Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Exploiting user feedback for online filtering in event-based systems☆

Fabio Petroni [a], Leonardo Querzoni [a,*], Roberto Beraldi [a], Mario Paolucci [b]

[a] Department of Computer Control and Management Engineering Antonio Ruberti, Sapienza University of Rome, Italy

[b] Institute of Cognitive Sciences and Technologies, CNR, Italy

ARTICLE INFO

Article history:
Received 29 April 2016
Received in revised form
10 October 2016
Accepted 13 October 2016
Available online xxxx

Keywords:
Event-based systems
Recommendation systems
Content filtering
Distributed systems

ABSTRACT

Modern large-scale internet applications represent today a fundamental source of information for millions of users. The larger is the user base, the more difficult it is to control the quality of data that is spread from producers to consumers. This can easily hamper the usability of such systems as the amount of low quality data received by consumers grows uncontrolled. In this paper we propose a novel solution to automatically filter new data injected in event-based systems with the aim of delivering only content consumers are actually interested in. Filtering is executed by profiling producers and consumers, and matching their profiles as new data is produced. Profiles are built by aggregating feedback submitted by consumers on previously received data.

1. Introduction

In the recent years the world of global information has been dramatically changed by the widespread participation of people to social networks, whose user base continue to grow, as of today, at an astonishing pace. In such systems users can freely exchange data in several forms: status updates, tweets, links, comments, etc. Users select data they are willing to receive using coarse grained selection methods like friend/follower lists, tag selection or free text searches; then the social network provides each user with a personalized data stream whose content takes into account user selection criteria, but also includes other data that is considered to be potentially interesting for the user or that simply is the output of some advertising campaign. Users are sometimes given the possibility to provide feedback on data they receive (e.g. "I like" on Facebook or re-tweets on Twitter), but this feedback is commonly taken into account only to increase the amount of data injected in the stream or to suggest users new selections and connections. This coarseness in the available selection criteria, mixed with the current data growth rate, starts to limit their effectiveness in conveying useful information to users: as user streams grow to unmanageable sizes they tend to contain more and more information whose perceived quality is low from the user standpoint. This problem, if not adequately addressed, could hamper the usability of these systems. An ideal system should be able to filter-out at run-time from each user stream all those data that the user would personally rate as "low quality", only delivering data that matches the user selection criteria and satisfies some personal quality constraints. Current solutions in the state of the art related to *reputation systems* or *collaborative filtering* cannot be easily adapted to solve this online filtering problem as they either lack generality, or are inherently centralized.

In this paper we propose a novel online filtering solution for event-based systems (social networks are, in fact, moving from batch data mining to on-the-fly event processing [1]). Our solution leverages feedback expressed by users on received events to profile data sources. Profiling is based on a reputation metric calculated as a result of a collaborative process performed among all destinations that received and evaluated previous events injected by a same source. The profiling process takes into account that a source can produce events with different quality levels on different topics. New events injected in the system are filtered online depending both on the profile associated with their sources and on profiles built for characterizing the minimum quality thresholds of each potential destination. A destination profile is built considering all the feedback each user expressed on previously delivered events.

In social networks like Facebook or Twitter, such a solution could be adopted to automatically filter incoming updates from the user timeline or tweet stream, allowing him to focus only on updates that shall, with high probability, capture his interest. A

☆ A preliminary version of this work appeared at the 31st ACM Symposium on Applied Computing.

* Corresponding author.
E-mail addresses: petroni@dis.uniroma1.it (F. Petroni), querzoni@dis.uniroma1.it (L. Querzoni), beraldi@dis.uniroma1.it (R. Beraldi), mario.paolucci@istc.cnr.it (M. Paolucci).

http://dx.doi.org/10.1016/j.future.2016.10.017
0167-739X/

different example could be represented by social news aggregation services, like *Feedly*, that aggregate syndicated content from multiple RSS feeds and propose it to their users; users could express their interest in a news by signaling it via Feedly's "add to knowledge board" functionality, or by simply sharing the content with other users; lack of interest in a news could be identified at browser level by tracking the amount of time each stream item remains visible in the browser window and then pushing back this info toward Feedly's servers; information on various RSS streams and topics (tags) collected from its user base could be used by Feedly to build profiles needed to filter out uninteresting items from user streams, or to promote more interesting news to the top of the stream.

We evaluated the performance of our solution through a simulation-based study whose results show that it is able to effectively profile both source and destination users and to automatically filter out a large percentage of events that would be negatively evaluated by their recipients with a low percentage of false positives (high quality events that are erroneously filtered out).

The rest of this paper is organized as follows: Section 2 presents related works; Section 3 defines the system model and states the problem; Section 4 presents our solution, evaluated in Section 5; finally, Section 6 concludes the paper.

2. Related work

The interest in reputation shown by economy and game theory (see for example [2]) was quickly followed by a surge of attention in ICT. As the number of online users and transactions increased, transcending geographical limitations and personal acquaintances, traditional one-to-one word of mouth proved insufficient.

Very soon, the first systems supporting reputation appeared online, proving themselves essential for trust maintenance and partner selection, and at the same time showing their vulnerabilities under specific attacks [3], pointing out the best statistics for reputation estimation [4] and using simulation to show the importance of cognitive aspects [5].

More recently, surveys on applications for reputation systems began to appear [6,7]. Finally, as the field consolidated in the most recent years, simplified recipes [8] have been proposed. The interaction between reputation and crowdsourcing has also been explored recently [9].

Reputation in large scale dynamic systems has been studied in EigenTrust [10] that offered a fully decentralized solution to calculate a global reputation rating for every peer in a P2P network, based on their previous behavior. While the fully distributed approach is common to our solution, we aim at building quality estimators that characterize each publisher in different ways depending on the content of events he publishes. The same goal could be accomplished in Eigentrust only by running an instance for each CBA, an approach that would run extremely inefficiently.

A different solution in the area of reputation system has been proposed by Ismail and Josang in [4] where they introduced the *Beta Reputation System*. Their solution builds reputation scores using the beta probability density function, thus creating trust on the basis of a sound statistical methodology. Differently from the *beta reputation system*, that is proposed as a centralized solution, the system proposed in this paper provides a fully decentralized approach that still bases its output on the same beta function.

The authors of [11] pursued our same goal (i.e., thin out the user streams) with a different approach. Their solution is tailored to the Twitter social network and consists in profiling user by using the categories extracted from URLs posted in their tweets, and then using these categories to filter information stream. Conversely, the solution proposed in this work is of general applicability for any event-based systems.

Similarly to our work the solution proposed in [12] aims at offering a personalized stream of randomized web pages to each user. Their mechanism is strongly tied to collaborative filtering [13,14], a thriving subfield of machine learning, which involves finding similarities between users and making recommendations based on what similar users like. Collaborative filtering solutions in the literature are often divided in two groups: *memory-based* and *model-based* [15]. Memory-based methods are used in a lot of real-world systems because of their simple design and implementation. However, they impose several scalability limitations that make their use impractical when dealing with large amounts of data. Model-based approaches have been investigated to overcome the shortcomings of memory-based algorithms. The most successful Model-based techniques are by far those based on low-dimensional factor models, as the Netflix Prize (www.netflixprize.com) established, in particular those based on *matrix factorization* [16]. The most popular matrix factorization solutions are *Alternating Least Squares* (**ALS**) and *Stochastic Gradient Descent* (**SGD**). Recently, collaborative filtering has been applied to the area of binary rating recommendations with an efficient solution based on statistical profiling of user feedbacks [17]. None of the solutions in the literature, to the best of our knowledge, has been applied in a event-based messaging scenario. Furthermore, all these solutions build a model from an online training phase by continuously updating a large matrix including data for any user in the system. This approach typically delivers great recommendations but is extremely difficult to decentralize making it unappealing for distributed event-based systems. Conversely, our solution has been designed with the aim of being easily and effectively deployable is a fully distributed setting.

A typical problem that affects many currently deployed reputation/recommendation systems is represented by malicious users whose aim is to subvert the normal system behavior such to drive its output toward their desired goals. While systems are typically protected from direct attacks at their protocols and algorithms, defending against potentially large groups of colluding users is extremely difficult. Colluding users may, in fact, submit perfectly legit feedbacks to purposely drive the rating of an item toward unnaturally large or small values. This phenomenon, studied for example by Lien et al. in the Maze file-sharing system [18], has several complex facets and nuances that have been extensively analyzed in the survey by Hoffman, Zage and Nita-Rotaru [19]. Negative effects of such attacks can be mitigated by avoiding user impersonation or multiple identities (i.e. sybil attacks), limiting the generation and spread of false feedbacks and avoiding short-term abuses of the system. Our solution, designed with the aim of providing maximum efficiency and decentralization, does not adopt any of these countermeasures, and is thus susceptible to colluding attacks; however, by its nature it is not incompatible with the techniques described in the cited literature, and could thus be enhanced with collusion resistance capabilities. This evolution is left for future work.

3. System model and problem statement

We consider a system where a set of users can produce or consume information. Users that exchange information adopt an *event-based publish/subscribe* communication model [20]. Without loss of generality, we assume that a user is either a producer of events (*publisher*) or an event consumer (*subscriber*). Each event is characterized by a *content-based address* (CBA), defined in an *event space*, and a *payload*. Publisher can publish events in the whole event space, i.e. there is no restriction on the CBA of events. A subscriber can select the events he wants to receive by issuing a *subscription* that defines a subset of the event space, and thus restricts the CBAs accepted by the subscriber. If an

Please cite this article in press as: F. Petroni, et al., Exploiting user feedback for online filtering in event-based systems, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.10.017