

The Contextual Turn: from Context-Aware to Context-Driven Recommender Systems

Roberto Pagano,
Paolo Cremonesi,
Massimo Quadrana
Politecnico Di Milano
Milan, Italy

<name.surname>@polimi.it

Balázs Hidasi,
Domonkos Tikk
Gravity R&D Inc.
Budapest, Hungary

<name.surname>@gravityrd.com

Martha Larson
Delft University of Technology
& Radboud University
Delft & Nijmegen, Netherlands
m.a.larson@tudelft.nl

Alexandros Karatzoglou
Telefonica I+D
Barcelona, Spain
alexandros.karatzoglou@telefonica.com

ABSTRACT

A critical change has occurred in the status of context in recommender systems. In the past, context has been considered ‘additional evidence’. This past picture is at odds with many present application domains, where user and item information is scarce. Such domains face continuous cold start conditions and must exploit session rather than user information. In this paper, we describe the ‘Contextual Turn’: the move towards context-driven recommendation algorithms for which context is critical, rather than additional. We cover application domains, algorithms that promise to address the challenges of context-driven recommendation, and the steps that the community has taken to tackle context-driven problems. Our goal is to point out the commonalities of context-driven problems, and urge the community to address the overarching challenges that context-driven recommendation poses.

Keywords

position paper; context; recommender systems

1. INTRODUCTION

In recent years, the recommender system community has discovered the benefits of context. A common characterization of context-aware recommender systems is that they ‘try to incorporate or utilize additional evidence (beyond information about users and items) to estimate user preferences on unseen items’ [1]. In this position paper, we declare that it is time for a *Contextual Turn*, which revisits this definition, and acknowledges that being *aware* of context is not

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '16, September 15 - 19, 2016, Boston, MA, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4035-9/16/09...\$15.00

DOI: <http://dx.doi.org/10.1145/2959100.2959136>

always enough. Instead, an increasing number of application domains call for recommender system algorithms that are actually *driven* by context. In contrast to context-aware recommendation, in context-*driven* recommendation users and items take a back seat.

For the philosophically minded, the Contextual Turn is valuable because it overthrows the assumption that personalization in recommender systems involves recommendation for specific individuals. Instead, a context-driven recommender system, ‘personalizes’ to users’ context states. In this way, it introduces a disassociation between users and their historical behavior, giving users room to develop beyond their past needs and preferences. Instead, users receive recommendations based on what is going on around them in the moment (situation) and on what they are trying to accomplish (intent). The Contextual Turn re-envision the original insight of user-user collaborative filtering by taking the position that people have more in common with other people in the same situation, or with the same goals, than they do with past versions of themselves.

The Contextual Turn is very much a phenomenon of this moment. Until recently, a system whose main driver is context, would not be conceivable. Context-driven recommendation is made possible by a confluence of developments: huge amounts of data generated by users in a wide range of contexts are available, new sources of context-data are provided by sensors (smart phones, Internet of Things), and the computational power to process this data is within our reach. Finally, recent years have seen a rapprochement of recommender systems and information retrieval (IR). In IR, relevance has long been acknowledged as situational, e.g., dynamic, multidimensional, and dependent on user goals and situation [3]. IR also realizes that users themselves fill the gap between the results list and the perfect item, i.e., using diversification of results lists in cases in which the user’s information need may not be completely clear [14]. Together, these new resources and new insights make *now* the right moment to launch consolidated effort towards context-driven recommendation.

Conceptually, the Contextual Turn can be seen as a turn away from what we call the *Immutable Preference* paradigm

(ImP) in recommender systems. ImP assumes that users goals, needs, and tastes do not develop, and that the set of items to be recommended remains relatively static. Current state-of-the-art models try to overcome the ImP assumptions by tweaks to existing models. For example, they introduce decay to prioritize recent preferences over past preferences, they filter the recommendation list to include only currently available items, or they adjust weights to account for seasonality or trends. Recommender systems after the Contextual Turn will reject the assumption of ImP in their algorithmic fabric, rather than as an afterthought.

We do not expect the Contextual Turn to introduce a discontinuity into recommender system research. In fact, the underlying challenges of context-driven recommendation can be seen as more extreme forms of familiar recommender system challenges. The conventional problem of sparse user-item interactions is driven to an extreme, since a context-driven recommender uses a context-state as a target, and may have no user ID whatsoever. Further, for the context-driven recommender systems the cold start challenge is not a start-up challenge, but rather user and item dynamics make the challenge *continuous cold start*.

We anticipate that the Contextual Turn will unify currently disparate strands in the recommender system community. Most obviously, context-aware recommendation is the direct progenitor of context-driven recommendation. However, other areas also have something to add. We mention session-based recommendation, which models context as a series of user interactions carried out within a session. Then, both content-based recommendation and stream-based recommendation also contribute because of their power to tackle cold start and sparse user interactions. The Contextual Turn is in line with the operational requirements for today’s commercial recommender systems. Due to the difficulties of session modeling, most online recommenders use item-item recommendations. Although they account for 80% of recommendation traffic only 0.1% of recsys research focuses on this topic [12]. The Contextual Turn puts focus on session modeling and brings research into better alignment with industry needs.

Even though we name the Contextual Turn, we do not claim to have invented it. The notion of *context-driven recommendation* has been in the air for some time. Our concern is that its current state of fragmentation, pursued, as it is, via isolated research efforts under different names, is not particularly productive. Our purpose here is to point out commonalities across a range of application domains that build a unified picture of context-driven recommender systems as a domain-overarching challenge for recommender systems. By discussing the potential of existing algorithms, and the challenges that still remain open, we aim to encourage the recommender system community to join forces and develop new breeds of algorithms specifically suited to context-driven recommendation. Section 2 provides a short recap of context-aware recommendation, Section 3 discusses domains which, at present, have emerged as needing context-driven recommendation. Then, Section 4 covers algorithms suitable for context-driven recommendation, and sets the stage for Section 5, discussing the future challenges faced in order to successfully execute the Contextual Turn.

2. PAST: CONTEXT-AWARE RECSYS

Conventionally, context-aware recommender systems fall

into three types: *contextual pre-filtering*, where context drives data selection; *contextual post-filtering*, where context is used to filter recommendations once they have been computed using a traditional approach; and *contextual modeling*, where context is integrated directly into the model [1]. An example of contextual pre-filtering is the so-called *user micro-profile*, in which a single user is represented by a hierarchy of possibly overlapping contextual profiles [2]. Post-filtering methods can use traditional approaches and then apply filtering or weighting. In their experimental evaluation, Panniello *et al.* [10] found that the choice of a pre-filtering or post-filtering strategy depends on the particular recommendation problem.

Context-aware recommendation has become increasingly sophisticated, since the first models, which simply considered items to be different items if the context was different, i.e., a restaurant was considered a different restaurant when it was raining that when it was sunny. The Contextual Turn can be considered to represent a return to the idea that the context has a strong determining force on the item.

3. NEW CONTEXT-CENTRIC DOMAINS

The trigger of the Contextual Turn is the rise of new application domains in which the ImP assumptions fail to hold, and which require context-driven recommendation. Here we cover key domains, and mention examples of existing work.

Domains with *seasonality* patterns are strongly context-centric: Linear TV consumption is periodical and strongly dependent on the channel rather than the program itself [16]. E-tourism faces a similar break down of ImP assumptions: touristic items are time-variant and price-sensitive products. For instance, prices and cancellation policies of flight tickets or hotel rooms are dynamic and heavily depend on both absolute time context (e.g., low-season vs high season) and relative time context (e.g., days before booking). Incorporating seasonality as a contextual variable has already been proven to deliver better results [16, 5].

Domains with *availability* patterns cannot assume that all items are always available, but rather are dependent on contextual factors such as the number of users who have ‘consumed’ items at any moment of time. Examples of *limited availability* of items (or bounded capacity), include e-tourism or online marketplaces. Research has suggested that profile-based approaches traditionally used in e-commerce domains do not necessarily work for e-tourism applications, because of the particular nature of the touristic products [17]. Examples in the e-tourism domain are seats in restaurants, or rooms in hotels. A hotel may become unavailable during high season or when the booking time is close to the desired time of usage. The quality of the offering of items is not constant: as the number of available items decreases, the quality of the remaining ones tends to decrease, because the items that are first sold are typically the “best” ones. In a context-driven setting, the availability of the items can be used as a contextual variable, useful for understanding the conditions (situations) that led the user to choose some product among the available ones.

The *turnover* of the items is related to availability. With turnover items disappear or are replaced, never to return. The turnover (or dynamic catalog) property of the domain can be found in the Linear TV, online marketplaces, news and user generated content recommendation domains. In these domains items are added and removed from the cat-

alog very frequently. This problem can be solved by the context-driven paradigm by considering the joint user-item-context information of previous interactions. Moreover the use of sessions can also address this problem, for example by favoring recent sessions (that therefore use recent and available items).

Finally, we turn to consider the dynamics of users. Here, we first point to *personal development*, the changes that users undergo because they learn or otherwise evolve over time. For example, recommending learning materials or jobs must take into account that users evolve over their lifetimes. A related phenomenon is *taste drift*, which also involves changing preferences for items, but is not necessarily a development pattern.

The *immediate task* of the user is crucial in domains such as music recommendation. The authors of [15] developed a learning-free recommender that was only exploiting contextual signals gathered from smartphone sensors. The system provides, e.g., workout music when the user is exercising at the gym.

Session is another contextual signal that plays a fundamental role in the music domain. Sessions are sequences of user interactions with the online service. A subset of these interactions is relative to the items, such as purchase or view. A session usually has a goal, such as finding a good TV program, watching short clips in a certain topic, or listening to music of a certain style or mood. There is an increasing number of session-aware algorithms that exploit the session for providing recommendation. In the music domain, [4] showed that the sequence of songs the user already listened to is a proxy for the kind of music the user want to listen to *in that particular situation*.

4. PRESENT: USEFUL ALGORITHMS

The notion of context key to context-driven recommendation is the user situation and the user intent. This abstract concept must be translated into concrete algorithmic solutions. We point out that context can still be regarded as either fully observable, partially observable or unobservable [1]. A context-driven recommender uses observable context (e.g., geo-location), directly, and relies on inference to exploit partially observable or unobservable contexts (e.g., activity, physical state, mood, user goal).

The existence of domains in which the ImP assumptions breakdown poses a serious challenge for recommender systems. However, the availability of increasing amounts of data from a variety of sources provides an opportunity to compensate. The challenge is to identify helpful information sources to exploit. The RecSys 2015 best paper [9] is an example of work that opens up new possibilities for exploiting different context-variables (or signals), even when there are no user interactions (continuous cold start situation). We also mention that collecting useful data may mean elicitation of information from users, either actively or by presenting them with opportunities to engage in informative interactions. Here, active learning strategies would offer themselves as helpful.

A context-driven algorithm merges characteristics of collaborative, content, and context-aware (and session-aware) algorithms. Of course it's context-aware, since it exploits contextual variables. It is collaborative, but the collaboration happens among context states rather than among users.

It is content-based, since the content of an item can be used to refine the system's understanding of the user context.

Some state-of-the-art algorithms can already be adapted to the context-driven paradigm with some modifications. Here we list some state-of-the-art algorithms and their ability to deliver context-driven recommendations.

Decision Trees (and Gradient Boosted Trees and Random Forest in particular) perfectly fit in the context-driven definition, since they usually do not include user and item in their training. In this way the tree can learn the best hierarchy among the contextual features.

General Factorization Framework (GFF) [8] is a powerful context-aware factorization algorithm that makes it possible to experiment with preference models. The preference model is provided to the algorithm as an input. The number and meaning of the input dimensions is arbitrary. Due to its flexibility, this framework already has the ability to deliver context-driven recommendation. For example, it can exclude the user ID from the preference model.

Factorization machines (FM) are another powerful factorization algorithm that can be applied to context-aware data [13]. Similarly to GFF, FMs can (but are not limited to) model pairwise interactions between the input dimensions, which in turn are not constrained to contain the user (or item). For this reason, they can be used in pure context-driven scenarios.

Session-based algorithms deserve special attention, because they are intrinsically *context-driven*. Recurrent neural networks (RNNs) deliver good performance for modeling and generating sequences. RNNs have been demonstrated in [6] to be useful for session-based recommendations, providing recommendations based on items in the active session of the user. Moreover, deep learning is able to exploit item content directly in order to derive semantic representations of the data and model user preferences. Adding other contextual variables as well as the content of the item and removing the user is straightforward and enables the neural network to model preference patterns in different contexts. Finally, the availability of an increasing amount of data and work in this area, makes Deep Learning a promising direction for context-driven recommendation.

5. FUTURE: CONTEXT-DRIVEN RECSYS

We conclude with a discussion of open issues left by algorithms in Section 4 and a discussion of additional benefits.

5.1 Open Challenges for Research

The Contextual Turn requires addressing a number of research challenges.

Continuous context. Factorization methods require discrete input variables. Context, however, is often continuous (time, geo-location). Methods are needed for integrating continuous information or packaging it into categorical form without damaging its value. Initial work in this direction was carried out by [7].

Missing values. With current state-of-the-art methods, each event must have a value for each input dimension, i.e. missing context is not handled. While this is a realistic requirement for e-commerce and online services, it is far less reasonable for sensory data.

Scalability. Since the recommendation depends on the value of contextual variables, there are two choices: compute the recommendation at the moment of request or pre-

compute the recommendations for every possible contextual state. The second option is feasible if the set of contextual states is not large, but becomes challenging when we are dealing with many contextual variables or sessions.

Evaluation In the simplest case, evaluation can be carried out using conventional approaches, but replacing the user with the appropriate contextual variables. However when dealing with sessions, the evaluation must be adapted, in order to take into account the sequential dependency between items. In some domains, recommendations are only useful to users if strict ordering is respected (e.g., app recommendation for a task). In others, a certain amount of flexibility is allowed (e.g., users tolerate, or even welcome, shuffle in their playlists). From the business perspective it is key that clients are able to understand the metrics, and put them to use.

5.2 Benefits of the Contextual Turn

The benefits of the Contextual Turn potentially transcend conventional requirements for recommender systems.

Popping the Bubble Recommenders focusing on context-state rather than on individuals can counter the dangers of hyper-personalization, i.e., the Filter Bubble problem [11]. Of course context gives rise to its own bubble effect. However, people arguably have more control over their context, the situation and goal of the moment (intent), than over their accumulated past history.

Leveling the playing field Issues of *bias* are closely related to the Filter Bubble. Recommendations for jobs, online educational courses, or scientific literature must be blind to user characteristics such as gender, ethnicity, and social economic background. Under context-driven recommendation, the focus of the recommender system is, by design, on situation and intent, and not on demographic group (i.e., something users do not choose themselves).

Protecting privacy Both user requirements for privacy and laws regarding the storage of personal information are in flux, cf. the European General Data Protection Regulation, adopted April 2016. Facing these calls requires algorithms that need as little data as possible, and, in particular, can avoid storing data associated with specific user ID for extended periods. By focusing on effective exploitation of context and not individuals, algorithms will be robust to future evolution of privacy requirements.

Adding transparency Context-driven recommendations can use context variables to explain recommendations to users. Establishing best practices would require a significant amount of additional research, but we point to the potential of context-based explains to be less ‘creepy’ and more readily understandable for users.

Companies already understood the potential of context for making recommendations. As an example Google Context Awareness API can already exploited to gather high quality contextual data. We hope that this paper will foster a stronger cooperation between academia and industry for better addressing new research paths in recommender systems, e.g., by organizing challenges or providing data for benchmarks.

In closing, we emphasize that we do not expect context-driven recommendation to replace context-aware recommendation or conventional personalization. Instead, our intent is that by explicitly pointing out the ongoing Contextual Turn, we can encourage the recommender system commu-

nity to look specifically at how context can be exploited as the primary source of information for recommendations.

As a final note, we mention that space constraints have kept our reference list here to a bare minimum. As a resource to the recommender system community we have published an extended bibliography¹.

6. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. *Context-Aware Recommender Systems*, pages 191–226. Springer, 2015.
- [2] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. In *CARS Workshop at ACM RecSys 2009*.
- [3] P. Borlund. The concept of relevance in IR. *JASIST*, 54(10):913–925, 2003.
- [4] A. Condorelli, P. Cremonesi, R. Pagano, M. Quadrana, and R. Turrin. Large scale music recommendation. In *LSRS Workshop at ACM RecSys 2015*.
- [5] P. Cremonesi, F. Garzotto, R. Pagano, and M. Quadrana. Recommending without short head. In *ACM WWW 2014 Companion*, pages 245–246.
- [6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *ICLR*, 2016.
- [7] B. Hidasi and D. Tikk. Approximate modeling of continuous context in factorization algorithms. In *ACM Workshop on CARR 2014*.
- [8] B. Hidasi and D. Tikk. General factorization framework for context-aware recommendations. *Data Min. Knowl. Disc.*, 30(2):342–371, 2015.
- [9] A. Q. Macedo, L. B. Marinho, and R. L. Santos. Context-aware event recommendation in event-based social networks. In *ACM RecSys 2015*, pages 123–130.
- [10] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *ACM RecSys 2009*, pages 265–268.
- [11] E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.
- [12] I. Pilászy, A. Serény, G. Dózsa, B. Hidasi, A. Sári, and J. Gub. Neighbor methods vs matrix factorization – case studies of real-life recommendations. In *LSRS Workshop at ACM RecSys 2015*.
- [13] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *ACM SIGIR 2011*, pages 635–644.
- [14] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *ACM WWW 2010*, pages 881–890.
- [15] A. Sen and M. Larson. From sensors to songs: A learning-free novel music recommendation system using contextual sensor data. In *LocalRec Workshop at ACM RecSys 2015*.
- [16] R. Turrin, A. Condorelli, P. Cremonesi, and R. Pagano. Time-based TV programs prediction. In *RecSysTV Workshop at ACM RecSys 2014*.
- [17] H. Werthner and F. Ricci. E-commerce and tourism. *Commun. ACM*, Dec. 2004.

¹<http://www.citeulike.org/groupfunc/20206/home>