Interactive Data-Driven Research: the place where databases and data mining research meet

Sihem Amer-Yahia, Vincent Leroy, Alexandre Termier, Martin Kirchgessner, Behrooz Omidvar-Tehrani

Interactive Data-Driven Research

the place where databases and data mining research meet

Sihem Amer-Yahia[†], Vincent Leroy[†], Alexandre Termier[◇],
Martin Kirchgessner[†], Behrooz Omidvar-Tehrani[†]
[†] Univ. Grenoble Alpes - LIG/CNRS, [◇] Univ. Rennes 1 - IRISA/INRIA
[†] *firstname.lastname@imag.fr*, [◇] *alexandre.termier@irisa.fr*

1. INTRODUCTION

Data-driven research, or the science of letting data tell us what we are looking for, is in many areas, the only viable approach to research. In some domains like adaptive clinical trials and emerging research areas such as social computing, useful results are highly dependent on the ability to observe and interactively explore large volumes of real datasets. Database management is the science of efficiently storing and retrieving data. Data mining is the science of discovering hidden correlations in data. Interactive data-driven research is a natural meeting point that presents a new research opportunity. The ability to conduct effective data-driven research requires to combine efficient indexing and querying from databases and pattern mining and classification from data mining to help analysts understand what lies behind large data volumes. In this paper, we explore key challenges and new opportunities in building robust systems for interactive data-driven research.

1.1 Data-Driven Research: Facts

We define interactive data-driven research as the science of storing data, discovering correlations, storing correlations, and interactively querying data and correlations. A recurring requirement that is found in various domains ranging from health management to social computing, is the need to perform querying and discovery efficiencly.

In oncology drug development, predictive biomarkers such as molecules or other features of a patient, are used to find subsets of the patient population who can benefit from a particular therapy. Given the large number of patient features, a recent trend in this domain has been to construct predictive classifiers from a combination of biomarkers and their use to identify groups of patients who are most likely to benefit from a particular treatment.[1] Recent efforts such as *PatientsLikeMe* in the United States [2] and Yellow Card in the United Kingdom,[3] focus on gathering and mining patients' experiences when taking drugs. Gathered data is used to build population models that are then shared with research and industry partners whose goal is to improve products and services. A recent study this year on the use of Yellow Card,[4] suggests that even for well known drugs such as Aspirin, patient reporting may provide a complementary contribution to that of health care professionals.

Similarly, research in social computing is characterized by a heavy reliance on large-scale analytics of users' activities (e.g., on Flickr or Twitter) in order to understand their needs and design appropriate content discovery methods [6, 22]. To do so, user attributes such as their age and gender, and user actions such as tagging photos or rating movies, are mined together to determine groups of users who have common properties (e.g., same age group) and exhibit similar behavior (e.g., like the same movies). Such groups are not always known in advance and, more importantly, whether a group is interesting or not, will depend on what is found.

1.2 Data-Driven Research: Opportunities

When data guides research, new opportunities appear. First, data that lends itself to being explored usually has a number of statistical properties (e.g., outliers, long tail), that make traditional mining semantics inappropriate and of their corresponding algorithms inefficient. To address that, we introduce *Shallow Mining* to account for properties in data. Second, the value of exploring data relies on a careful combination of letting data tell us what to look for and asking the analyst to guide the discovery process. We refer to that as *Interactive Exploration*.

A good interactive data-driven research system should harmoniously combine shallow mining and interactive exploration in an iterative process. Shallow mining is used to quickly provide interesting starting points for further exploration. There are two challenges for shallow mining: semantics and efficiency. Semantics is application-dependent and is used to define a small number of *relevant* and *diverse* exploration options. We will describe the use of *top-k semantics* in two different scenarios: per-item pattern mining and user group exploration. Efficiency is what enables the seamless integration of shallow mining into an interactive exploration that helps the analyst determine, on-the-fly, a direction to pursue the exploration.

[1] *http://www.ncbi.nlm.nih.gov/pubmed/23489587*
[2] *http://www.patientslikeme/about*
[3] *https://yellowcard.mhra.gov.uk/*
[4] *http://www.ncbi.nlm.nih.gov/pubmed/23444232*

We first describe real-world datasets from application domains where data-driven research is ubiquitous (Section 2.1) and summarize the general approach we advocate for data-driven research (Section 2.2). Then, we present our work and research directions for efficient and interactive shallow mining (Sections 3 and 4 respectively). Finally, we discuss two fundamental research directions: the development of a principled validation approach of systems for interactive data-driven research, and the need to formalize the end-to-end process of data preparation, an essential step in data-driven research.

2. OUR APPROACH TO DATA-DRIVEN RESEARCH

2.1 Datasets and Models

We are particularly interested in user-centric data and we will be using examples from the health management and the social web contexts. Both kinds of datasets have been extensively used in data-driven research. In health management, user data could be generated by users (as in *PatientsLikeMe* or *Yellow Card*) or by implantable sensors.[5] On the social web, user-generated data comes from collaborative tagging, rating, or reviewing sites such as del.icio.us, MovieLens, or Amazon, respectively.

We consider a set of users $\mathcal{U}$, a set of items $\mathcal{I}$, and a database $\mathcal{D}$ of tuples of the form $\langle u, i, l\rangle$ where each tuple designates the action of a user $u \in \mathcal{U}$ on an item $i \in \mathcal{I}$ with a label l in a set of labels $\mathcal{L}$ that includes reports on individuals' health, tags, ratings, reviews, comments on a product (e.g., camera or drug), or extracted sentiment from text.

For example, $\langle Mary, Aspirin, nausea\rangle$, on *PatiensLikeMe*, represents Mary who reports nausea when taking Aspirin, $\langle John, foodnsport, sports\rangle$ and $\langle John, foodnsport, diet\rangle$, in del.icio.us, represent user John who tagged the foodnsport website with the tags sports and diet. In MovieLens, $\langle Sue, Titanic, 5\rangle$ represents user Sue who rated the movie Titanic with the highest rating. On Twitter, the triple $\langle John, Obama, pos\rangle$ expresses that user John's sentiment on entity Obama is positive. Finally, in Flickr, $\langle Rob, 534, Louvre\rangle$, $\langle Rob, 534, Paris\rangle$ represent that user Rob tagged the same picture with Louvre and with Paris.

Each user u is also described with attributes drawn from a set of attributes $\mathcal{A}$ representing demographics information such as *Gender* and *Age*. We refer to each attribute in $\mathcal{A}$ as a_i and to its values as v_j^i.

The mining algorithms we are interested in output *groups* of users $g = \{u_1, ..., u_n\} \subseteq \mathcal{U}$. All users of a group g satisfy a description defined as a conjunction of terms $c_1 \wedge ... \wedge c_m$, where a term c_i can either be the presence of an item $i \in \mathcal{I}$ or an equality condition on a attribute $a \in \mathcal{A}$: $a = v$.

2.2 Shallow Mining and Interactive Exploration

The motivation behind interactive exploration is to establish a dialog between the analyst and the data. This is necessary in data-driven research where it is often the case that analysts do not know in advance what to look for. In this way, data talks to the analyst by providing some insights, and the analyst responds back by specifiying where she wants to delve into details. Two important components are necessary to enable an effective exploration: an algorithm that starts with very few assumptions on what to extract from data, and an analyst who is able to guide the exploration. In contrast with feedback learning, the goal is not to build models to be reused for subsequent explorations. Rather, it is *to optimize the current exploration.*

Shallow Mining. Mining algorithms traditionally rely on input parameters to exhaustively mine a specific part of the solution space. For instance, in the domain of itemsets mining, it is common to retrieve all itemsets that appear in over 1% of the records. The number of itemsets satisfying this condition is often very high as the algorithm returns *all* solutions. We refer to this type of mining as *deep*. It is also *narrow*: the results are specifically tailored to match specified parameters and are not always representative of the input dataset. In the itemsets mining example, returned itemsets all constitute a different combination of the very few most common items, thus ignoring the large majority of the items composing the dataset. This is exacerbated when data has statistical properties such as long tail and outliers. In addition, while there are situations in which an analyst may require a large number of results (in the case where they are fed to another process), current itemsets mining algorithms are not suitable for the exploration of a new dataset and cannot efficiently support data-driven research. We advocate the development of *wide* and *shallow* analysis and refer to that as *Shallow Mining*.

The purpose of shallow mining is to find a *wide* result set, i.e., representative results that reflect different aspects of the dataset currently explored and serve as a starting point to further exploration. The objective is to allow the analyst to target any part of the dataset, either focusing on frequent events, or selecting a more peculiar fraction of the data, which would have traditionally been considered as an anomaly and pruned. As the mining target widens, the goal should not be to return exhaustive results as that would require prohibitive execution times and overwhelm the analyst. Instead, mining is *shallow* and returns a sample of the results, selecting representative ones that enable and facilitate further exploration.

As related works, Sanders and Goethals [15] recently proposed an algorithm to sample maximal frequent itemsets while optimizing the coverage of items in the dataset. This approach constitutes an interesting example of shallow mining: the set of results returned is relatively small, but it describes well the entirety of the dataset. This sampling method favors large itemsets with a low frequency, close to the minimal support threshold.

Interactive Exploration. Interactive exploration consists in iteratively applying shallow mining and asking for user feedback in order to determine the next mining direction. Designing an interactive data exploration framework is a hot topic nowadays [5, 21]. Such a framework has 3 critical principles: (P_1) at any given step in the process, the analyst

[5] *http://circep.ahajournals.org/content/3/6/657.extract*

must be given alternatives but must not be overwhelmed with exploration options; (P_2) one or more holistic quality measures (e.g. space coverage, interestingness of an alternative) should be applied to select exploration alternatives; (P_3) given its online nature, each exploration step must be fast and executed in few seconds.

From these principles, we identified three challenges for data-driven research. Principle (P_1) requires to have few representative patterns for the data, and can be satisfied by shallow mining. The first challenge consists in defining the semantics of representativity in shallow mining. The second challenge, in line with principle (P_3), consists in finding representative results efficiently. The third challenge, in line with principle (P_2), consists in integrating shallow mining with interactive exploration.

For the first challenge, a top-k semantics where the analyst sees k exploration options is a natural choice in data-driven research. We discuss two different such semantics that will be used in the rest of this paper. The first semantics will be used to illustrate efficiency in Section 3 and the second to illustrate interactivity 4.

Alternative top-k semantics: We explore two scenarios for top-k semantics. The first scenario stems from the observation that most Web application only require a limited number of itemsets for each item. For instance, Amazon may use a few tens itemsets as a recommendation on the page of an article, but no more. Thus, it is possible to determine a value k such that the application requires a maximum of k itemsets for each item in the dataset. It is important to ensure that the k results sampled per item are among the most interesting ones. We hence choose to return, for each item, only its most frequent closed itemsets [17] to avoid redundancy.

In the second scenario, we consider that the analyst is exploring a search space of *groups* denoted $\mathcal{G} \subseteq 2^{\mathcal{U}}$, where each group represents a subset of users and has a description (defined in Section 2.1). The description conveys a characteristic common to all users in the group. For example if the data represents Flickr users, a group could be all users that went to London and took pictures of Big Ben and Tower Bridge. In medical data describing patients' reaction to a newly-tested drug, a group could be all patients that feel nauseous after the second take of the drug and whose body temperature is in the interval $[37.5, 38.0]$ at that time. Such groups and descriptions are found by data mining algorithms such as frequent pattern mining algorithms or subspace clustering. Due to the combinatorial number of possible groups, interactive exploration is one of the most promising techniques to exploit the richness of these analysis.

3. EFFICIENT SHALLOW MINING

In order to illustrate the efficiency of shallow mining, we focus on TopLCM [13], an instantiation of frequent itemsets mining used to find per-item representative patterns. Introduced 20 years ago by Agrawal et al. [1], frequent itemset mining (FIM) is a fundamental task in data mining, that aims at finding all the sets of items occurring at least ε times in a transactional dataset. These items represent correlation between attributes, and can also be used to derive association rules. The complexity of FIM is exponential in the number of items in the dataset. Hence, a large majority of FIM experiments are performed with relatively high values of ε, such that the number of frequent items remains low and the mining completes within reasonable time.

Web datasets containing user data are a rich source of information which can be used to extract itemsets for recommendation purposes. These datasets are often characterized by a "long tail" distribution [12], which means that the majority of items have a low frequency. Consequently, traditional frequent itemsets mining algorithm are unable to generate itemsets for the long tail, as this would require setting ε too low. The results obtained are too *narrow*, and fail to cover a large portion of the dataset. TopLCM aims at re-visiting the FIM problem by providing a *wide* set of representative results.

3.1 TopLCM problem definition

Given a dataset $\mathcal{D}$, a set of items $\mathcal{I}$, a frequency threshold ε and an integer k, TopLCM returns, $\forall i \in \mathcal{I}$, $top(i)$, the k most frequent closed itemsets containing i.

By limiting the size of the output to k itemsets for each frequent item, TopLCM ensures that the results are *shallow*. Some frequent itemsets mining algorithms, such as LCM [23] have a linear complexity with respect to the output size, and directly benefit from this bound. TopLCM is able to lower the frequency threshold ε to include a larger fraction of the items of the dataset (*wide* results set) without suffering from an exponential increase in the output size.

3.2 TopLCM algorithm

As its name suggests, TopLCM inherits the itemsets enumeration strategy of LCM [23], the fastest closed frequent itemsets mining algorithm. LCM recursively builds itemsets by augmenting existing ones, relying on the anti-monotony property of itemsets' frequency: augmenting a closed itemset always lowers the frequency. LCM ensures, through a "prefix-preserving test" that each itemset can only generated once, which avoid redundant work. In addition, items are explored by decreasing order of frequency, which limits the memory consumption.

TopLCM adapts techniques traditionally used in the top-k query evaluation of search engines and databases [9] to extend LCM with an early termination strategy. TopLCM stores the k most frequent itemsets of each item in a *top-k collector*. During the exploration, TopLCM uses a heuristic to determine whether an extension may recursively generate itemsets part of the top-k results of an item. If that is not the case, the exploration space may safely be pruned without affecting the results. Additional details and optimizations of TopLCM can be found in [13].

3.3 TopLCM evaluation

We evaluated TopLCM on usual FIM datasets as well as "long tail" Web datasets, respectively represented here by *Accidents* and *LastFM*. The first one lists the characteristics of 340,184 traffic accidents among 468 possibilities [10]. The second one was crawled from LastFM, a music recommendation website, and contains the 50 favorite artists (among

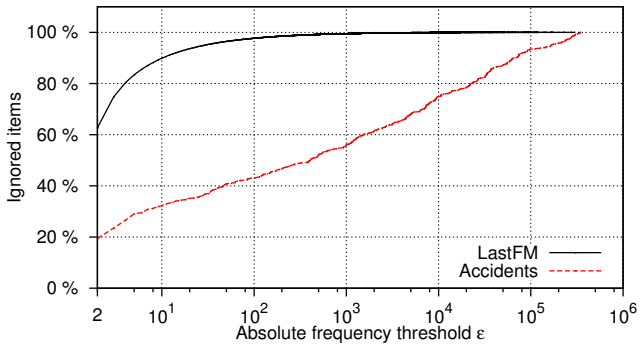

Figure 1: Portion of available items ignored by FIM with respect to the frequency threshold, for our examples' datasets. Traditional FIM uses thresholds much higher than 1000.

1.2 million) of 1.2 million users having a public profiles [4]. The profile of each user froms a transaction.

We summarize the difference between TopLCM and traditional frequent itemsets mining with the following example: On the *LastFM* dataset, given a fixed execution time of 5 minutes, an analyst can either *(i)* use a standard FIM algorithm with $\varepsilon = 500$ and obtain all the matching itemsets, covering only 0.82% of the items in the dataset (see Figure 1), or *(ii)* use TopLCM, with $\varepsilon = 2$ and $k = 100$, and obtain up to 100 itemsets for every item appearing at least twice in the dataset (37.4%).

TopLCM can also mine *Accidents* with $\varepsilon = 2$. When $k = 50$ TopLCM is able to identify 50 distinct closed itemsets for all items appearing 10 times or more. Other items are not sufficiently frequent to completely fill their top-k list, but TopLCM returns all the itemsets present in the dataset nonetheless. Accidents is considered as a very dense dataset, extremely costly to mine at low support. By opting for a shallow approach TopLCM identifies new itemsets for less frequent items which were unreachable using traditional approaches.

Shifting the FIM problem to an item-centric variant allows TopLCM to find itemsets in parts of the dataset that would be ignored by existing algorithms. Hence TopLCM does not replace nor outperform existing work from the FIM community, but rather proposes a different approach adapted new uses cases. In the context of data-driven research, we believe the output of TopLCM is much more suitable to feed an interactive exploration process as discussed in the next section.

4. INTERACTIVE EXPLORATION

Recent approaches for interactive exploration [5, 21] of user groups are iterative approaches and are formalized as an optimization problem. At each step, following principle (P_1) a limited set of groups $\{g_1, ..., g_k\}$ is shown to the analyst (where k can be a parameter of the algorithm). Following principle (P_2), this set of groups is the (approximated) result to an optimization problem exploiting a quality measure δ on the set of groups to show: $\{g_1, ..., g_k\} = argmax_{G \subseteq \mathcal{G} \mid |G|=k}(\delta(G))$. Existing quality measures combine an estimation of the *relevance* of the groups shown, and a *diversity* criteria among the groups shown to favor a wide exploration of the space of groups.

4.1 Evaluation alternatives

The choice of a good quality measure δ is tricky, and two different directions are found in recent works. One of them is to make the interactive process *statefull*, i.e. to progressively build a model the analyst expectations [5]. This model allows to progressively refine the quality measure estimating the set of groups, at the price of asking some more feedback to the analyst (explicitely state what is interesting and what is uninteresting in the groups shown). With the analyst's feedback, the problem can be expressed as an exploitation/exploration problem, where well known multi-armed bandit techniques [11, 5] can be used. This approach may lack resiliency when confronted with large *query drift* from the analyst. We consider it better adapted when the analyst has a relatively precise idea of what she is looking for, and needs to drill down deep inside the space of groups to find it.

The other direction, conversely, is to perform a *stateless* interactive process. This means that the estimation function is fixed, and does not model the analyst. This is what we proposed with the Ueca approach [21]. In this approach, a set of groups for further exploration is entirely determined by the group under consideration by the analyst. Contrarily to the statefull approach above, it is better adapted for highly exploratory situations, where the analyst has no idea of what she is looking for in the data. In the Ueca approach this is further supported by a strong weighting on the diversity of the results, allowing the analyst to see many different groups, and to easily shift her analysis focus.

Currently, works either present a statefull or stateless interactive exploration. However for a generic system that could accomodate a large range of analysis situations, an hybrid approach could be a more flexible solution. At the beginning of an analysis session, it is more important to allow unbounded exploration, with a stateless method. When the analyst starts to drill down on groups (which can be detected through her choice of groups), the exploration can switch to be statefull, in order to accelerate the drilling down by becoming more conservative in the choice of groups shown. If query drift is detected, signaling a shift of analyst interest, exploration can switch back to stateless.

4.2 Interactivity through fast computation

The exploration techniques proposed above, following principles (P_1) and (P_2) of interactive exploration, focus on what is shown to the analyst. However, to support interactive analysis, these techniques must not only give interesting exploration choices: they must also provide those choices *fast*. This is principle P_3: the train of thought of the analyst must not be broken. This requires computation steps to be at best instant (less than 1 second), and in the worst case to take few seconds (less than 1 minute).

Existing works have two different strategies to guarantee such fast results. The first one, found in the Ueca system [21], is to pre-compute all groups of potential interest in an offline phase, before the data analysis. These groups are

then stored and indexed in a database with efficient bitmap encoding, guaranteeing fast access and intersection operations. Thanks to this precomputation and indexing, the method that selects the set of groups to show to the analyst has a large pool of groups to choose from, with the potential to find very good groups. The size of this group pool is also a risk, as it makes the search space of the set of group selection method very large. Thus the computation time that has been saved by computing many groups beforehand can be lost in the selection of the most interesting set of groups. This problem can be alleviated using aggressive approximation techniques in the selection of the set of groups, at the risk of loosing the quality benefit of having many groups pre-computed.

The other strategy is to compute the groups and the set of groups on the fly, using approximation techniques to simplify group discovery. The TopLCM approach presented in Section 3 can be used in this setting, by giving it a low k value and if necessary by specifying a limited set of items for which the top-k closed frequent itemsets are needed. This result can then be passed to an optimization algorithm finding the best set of closed frequent itemsets to show the analyst. Sanders and Goethals [15] propose and integrated solution based on random sampling to discover directly a set of maximal frequent itemsets with low support and low overlap that describe well the data. Such result can be of interest for the analyst, especially at the initial stages of analysis of a new dataset. Boley et al. [5] are more agnostic on the mining results: their requirement is to have anytime and parameter free algorithms. Their method exploits the "idle" times during which the analyst reviews the results produced at one step to mine results of potential interest in the next step.

While the first strategy relies on legacy data mining algorithms and indexing techniques, the second one is likely to see the design of more and more fast algorithms adapted for interaction. Both can be efficiently combined: if there is some offline time between data acquisition and data analysis (for example, the night), heavy pre-computation can be performed and results indexed in order to provide near-instant interaction during the first steps of analysis. These first step are usually more difficult for online algorithms, as if the dataset is large loading it could consume most of the allocated time budget. Such algorithms are better adapted to the drill down steps, where smaller parts of the dataset are concerned and complex mining tasks can be conduced in few seconds.

In order to meet tight computation time constraints for interactivity, exploiting parallelism is an important direction of future research. We already started working in this direction with TopLCM, but this is limited to discovering groups of interest. Making new algorithms that exploit multi-core or the future many-cores processors in order to improve both group discovery and set of group computation will improve the responsiveness of algorithms, the quality of the solutions found, or both.

5. LOOKING FORWARD

5.1 Validation

Throughout this paper, we presented our view of data-driven research, which combines techniques from data mining and databases. There are already several efforts going in this promising direction. However, one key point is still missing to guarantee steady research progress in this area: a principled validation approach.

The challenge here is that, unlike in Information Retrieval where the notion of a ground truth exists or may be crafted, the notion of quality in interactive data-driven research is subjective and depends on the state of mind of the analyst at exploration time. The absence of a gound truth considerably limits the applicability of user studies. Moreover, due to the large size of the search space (much larger than a document space on the Web), it is difficult to create test cases where all the "good" answers (at least for a given analyst) can be known and labeled.

We advocate a hybrid scheme that combines quantitative and qualitative validations. Quantitative validation is based on measures to evaluate the exploration: for example, measures of coverage of the input dataset, measures of diversity of choices, number of steps to reach some specified nodes with a random walk. Qualitative validation is based on user studies, using analysts with diffreent goals, and gather feedback from different end-users.

A quantitative validation allows to have some guarantees on the interactive exploration proposed and help uncover its biases and the use cases where it is best adapted.

5.2 Toward a data processing framework

Often times, data processing in data-driven research is proprietary to each implementation and is mostly "encapsulated" in scripts that are hard to verify and modify. The opportunity here is to formalize data processing and develop appropriate tools to enable easy prototyping of data-driven applications. We advocate the development of a data model and algebra to express and optimize data processing.

To illustrate the challenges and opportunities behind the development of such a framework, we use the example of *Data Preparation*, a recurring pre-processing step in data-driven research. During data preparation, raw data is sanitized, normalized, enriched, pruned, and transformed to fit the required data structures and statistical properties of subsequent steps. That is the case both in developing adaptive clinical trials and in building social applications. In the clinical domain, data is often pruned to keep the most statistically significant observations or those corresponding to a specific combination of patient attributes (e.g., focus on infants).

In oncology drug development, statistical methods such as Principal Component Analysis are used to pre-determine a number of features that are deemed important in the analysis and that reduce the size of the search space. Such features are often identified using high-throughput genomic and proteomic technologies whose goal is to produce hundreds of potential biomarkers. Those biomarkers are then used to predict clinical responses to new drug candidates. We argue that the initial step of pre-processing the raw data to fit pre-defined models may miss potentially useful cases of clinical development. This argument is substantiated by the fact that extensively evaluated biomarkers that corre-

late with desired clinical outcomes, are generally favored to other less-known biomarkers, that present new opportunities for predictive clinical response and for speeding up provisional regulatory approval of a drug. A specific example is the case of cancer treatment where cance-drug labels are known for being less detailed in their specification of effects on patients symptoms and funcitoning. Only 25% of cancer-drugs list those effects [6]. According to the same reference, the Food and Drug Adminitration (FDA) has taken several recent steps in gathering patients' feedback (referred to as Patient Reported Outcomes). Such datasets are highly valuable and constitute an unprecedented source for mining drug effects on different user populations. Data pruning, either by specifying features to focus on, or by filtering out infrequent observations, may hurt the discovery process.

In building social applications, similar data preparation operations are needed to make raw social data ready-to-be-exploited by social applications. For website recommendation on del.icio.us, users' tagging actions are pre-processed in a preliminary step, in order to remove the long tail of tagging, extract topics from tags and cluster users together [14, 19, 20]. Similarly, in a movie recommendation application on MovieLens that caters to user groups, users' ratings are normalized and users are pre-grouped based on rating similarities in order to optimize the retrieval of movies for arbitrary user groups [3, 16, 18]. When doing sentiment extraction on Twitter, tweets are pre-processed to extract topics and entities [7]. In news article recommendation, articles are pre-processed to extract topics and sentiment extraction is applied to user comments on articles [2]. Finally, when Flickr photos are used to build touristic itineraries in a city [8], tags and geographic coordinates are used to map individual photos to landmarks thereby enriching raw social data. All those are examples of primitive data preparation operations that are repeatedly hard-coded in various applications.

In addition to being closed under a carefully chosen data model, an algebra for data processing must have a number of essential properties: expressivity, declarativity, an the ability to represent and a new property that we refer to as invertibility. Just like SQL or XQuery, closure ensures that each operation takes as input one or more instances of and outputs an instance of the data model. Therefore, operations can be composed any number of times and in any order. Invertibility aims to guarantee that each data processing operation admits an *inverse* allowing to undo and backtrack computation. This property is important to enable the evaluation of different semantics, without rebuilding applications from scratch. For instance, once may want to change the support value in pattern mining. Ensuring fine-grained invertibility (at the level of each operation in the language), will enable optimized execution plans where bulk pruning and aggregations are avoided.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.

[2] S. Amer-Yahia, S. Anjum, A. Ghenai, A. Siddique, S. Abbar, S. Madden, A. Marcus, and M. El-Haddad. Maqsa: a system for social analytics on news. In *SIGMOD Conference*, pages 653–656, 2012.

[3] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2(1):754–765, 2009.

[4] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. The gossple anonymous social network. In *Proceedings of the 11th International Middleware Conference*, pages 191–211, 2010.

[5] M. Boley, B. Kang, P. Tokmakov, M. Mampaey, and S. Wrobel. One click miningt-interactive local pattern discovery through implicit preference and performance learning. In *KDD workshop on Interactive Data Exploration and Analytics (IDEA)*, pages 28–36, 2013.

[6] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. Who tags what? an analysis framework. *PVLDB*, 5(11):1567–1578, 2012.

[7] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *COLING (Posters)*, pages 241–249, 2010.

[8] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 35–44, New York, NY, USA, 2010. ACM.

[9] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS*, pages 102–113, 2001.

[10] K. Geurts, G. Wets, T. Brijs, and K. Vanhoof. Profiling high frequency accident locations using association rules. In *Proceedings of the 82nd Annual Transportation Research Board, Washington DC. (USA), January 12-16*, page 18pp, 2003.

[11] D. Glowacka, T. Ruotsalo, K. Konyushkova, K. Athukorala, S. Kaski, and G. Jacucci. Directing exploratory search: reinforcement learning from user interactions with keywords. In *IUI*, pages 117–128, 2013.

[12] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *WSDM*, pages 201–210, 2010.

[13] M. Kirchgessner, V. Leroy, A. Termier, S. Amer-Yahia, and M.-C. Rousset. Toplcm: An efficient algorithm for mining item-centric top-k closed itemsets. Technical report, University of Grenoble, France, 2013.

[14] S. Maniu and B. Cautis. Taagle: efficient, personalized search in collaborative tagging networks. In *SIGMOD Conference*, pages 661–664, 2012.

[15] S. Moens and B. Goethals. Randomly sampling maximal itemsets. In *KDD workshop on Interactive Data Exploration and Analytics (IDEA)*, pages 80–87, 2013.

[16] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of user. In *ECSCW*, pages 199–218, 2001.

[17] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *ICDT*, pages 398–416, 1999.

[18] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen.

[6] *http://www.nejm.org/doi/full/10.1056/NEJMp1114649*

Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[19] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum. Efficient top-k querying over social-tagging networks. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 523–530, New York, NY, USA, 2008. ACM.

[20] J. Stoyanovich, S. Amer-Yahia, C. Marlow, and C. Yu. Leveraging tagging to model user interests in del.icio.us. In *AAAI Spring Symposium: Social Information Processing*, pages 104–109, 2008.

[21] B. O. Tehrani, S. Amer-Yahia, and A. Termier. Uniform exploration of user communities and activities in ueca. Technical report, University of Grenoble, France, 2013.

[22] M. Tsytsarau, S. Amer-Yahia, and T. Palpanas. Efficient sentiment correlation for large-scale demographics. In *SIGMOD Conference*, pages 253–264, 2013.

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