CrowdMiner: Mining association rules from the crowd

Yael Amsterdamer[1], Yael Grossman[1], Tova Milo[1], and Pierre Senellart[2]

[1]Tel Aviv University

[2]Télécom ParisTech & The University of Hong Kong

ABSTRACT

This demo presents *CrowdMiner*, a system enabling the mining of interesting data patterns from the crowd. While traditional data mining techniques have been used extensively for finding patterns in classic databases, they are not always suitable for the crowd, mainly because humans tend to remember only simple trends and summaries rather than exact details. To address this, *CrowdMiner* employs a novel crowd-mining algorithm, designed specifically for this context. The algorithm iteratively chooses appropriate questions to ask the crowd, while aiming to maximize the knowledge gain at each step. We demonstrate *CrowdMiner* through a *Well-Being* portal, constructed interactively by mining the crowd, and in particular the conference participants, for common health related practices and trends.

1. INTRODUCTION

Habits and practices of people are routinely analyzed by researchers and organizations alike for various purposes. Discovering statistically significant patterns in the crowd's habits is a challenging task, and traditional tools (interviews, polls, surveys) to collect data from individuals about their daily life are costly to implement. Moreover, it is often hard to know which are the best questions to ask. This is in part due to the fact that human knowledge forms an *open world* [6]; one often wants to use the crowd to find out what is *interesting and significant* about a particular topic, without full knowledge of the topic.

For classic databases, data mining techniques have been developed for identifying such patterns and inferring *association rules* among data items. However, when dealing with human behavior, a comprehensive database may not be available for mining. This is because, typically, the day-to-day actions of people are not recorded - except in their own memories, which are limited in terms of detail recollection. Indeed, social studies show that instead of full details, people often tend to recall information in the form of summaries [4].

Proceedings of the VLDB Endowment, Vol. 6, No. 12

To address this, we present the system prototype *CrowdMiner*. *CrowdMiner* relies on the crowd of Web users, and implements a novel algorithm to effectively (and efficiently) mine interesting data patterns from the crowd. This algorithm combines, for the first time, a *proactive* crowdsourcing approach of posing questions to the crowd, with the discovery of patterns in an unknown domain. While there has been vast research on data mining, and many recent developments of crowdsourcing platforms (e.g., [8, 6]), there has been no previous work integrating the two.

As a motivating example, consider the study of people's well-being practices, such as sports, dietary habits, alternative medicine, etc. The results of such a study may be important, e.g., for medical research or marketing purposes. In the context of the study, it may be useful to identify patterns that correlate items, such as "baking soda and lemon can be used for relieving heartburn" or "People mostly take an energy drink when they go jogging". To learn such patterns with current data mining techniques, one must supply, as input, a database of *transactions* [1]. These may correspond, in the well-being domain, to "events" such as a particular workout, meal, etc., including all relevant details. Unfortunately, this type of data is not systematically documented.

To overcome the lack of records, we may turn to the crowd to ask people about their personal habits. Clearly, it is unrealistic for them to remember all the details about their well-being "events". This means that, in this example, a transaction database cannot be constructed. However, even though people cannot recall all of their transactions, social studies show that they can often provide simple summaries, and even more complicated ones when asked targeted questions [4]. For example, they may be able to provide simple summaries like "I have a headache 1-2 times a week, and 90% of the time drinking coffee helps", which we can interpret as a *personal rule* correlating "headache" and "coffee" for some individual, with the frequency expressions ("90% of the time") being interpreted as indicators for the rule significance. People can also answer more complex but targeted questions, such as "How often, when you go jogging in the morning, do you take both sunglasses and an energy drink?" [4]. Consequently, the crowd-mining algorithm underlying *CrowdMiner* interleaves *open questions* ("Tell me of two activities you typically do together") with more targeted ones (termed *closed questions*), which are dynamically constructed as more information is revealed, to identify interesting correlations of varying complexity.

The technical background for *CrowdMiner*'s algorithm is detailed in a full paper by the current authors [2]. In Sec-

tions 3 and 4 we overview the parts of the solution in [2] which are used in our prototype system, described in Section 5.

Demonstration. We demonstrate the use of *CrowdMiner* for the construction of a unique *WellBeing* portal, focusing on common health and well-being-related practices. See Section 6 for full details.

Beyond manually constructed traditional health or well-being portals, some portals have recently started to embed crowd-based data (e.g., the award-winning CureTogether [5]). Users may be involved for posting health tips, voting for the suggestions of others (as in [5]), and so on.

Our *WellBeing* portal *takes crowd involvement to the next level*: using our crowd mining techniques we can identify where knowledge is lacking, and proactively engage portal users to contribute the missing data, rather than wait for them to provide it. To engage the users we ask them both closed and open questions while they are browsing our portal. The interpretations of answers are embedded into our evolving knowledge base.

In the demonstration, we will use two computers displaying the *WellBeing* portal: one in *standard* mode, and the other in *administrator* mode, which allows peeking into the underlying knowledge base. Audience members will be invited to browse through the portal in standard mode. They will be able to search for their favorite activities, foods or medicine ingredients; to follow links to automatically-generated portal pages, and find well-being information and tips, learned from their colleagues. As explained above, they will answer questions about their habits while browsing. In parallel to the standard browsing, we look at the administrator-mode portal. There, a list of rules will constantly be updated according to the collected knowledge. Through this list we will be able to observe, e.g., how answers affect certain rules' significance, and how questions are chosen (on which rules), providing insight into the operation of the underlying algorithm.

2. RELATED WORK

Crowdsourcing has recently gained the attention of the research community (see, e.g., [8, 6, 3]). However, our work in [2] is the first, to our knowledge, to consider the use of data-mining–inspired techniques for learning from the crowd.

Our work has strong connections with *association rule learning* [1]. A particularly relevant line of works [9, 11] mine databases based on *data samples*. While they sample transactions, such information is not available in our settings at all. One could sample rules, which corresponds to asking only open questions. However, this allows much less control over the collected information, and our experiments in [2] show that algorithms that interleave open and closed questions perform much better.

Our approach also relates to the discipline of *active learning* [7], where algorithms choose the data instances to be labeled by experts. While we also choose which rules to ask about, in our settings there are no "experts", and the absolute truth may only be estimated by aggregating many user answers.

3. TECHNICAL BACKGROUND

We are interested in asking users about rules that apply to them individually, and inferring from this overall important rules and general trends. For that, we start with basic definitions of association rules and their quality measurements per individual. We use these simple definitions, based on classic association rule theory [1], to capture the summarized manner in which people remember their data. This also guides us in the types of questions we ask the crowd and the interpretation we give to their answers.

Let $\mathcal{U}$ be a set of users. Define a *transaction* t as a subset of a fixed, non-empty item domain $\mathcal{I} = \{i_1, i_2, \dots\}$. Each user $u \in \mathcal{U}$ is associated with a *personal database* D_u, which is a bag (multiset) of transactions.

Let $A, B \subseteq \mathcal{I}$, $A \cap B = \emptyset$. Then $A \to B$ is used to denote an *association rule*, which signifies that in a personal database D_u, $A \subseteq t$ implies $B \subseteq t$, for any $t \in D_u$. We sometimes write $a \to b$ instead of $\{a\} \to \{b\}$, for brevity.

When mining well-being habits, for instance, items may represent activities, food, sports gear, etc. An association rule might be *jogging* $\to$ *sunglasses*, signifying that when a user goes jogging, she wears sunglasses.

We use the standard definitions of support and confidence [1] as a measure for the significance of a rule *per user*. Given a user u with database D_u and a rule $A \to B$, let the *user support* and *user confidence* of $A \to B$ in the transactions of u be defined, respectively, as: $supp_u(A \to B) := \frac{|\{t \in D_u | A, B \subseteq t\}|}{|D_u|}$ and $conf_u(A \to B) := \frac{|\{t \in D_u | A, B \subseteq t\}|}{|\{t \in D_u | A \subseteq t\}|}$.

Questions and answers. Data mining techniques generally rely on processing transactions for association rule learning. In contrast, in our settings, the personal database D_u may be *completely virtual*, and not available for mining directly. As explained in the Introduction, people tend to remember their data in the form of summaries. We model these as personal association rules, and ask users about their rules along with their significance indicators. We consider two types of questions, formalized as follows.

- **Closed questions.** Questions modeled as $A \to^? B$. We interpret the answer of a user u as the support and confidence of the rule $A \to B$ w.r.t. D_u, i.e., the pair $\langle supp_u(A \to B), conf_u(A \to B) \rangle$.
- **Open questions.** Questions modeled as $? \to^? ?$. The answer of a user u is interpreted as some A, B, along with the confidence and support of $A \to B$ in D_u.

We show in the sequel how these types of questions can be used for efficiently mining interesting rules from the crowd. In Section 5 we explain how questions are presented to human users and how the answers are collected.

The user support and confidence of *different* association rules may be dependent. For instance, the support of $A \cup B \to C$ is bounded by the support of $A \to C$. We consider these dependencies in asking about rules for which we have no samples (See Section 4), but not in our approximations (described below), for computation simplicity. This means that we gather information on each rule separately from the crowd. Note that this does not affect the algorithm correctness, but gathering information on several rules at once could be a further optimization left for future work.

Overall Rule Significance. We are interested in general trends, i.e., rules that have an overall significance in a group of users. For that, we need to aggregate user answers, to compute the *average user behavior*. Formally, we say that a rule $r = A \to B$ is *significant* in a group of users $\mathcal{U}$ if it satisfies $\text{avg}_{u \in \mathcal{U}}\, supp_u(r) \geqslant \Theta_s$ and $\text{avg}_{u \in \mathcal{U}}\, conf_u(r) \geqslant \Theta_c$, where $0 \leqslant \Theta_s, \Theta_c \leqslant 1$ are predetermined threshold values.

Using thresholds for the support and confidence is standard in data mining. The choice of avg as the aggregation function has two practical benefits. First, when averaging, inaccuracies in individual user answers tend to cancel out [2]. Second, it allows the development of robust estimations for the significance of rules, as explained next.

In practice, it is impossible to obtain the answers of all the users about a certain rule. Thus, we resort to sample-based empirical estimations (Section 4). For that reason, and for simplifying the estimation formulations we assume that our questions are posed to randomly-chosen crowd members.

4. ALGORITHM

We now give an overview of the algorithm at the core of *CrowdMiner* (see [2] for full details). This algorithm can return a set of estimated significant rules at any point in its execution, based on the knowledge collected thus far. Since the crowd is the most expensive resource, our algorithm aims to maximize the knowledge gain in each successive iteration, by a careful choice of crowd questions.

We first propose a method for choosing the next best *closed* questions based on sampling (open questions are considered afterwards). For simplicity, the description is restricted to the selection of one question at a time.

Rules and error. We model each user answer about a rule r as a *2D sample* for the rule, where the two dimensions are the user support and confidence values. Samples are assumed to be taken independently from some unknown distribution with mean $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$ (the distribution of user answers). By the central limit theorem, the sample mean f_r approaches a normal distribution. We thus approximate f_r by the bivariate normal distribution with mean vector $\tilde{\mu}$ and covariance matrix $\frac{1}{N}\tilde{\Sigma}$, where N is the sample size.

Both $\tilde{\mu}$ and $\tilde{\Sigma}$ are unknown, but, using the samples for rule r, we can approximate $\tilde{\mu}$ as the *sample average* μ and $\tilde{\Sigma}$ as the *sample covariance* Σ. Now the sample mean distribution f_r may be approximated as a normal distribution with mean μ and covariance $\frac{1}{N}\Sigma$.

For each rule r, according to f_r, the probability of an error $P_{\text{err}}(r)$ is defined as follows. If the probability for the mean to be above both thresholds is greater than 0.5, r is considered a *significant* rule. Then the probability of making an error is the probability of $\tilde{\mu}$ being actually below one of the thresholds. Otherwise, r is an insignificant rule, and $P_{\text{err}}(r)$ is the probability of being above both thresholds. We thus have, if r is significant: $P_{\text{err}}(r) = 1 - \int_{\Theta_c}^{\infty}\int_{\Theta_s}^{\infty} f_r(s,c)\,\mathrm{d}s\mathrm{d}c$ where s represents the support and c the confidence; and its complement for the case r is insignificant.

We next estimate the effect of one more question on r. For that, we assume that the sample distribution, denoted g_r, is bivariate normal with μ and Σ for mean and covariance. Using g_r, we compute the *expected next error probability* P'_{err} given one more sample for r. This is done by integrating the error of the new sample mean f'_r over all possible next samples according to g_r. The integral (omitted due to space constraints) cannot be computed directly, and thus we use a numerical Monte Carlo method to estimate its value.

The best rule to ask a closed question about is the one for which we obtain the largest expected error decrease, i.e., $\text{argmax}_r\, \mathbb{E}\left[P_{\text{err}}(r) - P'_{\text{err}}(r)\right] = \text{argmax}_r\, P_{\text{err}}(r) - \mathbb{E}\left[P'_{\text{err}}(r)\right]$.

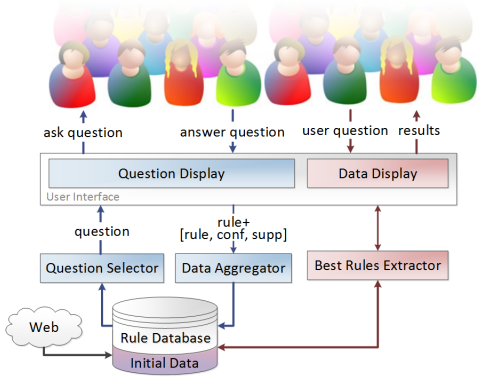


Figure 1: System Architecture

In order to handle rules without user samples, we allot each rule an initial set of samples by a new user $u^* \notin \mathcal{U}$. We select these samples such that the initial error is large.

Completing the picture. To optimize the selection of closed questions, we do not want to consider the entire association rule space, which may be huge. Instead, we want to consider, at each point, only an interesting subset of these rules, which we refer to as *enabled*. Following classic data mining, we note that two types of unknown rules have a higher chance of being significant: smaller rules, and rules similar to other rules that are known to be significant [1]. We thus only enable these types of rules (see [2] for full details).

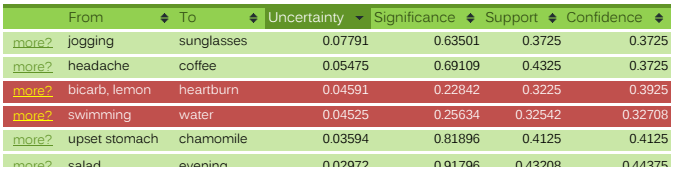
Finally, we discuss the use of open questions: Note that $\mathcal{I}$ is initially unknown to the algorithm, and discovering items can only be done via open questions. We essentially chose between (possibly) gaining new information by asking an open question, and using the existing information to reduce uncertainty, by asking a closed one. This relates to *sequential sampling*, where many strategies are proposed for balancing this type of tradeoff. Any one of the strategies described in the literature can be used, but empirical studies show that typically the trivial solution, where the decision is done based on flipping a (weighted) coin, is the most effective one [2]. Now, in each iteration of the algorithm, the system first decides between asking an open or closed question; if a closed question is chosen, it selects the question for the enabled rule that is expected to maximize the error reduction.

5. SYSTEM OVERVIEW

CrowdMiner is implemented in Python using a SQLite database. Figure 1 depicts the system architecture.

The main system workflow is shown in blue. The system asks crowd questions, analyzes the answers, and constructs further questions accordingly. The *Question Display* user interface shows users the current question to answer. Questions are received from the *Question Selector*.

This module operates over the *Rule Database*, a relational database which stores all the rule data currently known to us. Using the principles described in Section 4, the Question Selector constructs the question which is expected to supply the database with the greatest amount of information. The *Data Aggregator* is in charge of updating the database with user input. It takes the user answers, converts them to database format, and aggregates rule information.



	From	To	Uncertainty	Significance	Support	Confidence
more?	jogging	sunglasses	0.07791	0.63501	0.3725	0.3725
more?	headache	coffee	0.05475	0.69109	0.4325	0.3725
more?	bicarb, lemon	heartburn	0.04591	0.22842	0.3225	0.3925
more?	swimming	water	0.04525	0.25634	0.32542	0.32708
more?	upset stomach	chamomile	0.03594	0.81896	0.4125	0.4125
more?	salad	evening	0.02972	0.91796	0.43208	0.44375

Figure 2: Learned Rules List

The second workflow, shown in red, allows users to navigate the interactive portal and view the learned rules. The data is returned using the *Best Rules Extractor*. The *Portal Interface* enables embedding data on rules in structured Web pages.

Interacting with users. In practice, in order to interact with people, we need to phrase questions in natural language, and define the format of the answer input. Consider the closed question *swimming* $\rightarrow^?$ *energy drink*. In our case, asking a question about this rule does not require complex natural language processing: In short, we keep for each item a relevant phrase ("*go* swimming", "*have an* energy drink"). The phrases are placed in two *question templates*, corresponding to the support of "swimming" and the confidence of *swimming* $\rightarrow^?$ *energy drink*, respectively: "How often do you go swimming?" and "When you go swimming, how often do you have an energy drink?". We also have templates for open questions, such as "Complete the sentence: When I go [blank] I wear [blank]".

To simplify the processing of the answers, they are collected via structured forms. For instance, to express the frequency of a habit, the user is asked to provide the number of occurrences in a text field and choose the time period from a drop down list (for instance, "3 times a week"). This answer is then interpreted back into a support value (e.g., $\frac{3}{7}$). In the interpretation of answers to open questions, which include user-provided items, we use the WordNet ontology to correct spelling mistakes and unify synonyms [10].

6. DEMONSTRATION

As explained in the introduction, we present an interactively-constructed *WellBeing* portal as an application that uses an underlying *CrowdMiner* engine. In this portal, users can find useful tips and information about well-being habits such as sports, healthy diets, or natural medicine. While they are browsing the Web site, they are asked questions from time to time, their answers further enriching the knowledge base.

As a preparatory step to the demonstration we will construct an initial knowledge base, through the *WellBeing* portal, with the help of volunteers. Then, this initial knowledge will grow further with data from the conference participants.

During the demonstration, two laptops will be used to present the portal in two modes, *standard* and *administrator*. The conference participants will be invited to browse through our portal and fulfill the two-fold role of its users: data consumers and contributors. In parallel, we will inspect the administrator view and see how their actions affect the underlying knowledge base. Consider, for example, the following demonstration scenario. Alice, a conference participant, volunteers to try the *WellBeing* portal. She starts from the main page, that displays various well-being information, including new and interesting rules collected from her colleagues. Alice may then decide to perform a search for her favorite ingredient, say, garlic. In the search results page, Alice can find different items and rules related to her search, for instance, the item "garlic pills" or the rule "When I have the flu I take garlic". Alice may then choose to click on one of them, and be directed to a page containing relevant information, related items and rules, useful links and more.

During her browsing of the portal, a pop-up window may occasionally appear and ask Alice to provide data about her habits. For instance, to collect data about the *swimming*$\rightarrow$ *energy drink* rule, she may be asked "How often do you go swimming?" and "When you go swimming how often do you take an energy drink?". Alice can input her answers by checking the relevant option, e.g., "once a week" for swimming, and "never" for energy drink.

We can now turn to the administrator-mode portal to see the effect of Alice's answer, by going to the *Learned rules* page. In this page we can see a list of all the learned rules, painted green if they are estimated to be significant and red otherwise (See screenshot in Figure 2). We can now find the rule *swimming* $\rightarrow$ *energy drink* and see how Alice's answer changed its significance. We can also sort the rules to find, e.g., the most significant ones, or the ones with the highest uncertainty. To see on which rules the system poses questions, we can look at this list while a question pop-up is displayed. This provides intuition into the interactive nature of the underlying algorithm.

Acknowledgments. This work has been partially funded by the European Research Council under the FP7, ERC grants Webdam, agreement 226513, and MoDaS, agreement 291071, and by the Israel Ministry of Science.

7. REFERENCES

[1] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *VLDB*, pages 487–499, 1994.

[2] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd mining. In *SIGMOD*, pages 241–252, 2013.

[3] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. Tan. Asking the right questions in crowd data sourcing. In *ICDE*, pages 1261–1264, 2012.

[4] N. Bradburn, L. Rips, S. Shevell, et al. Answering autobiographical questions: The impact of memory and inference on surveys. *Science*, 236(4798):157–161, 1987.

[5] CureTogether. `http://curetogether.com/`.

[6] M. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *SIGMOD*, pages 61–72, 2011.

[7] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54(2):125–152, 2004.

[8] A. G. Parameswaran and N. Polyzotis. Answering queries using humans, algorithms and databases. In *CIDR*, pages 160–166, 2011.

[9] H. Toivonen. Sampling large databases for association rules. In *VLDB*, pages 134–145, 1996.

[10] WordNet. `http://wordnet.princeton.edu/`.

[11] M. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *RIDE*, pages 42–50, 1997.