattern $p = (c, u_1, u_2) \in \mathcal{P}$ a value $\text{IAS}(p) \in \mathbb{R}^+$. This similarity evaluates how the two groups of individuals (u_1, u_2) behave similarly given their outcomes w.r.t. the context c. In the scope of our study, we confine ourselves to IAS measures that can be expressed as weighted averages. The next definition, though limiting, is generic enough to handle a wide range of behavioral data.

Definition 5 (Inter-group Agreement Similarity Measure IAS**)** Let w be a function associating a weight to each triple from $(G_E \times 2^{G_I} \times 2^{G_I})$. The IAS of a pattern (c, u_1, u_2) ($\text{IAS} : \mathcal{P} \rightarrow \mathbb{R}^+$) is the weighted average of the similarities of the aggregated outcomes for each entity e supporting the context c.

$$\text{IAS}(c, u_1, u_2) = \frac{\sum_{e \in G_E^c} w(e, G_I^{u_1}, G_I^{u_2}) \times \text{sim}(\theta(G_I^{u_1}, e), \theta(G_I^{u_2}, e))}{\sum_{e \in G_E^c} w(e, G_I^{u_1}, G_I^{u_2})}$$

3.3 Examples of IAS Measures

By simply defining sim and θ, we present two instances of IAS measure that address two types of behavioral data with specific aims.

3.3.1 Behavioral Data With Numerical Outcomes

Collaborative Rating datasets are a classic example of behavioral data with numerical outcomes. Such datasets describe users who express numerical ratings belonging to some interval $O = [min, max]$ (e.g., 1 to 5 stars) over reviewees (*e.g. movies, places*). A simple and adapted measure for aggregating individual ratings over one entity is the weighted mean $\theta_{\text{wavg}} : 2^{G_I} \times G_E \rightarrow [min, max]$.

$$\theta_{\text{wavg}}(G_I^u, e) = \frac{1}{\sum_{i \in G_I^u} w(i)} \sum_{i \in G_I^u} w(i) \times o(i, e) \tag{1}$$

Weight $w(i)$ corresponds to the importance of ratings given by each individual $i \in G_I$. Such weight may depend on the confidence of the individual or the size of the sample population if fine granularity ratings (*rating of each individual*) are missing. If no weights are given, θ_{wavg} computes a simple average over ratings, denoted θ_{avg}. To measure agreement between two aggregated ratings over a single entity, we define $\text{sim}_{\text{rating}} : [min, max] \times [min, max] \rightarrow [0, 1]$.

$$\text{sim}_{\text{rating}}(x, y) = 1 - (\frac{|x - y|}{max - min}) \tag{2}$$

3.3.2 Behavioral Data with Categorical Outcomes

A typical example of such datasets are Roll Call Votes (RCVs)[7] datasets where voting members cast categorical votes. The outcome domain O is the set of all possible votes (e.g., $O = \{\text{For}, \text{Against}, \text{Abstain}\}$). To aggregate categorical outcomes we use the majority vote[8] θ_{majority}. We adapt its definition to handle potential ties (i.e., non unique majority vote). Hence, $\theta_{\text{majority}} : 2^{G_I} \times G_E \to 2^O$ returns all the outcomes that received the majority of votes.

$$\begin{aligned} \theta_{\text{majority}}(G_I^u, e) &= \{v \in O \ : \ v = \underset{z \in O}{\text{argmax}} \ \#\text{votes}(z, G_I^u, e)\} \\ \text{with } \#\text{votes}(z, G_I^u, e) &= |\{(i, e) : i \in G_I^u \ \wedge \ o(i, e) = z\}| \end{aligned} \tag{3}$$

We use a Jacquard index to assess the similarity between two majority votes x and y. Hence, $\text{sim}_{\text{voting}} \ : \ 2^O \times 2^O \to [0, 1]$ is defined as follows.

$$\text{sim}_{\text{voting}}(x, y) = \frac{|x \cap y|}{|x \cup y|} \ . \tag{4}$$

3.4 Discussion

We introduced above two simple similarity measures that can be used as part of the IAS measure to assess how similar two groups of individuals are. More sophisticated measures can be considered. For instance, in behavioral datasets with categorical outcomes, one can define an outcome aggregation measure which takes into account the empirical distribution of votes and then a similarity measure which builds up on a statistical distance (e.g. Kullback-Leibler divergence [17,39]). Such measures can also be used on behavioral datasets which involves numerical outcomes, for instance *Earth Mover Distance* measure was investigated in similarly structured dataset (rating dataset) in [3]. Several other measures can be relevant to analyze behavioral data with numerical outcomes depending on the aim of the study. In the empirical study, we investigate another similarity measure which relies on a ratio to highlight discrepancies between the medicine consumption rates of two subpopulations.

[7] Roll-Call vote is a voting system where the vote of each member is recorded, such as http://www.europarl.europa.eu (EU parliament) or https://voteview.com (US Congresses).

[8] The same measure is used by ***votewatch*** to observe the voting behaviors of groups of parliamentarians- http://www.votewatch.eu/blog/guide-to-votewatcheu/

4 A Branch and Bound Algorithm for Mining Relevant (Dis)Agreement Patterns

We address the design of an efficient algorithm for enumerating candidate patterns. First, we present how candidates are enumerated without redundancy by relying on a closure operator. Second, we detail the enumeration process, paying particular attention to the attributes domains depicted by a hierarchy. Next, we propose optimistic estimates for the quality measures. Eventually, these elements are used to define an efficient branch-and-bound algorithm which computes the complete set of relevant (dis)agreement patterns.

4.1 Enumerating Closed Patterns

Exploring the space of patterns (dis)agreement patterns $p = (c, u_1, u_2) \in \mathcal{D}_E \times \mathcal{D}_I \times \mathcal{D}_I$ is equivalent to explore $\mathcal{D}_E$ and $\mathcal{D}_I$ concurrently. Despite their differences, these two universes are formalized in a similar way, such that the exploration method presented below can be applied indifferently to any of them.

Let G be a collection of records (corresponding to entities G_E, or individuals G_I). We assume $\mathcal{A} = (a_1, a_2, ..., a_m)$ (corresponding to $\mathcal{A}_E$ or $\mathcal{A}_I$) to be the ordered list of attributes depicting its schema. Each attribute a_j has a domain of interpretation, noted $dom(a_j)$, which corresponds to all its possible values. In this section, we focus on attributes of two types: categorical attributes and numerical ones that cover a large number of practical cases. Nevertheless, our approach enables to introduce additional types without any modification of the general algorithm as illustrated in the section 4.2.

4.1.1 Description Language and Description Space Structure

A *description* $d = \langle r_1, r_2, ..., r_m \rangle$ is an ordered list of *conditions* on attributes' values. G^d denotes the *support* of d, i.e. the subset of G satisfying the conditions r_j of d. When a description d_1 is less specific than a description d_2, we note $d_1 \sqsubseteq d_2$. From a logical point of view $d_2 \Rightarrow d_1$. From this it follows that if $d_1 \sqsubseteq d_2$ then $G^{d_2} \subseteq G^{d_1}$. The form of a condition r_j depends on the type of its related attribute. A condition on a categorical (resp. numerical) attribute is an equality test (resp. a membership test) of the form $a_j = v$ (resp. $a_j \in [v..w]$) where v (and w) is (are) value(s) of the corresponding domain $dom(a_j)$.

Characterizing a given subset S of a collection G with a specific description, denoted $\delta(S)$, plays an important role in our process. $\delta(S)$ is built in a bottom-up fashion. Considering an element s of S, we obtain its characterization $\delta(s)$ by a composition of the characterizations of its values (one $\delta_j(s)$ for each considered attribute a_j of s). The description of S, $\delta(S)$, is then obtained by composing the characterizations of all of its elements. In the same spirit as in [26], we introduce some mapping functions $\delta_j(s)$ which provide a condition describing the value v of attribute a_j. This clearly depends on the type of a_j.

δ_j(s) is of the form $(a_j = v)$ (resp. $(a_j \in [v..v])$) when a_j is a categorical (resp. numerical) attribute. Applying such mappings to all the attributes leads to a complete description of s: $\delta(s) = \langle \delta_1(s), \delta_2(s), ..., \delta_m(s) \rangle$

Once the description δ(s) of a single record s is defined, the support G^d of description d can be expressed as $G^d = \{g \in G \mid d \sqsubseteq \delta(g)\}$.

It is then possible to obtain a description of S combining the descriptions of its elements. Each condition domain $\mathcal{D}_j$ related to an attribute a_j can be conceptualized as a *semi-lattice* with an *infimum operator* denoted $\sqcap$. Intuitively, such operator provides the tightest condition subsuming two conditions. The definition of the infimum operator depends on the type of the attribute. The infimum operator for a categorical (resp. numerical) attribute is denoted $\sqcap_c$ (resp. $\sqcap_n$). The infimum $\sqcap_n$ between interval conditions is defined following [40].

$$(a_j = v_1) \sqcap_c (a_j = v_2) = \begin{cases} a_j = v_1 & \text{if } v_1 = v_2 \\ \text{true}_{a_j} & \text{else} \end{cases}$$

$$(a_j \in [v_1..w_1]) \sqcap_n (a_j \in [v_2..w_2]) = a_j \in [min(v_1, v_2)..max(w_1, w_2)]$$

The Cartesian product of the m lattices related to attributes also forms a lattice [61] that we can also equip with an infimum operator $\sqcap$ which provides the *maximum common description* of two descriptions d, d':

$$d \sqcap d' = \langle r_1, ..., r_m \rangle \sqcap \langle r'_1, ..., r'_m \rangle = \langle r_1 \sqcap_1 r'_1, ..., r_m \sqcap_m r'_m \rangle \tag{5}$$

with $\sqcap_j$ is $\sqcap_n$ for numerical attributes and $\sqcap_c$ for categorical ones.

The *maximum common description* covering a subset S of a collection ($S \subseteq G$) is then obtained by: (where $\sqcap$ is commutative, associative and idempotent)

$$\delta(S) = \sqcap_{g \in S}(\delta(g)) \tag{6}$$

For example, in the dataset given in Table 1, individuals are described by both categorical and numerical attributes. The description $d = \langle \text{true}_{country}, \text{true}_{group}, age \in [20, 40] \rangle$ identifies the deputies whose age is between 20 and 40. Its support G_I^d is $\{i_1, i_2\}$. Interestingly, these two individuals share more elements in common w.r.t. their descriptors. Indeed, $d' = \delta(\{i_1, i_2\}) = \langle$ country='France', true_{group}, age$\in$ $[26, 30] \rangle$. Notice that (i) the two descriptions share the same support (G_I^d=$G_I^{d'}$) and (ii) $d \sqsubseteq \delta(G_I^d)$, as d is less specific than $\delta(\{i_1, i_2\}) = \delta(G_I^d)$.

Since we expect a support to be described only once in the result, the enumeration process has to prevent the enumeration of descriptions sharing the same support. More precisely, we expect to obtain the *closed description*, $clo(d)$, which is a description sharing its support with d ($G^d = G^{clo(d)}$), and which is also the most specific one (if G^d=$G^{d'}$, then $d' \sqsubseteq clo(d)$). This is not a constructive definition, but interestingly $clo(d) = \delta(G^d)$.This provides a method to obtain the desired description. This equality results from the fact that $\square^d$ and $\delta(\square)$ are a pair of Galois derivation operators between 2^G and $(\mathcal{D}, \sqsubseteq)$ providing as a consequence a closure operator: namely the composite operator $\delta(G^d)$. Proof and details of this equality is omitted as it is a well-known notion in pattern mining and formal concept analysis [27,26].

4.1.2 Description Enumeration

Our exploration algorithm is based on a depth-first enumeration method starting from the most general description $\langle \text{true}_{a_1}, \ldots, \text{true}_{a_m} \rangle$, shortly noted $*$. It proceeds by *atomic refinements* to progress, step by step, toward more specific descriptions. Intuitively, an *atomic refinement* of a description d produces a more specific description d' by reinforcing the condition of one attribute only. Furthermore, such refinement is minimal ($d \sqsubset d' \ \wedge \ \nexists e \in \mathcal{D} \ : \ d \sqsubset e \sqsubset d'$). Again, the atomic refinement of condition r_j takes different forms depending on the type of a_j. Since a condition over a numerical attribute is of the form $a_j \in [inf..sup]$[9], an atomic refinement corresponds to a left-minimal change with respect to existing values of the attribute a_j in G^d ($a_j \in [next_{G^d}(inf)..sup]$) or a right-minimal change ($a_j \in [inf..pred_{G^d}(sup)]$) on the interval bounds of the condition. An empty resulting interval means there is no possible refinement. Considering a categorical attribute, the atomic refinement of a condition true_{a_j} gives a condition of the form $a_j = v$, v being a value in $dom(a_j)$. Otherwise, a condition of the form $a_j = v$ does not admit any refinement. Whatever the type of the attribute is, $d \prec d'$ denotes the fact that d' results from an *atomic refinement* of condition d. This notion enables to easily progress from one description d to the next ones by relying on the following refining operator η.

$$\eta_j(d) = \{d' = \langle r'_1, \ldots, r'_m \rangle \in \mathcal{D} \ : \ r_j \prec r'_j \ and \ \forall k \in [1..j[\cup]j..m], r_k = r'_k\} \quad (7)$$

$$\eta(d) = \bigcup_{j \in [1..m]} \eta_j(d) \quad (8)$$

Algorithm 1 called EnumCC (*Closed Description ENumerator*), first introduced in [8], describes the exploration of the search space over a collection of records G defined by the attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$. Remind that such a collection may refer to either the collection of entities G_E or the collection of individuals G_I. *Closed Description ENumerator* enumerates once and only once all the *closed descriptions* that verify the support constraint $|G^c| \geq \sigma_G$ with σ_G a user defined threshold. EnumCC follows the same spirit as CloseByOne algorithm [44]. Note that EnumCC goes along the same lines of the enumeration algorithm *EnumCC* in our previous paper[8]. The main differences between the two algorithms reside on their implementation and the optimizations on which EnumCC relies. The implementation is available and maintained online[10].

Starting from a description d, EnumCC first computes its corresponding support G^d (line 1). If the size exceeds the threshold (line 2), the closure of d is computed (line 3). Subsequently, a *canonicity test* between *closure_d* and d is assessed (line 4). It allows to determine if a description after closure was already generated and to discard it, if appropriate, without addressing

[9] Note that true_{a_i} is equivalent to $a_j \in [MIN_{a_j}..MAX_{a_j}]$, MIN_{a_j}, (resp. MAX_{a_j}) being the minimal (resp. maximal) value of the domain of the attribute a_j.

[10] https://github.com/Adnene93/DEBuNk

the list of already generated closed descriptions requiring hence no additional storage. The canonicity test relies on an arbitrary order between attributes in $\mathcal{A}_G = \{a_1, a_2, ..., a_m\}$ indicating that, in the enumeration process, attribute conditions are refined following this arbitrary order. Let $d = \langle r_1, ..., r_f, ..., r_m \rangle$ a description resulting from the refinement of the fth condition of some preceding description, and $d' = \langle r'_1, ..., r'_f, ..., r'_m \rangle = clo(d)$ the closure of d. Following the arbitrary order between attributes, we expect for d', if it is the first tim e that it is encountered, that no condition before r'_f (i.e. $r'_1, ..., r'_{f-1}$) is refined; otherwise, $clo(d)$ was already generated after a refinement of preceding conditions and need thus to be discarded. The intuition behind the canonicity test being explained, a canonicity test rests essentially on a lectic order (cf. [27, p.66-68]) between d and its closure d' denoted $d \lessdot_f d'$ which is defined as follows: $d \lessdot_f d' \iff \forall i \in [1..f-1] \mid r_i = r'_i \ \wedge \ r_f \lessdot r'_f$. The latter condition, $r_f \lessdot r'_f$, corresponds to an analogous canonicity test between conditions and makes sense for multi-valued attributes types only (e.g. HMT in section 4.2). It does not need to be calculated for simple attributes (numerical, categorical). If the canonicity test is successful (line 4), *closure_d* is returned as a valid closed candidate (line 6). The algorithm then generates the neighbors by refining the attributes $\{a_f, ..., a_n\}$ continuing from d on the condition that *cnt_c* is not switched to *False* (lines 7-9). Flag f determines the index of the last attribute that was refined in the description d (operator η). Boolean cnt_c can be modified externally by some caller algorithm to prune the search space, for instance, when using optimistic estimates on the quality measures. Eventually, a recursive call is done to explore the sub search space related to d (lines 10-11). Hence, to enable the full exploration of search space $\mathcal{D}$ associated to a collection of records G, the algorithm is called with this initial parameters EnumCC$(G, *, \sigma, 1, true)$. Recall that $*$ is the description $\langle \text{true}_{a_1}, \text{true}_{a_2}, \ldots \text{true}_{a_m} \rangle$ having the complete collection G as its support.

Algorithm 1: EnumCC$(G,\ d,\ \sigma_G,\ f,\ cnt)$

Inputs : G is the collection of records depicted each by m descriptor attributes
d a description from $\mathcal{D}$, σ_G a support threshold,
$f \in [1, m]$ a refinement flag, `cnt` a boolean
Output: yields all closed descriptions, i.e. $clo[\mathcal{D}] = \{clo(d) \text{ s.t. } d \in \mathcal{D}\}$

```
1  if |G^d| ≥ σ then
2      closure_d ← δ(G^d)
3      if d ≪_f closure_d then
4          cnt_c ← copy(cnt)                ▷ cnt_c value can be modified by a caller algorithm
5          yield (closure_d, G^closure_d, cnt_c)   ▷ yield the results and wait for next call
6          if cnt_c then
7              foreach j ∈ [f, m] do
8                  foreach d' ∈ η_j(closure_d) do
9                      foreach (nc, G^nc, cnt_nc) ∈ EnumCC(G, d', σ_G, j, cnt_c) do
10                         yield (nc, G^nc, cnt_nc)
```

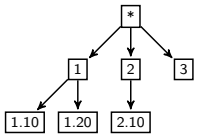
4.2 Hierarchical Multi-Tag Attribute (HMT)

Several votes and reviews datasets contain multi-tagged records whose tags are 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* 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*). For the sake of simplicity, we consider G a set of tagged records where each record g is described by a unique attribute *tags* which is a set of tags. Tags form a tree noted T. Fig. 2 depicts a dataset of tagged records.

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 enables 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 $t \prec u$ if $t < u$ and $\nexists e \in T \mid t < e < u$. Thus t is a parent of u denoted as $t = p(u)$.

A condition over an HMT attribute is assimilated as a membership in a set of tags $\{t_1, ..., t_n\}$. We denote the condition domain by $\mathcal{D}$ which is a subset of 2^T. Each object $g \in G$ is mapped by $\delta(g)$ to its corresponding condition in $\mathcal{D}$. Clearly, 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 generalizations $\uparrow t_1$ and $\uparrow t_2$ as shown in the flat representation in Fig. 2. Hence, an HMT restriction can be depicted by a rooted sub-tree of T and a record supports such restriction if it contains at least all tags of the sub-tree. It follows that, the partial order between two HMT conditions r, r' denoted $r \sqsubseteq r'$ (r' *more specific than* r) is valid if the sub-tree r covers the sub-tree r'. More formally, $r \sqsubseteq r'$ means $\forall t \in r \;\; \exists u \in r' \;\mid\; u \in \downarrow t$.

Two ways are possible to handle this attribute among the other attributes in the complex search space defined previously. One straightforward solution is to consider HMT attribute values as *itemsets* as depicted in the vector representation in Fig. 2. However, such a solution ignores the taxonomy T implying the enumeration of *chain descriptions*. For instance, a chain description $\{1, 1.20.01\}$ is regarded as a different description than $\{1.20.01\}$. This stems from the fact that items are unrelated from the viewpoints of itemsets solution. As a consequence, a larger search space is explored while determining the same closed descriptions. The same observation has been made for nu-



	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 tree of tags (left), a set of tagged items (middle) and its flat representation (right)

merical attributes [40]. To tackle such issue, we define an *HMT* description language. Similarly to the aforementioned attributes, we define the infimum operator between two conditions which computes the *maximum common sub-tree* covering a set of conditions. Let $r = \{t_1, ..., t_n\}$ and $r' = \{u_1, ..., u_m\}$ be two conditions of $\mathcal{D}$, we define $\sqcap$ as : $r \sqcap r' = max(\cup_{t \in r} \uparrow t \cap \cup_{u \in r'} \uparrow u)$ with $max : 2^T \rightarrow 2^T$ a function that maps a 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, $r \sqcap r'$ depicts the set of the maximum explicit or implicit tags shared by the two descriptions. For instance, if $c = \{1.10, 2\}$ and $d = \{1.20, 2.10\}$, $c \sqcap d = \{1, 2\}$. Fig. 3 illustrates the HMT infimum operator, where green and yellow tags represent respectively the explicit and the implicit tags of some given conditions in $\mathcal{D}$.

Similarly, a restriction r is an upper neighbor of r', that is $r \prec r'$ if either only one tag of r is refined in r' or a new tag is added in r' that shares a parent with a tag in r or with one of its ascendants. Formally:

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

Finally, we need to define the lectic order between two conjunctions of tags $r = \{t_1, ..., t_n\}$ and its closure $r' = \{u_1, ..., u_n, ..., u_m\}$ to assess the *canonicity test*. Given that r is generated after a refinement of the fth, the lectic order is defined as: $r \lessdot_f r' \Leftrightarrow \forall i \in [1..f-1] : t_i = u_i \wedge t_f \lessdot u_f$. The linear order $\lessdot$ between tags can be provided by the depth first search order on T. These concepts being defined, the mapping function δ and G^d can be extended easily to handle HMT among other attributes. Note that the HMT attributes support itemset attributes. This can be done simply by considering a flat tree T compound of all the items. Hence, HMT attributes can be seen as generalizations of itemset attributes, where implications between items are known.

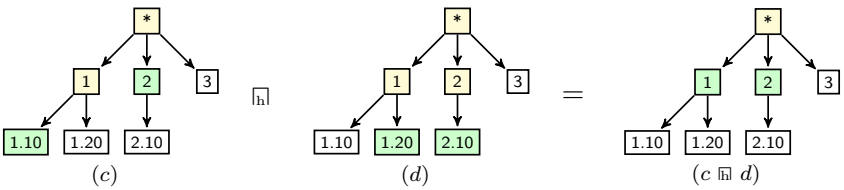


Fig. 3: Illustration of the infimum operator $\sqcap$

4.3 Optimistic Estimates on Quality Measures

The enumeration of closed patterns enables a non-redundant traversal of the search space without pruning based on the quality measure. We present some pruning properties based on bounds on $\varphi_{consent}$ and $\varphi_{dissent}$.

Let u_1, u_2 be two descriptions from $\mathcal{D}_I$ that respectively cover the two groups $G_I^{u_1}$, $G_I^{u_2}$. We consider optimistic estimates only with regards to the search space $\mathcal{D}_E$. We assume that u_1 and u_2 are instantiated a priori. Below, we give the definitions of an optimistic estimate [33].

Definition 6 (Optimistic Estimate) An optimistic estimate *oe* for a given quality measure φ is a function such that:

$$\forall \text{ contexts } c, d \in \mathcal{D}_E \,.\, c \sqsubseteq d \Rightarrow \varphi(d, u_1, u_2) \leq oe(c, u_1, u_2)$$

Tight optimistic estimates, defined in [33], offer more pruning abilities than simple optimistic estimate. Without loss of generality, we assume that the input domains of *oe* and φ are defined over both the pattern space $\mathcal{P}$ and over $2^E \times 2^I \times 2^I$. This is possible, since the quality measure only depends on extents.

Definition 7 (Tight Optimistic Estimate) An optimistic estimate *oe* is tight iff: $\forall c \in \mathcal{D}_E.\ \exists S \subseteq G_E^c \ :\ oe(G_E^c, G_I^{u_1}, G_I^{u_2}) = \varphi(S, G_I^{u_1}, G_I^{u_2})$.

Note that this does not require S to have a corresponding description in $\mathcal{D}_E$.

4.3.1 Lower Bound and Upper Bound for the IAS *Measure*

The two quality measures $\varphi_{consent}$ and $\varphi_{dissent}$ rely on the IAS measure. Since u_1 and u_2 are considered to be instantiated for optimistic estimates, we can rewrite the IAS measure for a context $c \in \mathcal{D}_E$ and its extent G_E^c:

$$\mathrm{IAS}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum\limits_{e \in G_E^c} w_e \times \alpha(e)}{\sum\limits_{e \in G_E^c} w_e} \; with \begin{cases} \alpha(e) = & \mathrm{sim}(\theta(G_I^{u_1}, e), \theta(G_I^{u_2}, e)) \\ w_e = & w(e, G_I^{u_1}, G_I^{u_2}) \end{cases} .$$

We can now define a lower bound LB and an upper bound UB for the IAS measure based on the following operators that are defined for any context $c \in \mathcal{D}_E$ and for $n \in \mathbb{N}$:

- $\mathrm{m}(G_E^c, n) = \mathrm{Lowest}_{e \in G_E^c}(\{w_e \times \alpha(e) \mid e \in G_E^c\}, n)$ returns the set of the n distinct records e from G_E^c having the lowest values of $w_e \times \alpha(e)$.
- $\mathrm{M}(G_E^c, n) = \mathrm{Highest}_{e \in G_E^c}(\{w_e \times \alpha(e) \mid e \in G_E^c\}, n)$ returns the set of the n distinct records e from G_E^c having the highest values of $w_e \times \alpha(e)$.
- $\mathrm{mw}(G_E^c, n) = \mathrm{Lowest}_{e \in G_E^c}(\{w_e \mid e \in G_E^c\}, n)$ returns the set of the n distinct records e from G_E^c having the lowest values of w_e.
- $\mathrm{Mw}(G_E^c, n) = \mathrm{Highest}_{e \in G_E^c}(\{w_e \mid e \in G_E^c\}, n)$ returns the set of the n distinct records e from G_E^c having the highest values of w_e.

Proposition 1 (Lower bound LB **for** IAS**)** *we define function* LB *as*

$$\mathrm{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum_{e \in \mathrm{m}(G_E^c, \sigma_E)} w_e \times \alpha(e)}{\sum_{e \in \mathrm{Mw}(G_E^c, \sigma_E)} w_e}$$

For any context c (corresponding to a subgroup G_E^c), LB *provides a lower bound for* IAS *w.r.t. contexts with σ_E a minimum context support threshold:*

$$\forall c, d \in \mathcal{D}_E.\ \ c \sqsubseteq d \Rightarrow \mathrm{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) \leq \mathrm{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2})$$

Proposition 2 (Upper bound UB **for** IAS**)** *we define function* UB *as*

$$\mathrm{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum_{e \in \mathrm{M}(G_E^c, \sigma_E)} w_e \times \alpha(e)}{\sum_{e \in \mathrm{mw}(G_E^c, \sigma_E)} w_e}$$

For any context c (corresponding to a subgroup G_E^c), UB *provides an upper bound for* IAS *w.r.t. contexts. i.e.*

$$\forall c, d \in \mathcal{D}_E.\ \ c \sqsubseteq d \Rightarrow \mathrm{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \mathrm{UB}(G_E^c, G_I^{u_1}, G_I^{u_2})$$

Now that both the lower bound and the upper bound of IAS are defined w.r.t. contexts, we define the optimistic estimates corresponding to φ_{consent} and φ_{dissent}. The proofs of the propositions are given in Appendix A.

4.3.2 Optimistic Estimates for Quality Measures

Proposition 3 (Optimistic estimate for φ_{consent} and φ_{dissent}) $\mathrm{oe}_{\text{consent}}$ *(resp.* $\mathrm{oe}_{\text{dissent}}$*) is an* ***optimistic estimate*** *for φ_{consent} (resp. φ_{dissent}) with:*

$$\mathrm{oe}_{\text{consent}}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \max(\mathrm{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) - \mathrm{IAS}(G_E, G_I^{u_1}, G_I^{u_2}), 0)$$

$$\mathrm{oe}_{\text{dissent}}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \max(\mathrm{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \mathrm{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}), 0)$$

The two defined optimistic estimates tight if the IAS measure is a simple average. i.e. all weights are equal to 1.

Proposition 4 *If $\forall(\{e\}, G_I^{u_1}, G_I^{u_2}) \subseteq G_E \times G_I \times G_I \ :\ w(e, G_I^{u_1}, G_I^{u_2}) = 1$,* $\mathrm{oe}_{\text{consent}}$ *(resp.* $\mathrm{oe}_{\text{dissent}}$*) is a* ***tight optimistic estimate*** *for φ_{consent} (resp. φ_{dissent}).*

4.4 Algorithm DEBuNk

DEBuNk is a Branch-and-Bound algorithm which returns the complete set of patterns as specified in the problem definition (Section 2). To this end, it takes benefit from the defined closure operator and optimistic estimates. Relying on algorithm EnumCC, DEBuNk starts by generating the couples of confronted groups of individuals that are large enough w.r.t. σ_I (lines 2-3). Then it computes the usual agreement observed between these two groups of individuals when considering all entities in G_E (line 4). Next, the context search space is explored to generate valid contexts c (line 5). Subsequently, the optimistic estimate *oe* is evaluated and the context sub search space is pruned if *oe* is lower than σ_φ (lines 7-8). Otherwise, the contextual inter-group agreement is computed and the quality measure is calculated (lines 9-10). If the pattern quality exceeds σ_φ then two scenarios are possible. Either the current pattern set P already contains a more general pattern, or it does not. In the former case, the pattern is discarded. In the latter, the new generated pattern is added to pattern set P while removing all previous generated patterns that are more specific than p w.r.t. extents (lines 11-14). Since the current pattern quality exceeds the threshold and all the remaining patterns in the current context sub search space are more specific than the current one, the sub search space is pruned (line 15). Eventually, if the quality measure is symmetric w.r.t. u_1 and u_2 (i.e. $\forall u_1, u_2 \in \mathcal{D}_I^2 \mid \varphi(c, u_1, u_2) = \varphi(c, u_2, u_1)$) there is no need to evaluate both qualities. As a consequence, it is possible to prune the sub search space of the couple descriptions (u_1, u_2) whenever $u_1 = u_2$ (lines 16-17).

Algorithm 2: DEBuNk($\langle G_I, G_E, O, o\rangle, \sigma_E, \sigma_I, \varphi, \sigma_\varphi$)

Inputs : $\langle G_I, G_E, O, o\rangle$ a behavioral dataset;
σ_E (resp. σ_I) minimum support threshold of a context (resp. group);
φ the quality measure; σ_φ quality threshold on the quality.
Output: P the set of exceptional (dis)agreement patterns.

1 $P \leftarrow \{\}$
2 **foreach** $(u_1, G_I^{u_1}, \mathit{cont}_{u_1}) \in$ EnumCC$(G_I, *, \sigma_I, 0, \text{True})$ **do**
3 **foreach** $(u_2, G_I^{u_2}, \mathit{cont}_{u_2}) \in$EnumCC$(G_I, *, \sigma_I, 0, \text{True})$ **do**
4 $\text{IAS}_{\text{ref}} \leftarrow \text{IAS}(*, u_1, u_2)$
5 **foreach** $(c, G_E, \mathit{cont}_c) \in$ EnumCC$(G_E^c, *, \sigma_E, 0, \text{True})$ **do**
6 **if** $\text{oe}_\varphi(c, u_1, u_2) < \sigma_\varphi$ **then**
7 $\texttt{cont}_c \leftarrow$ False ▷ Prune the unpromising sub-search space under c
8 **else**
9 $\text{IAS}_{\text{ref}} \leftarrow \text{IAS}(c, u_1, u_2)$
10 quality $\leftarrow \varphi(c, u_1, u_2)$ ▷ computed using IAS_{ref} and $\text{IAS}_{context}$
11 **if** quality $\geq \sigma_\varphi$ **then**
12 $p_{\text{new}} \leftarrow (c, u_1, u_2)$
13 **if** $\nexists p_{\text{old}} \in P \mid \text{ext}(p_{\text{new}}) \subseteq \text{ext}(p_{\text{old}})$ **then**
14 $P \leftarrow (P \cup p_{\text{new}}) \setminus \{p_{\text{old}} \in P \mid \text{ext}(p_{\text{old}}) \subseteq \text{ext}(p_{\text{new}})\}$
15 $\texttt{cont}_c \leftarrow$ False ▷ Prune the sub search space
16 **if** φ *is symmetric* ***and*** $u_1 = u_2$ **then**
17 ***break*** ▷ Prune the sub search space
18 ***return*** P

DEBuNk and DSC algorithm[8] differs on several levels. First, DEBuNk overcomes the limitations of lack of diversity of results provided by DSC which was designed to discover the top-k solutions. The present algorithm discards all patterns for which a generalization is already a solution. Second, DEBuNk handles a wider range of bounded quality measures (i.e. weighted mean IAS), in contrast to DSC algorithm which handles only a subset of these measures. Finally, DSC requires the prior definition of an aggregation level which makes it difficult to use and interpret. DEBuNk overcomes this issue by reducing the number of input parameters and integrating relevancy check between patterns. Hence, it requires less effort from the end-user both in terms of setting the parameters, and in terms of interpreting the quality of the resulting patterns.

5 Sampling (Dis)Agreement Patterns

DEBuNk returns the exact solutions according to the problem definition. Although relying on the enumeration of closed descriptions and pruning techniques, such an exploration may take a considerable time depending on the size and the complexity (i.e. attributes types) of the behavioral data. To address this concern, we devise a (dis)agreement pattern sampling approach called Quick-DEBuNk. It enables instant mining of (dis)agreement patterns by yielding approximate solutions that improve over time. Quick-DEBuNk relies on the two major steps to sample patterns, depicted in Fig. 4:

Frequency-Based Sampling(Step 1). A (dis)agreement pattern $p \in \mathcal{P}$ is drawn with a probability proportional to the size of its extent (i.e. $|\text{ext}(p = (c, u_1, u_2))| = |G_E^c| \times |G_I^{u_1}| \times |G_I^{u_2}|$). The key insight is to provide more chance to patterns supported by larger groups and contexts which are less likely to be discarded by more general ones generated by future iterations. This technique is inspired by the direct frequency-based sampling algorithm proposed in [12] which considers only Boolean attributed datasets.

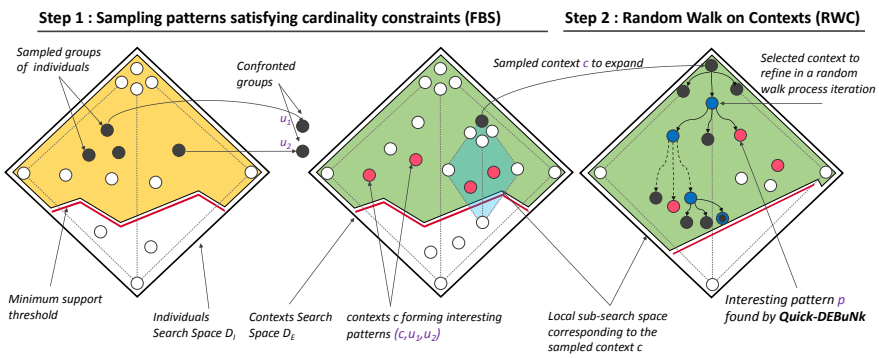


Fig. 4: *Quick-DEBuNk* approach in a nutshell

Here, this method is extended to handle more complex data with HMT, categorical and numerical attributes.

Random Walk on Contexts (step 2). Starting from a context obtained in step 1, a random walk traverses the search tree corresponding to the contexts description space $\mathcal{D}_E$. We introduce some bias to fully take advantage of the devised quality measures and the optimistic estimates , this being done to reward high quality patterns by giving them more chance to be sampled by the algorithm.

5.1 Frequency-Based Sampling (Step 1)

To sample patterns of the form $p = (c, u_1, u_2)$, we aim to draw description c, respectively u_1 and u_2, from description space $\mathcal{D}_E$, respectively $\mathcal{D}_I$, with a probability proportional to their respective support size. To this end, we devise the algorithm FBS (Frequency-Based Sampling).

In the following, for any $d \in \mathcal{D}$, $\downarrow d$ denotes the set of all descriptions subsuming d, i.e: $\downarrow d = \{d' \in \mathcal{D} \ : \ d' \sqsubseteq d\}$. Since $\mathcal{D}$[11]$= \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_m$, it follows that: $\downarrow d = \downarrow(r_1, r_2, ..., r_m) = \downarrow r_1 \times \downarrow r_2 \times ... \times \downarrow r_m$, where $\downarrow r_j$ is the set of conditions less specific than (implied by) r_j in the conditions space $\mathcal{D}_j$.

FBS generates a description d with a probability proportional to its frequency $\mathbb{P}(d) = \frac{|G^d|)}{\sum_{d' \in \mathcal{D}} |G^{d'}|}$ (formally defined in proposition 5). To this end, FBS performs two steps as depicted in Algorithm 3.

FBS starts by drawing a record g from G (line 1) with a probability proportional to the number of descriptions $d \in \mathcal{D}$ covering g (i.e: $| \downarrow \delta(g)|$). To enable this, each record $g \in G$ is weighted by $w_g = | \downarrow \delta(g)|$. For now, we use d^g to refer to $\delta(g)$. Knowing $d^g = (r_1^g, ..., r_m^g)$, the weight $w_g = | \downarrow d^g| = \prod_{j \in [1,m]} | \downarrow r_j^g|$ is the product of the numbers of restrictions subsuming each r_j^g. The size of $| \downarrow r_j^g|$ depends on the type of the related attribute a_j:

- *categorical attribute*: given that r_j^g corresponds to a value $v \in dom(a_j)$, we have $\downarrow r_j^g = \{*, v\}$ thus $| \downarrow r_j^g| = 2$.
- *numerical attribute*: given that r_j^g corresponds to an interval $[v, w]$ *with* $v, w \in dom(a_j)$, we have $\downarrow r_j^g$ is equal to the number of intervals having a left-bound $\underline{v} \leq v$ and a right-bound $\overline{w} \geq w$. More formally, $\downarrow r_j^g =$

Algorithm 3: FBS(G)

Input: G a collection of records which may be G_E or G_I

Output: a description d from $\mathcal{D}$ with $\mathbb{P}(d) = \frac{|G^d|}{\sum_{d' \in \mathcal{D}} |G^{d'}|}$

1 ***draw*** $g \backsim w_g$ ***from*** G ▷ with $w_g = | \downarrow \delta(g)|$
2 ***draw*** $d \backsim$ uniform($\downarrow \delta(g)$)
3 ***return*** d

[11]Cartesian product of the m lattices related to attributes conditions spaces forms a lattice[61]

$\{[\underline{v}, \overline{w}] \mid \underline{v} \leq v \wedge \overline{w} \geq w\}$. Hence, the cardinal of this set is $|\downarrow r_j^g| = |\{\underline{v} \in dom(a_j) \ : \ \underline{v} \leq v\}| \times |\{\overline{w} \in dom(a_j) \ : \ \overline{w} \geq w\}|$.

- *HMT attribute*: given that r_j^g corresponds to a set of tags $\{t_1, t_2, ...t_l\} \in dom(a_j)$, with $t_k \in T$ and T a tree, the condition r_j^g can be conceptualized as a rooted subtree of T where the leaves are $\{t_1, t_2, ...t_l\}$. Thus, $\downarrow r_j^g$ represents the set of all possible rooted subtrees of r_j^g. The latter cardinality can be computed recursively by starting from the root $*$ using $\text{nbs}(\text{tree}, \text{root}) = \prod_1^k (\text{nbs}(\text{tree}_i, \text{neighbor}_i) + 1)$ where neighbor_i returns the child tags of a given root and tree_i the subtree rooted on neighbor_i.

Given g the record returned from the first step and its corresponding description $d^g = \delta(g) = \langle r_1^g, ..., r_m^g \rangle$, FBS uniformly generates a description d from the set of descriptions covering g, that is $\downarrow d^g$. This can be done by uniformly drawing conditions r_j from $\downarrow r_j^g$, hence returning a description $d = \langle r_1, r_2, ..., r_m \rangle$. This comes from the fact that $\forall j \in [1, m] : \mathbb{P}(r_j) = \frac{1}{|\downarrow r_j^g|}$:

$$\mathbb{P}(d|g) = \prod_{j \in [1,m]} \mathbb{P}(r_j) = \frac{1}{\prod_{j \in [1,m]} |\downarrow r_j^g|} = \frac{1}{|\prod_{j \in [1,m]} \downarrow r_j^g|} = \frac{1}{|\downarrow d^g|} \ .$$

We now define the method used to uniformly draw a condition corresponding to an attribute a_j, according to its type:

- *categorical attribute*: given that $\downarrow r_j^g = \{*, v\}$ with $v \in dom(a_j)$, it is sufficient to uniformly draw an element r_j from $\{*, v\}$.
- *numerical attribute*: given that $\downarrow r_j^g = \{[\underline{v}, \overline{w}] \mid \underline{v} \leq v \wedge \overline{w} \geq w\}$, to generate an interval $[sv, sw]$ from $\downarrow r_j^g$ uniformly, one needs to uniformly draw a left-bound sv from the set $\{\underline{v} \in dom(a_j) \ : \ \underline{v} \leq v\}$ and a right-bound sw from the set $\{\overline{w} \in dom(a_j) \ : \ \overline{w} \geq w\}$.
- *HMT attribute*: given that $\downarrow r_j^g$ represents the set of rooted subtrees of r_j^g, we have to uniformly draw such rooted subtrees. A first way is to generate all the possible rooted subtrees and then uniformly draw an element from the resulting set. This does not scale. Hence we devised another method, relying on a stochastic process using the aforementioned function *nbs* (which counts the number of subtrees rooted on some given node). The algorithm takes the root $*$ as a starting tree. Next, the resulting subtree is augmented by a child c of $*$ with a chance equal to the number subtrees of $\downarrow r_j^g$ containing c. That is $\frac{\text{nbs}(r_j^g, *) - \text{nbs}(r_j^g - \{c\}, *)}{\text{nbs}(r_j^g, *)}$. Recursively, the algorithm continues from a drawn candidate child c.

Proposition 5 *A description $d \in \mathcal{D}$ has a probability of being generated by* FBS *equal to* $\mathbb{P}(d) = \frac{G^d}{\sum_{d' \in \mathcal{D}} |G^{d'}|}$. *(see Appendix A for proofs)*

FBS algorithm makes it possible to generate valid patterns $p = (c, u_1, u_2)$ from the pattern space $\mathcal{P} = \mathcal{D}_E \times \mathcal{D}_I \times \mathcal{D}_I$. This is achieved in the first step of Quick-DEBuNk (lines 3-6 in *Algorithm 5*) by sampling two group descriptions u_1, u_2 from $\mathcal{D}_I$ and a context c from $\mathcal{D}_E$ followed by assessing if the three descriptions satisfy the cardinalities constraints $\mathcal{C}$ (min. support thresholds).

Proposition 6 *Given the cardinality constraints $\mathcal{C}$, every valid pattern p is reachable by the first step of* Quick-DEBuNk. *i.e.* $\forall p \in \mathcal{P} : p$ *satistifies* $\mathcal{C} \Rightarrow \mathbb{P}(p) > 0$ *(see Appendix A for proofs)*

Step 1 of Quick-DEBuNk does not favor the sampling of high quality patterns as it does not involve an exploitation phase. The random walk process on contexts used in Step 2 enables a smarter traversal of the search space while taking into account the devised quality measures and optimistic estimates.

5.2 Random Walk on Contexts (RWC)

RWC (Algorithm 4) enumerates contexts of the search space corresponding to $\mathcal{D}_E$ while considering closure and optimistic estimates. *RWC* takes as input two confronted groups of individuals described by u_1,u_2 for which it looks for relevant contexts (i.e., to form an (dis)agreement pattern) following a random walk process starting from a context c. Mainly, RWC has two steps that are recursively executed until a terminal node is reached. RWC starts by generating all neighbors d of the current context c (line 2). Next, RWC assesses whether the size of the corresponding support G_E^c and the optimistic estimates respectively exceed the support threshold σ_E and the quality threshold σ_φ (line 3). If appropriate, the closed description d is computed (line 4). The algorithm proceeds by evaluating the quality of pattern (line 5). If the quality exceeds the threshold σ_φ, the pattern is valid and is hence yielded (line 6). Otherwise, the pattern is added to NtE (*Neighbors to be Explored*) (line 8) as its related sub search space may contain interesting patterns (*i.e* $\mathrm{oe}_\varphi(d, u_1, u_2) \geq \sigma_\varphi$). The second step of RWC consists in selecting a neighbor from NtE to be explored with a probability proportional to its quality (lines $10 - 12$). This process is recursively repeated until a terminal node is reached (i.e. $\mathrm{NtE} = \emptyset$).

Algorithm 4: RWC($\langle G_I, G_E, O, o\rangle, c, u_1, u_2, \sigma_E, \varphi, \sigma_\varphi$)

Inputs : $\langle G_I, G_E, O, o\rangle$ a behavioral dataset; c the current context;
(u_1, u_2) couple of confronted group descriptions of individuals;
σ_E threshold on support; φ the quality measure; σ_φ quality threshold.
Output: yield valid patterns (c, u_1, u_2)

```
1  NtE ← {}
2  foreach d ∈ η(c) do
3      if |G_E^d| ≥ σ_E and oe_φ(d, u1, u2) ≥ σ_φ then
4          closure_d ← δ(G_E^d)
5          if φ(d, u1, u2) ≥ σ_φ then
6              yield closure_d
7          else
8              NtE ← NtE ∪ {d}
9  if NtE ≠ ∅ then
10     draw next ∽ φ(next, u1, u2) from NtE
11     foreach c_next ∈ RWC(⟨G_I, G_E, O, o⟩, next, σ_E, φ, σ_φ, u1, u2) do
12         yield c_next
```

5.3 Quick-DEBuNk

Quick-DEBuNk (Algorithm 5) samples patterns from the full search space $\mathcal{D}_E \times \mathcal{D}_I \times \mathcal{D}_I$. It is based on FBS and RWC. It takes as input the same parameters as DEBuNk in addition to a *timebudget*. It starts by generating a couple of closed group descriptions of individuals u_1, u_2 that fulfill the support constraint (lines $3-5$) using FBS. Next, Quick-DEBuNk generates a context while only considering entities having a quality greater than the threshold σ_φ (line 6). The reason behind considering only $G_E^{\geq\sigma_\varphi}$ is clear: we have $\forall p \in \mathcal{P}$ *p satisfies* $\mathcal{C}$ and $\varphi(p) \geq \sigma_\varphi \Rightarrow \exists e \in G_E^c : \varphi(\{e\}, G_I^{u_1}, G_I^{u_2}) \geq \sigma_\varphi$ (since the quality measure is a weighted mean). If the context fulfills the cardinality constraint and its evaluated optimistic estimate is greater than the quality threshold (line 7), the algorithm then evaluates the quality of the sampled pattern (line 8). If this quality is greater than the threshold σ_φ, the pattern is appended to the resulting pattern set if and only if it is not more specific of an already found pattern w.r.t. extents (lines $9-11$). Otherwise, a random walk is launched starting from context c (line 13). This is done by relying on RWC. The algorithm continues by updating the resulting pattern set by each pattern yielded by RWC, as long as there is no more general pattern in the current pattern set P (lines $14-16$). Otherwise, RWC is interrupted (line 18). The process is repeated as long as the time budget allows.

Algorithm 5: Quick-DEBuNk($\langle G_I, G_E, O, o\rangle, \sigma_E, \sigma_I, \varphi, \sigma_\varphi$,*timebudget*)

Inputs : $\langle G_I, G_E, O, o\rangle$ a behavioral dataset;
σ_E (resp. σ_I) minimum support threshold of a context (resp. group);
φ the quality measure; σ_φ threshold on the quality;
timebudget the maximum amount of time given to the algorithm.
Output: P the set of local relevant (dis)agreement patterns

```
1  P ← {}
2  while executionTime < timebudget do
3      u1 ← clo(FBS(G_I))
4      u2 ← clo(FBS(G_I))
5      if |G_I^{u1}| ≥ σ_I ∧ |G_I^{u2}| ≥ σ_I then
6          c ← clo(FBS(G_E^{≥σ_φ}))          ▷ G_E^{≥σ_φ} = {e ∈ G_E | φ({e}, I_{u1}, I_{u2}) ≥ σ_φ}
7          if |G_E^c| ≥ σ_E ∧ oe_φ(c, u1, u2) ≥ σ_φ then
8              if φ(c, u1, u2) ≥ σ_φ then
9                  p_new ← (c, u1, u2)
10                 if ∄p_old ∈ P | ext(p_new) ⊆ ext(p_old) then
11                     P ← (P ∪ p_old) \ {p_old ∈ P | ext(p_old) ⊆ ext(p_new)}
12             else
13                 foreach d ∈ RWC(⟨G_I, G_E, O, o⟩, c, u1, u2, σ_E, φ, σ_φ) do
14                     p_new ← (d, u1, u2)
15                     if ∄p_old ∈ P | ext(p_new) ⊆ ext(p_old) then
16                         P ← (P ∪ p_new) \ {p_old ∈ P | ext(p_old) ⊆ ext(p_new)}
17                     else
18                         break
19                     if executionTime ≥ timebudget then
20                         return P
21 return P
```

6 Empirical Study

In this section, we report on both quantitative and qualitative experiments over the implemented algorithms. For reproducibility purposes, source code (in Python) and data are made available in our companion page[12].

6.1 Aims and Datasets

The experiments aim to answer the following questions:

- How effective is DEBuNk compared to State-of-the-Art algorithms?
- How well can DEBuNk and Quick-DEBuNk identify exceptional inter-group agreement patterns in synthetic data?
- Are the closure operators and optimistic estimate based pruning, efficient?
- How effective is HMT closed description enumeration compared against closed itemset enumeration?
- Does DEBuNk scale w.r.t. different parameters?
- How effective is Quick-DEBuNk at sampling interesting patterns in limited time budgets?
- Do the algorithms provide interpretable patterns?

Most of the experiments were carried out on four real world behavioral datasets whose general characteristics are summarized in Table 2. Each of the considered behavioral datasets figures out entities with an HMT (H) attribute together with categorical (C) and numerical (N) ones, while the individuals have numerical and categorical attributes.

EPD8[13] features voting information of the eighth European Parliament about the 958 members who were elected in 2014 or after. The dataset records $2.7M$ tuples indicating the outcome (For, Against, Abstain) of a member's voting during one of the 4161 sessions. Each session is described by its themes (H), its voting date (N) and its organizing committee (C). Individuals are described by their national party (C), political group (C), age group (C), country(C) and additional information about countries (date of accession to EU (N) and the country currency (C)). To analyze exceptional inter-group agreement in this dataset, we consider the measure IAS_{voting}. This measure is defined using $\theta_{majority}$ and sim_{voting}.

	$\lvert G_E \rvert$	$\mathcal{A}_E$	$\lvert G_I \rvert$	$\mathcal{A}_I$	Outcomes
EPD8	4161	$1H + 1N + 1C$	958	$1N + 5C$	$2.7M$
Movielens	1681	$1H + 1N$	943	$3C$	$100K$
Yelp	$127K$	$1H + 1C$	18	$3C$	$750K$
Openmedic	12 219	$1H$	78	$3C$	$500K$

Table 2: Main characteristics of the behavioral datasets

[12] https://github.com/Adnene93/DEBuNk

[13] http://parltrack.euwiki.org/, last accessed on 17 November 2017

Movielens[14] is a movie review dataset [34] consisting of $100K$ ratings (ranging from 1 to 5) expressed by 943 users on 1681 movies. The movies are characterized by their genres (H) and release date (N), while individuals are described with demographic information such as age group (C), gender (C) and occupation (C). To investigate (dis)agreement patterns, we use the adapted measure $\text{IAS}_{ratings}$. Its definition relies on θ_{wavg} and $sim_{ratings}$.

Yelp[15] is a social network dataset featuring individuals who rate (scores ranging from 1 to 5) places (stores, restaurants, clinics) characterized by their categories (H) and their state (C). The dataset originally contains 1M users. We preprocessed the dataset to constitute 18 groups of individuals based on the size of their friends network (C), their seniority (C) in the platform and whether users are elites or not (C). This preprocessing has been done to allow the discovery of interpretable patterns. The same $\text{IAS}_{ratings}$ measure is used to analyze this dataset.

Openmedic[16] is a drug consumption monitoring dataset that has been recently made available by *Ameli*[17]. This dataset inventories the number of drug boxes (described by their ATC classification (H)) yearly administered to individuals (2014, 2015 and 2016). Individuals are described with demographic information such as age group (C), gender (C) and region (C). In the qualitative results, we discuss an adapted IAS measure.

Comparing the size and the complexity of these datasets is difficult because of the heterogeneity of the attributes. In particular, the hierarchies of the HMT attributes are very different, as well as the range of the numerical ones. To enable a fair comparison, we operate a (kind of) conceptual scaling [27].

		Entities ($\mathcal{A}_E$)	Individuals ($\mathcal{A}_I$)
EPD8	attribute types	$1H + 1N + 1C$	$1N + 5C$
	size after scaling	$347 + 26 + 40 = 413$	$16 + 285 = 301$
	average scaling values in a tuple	20.44	14
Movielens	attribute types	$1H + 1N$	$3C$
	size after scaling	$20 + 144 = 164$	$4 + 2 + 21 = 27$
	average scaling values in a tuple	75.72	3
Yelp	attribute types	$1H + 1C$	$3C$
	size after scaling	$1175 + 29 = 1204$	$3 + 2 + 3 = 8$
	average scaling values in a tuple	5.77	3
Openmedic	attribute types	$1H$	$3C$
	size after scaling	14094	$2 + 13 + 3 = 18$
	average scaling values in a tuple	7	3

Table 3: Behavioral Datasets Characteristics After Scaling

[14] https://grouplens.org/datasets/movielens/100k/

[15] https://www.yelp.com/dataset/challenge, last accessed on 25 April 2017

[16] http://open-data-assurance-maladie.ameli.fr/, last accessed on 16 November 2017

[17] ***Ameli*** - France National Health Insurance and Social Security Organization

Dataset	Transactions	Items	AverageSize
EPD8	1 727 032 585	1 015	34.48
Movielens	16 807 109	218	79.37
Yelp	5 860 354	1 220	9.00
Openmedic	28 512 418	14 130	10.00

Table 4: Characteristics of datasets considered as plain collections of itemsets records - the plain collections correspond to $G_E \times G_I \times G_I$ while considering only pairable individuals (pairs of individuals who both expressed outcomes/actions)

The attributes are "projected" to a set of items by transforming each one to a Boolean representation. Each possible value of a categorical attribute provides a single item (e.g. *gender* gives *male*, *female* and *unknown*). The items corresponding to an HMT attribute are all the nodes of the tag tree (T). Each numerical attribute is transformed to an itemset with an *interordinal scaling* [40]. To a given set of values $[v_1, v_2, ...v_n]$, we associate $2n$ items $\{\leq v_1, \leq v_2, ... \leq v_n, \geq v_1, \geq v_2, ... \geq v_n\}$. Table 3 illustrates this step, while Table 4 shows the obtained comparable characteristics.

Some questions we aim to answer require data for which the ground truth is known. Since it is notoriously difficult to obtain such data, we designed an artificial behavior data generator. The generator works as follows. It first generates `nb_hidden_patterns` (dis)agreement patterns. Each pattern is represented by two group descriptions (u_1, u_2) and a context (c) where u_1, u_2 and c are defined over random categorical descriptions and are of random size. For each pattern, the extent is generated (i.e., `context_support_size` entities for the context and the two groups involving `group_support_size` individuals). The extents are generated as follows: first, a random description **ds** is uniformly drawn from $\mathcal{D}_E$ (resp. $\mathcal{D}_I$). Next, `support_size` records are generated, which have a description equal to or subsumed by **ds**. This process is repeated for each component of the pattern so as to built `nb_hidden_patterns` (dis)agreement

Parameter	Description	Default value
$\lvert G_E \rvert$ (`nb_entities`)	Number of entities	2000
$\lvert G_I \rvert$ (`nb_individuals`)	Number of individuals	500
$\lvert O \rvert$	Number of possible categorical outcomes	2
$\lvert \mathcal{A}_E \rvert$	Number of categorical attributes for entities	2
$\lvert dom(a_j) \rvert$ with $a_j \in \mathcal{A}_E$	Domain size of a categorical attribute $a_j \in \mathcal{A}_E$	4
$\lvert \mathcal{A}_I \rvert$	Number of categorical attributes for individuals	2
$\lvert dom(a_j) \rvert$ with $a_j \in \mathcal{A}_I$	Domain size of a categorical attribute $a_j \in \mathcal{A}_I$	4
`nb_hidden_patterns`	Number of planted conflictual patterns	3
`context_support_size`	Support size of a hidden pattern context	5
`group_support_size`	Support size of a hidden pattern group	5
`noise_rate`	Noise rate in/out the ground truth patterns	0
`data_sparsity`	Probability of an individual not to cast an outcome	0.33

Table 5: Default Parameters Used for Generating Artificial Behavioral Data

patterns. Note that the pattern generation process avoids overlapping between groups and contexts between differents patterns. These patterns describe conflictual situations, i.e., the individuals of one group in the pattern context express a voting outcome which is different from the other group' voting outcome. Conversely, the two groups are in agreement in the usual case, i.e., their votes over the entities outside the pattern context are similar. To achieve this, for each planted pattern (c, u_1, u_2) and for each entity $e \in G_E$, a random outcome **os** is drawn from the pool of possible outcomes (in here we consider $O = \{\text{Yes}, \text{No}\}$). Subsequently, each member comprising $G_I^{u_1}$ votes **os** for the entity e. Accordingly, individuals from $G_I^{u_2}$ cast a different outcome, if e is described by the context c. Otherwise, they cast the same outcome **os**. Once these patterns are generated, the rest of the dataset is generated by adding entities and individuals randomly while preserving the exceptionality of the patterns (i.e., the patterns must remain the most general exceptional patterns) till the desired size of the dataset is reached (i.e. $|G_E| =$ `nb_entities` and $|G_I| =$ `nb_individuals`). As described, the hidden patterns are pure. A last step enables to add noise within the patterns. For each pattern, the expressed outcome of individuals are randomly replaced with a `noise_rate` probability. Similarly, noise is added outside the patterns. Eventually, to get as close as possible to real-world behavioral dataset, we add sparsity in the data. To perform this task, each outcome of each pair $(e, i) \in G_E \times G$ have a probability of `data_sparisty` to be removed from the generated artificial behavioral dataset.. The parameters used are summarized in Table 5.

The rest of this section is organized as follows. We qualitatively compare DEBuNk and Quick-DEBuNk with state-of-the-art methods on artificial data (subsect. 6.2). We then study their ability in noisy data (subsect. 6.3). A full performance study of our two algorithms is reported in subsect. 6.4. Qualitative results on the four real-world datasets are provided in subsect. 6.5. A discussion on the limitations of our algorithms closes this section.

6.2 Comparison to state-of-the-art techniques

To put DEBuNk and its sampling alternative Quick-DEBuNk to the test, we investigates the ability of classical SD/EMM techniques to tackle the problem of discovering exceptional disagreement among groups of individuals in Section 6.2.1 and we compare their efficiency and effectiveness against our first attempt[8] implemented by DSC Algorithm in Section 6.2.2.

6.2.1 Comparison to SD/EMM methods

We aim to study how the SD/EMM methods are able to discover relevant (dis)agreement patterns. SD algorithms available in public implementations (e.g., Vikamine[5], Cortana [54], PySubgroup [50]) only consider one flat table with a target attribute. However, behavioral datasets involve three relations (Entities, Individuals, Outcomes) which are all processed by DEBuNk and its

sampling alternative Quick-DEBuNk to discover the interesting (dis)agreement patterns. To handle the problem we defined with a classical SD algorithm, we need to preprocess the data. We discuss and compare several problem adaptations.

SD-Majority: SD to discover contextual disagreements with the majority. The most direct way to apply SD on behavioral data is to consider the discovery of *groups* of individuals who express disagreement with the majority vote. This enables to discover patterns (c, g_1) where c is a context describing a set of entities and g_1 is a description of a group of individuals. To this end, we preprocess the behavioral data to obtain a Flat Behavioral Dataset (FBD) with a single table and a singe target class `SAME_AS_MAJORITY` as following: (1) we combine the entities and individuals tables using a join operation with the outcomes collection. (2) We compute the majority vote by aggregating the votes expressed on each entity. (3) We use this information to extend each record in the newly generated FBD with the attribute `SAME_AS_MAJORITY` which is equal to $+$, indicating that the individual voted in agreement with the majority in the considered entity. Otherwise `SAME_AS_MAJORITY` is equal to $-$. Example of FBD after such preprocessing is given in Table 6. Having this FBD augmented with the target class `SAME_AS_MAJORITY` offers the possibility to run common SD techniques to identify subgroups with a high prevalence of disagreement with the majority (Target label $= '-'$). The most adapted interestingness measure in this case is the precision gain [25], i.e. $Precision(subgroup) - \alpha^-$, which is high when there is a high disagreement in a subgroup compared to the disagreement observed in the full dataset. Note that this model does not fit perfectly our problem setting. It enables only the discovery of bi-set patterns (c, g_1) rather than the desired three-set patterns (c, g_1, u_2). Nevertheless, highlighting this type of pattern may help to partially identify interesting (dis)agreement patterns in a behavioral dataset. Furthermore, this adaptation does not takes into account the usual behavior of the group against the majority. This might clearly lead to the discovery of obvious patterns highlighting the individuals that are known to be a systematic opposition.

Entities			***Individuals***			***Outcomes***	
ide	theme	date	idi	country	group	outcome	`SAME_AS_MAJORITY`
e_1	1.20 Citizen's rights	20/04/16	i_1	France	S&D	For	+
...	...	...	...	...	...	...	...

Table 6: Example of input data format for SD-Majority after transforming the behavioral dataset given in Table 1.

SD-Cartesian: SD to discover contextual disagreement between two groups. We propose a second modeling to enable the discovery of three-set patterns (c, u_1, u_2) with SD techniques. To this end, the behavioral dataset is transformed into a flat table equivalent to the Cartesian product $G_E \times G_I \times G_I$.

This flat table is then augmented with a target class attribute SAME_VOTE which captures the (dis-)agreement between each couple of individuals on each entity for which both expressed an outcome. SAME_VOTE is thus equal to + if both individuals expressed the same outcome for the entity, − otherwise. This modeling – illustrated in Table 7 – makes it possible to discover patterns (c, u_1, u_2) which identify two groups of individuals and a context regrouping a set of entities over which the individuals in the first group disagrees with the ones composing the second group. This can be done using the precision gain as the interestingness measure. Even if the syntax of the patterns is similar to ours, the usual agreement between the two selected groups is not take into account. Hence, the semantics conveyed by these patterns is different from ours. Another major drawback of such modeling is the size of the table resulting from the Cartesian product. For instance, a small behavioral dataset with 200 entities and 100 individuals can contain up to 2×10^6 records which clearly make this setting not adapted and not scalable for real-world behavioral data.

Entities		***Individuals***		***Individuals***		***Outcomes***		
ide	theme	idi_1	country_1	idi_2	country_2	$outcome_1$	$outcome_2$	SAME_VOTE
e_5	7.30	i_1	France	i_2	France	*For*	*For*	+
e_5	7.30	i_1	France	i_3	France	*For*	*Against*	-
...	...	...	...	...	...	...	...	

Table 7: Example of input data format for SD-Cartesian after transforming the behavioral dataset given in Table 1 to a Cartesian product $G_E \times G_I \times G_I$.

Exceptional Contextual Subgraph Mining to discover contextual disagreement between two groups. Applying SD in the two aforementioned modelings does not allow to take into account the usual inter-group agreement in the model. A way to overcome this issue is to model the behavioral dataset as an attributed graph and looking for exceptional contextual subgraphs [41]. The so-called COSMIC algorithm is rooted in SD/EMM and aims at discovering contextual subgraphs whose edges have weights larger than expected. To this end, we transform the behavioral dataset to the Cartesian product $G_E \times G_I \times G_I$ extended with SAME_VOTE attribute like in *SD-Cartesian* formalization. This table is then used to build a bipartite graph where each side represents the collection of individuals G_I and an edge is instantiated between two vertices (individuals) for each entity on which the two individuals expressed conflicting outcomes. The set of transactions from $G_E \times G_I \times G_I$ where two individuals are disagree are associated to the edge between the two corresponding vertices (see Fig. 5). Once this transactions set obtained, COSMIC algorithm can be used to obtain exceptional contextual subgraphs. Note that, in this problem setting, an exceptional contextual subgraph corresponds to two groups of individuals which exhibit a higher disagreement rate in the considered context compared to the disagreement expected in a similar sized subgraph. Several interestingness measures have been proposed in the

COSMIC framework [9,41]. For the aim of this study, the lift measure is the most adapted: $\varphi(S) = \frac{\mathbb{P}(S|C)}{\mathbb{P}(S)}$ with S is the connected contextual subgraph induced by the selection performed by the description C. Note that: $\mathbb{P}(S|C)$ is the probability that a random drawn edge from all the edges in the full graph supporting the selection C falls in the induced contextual subgraph, $\mathbb{P}(S)$ is the relative weights in terms of the number of edges of the full subgraph S (the subgraph with the most general context). Note that a post-processing is necessary to transform exceptional contextual subgraphs into (dis)agreement patterns (c, u_1, u_2). Applying contextual subgraph mining given this modeling has some limitations: (1) the expected disagreement between two groups is computed from all the individuals instead of the individuals of the two groups. This can lead to the discovery of obvious patterns. (2) it considers as an input a transaction dataset computed from the Cartesian product $G_E \times G_I \times G_I$ which limits its usage, even for relatively small behavioral dataset.

We aim to compare how state-of-the-art methods perform in this three modelings and compare them to DEBuNk and Quick-DEBuNk. To this end, we generated 81 artificial dataset with 3 hidden patterns by varying several parameters (see Fig. 6). Note that the behavioral datasets are relatively small to be sure to obtain results for each modeling, especially ones that requires to build a Cartesian product. For SD-Majority and SD-Cartesian modelings, we used PySubgroup[50] to discover subgroups for the following reasons: the implementation is available online[18] as well as the easiness of its use. We ran the exhaustive search algorithm BSD[51] which is tailored to find relevant subgroups [28], this choice is also motivated by the fact that the selected interestingness measure is the Precision gain. For the attributed graph modeling, we used an implementation of COSMIC algorithm provided by the authors [41].

ide	themes	date
e_1	1.20	20/04/16
e_2	2.10	16/05/16
e_3	1.20; 7.30	04/06/16
...	...	...

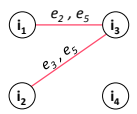
(a) Entities

idi	country	group	age
i_1	France	S&D	26
i_2	France	PPE	30
...	...	...	

(b) Individuals

id_edge	ide	idi_1	idi_2
t_1	e_2	i_1	i_3
t_2	e_5	i_1	i_3
t_3	e_3	i_2	i_3
t_4	e_5	i_2	i_3

(c) Transactions set (edges)



(d) Augmented Graph

Fig. 5: Example of input data format for Cosmic after transforming the behavioral dataset given in Table 1 to an augmented graph and its corresponding transactions set according to the observed discords.

[18] https://bitbucket.org/florian_lemmerich/pysubgroup

To evaluate the ability of the different approaches of uncovering planted patterns, we first define a similarity measure $sim_{\mathcal{P}}$ between two patterns $p = (c, u_1, u_2)$ and $p' = (c', u'_1, u'_2)$ from $\mathcal{P}$. It captures to what extent two patterns provide similar insights about changes of inter-group agreement.

$$\text{sim}_{\mathcal{P}}(p, p') = \sqrt{J(G_E^c, G_E^{c'}) \times \frac{1}{2} \cdot \left(J(G_I^{u_1}, G_I^{u'_1}) + J(G_I^{u_2}, G_I^{u'_2}) \right)} \text{ with } J(G, G') = \frac{|G \cap G'|}{|G \cup G'|} .$$

Note that, the quantity $(J(G_I^{u_1}, G_I^{u'_1}) + J(G_I^{u_2}, G_I^{u'_2}))$ is replaced by the following measure if the quality measure φ is symmetric:

$$\max(J(G_I^{u_1}, G_I^{u'_1}) + J(G_I^{u_2}, G_I^{u'_2}), J(G_I^{u_1}, G_I^{u'_2}) + J(G_I^{u_2}, G_I^{u'_1})) .$$

For comparing two pattern sets P, P' returned by respectively DEBuNk and Quick-DEBuNk, we use an F_1 score defined as follows.

$$F_1(P, P') = 2 \cdot \frac{\text{precision}(P, P') \cdot \text{recall}(P, P')}{\text{precision}(P, P') + \text{recall}(P, P')} , \tag{9}$$

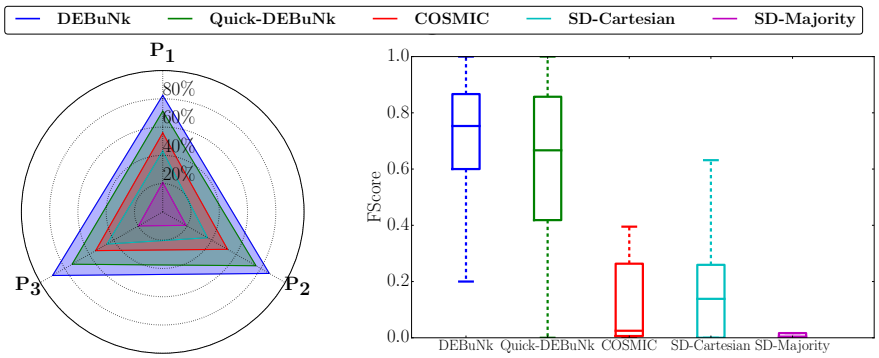
$$\text{with} \begin{cases} \text{precision}(P, P') & = \dfrac{\sum_{p \in P} \max(\{\text{sim}_{\mathcal{P}}(p, p') \mid p' \in P'\})}{|P|} , \\ \text{recall}(P, P') & = \dfrac{\sum_{p' \in P'} \max(\{\text{sim}_{\mathcal{P}}(p', p) \mid p \in P\})}{|P'|} . \end{cases}$$

A similar measure to the recall has been proposed by the authors in [14] to evaluate the ability of their algorithm to retrieve the ground-truth patterns. We extend this measure with the precision to evaluate not only that all the hidden patterns have been discovered by an algorithm (i.e. recall=1.) but also the conciseness of the returned set (i.e. precision=1 if and only if all returned patterns are actually present in the behavioral dataset).

We report in Figure 6a the comparative experiments between DEBuNk, Quick-DEBuNk, SD-Cartesian, SD-Majority and COSMIC in terms of their ability to retrieve each planted pattern in synthetic behavioral datasets. We report for each method the average similarity (over the 81 artificial data) between one of the three hidden patterns and its nearest representative in the result set. As expected, DEBuNk and Quick-DEBuNk outperforms other approaches. Moreover, the order between the approaches/modelings is sound. Majority-SD has the worst results due to the fact that this method, in the best case scenario, is only able to identify two of the three restrictions of a (dis)agreement pattern which impact on its performance. COSMIC performs slightly better than its alternative SD technique over the Cartesian product $G_E \times G_I \times G_I$ thanks to a more sophisticated model to capture the usual behavior.

Figure 6b summarizes the results obtained after running the five approaches. For a fair comparison (i.e., the problem of setting the good thresholds), we report the average F-Score of the only top-10 results for each approach. We

observe that DEBuNk and Quick-DEBuNk achieves to return high-quality results compared to the other approaches. Interestingly, COSMIC adaptation is of less quality than SD-Cartesian adaptation when analyzing both their conciseness and exactitude in terms of hidden pattern identification. Finally SD-Majority performs the worst due to its fundamental difference with the other approaches when comparing the provided patterns format.



(a) Average similarity between the planted patterns and their representatives returned by each method.

(b) Boxplots of F-score comparing the top-10 discovered patterns set by each method on each generated artificial data and the corresponding ground truth.

Fig. 6: Comparative qualitative performance study between DEBuNk ($\sigma_E = 3$, $\sigma_I = 3$, $\sigma_\varphi = 0.5$ and the quality measure $\varphi_{dissent}$), Quick-DEBuNk (same parameters as $DEBuNk$ with $timebudget = 5$ $seconds$), SD-Majority (`resultSetSize`= 50, i.e. Top-50), SD-Cartesian (`resultSetSize`= 25, i.e. Top-25) and Cosmic (Default parameters) performed over 81 artificial behavioral data with 3 hidden patterns by varying the number of individuals in $[100, 125, 150]$, the number of entities in $[100, 150, 200]$, the sparsity factor in $[0., 0.25, 0.5]$ and the noise in $[0., 0.2, 0.4]$.

6.2.2 Comparison to DSC

In this subsection, we report the results of experiments conducted to compare DEBuNk against first attempt [8] implemented by algorithm DSC.

We recall that DSC solves the problem of discovering top-k patterns that elucidate exceptional (dis)agreement between groups of individuals. In addition, as aforementioned in Section 2, for a sufficiently large k, DSC solves the core problem tackled in this paper limited to the two first conditions. To compare between DEBuNk and DSC, we designed experiments to answer to the two following questions. Note that, we disable the aggregation dimension parameter for DSC to obtain comparable pattern sets.

Q1. How concise is the patterns set provided by DEBuNk compared to the one provided by DSC?
Q2. How diversified is the patterns set, limited to k patterns, provided by DEBuNk compared to the one provided by DSC?

In order to answer to (Q1), we evaluate the number of patterns returned by DEBuNk and DSC when looking for the complete pattern set P (i.e. k sufficiently large for DSC). For this, we run both methods on EPD8 with various[19] quality thresholds σ_φ and descriptive attributes $\mathcal{A}_E$, $\mathcal{A}_I$. Figure 7 reports the results of these experiments. Results demonstrate that DEBuNk compresses considerably the desired pattern set while ensuring that each pattern returned by DSC is represented by a pattern returned by DEBuNk (according to the problem definition). In average, DSC returns $\times 38$ more patterns than DEBuNk. Moreover, DEBuNk achieves better performance than DSC in terms of run time . Thanks to (i) the model simplification which reduces the complexity of computing the interestingness measure and (ii) the pruning property implemented by DEBuNk supported by condition (3) of the problem definition.

So far, we compared DEBuNk against DSC when looking for the complete pattern set. Experiments discussed above clearly demonstrated the fact that in such setting DSC returns an overwhelmingly large results set. To tackle such problem, DSC implemented a top-k algorithm to control the size of the provided pattern set. Of course, the main drawback of using a top-k algorithm is the lack of diversity even when redundancy is avoided by closure operators. This lack of diversity is induced by the fact that, most likely, the patterns observing the highest qualities are condensed in small region of the dataset.

In order to fairly evaluate the diversity of patterns returned by both DSC and DEBuNk (Q2). We run both algorithms for several parameters[20] and compare the size of the datasets regions covered by both returned patterns set. This quantity can be captured by the number of outcomes covered by a results set, that is $|o[P^k]| = |\{(i,e) \in G_I \times G_E \text{ s.t. } o(i,e) \text{ is expressed}\}|$ with P^k an arbitrary pattern set containing k patterns. For a fair comparison, we compare $|o[P^k_{\text{DSC}}]|$ (top-k patterns) against $|o[P^k_{\text{DEBuNk}}]|$. To obtain the latter quantity, we run DEBuNk so as to obtain the complete pattern set P_{DEBuNk}. Next, we draw 100 k-sized samples drawn uniformly from the obtained P_{DEBuNk} and then compute the average $|o[P^k_{\text{DEBuNk}}]|$. It is worth mentioning that comparison can be made also by taking the top-k patterns P_{DEBuNk} rather than an arbitrary k-sized sample. We resolved to study the latter scenario, since the philosophy of DEBuNk is to retrieve the complete patterns set summarizing exceptional (dis)agreement in an underlying behavioral dataset.

Figure 8 sums up the results of the experiments. Clearly, DEBuNk' k-sized pattern set covers larger (and different) parts of the dataset compared to DSC' top-k pattern set. We observe that DEBuNk surpasses DSC by one order of magnitude ($\times 12.5$ in average) when comparing the portions of the dataset covered by their respective k-sized pattern set. Simply put, when the pattern set related to DEBuNk covers 10% of the dataset, DSC patterns cover less than 1% of the underlying dataset records.

[19] 27 runs for each method by varying $(|\mathcal{A}_E|, |\mathcal{A}_I|, \sigma_\varphi) \in [[1,2,3], [1,2,3], [0.2, 0.4, 0.6]]$

[20] 81 runs by varying $(k, |\mathcal{A}_E|, |\mathcal{A}_I|, \sigma_\varphi) \in [[10, 50, 100], [1,2,3], [1,2,3], [0.2, 0.4, 0.6]]$

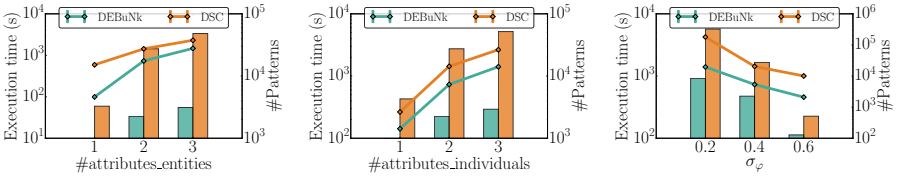


Fig. 7: Comparison between DEBuNk and DSC for the task of discovering the complete set of the desired patterns. Experiments consider the full EPD8 Dataset with the following default parameters: $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 2$, $\sigma_\varphi = 0.4$, $\sigma_E = 40$, $\sigma_I = 10$ and $\varphi_{dissent}$. Lines correspond to the execution time and bars correspond to the number of returned patterns.

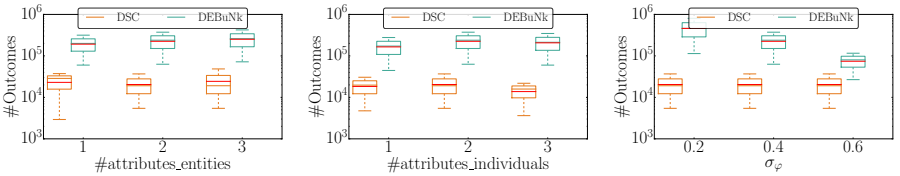


Fig. 8: Comparison between DEBuNk and DSC (top-k) for the task of discovering k-sized pattern set. Experiments consider the full EPD8 Dataset with the following default parameters: $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 2$, $\sigma_\varphi = 0.4$, $\sigma_E = 40$, $\sigma_I = 10$ and $\varphi_{dissent}$. Box plots correspond to the size of $O[P^k]$ when varying k in $[10, 50, 100]$.

6.3 Robustness to noise and ability to discover hidden patterns

We now study the ability of DEBuNk and Quick-DEBuNk to discover hidden patterns for larger behavioral datasets as well as their robustness to noise. To this end, we carried out DEBuNk and Quick-DEBuNk over several artificial datasets varying the noise rate from 0 to 0.8. The results illustrated in Figure 9 demonstrates that the exhaustive search approach DEBuNk is able to discover almost exclusively all the hidden patterns ($F1_Score > 0.8$) even jf the noise rate is rather high (≤ 0.6). Indeed when the noise rate is substantially high, $DEBuNk$ does not retrieve the noisy hidden patterns. This clearly results

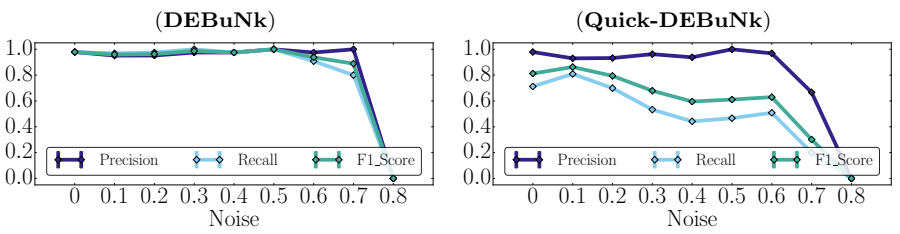


Fig. 9: Efficiency of DEBuNk ($\sigma_E = 7$, $\sigma_I = 7$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$) and Quick-DEBuNk ($\sigma_E = 7$, $\sigma_I = 7$, $\sigma_\varphi = 0.5$, $timebudget =$ 3 mn and $\varphi_{dissent}$) performed over 21 behavioral artificial data generated with the following default parameters ($|G_E| = 2000$, $|G_I| = 500$, $|\mathcal{A}_E| = |\mathcal{A}_I| = 3$, ***size_dom_entities_attributes*** = ***size_dom_individuals_attributes*** =4, ***nb_hidden_patterns*** =5, ***context_support_size*** =10, ***group_support_size*** =10).

from the evidence that several planted patterns disappear in the underlying artificially generated data after adding too much noise. This is an advantage for DEBuNk since the quality threshold is able to remove nonsensical patterns from the final set. In contrast, from these experiments, we observe that Quick-DEBuNk less robust to noise than DEBuNk. The performance of Quick-DEBuNk in terms of finding hidden patterns decreases faster with regard to the noise rate compared to DEBuNk. This is mainly due to the random walk procedure (RWC) which considers other sub search space than the one actually containing a hidden context as the noise reduces the quality of its subsuming parents. Still, it is worth mentioning that Quick-DEBuNk is able to retrieve partially planted patterns even when the noise is rather high. Interestingly, the sampling approach achieves a comparable precision to the exhaustive approach, this demonstrates that most of returned patterns are valid.

6.4 Performance study

6.4.1 Efficiency of closure operators and optimistic estimates

To evaluate the efficiency of closure operators and optimistic estimates, we resolve to compare DEBuNk against two baseline algorithms. The first baseline, named *Baseline*, is obtained by disabling both closure operators and the pruning properties supported by the defined optimistic estimates. Thus, *Basline* only pushes the anti-monotonic constraints. The second baseline, dubbed *Baseline+Closed*, is proposed to study more precisely the efficiency of the optimistic estimates. Thus, it is obtained by disabling the optimistic estimate based pruning. In this experiments, we interrupt a method if its execution time exceeds ten hours. Figures 10, 11 and 12 report the execution time and the number of evaluated patterns by each of the three methods (i.e. *Baseline*, *Baseline+Closed*, DEBuNk) when carried out on respectively EPD8, Movielens and Yelp datasets.

Experiments show evidence that the closure operator and the canonicity tests performed by EnumCC are effective as they substantially reduce the number of evaluated patterns. Additionally, DEBuNk is about one order of magnitude faster than *Baseline+Closed* algorithm, thanks to the optimistic estimate-based pruning. This especially happens when the IAS measure is a simple average, which is the case of the IAS measure used for EPD8, Yelp and Movielens. This is explained by the fact that the corresponding optimistic estimate is tight. Additional performance experiments on Openmedic are reported in Appendix B.

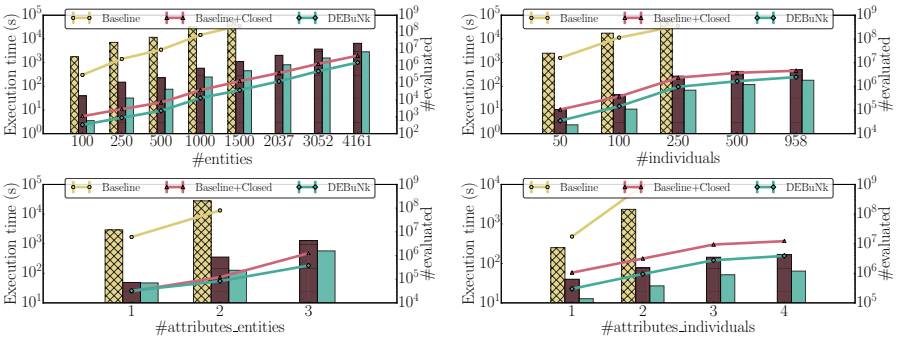


Fig. 10: Effectiveness of *DEBuNk* considering EPD8 Dataset with $|G_E| = 2000$, $|G_I| = 500$, $|Outcomes| = 750k$, $|\mathcal{A}_E| = 3$, $|\mathcal{A}_I| = 4$, $\sigma_E = 40$, $\sigma_I = 10$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$. Lines correspond to the execution time and bars correspond to the number of evaluated patterns.

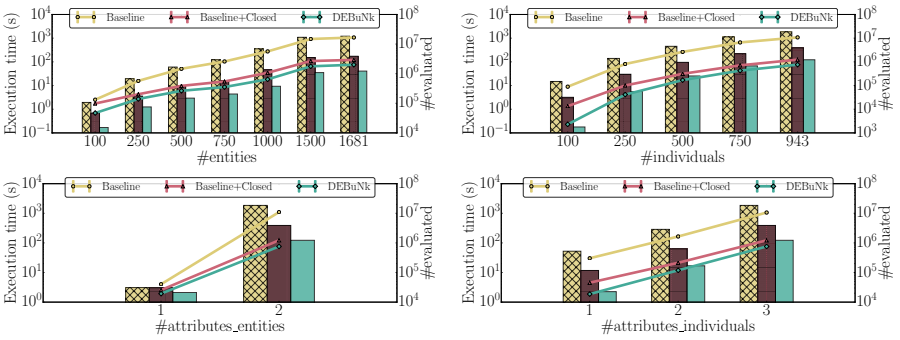


Fig. 11: Effectiveness of *DEBuNk* considering Movielens Dataset with $|G_E| = 1681$, $|G_I| = 943$, $|Outcomes| = 100k$, $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 3$, $\sigma_E = 8$, $\sigma_I = 50$, $\sigma_\varphi = 0.2$ and the quality measure $\varphi_{dissent}$

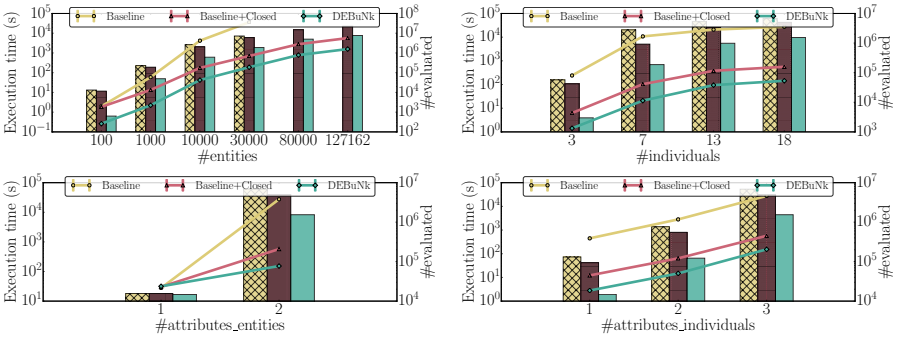


Fig. 12: Effectiveness of *DEBuNk* considering Yelp Dataset with $|G_E| = 25000$, $|G_I| = 18$, $|Outcomes| = 146k$, $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 3$, $\sigma_E = 5$, $\sigma_I = 1$, $\sigma_\varphi = 0.5$ and the quality measure $\varphi_{dissent}$

6.4.2 Efficiency of HMT closed descriptions vs. closed itemsets enumeration

In order to evaluate the performance of the closed descriptions enumeration in the presence of a taxonomy linking the tags (items), we study the behavior of DEBuNk (i.e. execution time and the number of explored patterns) both with and without leveraging the hierarchy between items. The latter can be done by scaling the HMT values (as illustrated in Fig. 2) using a vector representation for each tagged record. Experiments are carried out on EPD8 and Yelp datasets which entities are characterized respectively by a hierarchy of 347 tags and 1175 tags. To vary the number of items/tags constituting the hierarchy, we remove tags from the tree in a bottom-up fashion until the desired number of tags/items is reached, followed by replacing the HMT values of each entity by the set of ascendants tags remaining in the obtained tree.

Experiments reported in Figure 13 demonstrate that taking into account the hierarchy of tags significantly improves the performance of DEBuNk (5× faster). This results from the fact that, in contrast to itemsets enumeration, HMT descriptions enumeration exploits the structure of the hierarchy and therefore avoids considering chain descriptions (e.g. $\{1,\ 1.10.40\}$). Note that the bars depict the number of patterns that are visited by EnumCC used in DEBuNk to generate the closed patterns. Obviously, the HMT and Itemset closed description enumeration return the same number of closed patterns. We choose to represent the number of visited patterns rather than the number of closed patterns to explicit the differences between the HMT and Itemset enumeration in terms of the size of the explored search space.

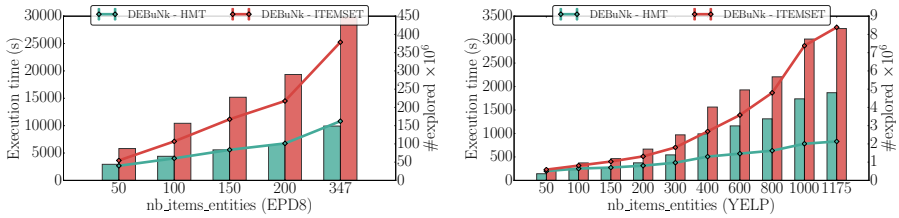


Fig. 13: Efficiency of HMT against itemsets closed descriptions enumeration according to the number of items/tags constituting the hierarchy for the two datasets EPD8 (left) and Yelp (right). For both datasets we only consider the HMT attribute for entities $|\mathcal{A}_E| = 1$. The used parameters for EPD8 are: $|\mathcal{A}_I| = 6$, $\sigma_E = 1$, $\sigma_I = 10$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$. The used parameters for Yelp are: $|\mathcal{A}_I| = 3$, $\sigma_E = 5$, $\sigma_I = 1$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$.

6.4.3 Performance study of DEBuNk

We now focus on the study of DEBuNk according to the size of the description spaces ($\mathcal{D}_E$, $\mathcal{D}_I$), the support thresholds, the quality threshold and the quality measures. To conduct the study of DEBuNk according to the size of the description spaces, we choose to vary the number of items resulting from

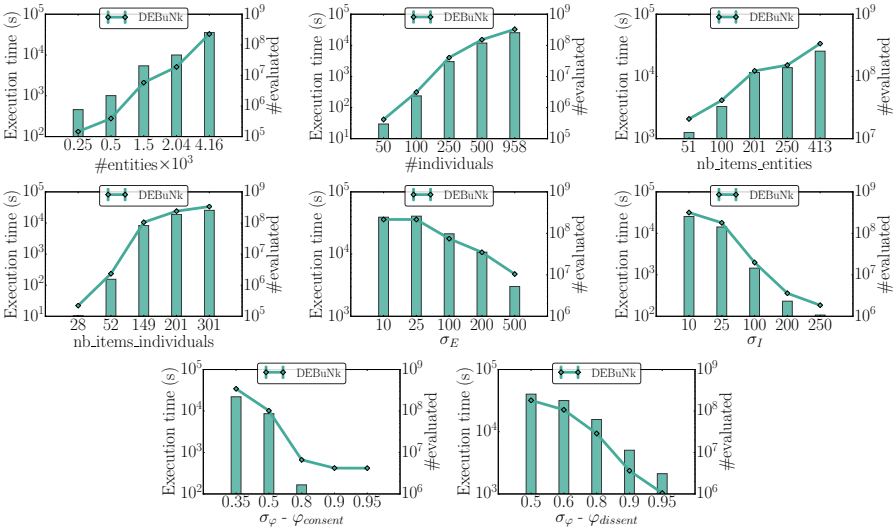


Fig. 14: Effectiveness of *DEBuNk* over EPD8 according to the sizes of E, I, $\mathcal{D}_E$, $\mathcal{D}_I$, the supports and quality measures thresholds. Considering by default $|G_E| = 4161$, $|G_I| = 958$, $|Outcomes| = 750k$, $|\mathcal{A}_E| = 3$, $|\mathcal{A}_I| = 6$. $\sigma_E = 40$, $\sigma_I = 10$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$.

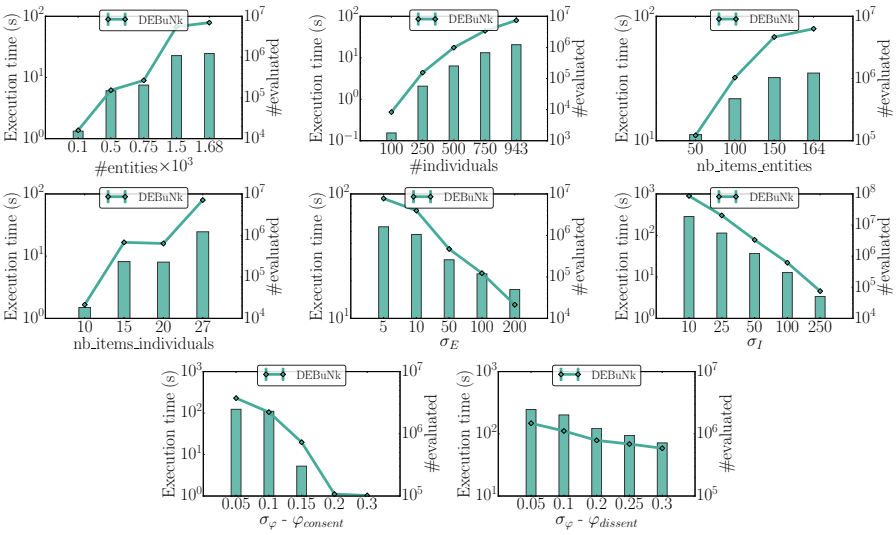


Fig. 15: Effectiveness of *DEBuNk* over Movielens Dataset according to the sizes of E, I, $\mathcal{D}_E$, $\mathcal{D}_I$, the supports and quality measures thresholds. Considering by default the full dataset. $\sigma_E = 8$, $\sigma_I = 50$, $\sigma_\varphi = 0.2$ and the quality $\varphi_{dissent}$

projecting the attributes values of each record (entity/individual) to their corresponding vector representation. To this end, we select values from each attribute according to the size of its corresponding domain so as to obtain the required number of items. We follow the same approach as in the experiments reported in Figure 13 to select the required number of tags for an HMT attribute. Numerical attributes domains are discretized according to the required number of items. Subsets of values of categorical attributes are regrouped under single categories in order to obtain the desired number of values.

Figures 14,15 and 16 report the behavior of DEBuNk when run on EPD8, Movielens and Yelp. Clearly, the number of evaluated patterns and the execution time increase with regards to the size of description spaces $\mathcal{D}_I$ and $\mathcal{D}_E$. The reported experiments confirm that pushing monotonic constraints (i.e. supports threshold σ_E, σ_I) improves drastically the efficiency of DEBuNk. Finally, a greater threshold on the quality σ_φ leads to an important reduction of the number of visited patterns and therefore to a better execution time. This demonstrates the effectiveness of the pruning properties enabled by the use of optimistic estimates. We also notice that $\varphi_{consent}$ performs slightly better than $\varphi_{dissent}$. This effect arises mainly from the fact that in the parliament, Movielens and Yelp datasets the overall observed inter-group agreement between groups of individuals is rather consensual. Similar results where observed when carrying out the experiments on Openmedic dataset. For more details, see Appendix B.

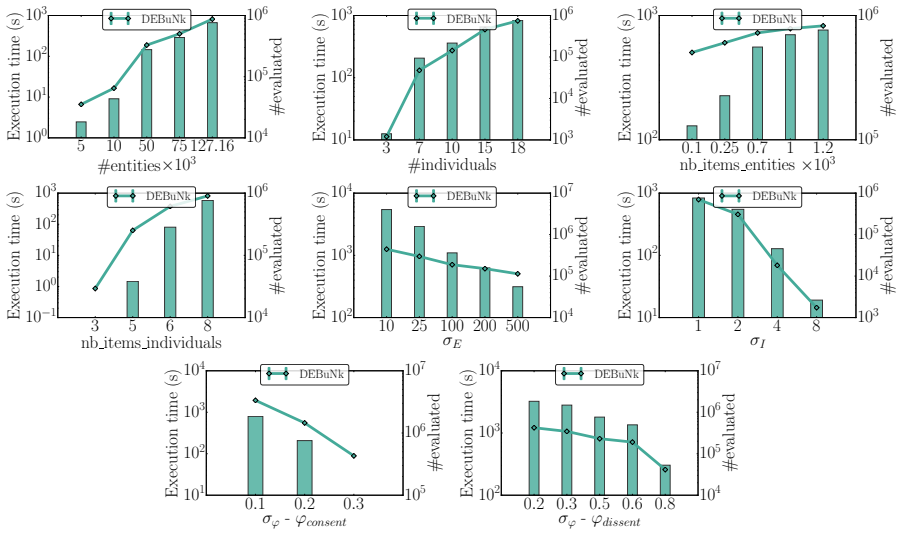
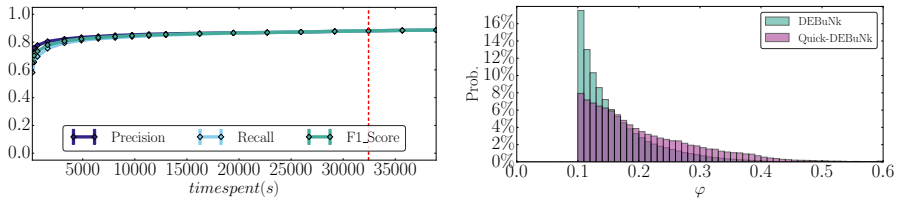


Fig. 16: Effectiveness of *DEBuNk* over Yelp Dataset according to the size of E, I, $\mathcal{D}_E$, $\mathcal{D}_I$, the supports and the quality measures thresholds. Considering by default the full dataset. $\sigma_E = 50$, $\sigma_I = 1$, $\sigma_\varphi = 0.5$ and the quality $\varphi_{dissent}$

6.4.4 Quick-DEBuNk vs. DEBuNk

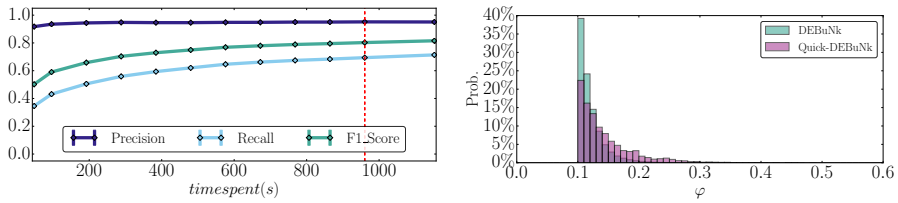
Since it is impossible to provide a ground-truth over real-world behavioral dataset, we evaluate the efficiency of *Quick-DEBuNk* over these type of datasets by comparing its results set against the one returned by the exhaustive search algorithm DEBuNk, this according to different time budgets. We use F-score measure, presented earlier in Equation 9, to capture to what extent the two patterns set returned by DEBuNk and Quick-DEBuNk provide similar insights about the change of pairwise behavior.

Figures 17a, 18a and 19a report the comparative study between DEBuNk and Quick-DEBuNk carried out on respectively EPD8, Movielens and Yelp. We notice that in all situations, Quick-DEBuNk is able to promptly returning high



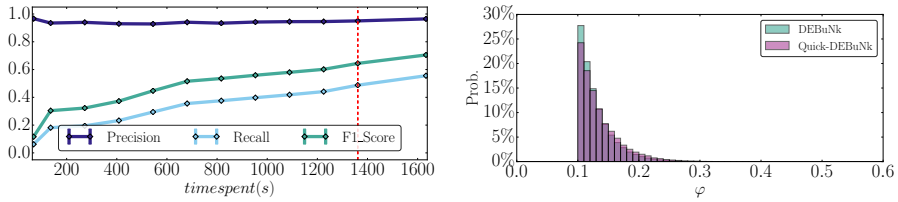
(a) $F_1(P_{\text{Quick-DEBuNk}}, P_{\text{DEBuNk}})$ **(b)** Empirical Distribution

Fig. 17: Efficiency of Quick-DEBuNk compared to DEBuNk over EPD8. Parameters used are $\sigma_E = 40$, $\sigma_I = 10$, $\sigma_\varphi = 0.5$ and $\varphi_{dissent}$. The red line in each figure correspond to the required time by DEBuNk to perform an exhaustive search.



(a) $F_1(P_{\text{Quick-DEBuNk}}, P_{\text{DEBuNk}})$ **(b)** Empirical Distribution

Fig. 18: Efficiency of Quick-DEBuNk compared to DEBuNk over MOVIELENS. Parameters used are $\sigma_E = 5$, $\sigma_I = 10$, $\sigma_\varphi = 0.25$ and $\varphi_{dissent}$. The red line in each figure correspond to the required time by DEBuNk to perform an exhaustive search.



(a) $F_1(P_{\text{Quick-DEBuNk}}, P_{\text{DEBuNk}})$ **(b)** Empirical Distribution

Fig. 19: Efficiency of Quick-DEBuNk compared to DEBuNk over Yelp. Parameters used are $\sigma_E = 15$, $\sigma_I = 1$, $\sigma_\varphi = 0.1$ and $\varphi_{dissent}$. The red line in each figure correspond to the required time by DEBuNk to perform an exhaustive search.

quality patterns. Interestingly, some differences can be observed from one to another dataset. For instance, Quick-DEBuNk is less efficient on Yelp dataset. We argue that this is due to the fact that the corresponding context search space is much larger than the three other behavioral datasets (see Table 3) which might impede random walk step RWC for finding high quality patterns.

We investigate also the empirical distribution from which the patterns are sampled from when using Quick-DEBuNk. This requires the true distribution of the qualities of valid patterns in the corresponding datasets. In this respect, we run DEBuNk by disabling the generality condition (see Problem definition). This is done to make it possible to identify all interesting (dis)agreement patterns in the dataset. In these experiments, we choose an arbitrary threshold set to $\sigma_\varphi = 0.1$. Similarly, we run Quick-DEBuNk so as to obtain a sufficiently large pattern set, and calculate the sampling distribution from the retrieved patterns' qualities. Clearly, we observe from the empirical distributions depicted in Figures 17b, 18b and 19b that Quick-DEBuNk rewards high quality patterns by giving them more chance to be sampled.

Finally, to evaluate the importance of RWC (Random Walk on Contexts) step in Quick-DEBuNk, we perform the same experiments with the same time budgets with RWC step disabled. In such configuration, Quick-DEBuNk without RWC returned only 3 472, 389 and 120 valid patterns compared to 408 610, 64 198 and 75 398 valid patterns when carried out on, respectively, EPD8, Movielens and Yelp. In average, Quick-DEBuNk without RWC retrieved 20× less valid patterns than the original Quick-DEBuNk. This clearly indicates that RWC improves the performance of Quick-DEBuNk. This stems from the fact that, when the first step (FBS step) generates a pattern, most of the time, the pattern is not of a sufficient quality. RWC tackles this issue by locally searching for interesting patterns, starting from the generated pattern.

6.5 Qualitative Results

We now focus on reporting example patterns discovered by the algorithm DEBuNk. We demonstrate the actionability of the provided patterns with three real world case studies: (i) In collaborative rating platforms (Yelp, Movielens), we study the affinities between groups of users with regard to their expressed ratings. (ii) In the voting system (European Parliament Dataset), we show how the voting behavior of parlementarians can provide interesting insights about the cohesion and the polarization between groups of parliamentarians in different contexts. Such information can be valuable for journalists and political analysts. (iii) Eventually, we give some example patterns reporting substantial differences in medicine consumption behavior between groups of individuals. Such results can be leveraged by epidemiologists to study comparative prevalence of sicknesses among different groups of individuals.

6.5.1 Study of Collaborative Rating Data

Table 8 describes some patterns returned by DEBuNk when applied over Movielens Dataset when looking for contexts that lead to a disagreement between groups of individuals labeled by their professional occupations. The first pattern describes that, while Student and Health professionals agree 74% of the time, they tend to disagree for Horror and Comedy Movies released between 1986 and 1994 (e.g. *Evil Dead II*, *Braindead*). Figure 20 illustrates the usual and the contextual ratings distribution of each of the two groups. We observe from this rating distributions, that the students like the movies highlighted by the pattern, whereas the healthcare professionals dislike them.

In Table 9, we present some results provided by DEBuNk over the Yelp dataset. The groups of individuals are labeled by the size of their friend network and their seniority in the Yelp platform. Notice that additional demo-

	(c, u_1, u_2)	$\lvert G_E^c \rvert$	$\lvert G_I^{u_1} \rvert$	$\lvert G_I^{u_2} \rvert$	$o(i,e)$	$\varphi_{dissent}$
1	Student vs. Healthcare in ['11 Horror', '5 Comedy'] [1986, 1994]	6	196	16	106	$0.42 = 0.74 - 0.33$
2	Student vs. Healthcare in ['5 Comedy'] [1991, 1991]	5	196	16	40	$0.41 = 0.74 - 0.33$
3	Healthcare vs. Artist in ['5 Comedy', '8 Drama'] [1987, 1993]	5	16	28	28	$0.42 = 0.73 - 0.3$
4	Lawyer vs. Executive in ['4 Children'] [1997, 1997]	5	12	32	27	$0.42 = 0.8 - 0.38$
5	Executive vs. Artist in ['7 Documentary'] [1996, 1997]	8	32	28	24	$0.41 = 0.77 - 0.36$

Table 8: Top-5 w.r.t. number of expressed outcomes ($o(i,e)$ *column*) of Relevant (dis)agreement patterns discovered over Movielens considering by default the full dataset, $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 1$, $\sigma_E = 5$, $\sigma_I = 10$ and $\sigma_\varphi = 0.4$.

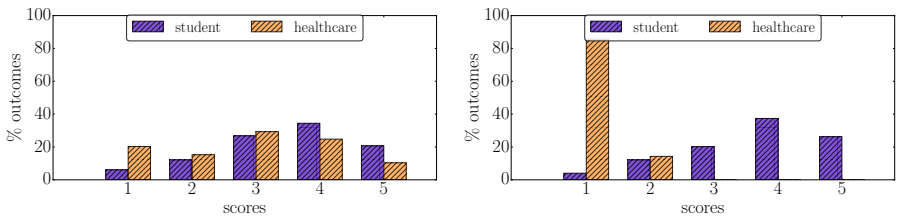


Fig. 20: Pattern 1 Illustration - distribution of ratings of individuals constituting the group of students versus distribution of ratings of individuals constituting the group of health professionals. Left figure corresponds to the usual distribution observed over all movies. Right figure corresponds to the contextual distribution observed over the context highlighted by pattern 1

graphic data about users are missing. This prevents DEBuNk from obtaining concrete results similar to the ones obtained in Movielens. The resulting patterns highlight the places for which groups of individuals have divergent opinions. For example, the second pattern in Table 9 states that Senior Yelp users (registered in Yelp before 2010) having a friend network of medium size (less than 100 friends) disagree with users registered in Yelp before 2015 having a large friend network (more than 100 friends) on Internal Medicines Clinics in Nevada (e.g. University Urgent Care, Las Vegas Urgent Care), contrary to the usual, where these two groups roughly share the same opinions about places in general (81% of the time). Figure 21 gives the overall rating distribution corresponding to the second pattern. Note that in Table 9, the values shown in columns ($|G_I^{u_1}|$, $|G_I^{u_2}|$) designating respectively the size of the groups u_1, u_2 correspond to the number of their composing aggregated groups. For instance, the (Senior, Medium) group is made up of two high granularity groups (elite and non-elite) which in reality is comprised of 44927 registered users.

	(c, u_1, u_2)	$\|G_E^c\|$	$\|G_I^{u_1}\|$	$\|G_I^{u_2}\|$	$o(i,e)$	$\varphi_{dissent}$
1	(Newcomer,*) vs. (Middler,*) in ['03 Automotive', '14.22 Electronics Repair', '22.06 Battery Stores', '22.21 Electronics'] *	10	6	6	43	$0.4 = 0.8 - 0.4$
2	(Senior, Medium) vs. (Middler, Large) in ['10.55.21 Internal Medicine'] NV	15	2	2	39	$0.43 = 0.81 - 0.38$
3	(Newcomer, Medium) vs. (Middler, Large) ['11.59.01 Apartments', '11.59.18 University Housing'] AZ	14	2	2	30	$0.4 = 0.78 - 0.38$
4	(*, Small) vs. (Middler, Large),in ['10.55.50 Urologists'] *	10	6	2	30	$0.43 = 0.79 - 0.36$
5	(*, Large) vs. (Newcomer,*) in ['08 Financial Services', '22 Shopping'] AZ	12	6	6	30	$0.4 = 0.79 - 0.39$

Table 9: Top-5 w.r.t. number of expressed outcomes ($o(i,e)$ *column*) of Relevant (dis)agreement patterns discovered over Yelp considering by default the full dataset, $|\mathcal{A}_E| = 2$, $|\mathcal{A}_I| = 2$, $\sigma_E = 10$, $\sigma_I = 1$ and $\sigma_\varphi = 0.4$.

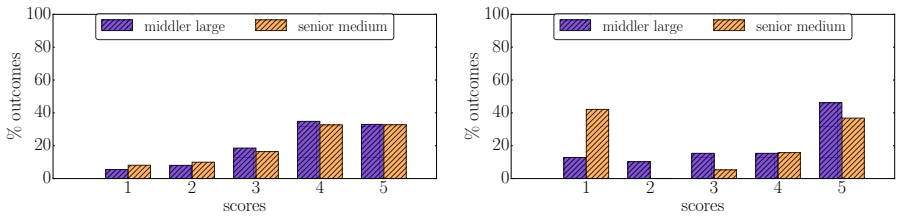


Fig. 21: Pattern 2 Illustration - distribution of ratings of individuals constituting the group of Yelp user registered in between 2010-2015 having a large friend network versus distribution of ratings of individuals constituting the group of Yelp users registered before 2010 having a medium friend network. Left figure corresponds to the usual distribution observed over all Yelp places. Right figure corresponds to the contextual distribution observed over Nevada's Internal Medicines Places.

6.5.2 Analysis of the Voting Behavior in the European Parliament Dataset

Table 10 exposes patterns obtained by DEBuNk where the aim is to find contexts (subsets of ballots) that lead groups of parlementarians (labeled by their countries and their corresponding date of accession to the European Union) to strong disagreement compared to the usual observed inter-group agreement. Such analysis can be valuable to political analysts and journalists as it

	(c, u_1, u_2)	$\lvert G_E^c \rvert$	$\lvert G_I^{u_1} \rvert$	$\lvert G_I^{u_2} \rvert$	$o(i,e)$	$\varphi_{dissent}$
1	([1973, 1973] United Kingdom) vs. (*,*) ['4 Economic, social & territorial cohesion', '8.70 Budget of the Union']	47	88	958	30255	0.54 = 0.68 − 0.14
2	([1973, 1973] United Kingdom) vs. (*,*) ['4.15.05 Industrial restructuring, job losses, Globalization Adjustment Fund']	47	88	958	30250	0.54 = 0.68 − 0.14
3	([1958, 1958] Italy) vs. ([1981, 2013] *) ['3.40 Industrial policy', '6.20.02 Export /import control, trade defence']	79	99	433	29501	0.51 = 0.87 − 0.35
4	([1958, 1995] *) vs. ([1973, 2013] *) ['3.40.16 Raw materials']	44	709	547	28989	0.55 = 0.91 − 0.36
5	([1958, 1995] *) vs. ([1973, 2013] *) ['6.20 Common commercial policy' , '6.30 Development cooperation']	38	709	547	25268	0.51 = 0.91 − 0.39

Table 10: Top-5 w.r.t. number of expressed outcomes ($o(i,e)$ *column*) of Relevant (dis)agreement patterns discovered over EPD8 considering by default the full dataset, $|\mathcal{A}_E| = 1$, $|\mathcal{A}_I| = 2$, $\sigma_E = 25$, $\sigma_I = 1$ and $\sigma_\varphi = 0.5$ using $\varphi_{dissent}$. It is important to note that we choose carefully $\sigma_E \geq 25$ to reach subgroups of the third level of the themes hierarchy which on average contain $\backsim 25$ ballots.

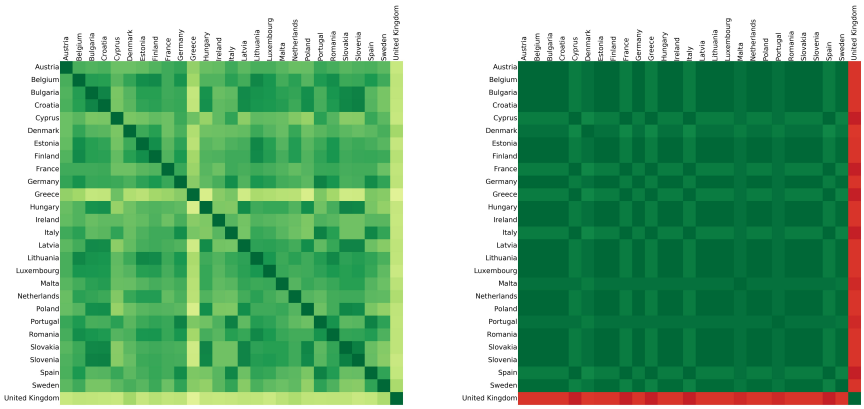


Fig. 22: Illustration of Pattern 2 reported in Table 10. Left matrix depicts the inter-group agreement observed in general between countries when considering all ballots. The right matrix correspond to the itner-agreement between countries for the context pointed out by Pattern 1 = (['4.15.05 Industrial restructuring, job losses and EGF'], UK, *)

enables to uncover subjects/thematics of votes on which countries have divergent points of view. For instance, the second pattern in Table 10 illustrated in Figure 22, states that the ballots concerning theme 4.15.05 (Industrial Restructuring, job losses, EGF) lead to strong disagreements of parlementarians from the United Kingdom with their peers. In Figure 22, we choose to visualize the second pattern by a similarity matrix where each cell represents the similarity between two countries. This can be seen as post-processing step where the end-user chooses to enrich the pattern with more related information (similarities between other countries). Such augmented visualization brings more context to the pattern. While the second pattern conveys that UK parlementarians are in strong disagreement with their peers, the visualization goes beyond by reporting that all other countries formed a coalition against UK.

Algorithms elaborated in this work also enable the discovery of consensual subjects, thanks to the quality function $\varphi_{consent}$. In Table 11 , we report patterns where groups of parliamentarians agree more than what is observed in general. For example, pattern 1 of Table 11 shows that while *Socialists and Democrats* (S&D - *left-wing*) parlementarians are usually in disagreement ($\text{IAS}_{voting} = 0.41$) with European Conservatives and Reformists (ECR - *right-wing*), they tend to have convergent opinions ($\text{IAS}_{voting} = 0.9$) on ballots concerning theme 6.20.03 (bilateral agreement and relations with countries external to the union, e.g. Implementation of the Free Trade Agreement between the European Union and the Republic of Korea). In Figure 23, we illustrate the similarities between political groups for pattern 3 reported in Table 11. It is worth to note that, as part of a collaboration with political journalists, we provide an online tool[21], dubbed ANCORE[45], which makes it possible to analyze European parliament voting sessions.

	(c, u_1, u_2)	$\lvert G_E^c \rvert$	$\lvert G_I^{u_1} \rvert$	$\lvert G_I^{u_2} \rvert$	$o(i,e)$	$\varphi_{consent}$
1	S&D vs. ECR in ['6.20.03 Bilateral economic and trade agreements and relations']	185	211	103	43162	$0.41 = 0.9 - 0.49$
2	PPE vs. GUE/NGL ['8.70.03.03 2013 discharge']	137	263	60	33664	$0.41 = 0.85 - 0.43$
3	ENF vs. * ['3', '8 State & evolution of the Union']	42	48	958	27191	$0.4 = 0.69 - 0.29$
4	GUE/NGL vs. * ['4.10.04 Gender equality', '4.15.08 Employment, wages and salaries']	41	60	958	25553	$0.41 = 0.98 - 0.57$
5	ECR vs. * ['1.20.09 Protection of privacy', '7 Area of freedom, security & justice']	39	103	958	25189	$0.4 = 0.97 - 0.57$

Table 11: Top-5 w.r.t. number of expressed outcomes ($o(i,e)$ *column*) of relevant (dis)agreement patterns discovered over European Parliament Dataset considering by default the full dataset, $|\mathcal{A}_E| = 1$, $|\mathcal{A}_I| = 1$, $\sigma_E = 15$, $\sigma_I = 1$ and $\sigma_\varphi = 0.4$ using $\varphi_{consent}$.

[21] http://contentcheck.liris.cnrs.fr

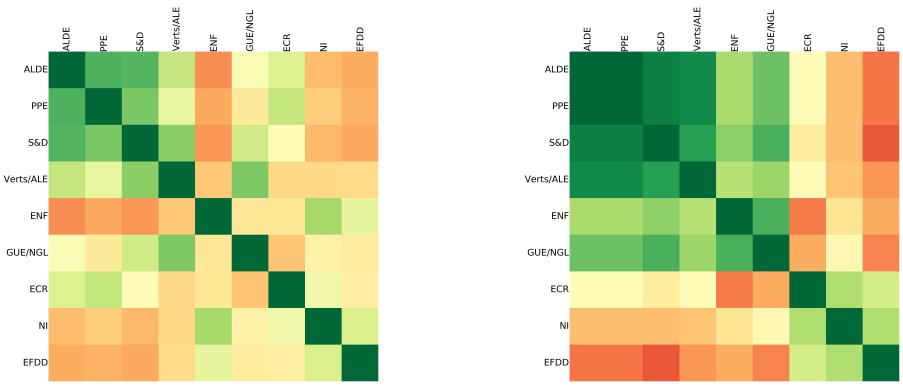


Fig. 23: Illustration of pattern 3 reported in Table 11. The left matrix depicts the agreement observed in general between political groups when considering all ballots. The right matrix corresponds to the inter-group agreement between groups for the context pointed out by pattern 3. We observe that group ENF is in disagreement with ALDE, PPE and S&D who hold 63% of the seats in the 8th European Parliament. The context of Pattern 3, which mainly covers EGF (European Globalisation Adjustment Fund) ballots, suggests an agreement between group ENF and the majority.

6.5.3 Illnesses Prevalence on the Basis of Medicine Consumption

One interesting analysis task to be carried out on *Openmedic* Dataset is to look for subgroups of drugs where the ratio of consumption between two groups of individuals is substantially different than the one usually observed. For instance, we found that while *Females* takes 1.32× more drugs than *Males* in overall terms, this ratio icreases up to 5× when considering drugs prescribed for *Hyperthyroidism* (see Pattern 3 in Table 12). These results are similar to what reports an epidemiology study by Wang et Al. in [66]. Such task can provide insightful hypothesis regarding some sicknesses prevalence for particular groups of individuals. In the behavioral dataset *Openmedic*, the outcomes of individuals are depicted by numerical values reporting the count of drug boxes. As we are interested in characterizing the inter-group agreement by the consumption ratio, we instantiate IAS as follows:

$$\mathrm{IAS}_{ratio}(c, u_1, u_2) = \frac{\sum_{e \in G_E^c} \theta_{avg}(G_I^{u_1}, e)}{\sum_{e \in G_E^c} \theta_{avg}(G_I^{u_2}, e)}$$

This ratio falls under the definition of IAS considered in *Definition 5* as it can be rewritten in the form of a weighted average.

$$\begin{aligned}\mathrm{IAS}_{ratio}(c,u_1,u_2) &= \frac{\sum_{e\in G_E^c}\theta_{avg}(G_I^{u_2},e)\times\frac{\theta_{avg}(G_I^{u_1},e)}{\theta_{avg}(G_I^{u_2},e)}}{\sum_{e\in G_E^c}\theta_{avg}(G_I^{u_2},e)}\\ &= \frac{\sum\limits_{e\in G_E^c} w(e,G_I^{u_1},G_I^{u_2})\times sim_{ratio}(\theta_{avg}(G_I^{u_1},e),\theta_{avg}(G_I^{u_2},e))}{\sum_{e\in G_E^c} w(e,G_I^{u_1},G_I^{u_2})}\\ &\textit{with } w(e,G_I^{u_1},G_I^{u_2}) = \theta_{avg}(G_I^{u_2},e) \textit{ and } sim_{ratio}(x,y) = \frac{x}{y}\end{aligned}$$

In order to provide interpretable patterns according to the aim of the study, we define an adapted quality measure φ_{ratio} as :

$$\varphi_{ratio}(p) = \frac{\mathrm{IAS}_{ratio}(p)}{\mathrm{IAS}_{ratio}(p^*)} \quad \textit{with } p = (c,u_1,u_2) \in \mathcal{P} \textit{ and } p^* = (*,u_1,u_2)$$

Drug boxes are labeled by tags in the ATC[22] classification system. We aim at leveraging the medical consumption differences between groups of individuals to investigate the comparative prevalence[23] of illnesses between gender groups. Table 12 gives an example of patterns returned by DEBuNk when applied on Openmedic. For instance, Pattern 4 states that, for drugs prescribed for *Gout sickness*[24], Men consume 3× more drugs than Women, whereas in overall terms, Men consume 0.76× less drugs than Women. Similar results were reported by an epidemiology study of *Gout* in [60] giving an incidence of gout per 1,000 person-years of 1.4 in women and 4.0 in men. Patterns 3 and 4, depicted in Figure 24, report details on the differences between the two gender groups in terms of population size and number of medicines consumed both in overall and in the context highlighted by the pattern.

	(c,u_1,u_2)	$\lvert G_E^c\rvert$	$\lvert G_I^{u_1}\rvert$	$\lvert G_I^{u_2}\rvert$	$o(i,e)$	φ_{ratio}
1	Men vs. Women in N07B - Drugs used in addictive disorders	138	39	39	4195	$4.59 = \frac{3.48}{0.76}$
2	Women vs. Men in A12A - Calcium	54	39	39	3174	$3.96 = \frac{5.21}{1.32}$
3	Women vs. Men in H03 - Thyroid Therapy	31	39	39	1981	$3.89 = \frac{5.13}{1.32}$
4	Men vs. Women in M04A - Antigout preparations	42	39	39	1940	$3.91 = \frac{2.97}{0.76}$

Table 12: Top-4 w.r.t. the number of expressed outcomes on Openmedic considering by default the full dataset, $|\mathcal{A}_E| = 1$, $|\mathcal{A}_I| = 1$, $\sigma_E = 10$, $\sigma_I = 1$ and $\sigma_\varphi = 3.5$ using φ_{ratio}. It is important to note that we choose carefully $\sigma_E \geq 10$ to reach subgroups of medicines of the fifth level of ATC tree which on average contain $\backsim$ 10 medicines.

[22] ATC: the Anatomical Therapeutic Chemical classification system classifies therapeutic drugs according to the organ or system on which they act and their chemical, pharmacological and therapeutic properties - http://www.who.int/classifications/atcddd/en/

[23] http://www.med.uottawa.ca/sim/data/epidemiology_rates_e.htm

[24] https://www.medicinenet.com/gout_gouty_arthritis/article.htm

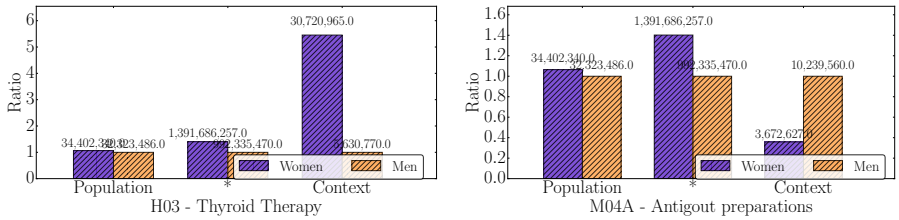


Fig. 24: Drugs consumption behavior of gender groups in Patterns 3 (left) and 4 (right).

6.6 Discussion

DEBuNk scales well w.r.t. the size of the search space corresponding to the entities collection thanks to the defined optimistic estimates which enable to prune unpromising parts of the search space. However, DEBuNk does not scale according to the size of the description spaces related to the individuals. This limits its application when behavioral datasets have a large number of individuals described with many attributes. This is due the need of taking into account the usual inter-group agreement in the interestingness measures. As a consequence, it is notoriously difficult to define an optimistic estimate which not only works on the entities related search space, but also on the one corresponding to the confronted couples of groups of individuals. This should be the scope of future research, starting with definition of bounds on the usual agreement quantity. Algorithm Quick-DEBuNk partially addresses this scalability issue by sampling the couples of groups directly from the patterns space rather than starting from the search tree root. Interestingly, the experiments demonstrated that Quick-DEBuNk makes it possible to retrieve most of the interesting patterns in a relatively small amount of time (i.e. compared to what returns the exhaustive search algorithm DEBuNk and the ground truth in artificial data). This is particularly observed for EPD8 dataset involving the largest descriptions space $\mathcal{D}_I \times \mathcal{D}_I$, hence empirically demonstrating its interest. Nevertheless, Quick-DEBuNk does not have theoretical guarantees on the distribution of the sampled patterns (we only proved that all valid patterns are reachable and are generated proportionally to their size). This shortcoming is due to two reasons. On the one hand, the three-set format of the patterns makes them challenging to be sampled proportionally to their interestingness measure since the value is computed only when the context is known (no information is available before the instantiation of the two groups). On the other hand, quality measures that are expressed as average functions are complex to apprehend under direct pattern sampling framework. Dealing with this two issues is required to obtain theoretical guarantees.

To avoid misleading interpretations, it is important to be aware of the data sparsity. Remind that the proposed approaches enable to discard some patterns that involve too small subset of entities on which the two confronted groups haven't expressed enough outcomes. Moreover, the strength of the claim related to the pattern should be assessed according not only to the data sparsity

but also to the representativeness of the two subpopulation of interest (e.g. the claims drawn from the EU parliament votes are usually consistent even though the data are fairly sparse).

7 Related Work

Scientists have always seen Exploratory Data Analysis (EDA) as an important research area since its introduction [65]. Among the various EDA techniques that aim to maximize insight into datasets and uncover underlying structures, Subgroup Discovery (SD) [42,67,4,36] is a generic data mining task concerned with finding regions in the data that stand out with respect to a given target. Many other data mining tasks have similar goals as SD, e.g., emerging patterns [19], significant rules [64], contrast sets [7] or classification association rules [53]. However, among these different tasks, SD is known as the most generic one, especially SD is agnostic of the data and the pattern domain. For instance, subgroups can be defined with a conjunction of conditions on symbolic [46] or numeric attributes [32,6] as well as sequences [31]. Furthermore, the single target can be discrete or numeric [49]. Exceptional Model Mining (EMM) [48], while sharing exactly the same exploration space (i.e., the description space), extends SD by offering the possibility to handle complex targets, e.g., several discrete attributes [47,22,21], graphs [41,10,9], two numeric targets [20] and preferences [63,62]. Our method is rooted in the SD/EMM framework. Nevertheless, the problem we tackle cannot be directly addressed with an instance of SD/EMM because a target space is provided instead of explicit targets. As a consequence, the discovery of (dis)agreement patterns with a SD/EMM instance would consist in performing a SD discovery algorithm per pair of confronted groups of individuals, which is not feasible in practice due to the exponential number of possible pairs of groups. Dynamic EMM/SD (i.e., EMM/SD with a non-fixed model) has been recently investigated for different aims. Bosc et al. [15] propose a method to handle multi-label data where the number of labels per record 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 [41,10,9]. This work, which is an extension of [8], is the first attempt to discover (dis)agreement patterns with a method rooted in dynamic SD/EMM. [8] is extended on many levels: (1) we provide an easier to use framework to discover exceptional (dis)agreement between groups which requires less parameter setting and interpretation efforts by the end-user. (2) Our proposal enables to use a wider spectrum of interestingness measures that can be enriched by relying on the building blocks discussed in Section 3. (3) This work provides a more elaborate exhaustive search algorithm compared to the former algorithm as discussed in Section 4 and (4) An alternative sampling approach Quick-DEBuNk is proposed. It enables instant mining of (dis)agreement patterns, which sets the ground for interactive pattern mining tools.

Behavioral data analysis has received a wide interest in the last two decades allowing the development of new services for consumers, citizens, companies and organizations. In [18], the problem of discovering meaningful ratings interpretation is introduced. It can be formalized as a SD problem, the authors' aim is to identify groups of users that substantially agree or disagree w.r.t. some given subset of entities while using a mono-objective measure (ratings average). Extensions have been proposed to enable multi-objective groups identification thanks to more complex statistical measure (rating distribution) [3,57]. This makes it possible the discovery of intra-group behavior patterns such as polarized and homogeneous opinions. The main differences with our work are: (i) these methods consider intra-group agreement only (no inter-group agreement) without taking into account the usual agreement observed between the individuals; (ii) the set of reviewees on which the study is performed is given in prior, in contrast to our SD/EMM based approach, which discovers relevant contexts by leveraging the reviewees dimension.

Similarly, the two past decades have witnessed an increasing emergence of Open Government Data[25] (OGD) promoting transparency and accountability in public institutions. Consequently, many researchers from different fields (e.g., information science, political and social sciences, data mining and machine learning) have studied such data [16]. For instance, [38] uses hierarchical clustering and PCA to identify cohesion blocs and dissimilarity blocs of voters within the US Senate. Similar work was done on the Finnish [59], the Italian [2] and the Swiss [24] parliaments to study the polarization and cohesion between parliamentarians. In the same spirit, [30] investigates the voting behavior of citizens instead of politicians relying on subgroup discovery. Our work goes further and supports the discovery of new insights in such data.

Monitoring the disease prevalence is an important task. Many researchers dedicated their effort to analyze the prevalence of diseases considering different sources of data. In [58], the authors highlight the importance of considering outpatient data (e.g. medical prescriptions) in such epidemiology studies. This motivates the analysis task proposed over Openmedic data. It provides an interesting additional tool in epidemiology surveillance applications by revealing substantial differences in medicine consumption between subpopulations.

The discovery of the complete set of interesting patterns (e.g., frequent, discriminant) has two disadvantages that limit the use of such methods in practice. It is time consuming to compute the complete set of solutions. Furthermore, this set can be absolutely huge and non-manageable for a human expert. To overcome this limitation, many approaches that can effectively sample the pattern space for interesting patterns have been proposed for a decade. These methods address some frequent or discriminant itemset mining tasks [12,29,52,56] offering some theoretical guarantees on the sampling quality or more generic ones [23,11,1]. In [23], the authors define the problem of sampling pattern sets and propose a method based on a SAT solver sampling solution. However, this approach only supports pattern languages that can be compactly

[25] http://www.oecd.org/gov/digital-government/open-government-data.htm

represented by binary variables such as itemsets. It requires the discretization of numerical attributes. Authors in [11,1] use a MCMC (Monte-Carlo Markov-Chain) based algorithm to achieve sampling with guarantees according to a desired probability distribution. Despite the generic nature and the interesting guarantees that MCMC algorithms provide, it requires a number of steps that grows exponentially in the input size to generate a single pattern [11]. This may prevent the user to obtain instant results. The problem we are interested in has several specificities. First, the search space involves attributes of different types (i.e., numerical, symbolical, HMT attributes) which prevents us to use sampling techniques based on itemset language. Second, the quality measure is not considered in the state-of-the-art methods that mainly support frequency and discriminative measures [12,13]. Finally, the method proposed in [55] for EMM is not suited to our problem since we have to simultaneously consider both description space and target space. Algorithm Quick-DEBuNk handles the specificity of the problem by combining an exploration step (i.e. direct sampling step from the pattern space) and an exploitation step while taking profit of the quality measures properties (i.e. random walk step on contexts search space). However, we have no theoretical guarantee on the quality of the sampled (dis)agreement patterns.

8 Conclusion

In this paper, we defined the problem of discovering exceptional (dis)agreement in behavioral data. The generic definition of behavioral data enables to encompass datasets featuring individuals and their outcomes on some entities whatever the application domain. The exceptional (dis)agreement patterns discovery is rooted in SD/EMM with a novel pattern domain and associated quality measures. However, the targets are not specified and have to be enumerated in our framework. We defined DEBuNk, a branch and bound algorithm which takes benefit from closure operators, properties of some descriptions space (as for HMT attributes) and (tight) optimistic estimates to efficiently enumerate the patterns. Alternatively, we devised Quick-DEBuNk, an algorithm that does not return the complete set of (dis)agreement patterns anymore but rather samples the space of patterns. We investigated several quality measures to assess inter-group agreement. The extensive experimental study we reported demonstrates the efficiency of our algorithms as well as their ability to provide new insights in three case-studies: (i) the investigation of contexts that impact the inter-agreement between parliamentarians, (ii) the characterization of affinities and contrasted opinions between reviewers in rating platforms and (iii) the study of prevalence of certain sicknesses that can be pointed out by high discrepancies between the medicine consumption rates of two subpopulations.

We believe that this work opens new directions for future research. This generic framework can be extended by paying a particular attention to the analysis of intra-group agreement within a group of individuals. It may sup-

port the discovery of contexts that divide a political group. This requires the definition and the integration of adapted similarity measures into the IAS (Inter-group Agreement Similarity) measure. For instance, the cohesion of a political group can be assessed by the "agreement index" [37], which is an application-specific measure to the study the European parliament. More generic measures, such as Krippendorff's alpha coefficient [35], could also be investigated. While our method is able to analyse behavioral datasets with large collections of entities (e.g., Yelp), tackling large collections of individuals still remains challenging to assure the scalability of both DEBuNk and Quick-DEBuNk. Indeed, the search space related to individuals does not have, according to our problem definition, properties that can be leveraged to prune unpromising parts of this search space. Another interesting future direction is to take into account the temporal dimension into the analysis of behavioral data. This can offer the opportunity to investigate how the relationship (e.g. inter-group agreement) between groups of individuals evolves through time. The study of this dynamics makes it possible to discover new insights in behavioral data.

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

References

1. Al Hasan, M., Zaki, M.J.: Output space sampling for graph patterns. Proceedings of the VLDB Endowment **2**(1), 730–741 (2009)
2. Amelio, A., Pizzuti, C.: Analyzing voting behavior in italian parliament: Group cohesion and evolution. In: ASONAM, pp. 140–146. IEEE (2012)
3. Amer-Yahia, S., Kleisarchaki, S., Kolloju, N.K., Lakshmanan, L.V., Zamar, R.H.: Exploring rated datasets with rating maps. In: WWW (2017)
4. Atzmueller, M.: Subgroup discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **5**(1), 35–49 (2015)
5. Atzmueller, M., Lemmerich, F.: Vikamine–open-source subgroup discovery, pattern mining, and analytics. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 842–845. Springer (2012)
6. Atzmüller, M., Puppe, F.: Sd-map - A fast algorithm for exhaustive subgroup discovery. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 6–17 (2006)
7. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. Data mining and knowledge discovery **5**(3), 213–246 (2001)
8. Belfodil, A., Cazalens, S., Lamarre, P., Plantevit, M.: Flash points: Discovering exceptional pairwise behaviors in vote or rating data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 442–458. Springer (2017)
9. Bendimerad, A., Cazabet, R., Plantevit, M., Robardet, C.: Contextual subgraph discovery with mobility models. In: International Workshop on Complex Networks and their Applications, pp. 477–489. Springer (2017)
10. Bendimerad, A.A., Plantevit, M., Robardet, C.: Unsupervised exceptional attributed sub-graph mining in urban data. In: ICDM, pp. 21–30 (2016). DOI doi:10.1109/ICDM.2016.0013. URL https://doi.org/10.1109/ICDM.2016.0013
11. Boley, M., Gärtner, T., Grosskreutz, H.: Formal concept sampling for counting and threshold-free local pattern mining. In: Proceedings of the 2010 SIAM International Conference on Data Mining, pp. 177–188. SIAM (2010)

12. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 582–590. ACM (2011)
13. Boley, M., Moens, S., Gärtner, T.: Linear space direct pattern sampling using coupling from the past. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 69–77. ACM (2012)
14. Bosc, G., Boulicaut, J.F., Raïssi, C., Kaytoue, M.: Anytime discovery of a diverse set of patterns with monte carlo tree search. Data Mining and Knowledge Discovery **32**(3), 604–650 (2018)
15. Bosc, G., Golebiowski, J., Bensafi, M., Robardet, C., Plantevit, M., Boulicaut, J.F., Kaytoue, M.: Local subgroup discovery for eliciting and understanding new structure-odor relationships. In: International Conference on Discovery Science, pp. 19–34. Springer (2016)
16. Charalabidis, Y., Alexopoulos, C., Loukis, E.: A taxonomy of open government data research areas and topics. Journal of Organizational Computing and Electronic Commerce **26**(1-2), 41–63 (2016)
17. Csisz, I., et al.: Information-type measures of difference of probability distributions and indirect observations. Studia Sci. Math. Hungar. **2**, 299–318 (1967)
18. Das, M., Amer-Yahia, S., Das, G., Yu, C.: Mri: Meaningful interpretations of collaborative ratings. PVLDB **4**(11), 1063–1074 (2011)
19. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 43–52. ACM (1999)
20. Downar, L., Duivesteijn, W.: Exceptionally monotone models—the rank correlation model class for exceptional model mining. Knowledge and Information Systems **51**(2), 369–394 (2017). DOI 10.1007/s10115-016-0979-z. URL https://doi.org/10.1007/s10115-016-0979-z
21. Duivesteijn, W., Feelders, A.J., Knobbe, A.: Exceptional model mining. Data Mining and Knowledge Discovery **30**(1), 47–98 (2016)
22. Duivesteijn, W., Knobbe, A.J., Feelders, A., van Leeuwen, M.: Subgroup discovery meets bayesian networks – an exceptional model mining approach. In: ICDM, pp. 158–167 (2010). DOI doi:10.1109/ICDM.2010.53
23. Dzyuba, V., van Leeuwen, M., De Raedt, L.: Flexible constrained sampling with guarantees for pattern mining. Data Mining and Knowledge Discovery **31**(5), 1266–1293 (2017)
24. Etter, V., Herzen, J., Grossglauser, M., Thiran, P.: Mining democracy. In: A. Sala, A. Goel, K.P. Gummadi (eds.) Proceedings of the second ACM conference on Online social networks, COSN 2014, Dublin, Ireland, October 1-2, 2014, pp. 1–12. ACM (2014). DOI 10.1145/2660460.2660476. URL https://doi.org/10.1145/2660460.2660476
25. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of rule learning. Springer Science & Business Media (2012)
26. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: H.S. Delugach, G. Stumme (eds.) Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings, *Lecture Notes in Computer Science*, vol. 2120, pp. 129–142. Springer (2001). DOI 10.1007/3-540-44583-8_10. URL https://doi.org/10.1007/3-540-44583-8_10
27. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
28. Garriga, G.C., Kralj, P., Lavrač, N.: Closed sets for labeled data. Journal of Machine Learning Research **9**(Apr), 559–580 (2008)
29. Giacometti, A., Soulet, A.: Frequent pattern outlier detection without exhaustive mining. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 196–207. Springer (2016)
30. Grosskreutz, H., Boley, M., Krause-Traudes, M.: Subgroup discovery for election analysis: a case study in descriptive data mining. In: International Conference on Discovery Science, pp. 57–71. Springer (2010)

31. Grosskreutz, H., Lang, B., Trabold, D.: A relevance criterion for sequential patterns. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 369–384 (2013). DOI doi:10.1007/978-3-642-40988-2_24. URL https://doi.org/10.1007/978-3-642-40988-2_24
32. Grosskreutz, H., Rüping, S.: On subgroup discovery in numerical domains. Data Min. Knowl. Discov. **19**(2), 210–226 (2009). DOI doi:10.1007/s10618-009-0136-3. URL https://doi.org/10.1007/s10618-009-0136-3
33. Grosskreutz, H., Rüping, S., Wrobel, S.: Tight optimistic estimates for fast subgroup discovery. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 440–456. Springer (2008)
34. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) **5**(4), 19:1–19:19 (2016). DOI 10.1145/2827872
35. Hayes, A.F., Krippendorff, K.: Answering the call for a standard reliability measure for coding data. Communication methods and measures **1**(1), 77–89 (2007)
36. Herrera, F., Carmona, C.J., González, P., Del Jesus, M.J.: An overview on subgroup discovery: foundations and applications. Knowledge and information systems **29**(3), 495–525 (2011)
37. Hix, S., Noury, A., Roland, G.: Power to the parties: cohesion and competition in the european parliament, 1979–2001. British Journal of Political Science **35**(2), 209–234 (2005)
38. Jakulin, A.: Analyzing the us senate in 2003 : Similarities , networks , clusters and blocs (2004). URL Availableathttp://kt.ijs.si/aleks/Politics/us_senate.pdf
39. Johnson, D., Sinanovic, S.: Symmetrizing the kullback-leibler distance. IEEE Transactions on Information Theory (2001)
40. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Information Sciences **181**(10), 1989–2001 (2011)
41. Kaytoue, M., Plantevit, M., Zimmermann, A., Bendimerad, A., Robardet, C.: Exceptional contextual subgraph mining. Machine Learning pp. 1–41 (2017)
42. Klosgen, W.: Explora: A multipattern and multistrategy discovery assistant. Advances in knowledge discovery and data mining (1996)
43. Kralj Novak, P., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. J. Mach. Learn. Res. **10** (2009)
44. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence **14**(2-3), 189–216 (2002)
45. de Lacombe, C., Morel, A., Belfodil, A., Portet, F., Labbé, C., Cazalens, S., Plantevit, M., Lamarre, P.: Analyse de comportements relatifs exceptionnels expliquée par des textes : les votes du parlement européen. In: Extraction et Gestion des connaissances, EGC 2019, Metz, France, January 21-25, 2019, pp. 437–440 (2019)
46. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with cn2-sd. Journal of Machine Learning Research **5**(Feb), 153–188 (2004)
47. van Leeuwen, M., Knobbe, A.J.: Diverse subgroup set discovery. Data Min. Knowl. Discov. **25**(2), 208–242 (2012). DOI doi:10.1007/s10618-012-0273-y. URL https://doi.org/10.1007/s10618-012-0273-y
48. Leman, D., Feelders, A., Knobbe, A.: Exceptional model mining. In: ECMLPKDD, pp. 1–16. Springer (2008)
49. Lemmerich, F., Atzmueller, M., Puppe, F.: Fast exhaustive subgroup discovery with numerical target concepts. Data Min. Knowl. Discov. **30**(3), 711–762 (2016). DOI doi:10.1007/s10618-015-0436-8. URL https://doi.org/10.1007/s10618-015-0436-8
50. Lemmerich, F., Becker, M.: pysubgroup: Easy-to-use subgroup discovery in python. In: U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, N. Hurley (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part III, *Lecture Notes in Computer Science*, vol. 11053, pp. 658–662. Springer (2018). DOI 10.1007/978-3-030-10997-4_46. URL https://doi.org/10.1007/978-3-030-10997-4_46

51. Lemmerich, F., Rohlfs, M., Atzmueller, M.: Fast discovery of relevant subgroup patterns. In: FLAIRS Conference (2010)
52. Li, G., Zaki, M.J.: Sampling frequent and minimal boolean patterns: theory and application in classification. Data Mining and Knowledge Discovery **30**(1), 181–225 (2016)
53. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD, pp. 80–86 (1998). URL http://www.aaai.org/Library/KDD/1998/kdd98-012.php
54. Meeng, M., Knobbe, A.J.: Flexible enrichment with cortana – software demo. In: Proceedings Benelearn, pp. 117—119 (2011)
55. Moens, S., Boley, M.: Instant exceptional model mining using weighted controlled pattern sampling. In: International Symposium on Intelligent Data Analysis, pp. 203–214. Springer (2014)
56. Moens, S., Goethals, B.: Randomly sampling maximal itemsets. In: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, pp. 79–86. ACM (2013)
57. Omidvar-Tehrani, B., Amer-Yahia, S., Dutot, P.F., Trystram, D.: Multi-objective group discovery on the social web. In: ECMLPKDD (2016)
58. Orueta, J.F., Nuño-Solinis, R., Mateos, M., Vergara, I., Grandes, G., Esnaola, S.: Monitoring the prevalence of chronic conditions: which data should we use? BMC health services research **12**(1), 365 (2012)
59. Pajala, A., Jakulin, A., Buntine, W.: Parliamentary group and individual voting behavior in finnish parliamentin year 2003 : A group cohesion and voting similarity analysis (2004)
60. Roddy, E., Doherty, M.: Epidemiology of gout. Arthritis research & therapy **12**(6), 223 (2010)
61. Roman, S.: Lattices and ordered sets. Springer Science & Business Media (2008)
62. de Sá, C.R., Duivesteijn, W., Azevedo, P., Jorge, A.M., Soares, C., Knobbe, A.: Discovering a taste for the unusual: exceptional models for preference mining. Machine Learning **107**(11), 1775–1807 (2018)
63. de Sá, C.R., Duivesteijn, W., Soares, C., Knobbe, A.J.: Exceptional preferences mining. In: DS, pp. 3–18 (2016). DOI doi:10.1007/978-3-319-46307-0_1
64. Terada, A., Okada-Hatakeyama, M., Tsuda, K., Sese, J.: Statistical significance of combinatorial regulations. Proceedings of the National Academy of Sciences **110**(32), 12,996–13,001 (2013)
65. Tukey, J.W.: Exploratory data analysis. Addison-Wesley series in behavioral science : quantitative methods. Addison-Wesley (1977). URL http://www.worldcat.org/oclc/03058187
66. Wang, C., Crapo, L.M.: The epidemiology of thyroid disease and implications for screening. Endocrinology and Metabolism Clinics **26**(1), 189–218 (1997)
67. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: European Symposium on Principles of Data Mining and Knowledge Discovery, pp. 78–87. Springer (1997)

Symbol	Definition
G_E	A finite collection of records depicting entities
G_I	A finite collection of records depicting individuals
O	the domain of possible outcomes
o	function returning the outcome of an individual over an entity
$\langle G_I, G_E, O, o\rangle$	A behavioral dataset
$\mathcal{A}_E$	Descriptive attributes of entities
$\mathcal{A}_I$	Descriptive attributes of individuals
θ	An outcome aggregation measure
sim	a similarity function between two aggregated outcomes
IAS	Inter-group Agreement Similarity Measure
φ	An interestingness measure
$\mathcal{D}_E$	The description domain of entities containing all possible contexts
$\mathcal{D}_I$	The description domain of individuals
G_E^d	A subgroup of entities supporting a description $d \in \mathcal{D}_E$
G_I^u	A subgroup of individuals supporting a description $u \in \mathcal{D}_I$
δ	A mapping function that binds an entity from G to a description in $\mathcal{D}$
$\mathcal{P}$	$= \mathcal{D}_E \times \mathcal{D}_I \times \mathcal{D}_I$ and denotes the pattern space
p	$= (c, u_1, u_2) \in \mathcal{P}$ depicts a (dis)agreement pattern
p^*	$= (*, u_1, u_2) \in \mathcal{P}$ depicts the referential (dis)agreement pattern related to some pattern $p = (c, u_1, u_2)$
P	$\subseteq \mathcal{P}$ denotes a pattern set
$\sqsubseteq$	read "less restrictive than" is a partial order between descriptions (resp. patterns) in some descriptions space $\mathcal{D}$ (resp. patterns space $\mathcal{P}$)
$\wedge$	the conjunction operator between descriptions in $\mathcal{D}$
η	the refinement operator which applies atomic refinements of a given description $d \in \mathcal{D}$, thereby yielding neighbor descriptions of d w.r.t. $\sqsubseteq$

Table 13: Symbol table

A Appendix: Proofs of Theorems and Propositions

Before giving the proof of the proposition 1 we present the following lemma.

Lemma 1 *Let $n \in \mathbb{N}^*$, $A = \{a_i\}_{1 \leq i \leq n}$ and $B = \{b_i\}_{1 \leq i \leq n}$ such that:*

$$\forall i \in 1..n-1 : 0 \leq a_i \leq a_{i+1}$$
$$\forall i \in 1..n-1 : 0 < b_{i+1} \leq b_i$$

we have:

$$\forall k \in 1..n : \frac{\sum_{i=1}^{k} a_i}{\sum_{i=1}^{k} b_i} \leq \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \leq \frac{\sum_{i=n-k+1}^{n} a_i}{\sum_{i=n-k+1}^{n} b_i}$$

Proof (Lemma 1) Using the same notations of the lemma, we know that:

$$\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} - \frac{\sum_{i=1}^{k} a_i}{\sum_{i=1}^{k} b_i}$$

is of the same sign of:

$$\left(\sum_{i=1}^{n} a_i\right) \times \left(\sum_{i=1}^{k} b_i\right) - \left(\sum_{i=1}^{k} a_i\right) \times \left(\sum_{i=1}^{n} b_i\right)$$

This above quantity is equal to:

$$\left(\sum_{i=1}^{k} a_i + \sum_{i=k+1}^{n} a_i\right) \times \left(\sum_{i=1}^{k} b_i\right) - \left(\sum_{i=1}^{k} a_i\right) \times \left(\sum_{i=1}^{k} b_i + \sum_{i=k+1}^{n} b_i\right)$$

Which is equal to

$$\left(\sum_{i=k+1}^{n} a_i\right) \times \left(\sum_{i=1}^{k} b_i\right) - \left(\sum_{i=1}^{k} a_i\right) \times \left(\sum_{i=k+1}^{n} b_i\right)$$

Using the lemma hypotheses (orders between a_i's and b_i's), we have:

$$\sum_{i=k+1}^{n} a_i \geq (n-k) \times a_k$$
$$\sum_{i=1}^{k} b_i \geq k \times b_k$$
$$\sum_{i=1}^{k} a_i \leq k \times a_k$$
$$\sum_{i=k+1}^{n} b_i \leq (n-k) \times b_k$$

Thus:

$$\left(\sum_{i=k+1}^{n} a_i\right) \times \left(\sum_{i=1}^{k} b_i\right) \geq (n-k) \times k \times a_k \times b_k$$
$$\left(\sum_{i=1}^{k} a_i\right) \times \left(\sum_{i=k+1}^{n} b_i\right) \leq (n-k) \times k \times a_k \times b_k$$

We conclude that

$$\left(\sum_{i=k+1}^{n} a_i\right) \times \left(\sum_{i=1}^{k} b_i\right) - \left(\sum_{i=1}^{k} a_i\right) \times \left(\sum_{i=k+1}^{n} b_i\right) \geq 0$$

Hence, we have:

$$\forall k \in 1..n : \frac{\sum_{i=1}^{k} a_i}{\sum_{i=1}^{k} b_i} \leq \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}$$

Similarly the inequality $\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \leq \frac{\sum_{i=n-k+1}^{n} a_i}{\sum_{i=n-k+1}^{n} b_i}$ can be easily proved following the same line of reasoning of the proof of the first part of the inequality.

□

Proof (Proposition 1) By a straightforward application of Lemma 1 we obtain for any d s.t. $|G_E^d| \geq \sigma_E$ the following inequality.

$$LB(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \tag{10}$$

This stems from the fact that $LB(G_E^d, G_I^{u_1}, G_I^{u_2})$ takes the sum of the lowest σ_E quantities constituting the numerator of $\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2})$ and divides them by the sum of the greatest σ_E quantities forming the denominator of $\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2})$.

Moreover, we have that LB is monotonic w.r.t. $\sqsubseteq$ of $\mathcal{D}_E$. i.e.

$$c \sqsubseteq d \Rightarrow LB(G_E^c, G_I^{u_1}, G_I^{u_2}) \leq LB(G_E^d, G_I^{u_1}, G_I^{u_2}) \tag{11}$$

This results from $c \sqsubseteq d \Rightarrow G_E^d \subseteq G_E^c$. Hence, if we reorder values of G_E^c and G_E^d where $G_E^c = \{e_1^c, ..., e_{|G_E^c|}^c\}$ and $G_E^d = \{e_1^d, ..., e_{|G_E^d|}^d\}$ as such:

$$\begin{cases} w_{e_1^c}.\alpha(e_1^c) \leq w_{e_2^c}.\alpha(e_2^c) \leq ... \leq w_{e_{\sigma_E}^c}.\alpha(e_{\sigma_E}^c) \leq ... \leq w_{e_{|E_c|}^c}.\alpha(e_{|G_E^c|}^c) \\ w_{e_1^d}.\alpha(e_1^d) \leq w_{e_2^d}.\alpha(e_2^d) \leq ... \leq w_{e_{\sigma_E}^d}.\alpha(e_{\sigma_E}^d) \leq ... \leq w_{e_{|G_E^d|}^d}.\alpha(e_{|G_E^d|}^d) \end{cases}$$

Given that $G_E^d \subseteq G_E^c$, it is clear that: $\forall i \leq \sigma_E \mid w_{e_i^c}.\alpha(e_i^c) \leq w_{e_i^d}.\alpha(e_i^d)$. Having that $m(G_E^c, \sigma_E) = \{e_1^c, ..., e_{\sigma_E}^c\}$ and $m(G_E^d, \sigma_E) = \{e_1^d, ..., e_{\sigma_E}^d\}$, it follows that:

$$\sum_{e \in m(G_E^c, \sigma_E)} w_e \times \alpha(e) \leq \sum_{e \in m(G_E^d, \sigma_E)} w_e \times \alpha(e) \tag{12}$$

Similarly, if we reorder entities e in descending order w.r.t the weights w_e we have $\forall j \leq \sigma_E \mid w_{e_j^d} \leq w_{e_j^c}$. Resulting in:

$$\sum_{e \in Mw(G_E^c, \sigma_E)} w_e \geq \sum_{e \in Mw(G_E^d, \sigma_E)} w_e \tag{13}$$

Hence, from (12) and (13) we have $\text{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) \leq \text{LB}(G_E^d, G_I^{u_1}, G_I^{u_2})$ and provided that $LB(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2})$ from (10), we have: $\forall c, d \in \mathcal{D}_E.\ \ c \sqsubseteq d \Rightarrow \text{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) \leq \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2})$

□

Proof (Proposition 2) This proof is similar to the proof of *Proposition 1*. For the sake of brevity, we give a *proof sketch*. By a direct application of Lemma 1, it is clear that for any d s.t. $|G_E^d| \geq \sigma_E$.

$$\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{UB}(G_E^d, G_I^{u_1}, G_I^{u_2}) \tag{14}$$

We have that UB is anti-monotonic w.r.t. $\sqsubseteq$ of $\mathcal{D}_E$. i.e.

$$c \sqsubseteq d \Rightarrow \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) \geq \text{UB}(G_E^d, G_I^{u_1}, G_I^{u_2}) \tag{15}$$

This results from $c \sqsubseteq d \Rightarrow G_E^d \subseteq G_E^c$. Thus,

$$\sum_{e \in M(G_E^c, \sigma_E)} w_e \times \alpha(e) \geq \sum_{e \in M(G_E^d, \sigma_E)} w_e \times \alpha(e) \textit{ and } \sum_{e \in mw(G_E^c, \sigma_E)} w_e \leq \sum_{e \in mw(G_E^d, \sigma_E)} w_e$$

Hence, given (14) and (15) it follows that:

$$\forall c, d \in \mathcal{D}_E.\ \ c \sqsubseteq d \Rightarrow \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) \qquad \square$$

Proof (Proposition 3) given $c, d \in \mathcal{D}_E$ such that $c \sqsubseteq d$, using *proposition 1* we have:

$$\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2})$$
$$\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) \leq \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2})$$

Since $\varphi_{\text{consent}}(G_E^d, G_I^{u_1}, G_I^{u_2}) = \max(\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}), 0)$ thus $\varphi_{\text{consent}}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{oe}_{\text{consent}}(G_E^c, G_I^{u_1}, G_I^{u_2})$
Similarly we have:

$$\text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \geq \text{LB}(G_E^c, G_I^{u_1}, G_I^{u_2})$$
$$\text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \text{LB}(G_E^c, G_I^{u_1}, G_I^{u_2})$$

Since $\varphi_{\text{dissent}}(G_E^d, G_I^{u_1}, G_I^{u_2}) = \max(\text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E^d, G_I^{u_1}, G_I^{u_2}), 0)$ we get $\varphi_{\text{dissent}}(G_E^d, G_I^{u_1}, G_I^{u_2}) \leq oe_{dissent}(G_E^c, G_I^{u_1}, G_I^{u_2})$ $\square$

Proof (Proposition 4) Given that $\forall (e, G_I^{u_1}, G_I^{u_2}) \in E \times 2^I \times 2^I\ :\ w(e, G_I^{u_1}, G_I^{u_2}) = 1$, we have for any $c \in \mathcal{D}_E$ s.t. $|G_E^c| \geq \sigma_E$.

$$\text{IAS}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum_{e \in G_E^c} \alpha(e)}{|G_E^c|} \textit{ and } \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum_{e \in M(G_E^c, \sigma_E)} \alpha(e)}{\sigma_E}$$

It follows from the fact that $M(G_E^c, \sigma_E) \subseteq G_E^c$:

$$\begin{aligned}
\exists S \subseteq G_E^c\ :\ & \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \text{IAS}(S, G_I^{u_1}, G_I^{u_2}) \\
& \text{UB}(G_E^c, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) = \\
& \qquad \text{IAS}(S, G_I^{u_1}, G_I^{u_2}) - \text{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) \\
& \text{oe}_{\text{consent}}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \varphi_{\text{consent}}(S, G_I^{u_1}, G_I^{u_2})
\end{aligned}$$

The subset S being for example the set $M(G_E^c, \sigma_E)$ itself. The same reasoning applies when considering $\text{oe}_{\text{dissent}}$. In this case we consider the lower bound LB. We have:

$$\text{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \frac{\sum_{e \in m(G_E^c, \sigma_E)} \alpha(e)}{\sigma_E}$$

Given that $m(G_E^c, \sigma_E) \subseteq E$, we have:

$$\begin{aligned}\exists S \subseteq G_E^c \ : & \mathrm{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \mathrm{IAS}(S, G_I^{u_1}, G_I^{u_2}) \\ & \mathrm{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \mathrm{LB}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \\ & \qquad \mathrm{IAS}(G_E, G_I^{u_1}, G_I^{u_2}) - \mathrm{IAS}(S, G_I^{u_1}, G_I^{u_2}) \\ & \mathrm{oe}_{\mathrm{dissent}}(G_E^c, G_I^{u_1}, G_I^{u_2}) = \varphi_{\mathrm{dissent}}(S, G_I^{u_1}, G_I^{u_2})\end{aligned}$$

This proves that, if IAS is a simple mean, for any $c \in \mathcal{D}_E$ s.t. $|G_E^c| \geq \sigma_E$:

$$\exists S, S' \subseteq G_E^c \ : \ \begin{cases} \mathrm{oe}_{\mathrm{consent}}(G_E^c, G_I^{u_1}, G_I^{u_2}) & = \varphi_{\mathrm{consent}}(S, G_I^{u_1}, G_I^{u_2}) \\ \mathrm{oe}_{dissent}(G_E^c, G_I^{u_1}, G_I^{u_2}) & = \varphi_{\mathrm{dissent}}(S', G_I^{u_1}, G_I^{u_2}) \end{cases}$$

Hence $\mathrm{oe}_{\mathrm{consent}}$ and $oe_{dissent}$ are tight optimistic estimates for respectively $\varphi_{\mathrm{consent}}$ and $\varphi_{\mathrm{dissent}}$ if the underlying IAS is a simple average. □

Before giving the proof of the proposition 5 we present the following lemma.

Lemma 2 *The sums of the number of all descriptions covering each record in G is equal to the sum of the supports of all descriptions in $\mathcal{D}$. That is:*

$$\sum_{g \in G} |\downarrow\delta(g)| = \sum_{d \in \mathcal{D}} |G^d|$$

Proof (Lemma 2) For $g \in G$, we have $\downarrow\delta(g) = \{d \in \mathcal{D} \ : \ d \sqsubseteq \delta(g)\}$ and for $d \in \mathcal{D}$, we have $G^d = \{g \in G \mid d \sqsubseteq \delta(g)\}$. Let us define the indicator function on $\mathcal{D} \times G$:

$$\mathbb{1}_{\sqsubseteq}(d, g) = \begin{cases} 1 & \text{if } d \sqsubseteq \delta(g) \\ 0 & \text{else} \end{cases}$$

Hence, we have $|\downarrow\delta(g)| = \sum_{d \in \mathcal{D}} \mathbb{1}_{\sqsubseteq}(d, g)$ and $|G^d| = \sum_{g \in G} \mathbb{1}_{\sqsubseteq}(d, g)$ thus:

$$\sum_{g \in G} |\downarrow\delta(g)| = \sum_{g \in G} \sum_{d \in \mathcal{D}} \mathbb{1}_{\sqsubseteq}(d, g) = \sum_{d \in \mathcal{D}} \sum_{g \in G} \mathbb{1}_{\sqsubseteq}(d, g) = \sum_{d \in \mathcal{D}} |G^d| \qquad \square$$

Proof (Proposition 5) We denote by **gs** the random record drawn in line 1 and by **ds** the random description drawn in line 2 of FBS.

$$\begin{aligned}\mathbb{P}(\mathbf{ds} = d) &= \sum_{g \in G} \mathbb{P}(\mathbf{gs} = g)(\mathbf{ds} = d|g) \\ &= \sum_{g \in G^d} \frac{1}{|\downarrow\delta(g)|} \times \underbrace{\frac{|\downarrow\delta(g)|}{\sum_{i \in G} |\downarrow\delta(i)|}}_{\text{weight } w_g \text{ normalized}} = \frac{|G^d|}{\sum_{g \in G} |\downarrow\delta(g)|}\end{aligned}$$

It follows that from *Lemma 2* that $\mathbb{P}(\mathbf{ds} = d) = \dfrac{|G^d|}{\sum_{d' \in \mathcal{D}} |G^{d'}|}$ □

Proof (Proposition 6) Given *Proposition 5*, it is clear that $\forall p \in \mathcal{P} : p = (c, u_1, u_2)$ *satisfies* $\mathcal{C} \Rightarrow \mathbb{P}(p) = \frac{|\mathrm{ext}(p)|}{Z} > 0$. with $|\mathrm{ext}(p)| = |G_E^c| \times |G_I^{u_1}| \times |G_I^{u_2}|$ and $Z = \sum_{p' \in \mathcal{P}} |\mathrm{ext}(p')|$ a normalizing factor. □

B Appendix: Additional Performance Experiments for Openmedic

In this appendix we report additional performance experiments over OpenMedic. Figure 25 depicts a comparative performance study between the baseline approach performing an exhaustive search without using closure operators, a Baseline+Closed approach and DEBuNk. In contrast with what we observed with the comparative performance experiments over EPD8, Yelp and Movielens, the performance gain for Openmedic between the three method is negligible. This is mainly due to the structure of OpenMedic Data and the type of study we conduct on, more precisely: (1) the ratio between closed description/all description is rather small, (2) the used IAS is a weighted average with a very scattered distribution of weights, (3) the optimistic estimate proposed for weighted averages IAS measures is not tight.

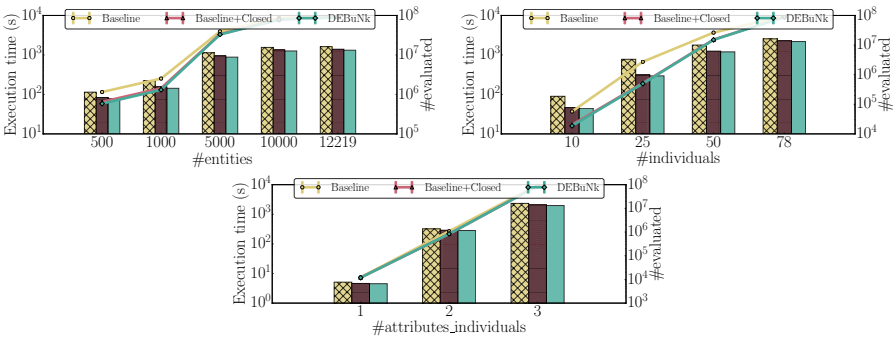


Fig. 25: Effectiveness of *DEBuNk* over Openmedic Dataset considering $|G_E| = 12219$, $|G_I| = 78$, $|Outcomes| = 500k$, $|\mathcal{A}_E| = 1$, $|\mathcal{A}_I| = 3$, $\sigma_E = 5$, $\sigma_I = 1$, $\sigma_\varphi = 5$ and the quality measure φ_{ratio}

Figure 26 illustrates the behavior of DEBuNk when conducted on OpenMedic according to different parameters. Bottom line, the experiments shows evidence that the number of descriptive attributes (complexity of descriptions space) are the ones which impact the most the efficiency of the algorithm. Moreover, the same conclusion on the experiments on EPD8, Movielens and Yelp can be drawn regarding the behavior of the algorithm according to the cardinality and quality thresholds.

Finally, we report in figure 27 a comparative performance between DEBuNk and Quick-DEBuNk over Openmedic dataset. In the first stages, the sampling algorithm achieves to find a good portion of the patterns which are returned by the exhaustive search algorithm DEBuNk. However, the method seems to converges less slowly, compared to the its performance in the other datasets, to the full results set. This results from the fact that the quality measure φ_{ratio} applied over openmedic is not symmetric, which requires to the algorithm to look twice for each couple of confronted groups of individuals, since the order matter between the two groups for the used interestingness measure.

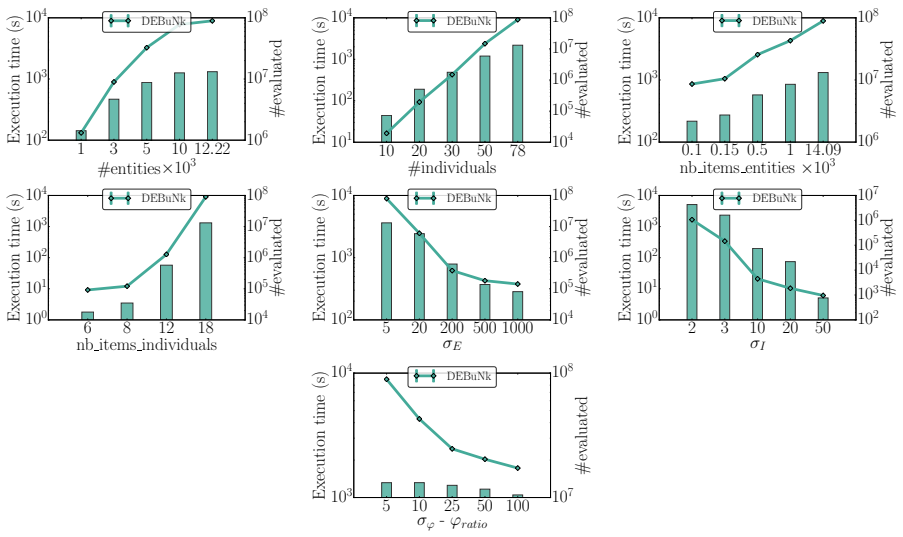


Fig. 26: Effectiveness of *DEBuNk* over Openmedic Dataset according to the sizes of E, I, $\mathcal{D}_E$, $\mathcal{D}_I$, the supports and quality measures thresholds. Considering by default the full dataset. $\sigma_E = 5$, $\sigma_I = 1$, $\sigma_\varphi = 5$ and the quality φ_{ratio}

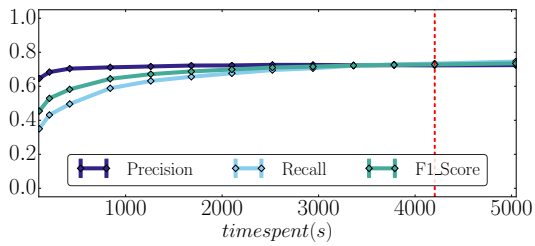


Fig. 27: Efficiency of Quick-DEBuNk compared to DEBuNk over Openmedic. We consider the full dataset (i.e. all attributes and all records), the parameters used for Openmedic are $\sigma_E = 5$, $\sigma_I = 1$, $\sigma_\varphi = 5$ and φ_{ratio}.