Information Overload In Social Media Streams And The Approaches To Solve It

Maria Grineva
Yandex Labs
299 S California Ave #200
Palo Alto, CA 94306-1914, USA
mariagrineva@yandex-team.ru

Maxim Grinev
Yandex Labs
299 S California Ave #200
Palo Alto, CA 94306-1914, USA
maximgrinev@yandex-team.ru

ABSTRACT

We discuss the problem of information overload in social media streams. We identify two groups of approaches to solve the problem. The first group is based on filtering social media streams. These methods are already quite mature and successfully used in practice. The second group of approaches proposes completely different paradigms for information sharing and consumption rather than stream. Such approaches only start to emerge, and we discuss opportunities and research challenges that they raise. One of the alternative paradigms described in this paper, *passive information consumption via the push model*, is currently being developed by the authors at Yandex Labs.

1. INTRODUCTION

Streams have become a conventional model for both distribution and consumption of information shared on social media sites. In the beginning of the Web 2.0 era, RSS streams boosted the popularity of blogging by providing a uniform way to connect bloggers with their readers. Bloggers got a tool to syndicate their frequent blog entries automatically, while readers were able to easily subscribe to several RSS feeds.

Later, social networking sites adopted the stream model for the information shared by their users. Facebook has registered its users friends relationships building global social graph, Twitter has introduced nonreciprocal following relationship thus building global "interest graph". Users of social networking sites continuously produce a stream of shared status updates, photos and links and receive a united stream from all their friends or followees. The role of a social graph is to define the routes for the streams of information shared by the users.

However, today a lot of active social media users complain that their streams have become too overloaded and hard to extract useful information from. There are several reasons for this overload. Besides the general increase of usage and amounts of data shared every day on social networks, we distinguish the following two reasons: the growth of the number of connections in the social graph and automatic updates coming from applications.

Social graph has become too general. Facebook's social graph started originally as a tool to keep in touch with friends, has now become too unspecific and dense. Many Facebook users have more than 200 friends ranging from their real friends and relatives to people they have met at professional events and feeds from favorite brands and news sites. According to the recent study of Facebook social graph [24], the average distance between vertices of its giant component was found to be 4.7, that indicates that individuals on Facebook have potentially tremendous reach. Shared content only needs to advance a few steps across Facebook's social network to reach a substantial fraction of the world's population. This growth of connections density is a natural process, because as soon as the user discovers another interesting source of information, they like to subscribe to keep up with it. But all those streams put together into a common timeline represent a huge, messy, hard to consume stream, while, in its depths, it still contains relevant information.

Streams of updates automatically generated by applications. In late 2011 Facebook introduced so called *frictionless sharing* for applications. Now applications can automatically post *verb-based* updates on behalf of its users about what users are doing within the application, without the user having to manually push a *"Share"* button. So that applications are able to post on users' walls as soon as they, for example, *listen* to a song on Spotify, *look up a recipe* on Foodily, *buy tickets* on Ticketmaster, or *go for a run* using Nike+. This move by Facebook provoked a lot of criticism: obviously, such amounts of data pollute an already overloaded stream. Moreover, it raises privacy issues. We believe that this phenomenon is inevitable because, on one hand, applications are interested to spread their updates on Facebook to reach more users and, on the other hand, users benefit by getting more information from their friends. And it is not a question if to allow or prohibit applications to post what they want, but rather a question of creating the right tools to make use of this new kind of data.

We discuss the approaches to turn flooded social media streams into manageable sources of useful and relevant information. We divide all existing approaches into two big groups. The first group aims to **filter** out the noise in the stream and to pick out the most important and relevant items. A number of solid solutions based on filtering have been provided by the research community as well as by several start-up companies. We give an overview of these solutions. Approaches of the second group give up the stream model and propose **different paradigms** for information sharing and consumption in social media. We point out interesting research challenges raised by these approaches.

2. FILTERING THE NOISY STREAM

As Clay Shirky, a popular social media theorist, once stated: creating effective filters is the way to deal with information overload [23]. We discuss filtering from the three following perspectives: (1) algorithmic filtering, (2) dividing the stream into sub-streams by topic (3) hand-curated streams.

2.1 Algorithmic Filtering

Filtering social streams has attracted a fair amount of attention in the research community and in the industry. In [10, 12], the authors propose content recommendation system for Twitter that tackles the problem of information overload helping users both to filter the stream down to those items that are indeed of interest and to discover interesting content from outside of their stream. Users set their topics of interest, the system identifies URLs relevant to users' interests, and ranks the URLs. The ranking is based on social voting that considers social interaction between users who have mentioned the URL. In [21], authors present a system for discovering news related posts on Twitter and clustering these messages based on their location. In Twahpic system [1] developed in Microsoft Research each post is examined using topic models [20, 19] built on recently fetched posts from the Twitter stream. Each topic is labeled by one of the categories (*Substance*, *Social*, *Status*, or *Style*). The system can then evaluate each post and decide the ratio of each category in that post.

Outside of academic research, several start-up companies provide stream filtering or recommendation services, such as my6sense [2], Zite [7], and The Tweeted Times [6]. Both my6sense and Zite implement topic-based filtering of RSS feeds and Twitter streams using feedback from the user. The Tweeted Times has been built by the authors of this paper and now it is being integrated with Yandex' services in Yandex Labs. As apposed to filtering by topic relevance, The Tweeted Times takes a different approach: it identifies the most frequently mentioned URLs among user's followees on Twitter and turns the overloaded stream into a concise personalized newspaper. We believe that friends in social networks are the best filter for news recommendations: if you follow someone on Twitter that means you are interested in links they post. Rather than asking the user to provide her interests as keywords, we use her immediate social graph which best describes them implicitly. As soon as the user's interests have changed, she discovers new people to follow, thus her social graph reflects it.

Facebook also uses social graph to filter posts. Its EdgeRank algorithm [15] determines what stories from their friends the user sees on their news feed when they log in to Facebook. Every item that shows up in user's news feed is considered an *object*. Whenever the user interacts with an object she creates what Facebook calls an *edge*, which includes actions like *tagging* or *commenting*. Each edge has three components important to Facebook's algorithm: (1) affinity score between the viewing user and the object's creator, which bases on how often they communicate on the site, (2) weight given to each type of edge (for example, a *comment* have higher weight than a *like*) and (3) time - the older an edge is, the less important it becomes. Summing up the multiplication of these factors gives an object's EdgeRank. And the higher is the rank, the more likely the object is to appear in the user's news feed.

2.2 Dividing The Stream Into Substreams

Dividing the united stream into smaller substreams by topic and/or importance seems to be a natural step towards better stream management. All major social networking sites provide instrumentation to do it: Twitter has lists, Facebook allows to organize groups of friends, and Google has introduced Google+ circles.

Twitter list is a manually created set of Twitter users who often tweet on a certain topic. For example, anyone can create a Twitter list of users who are working on research of recommender systems and often tweet about it (such as this list: *https://twitter.com/alisohani/recommender-system*). The stream produced by such a list is supposed to contain news and articles on recommender systems. It does contain a lot of relevant content about recommender systems. However it is still quite noisy because the list members have other interests too and do not tweet only about the recommender systems. Twitter lists (and similar mechanisms) need filtering to become really useful.

Our research project described in [14] proposes a method for filtering thematic Twitter lists. It has shown that for most thematic Twitter lists it is possible to automatically identify central topic using LDA topic modeling. The method proposed can incrementally maintain the topic once it was identified. It is a necessary requirement because the central topic of a Twitter list may change over time. As soon as the central topic is identified, new coming tweets are classified as relevant or irrelevant to the topic of the list and irrelevant tweets are filtered out. Experiments show that the method filters out the irrelevant tweets with 86% accuracy.

2.3 Hand-Curated Streams

Another approach is to rely on an authoritative curator that would filter the news providing only the most relevant news. Dozens of news curation tools have been proposed by start-ups recently. Twitter is often used by journalists for news curation. Mathew Ingram *@mathewi*, for example, curates news in technology. Twitter retweets is a unique instrumentation due to which news can spread in minutes all over the world. Reweet preserves the original tweet's authority. Among other popular curation tools are Storify, which allows combining social media posts into pictorially presented stories, and Instapaper, a bookmarking service, which can also be used to follow links bookmarked by other users.

3. ALTERNATIVE PARADIGMS

We discuss several alternative approaches that give up the stream model and propose different paradigms for information sharing and consumption in social media.

3.1 Information Boards vs. Streams

As apposed to sharing into a stream, users can add new items into a structured information space thus incrementally building and improving it. Pinterest [3], a new social bookmarking site, lets its users to organize items into topical boards. Thus, as apposed to sharing a new item into a stream, it is "pinned" onto an appropriate board. A number of items on a board can grow infinitely (like, for example, on a board *"cities I want to visit"*) or until its logical completeness (*"an outfit for a Christmas party"*). Polyvore [4] is another example of this idea. On Polyvore users create collections of clothes and accessories that best fit all together.

3.2 Building Flash Mobs for Real-Time Communication on Social Sites

One implication of the overloaded news feed on Facebook is that users tend to read some latest updates only and do not read the stream far back. As soon as they have looked through the latest updates, they start communicating around them and stick at the top of the stream. Often such communication happens in real time as friends who have posted the recent posts are still online. To our opinion, one of the reasons Facebook popularity grows faster than Twitter's one is that Facebook is a mix of stream and real-time communication platform (users can see who is online and start chatting, play games, etc.) while Twitter is a pure stream platform. Opening Facebook is like entering a room where continuous discussion is always going on and seeing who is there right now discussing what. The problem is that if the current discussion on the news feed is not interesting to the user they do not have instrumentation to "change the room" to join another group of people. We need a service that would dynamically fetch small groups of people from the user's big general contact list, specific to their current context. So that the user can communicate/interact within the group in real time and then move on. We call such a group a flash mob.

Such services start to appear. For example, on Turntable.fm [5] people get together in virtual rooms to listen to music and play music for each other, communicate and interact synchronously. On Turntable.fm, user's Facebook contact list serves as a serendipitous guide to choose the room: one can check out the rooms where their Facebook friends are virtually present right now. Friends-based recommendation serves only as a hint to start. Alternatively, one can choose a room that suits best her current situation, like *"Coding Soundtracks"* or *"90's hits"*.

Another example is Google+ Hangouts - online video chat with friends on Google+. From GigaOm's post [16]: "It isn't a chat (in the traditional Internet sense) and it isn't a conference call. Hangout with folks you want to connect, even for a few seconds, enjoy an immersive interaction and then move on."

The main challenge for services like these is to develop effective people recommendation algorithms to build a flash mob. In [13] authors give an overview of people recommendation algorithms: some of them are based on matching user's profile to the topic of the group or profiles of others while other methods consider social connections between the users. To provide effective recommendations for flash mobs these algorithms should also take into account real time information about users (such as the probability of users' availability at the moment and the mood of the users) similar to how it is used in time-aware music recommendation systems [22, 8].

3.3 Passive Information Consumption Via The Push Model

What if to stop worrying about missing something important in the stream and rely on a data processing tool that indexes everything that your friends posted in social networks and pushes you back with an important piece of information at the right time or at the right place.

Our group at Yandex Labs is working on tools that collect user's personal data from social networks in order to index it and to make it available for the user later when they appear in the appropriate context. We see interesting research challenges in developing tools for understanding the user's current context. Under the context we understand current user's activity such as *"eating"*, *"shopping"*, *"watching TV"*, *"on a business meeting"*, *"meeting with friends"*, *"reading"*, *"working"* and other. In [18] authors propose methods to infer the current context from mobile censor data (geolocation, time), user's calendar and search queries. We extend these methods using (a) the information about people who are currently located near the user and (b) user's activity in social networks and various applications (e.g. email). Another interesting challenge is to predict user's future context based on the previous user activity [17]. For example, the system should recommend a movie to watch before the user already started to watch something. How to model the combination of factors when many different contextual factors have been extracted is discussed in [11, 9].

It is important to note that in the most of the existing research works, user's activity in social networks does not play a major role. While with the recent launch of the Facebook's frictionless sharing discussed in the introduction, a lot of data about the user's current activity is now pouring into Facebook. This data is available upon the user's permission and has a huge potential in identifying the user's current context.

4. CONCLUSION

The growth of social media challenges its main mechanism of information distribution and consumption - streams. Along with well-established stream filtering methods there emerge new approaches that proposes alternatives to the stream model. We have discussed the new approaches. To our opinion, push-based recommendation that take into account the user's current context seems to be the most general and promising one.

5. REFERENCES

[1] Microsoft Research Twahpic. `http://twahpic.cloudapp.net/About.aspx`.

[2] my6sense - a service for filtering RSS and social streams. `http://www.my6sense.com`.

[3] Pinterest - a visual bookmarking service. `http://pinterest.com/`.

[4] Polyvore - a fashion community Web portal.

[5] Turntable.fm - a real-time music listening service. `http://turntable.fm/`.

[6] The Tweeted Times - a personalized newspaper generated from your Twitter account. `http://tweetedtimes.com/`.

[7] Zite - a personalized magazine for iPad and iPhone. `http://zite.com/`.

[8] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. *Workshop on ContextAware Recommender Systems CARS 2009 in ACM Recsys*, 2009:1–5.

[9] L. Baltrunas, M. Kaminskas, F. Ricci, L. Rokach, B. Shapira, and K. H. Luke. Best usage context prediction for music tracks. *idscsomumnedu*, 2010.

[10] M. S. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. H. Chi. Eddi : Interactive topic-based browsing of social status streams. *Fortune*, pages 303–312, 2010.

[11] T. Bogers. Movie recommendation using random walks over the contextual graph. *Search*, 2010.

[12] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. H. Chi. Short and tweet : Experiments on recommending content from information streams. *Scanning*, pages 1185–1194, 2010.

[13] E. M. Daly, W. Geyer, and D. R. Millen. The network effects of recommending social connections. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 301–304, New York, NY, USA, 2010. ACM.

[14] B. Guc. Information filtering on micro-blogging services. *Mather thesis*, 2010.

[15] J. Kincaid. Edgerank: The secret sauce that makes Facebook's news feed tick. *TechCrunch*, April 2010.

[16] O. Malik. Google hangouts gives the.

[17] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, and F. Hattori. A recommendation system considering users' past/current/future contexts. *idscsomumnedu*, pages 3–7, 2010.

[18] K. Partridge and B. Price. Enhancing mobile recommender systems with activity inference. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH*, UMAP '09, pages 307–318, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. *International AAAI Conference on Weblogs and Social Media*, 5(4):130–137, 2010.

[20] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda : A supervised topic model for credit attribution in multi-labeled corpora. *Language*, 13(August):248–256, 2009.

[21] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. *Information Storage and Retrieval*, pages 42–51, 2009.

[22] J. Seppanen and J. Huopaniemi. Interactive and context-aware mobile music experience. In *Proc. of the 1th Int. Conference on Digital Audio Effects (DAFx-08)*, 2008.

[23] C. Shirky. It's not information overload. It's filter failure. *Web 2.0 Expo*, September 2008.

[24] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the Facebook social graph. *CoRR*, abs/1111.4503, 2011.