Crowd-powered recommendation for continuous digital media access and exchange in social networks

FP7-610594

D3.1

Scalable stream recommendation algorithms

Dissemination level:	Public
Contractual date of delivery:	Month 6, 31 March 2014
Actual date of delivery:	Month 10, 21 July 2014
Workpackage:	WP3. Stream Recommendation algorithms
Task:	T3.1 - User Context Stream Analysis T3.2 - Social and Community Stream Analysis T3.3 - User Interaction Stream Analysis T3.4 - Real-time Stream Recommendation
Type:	Report
Approval Status:	Final
Version:	1.0
Number of pages:	19
Filename:	CrowdRec_WP3_D3.1_MOVIRI_21072014_V1.0

Abstract

This deliverable reports the results achieved in WP3 "Stream Recommendation algorithms" at the end of the first iteration (out of three). The algorithms integrate real-time feedback from different sources of data: user interactions (e.g., user feedback), user context (e.g., time, location), and social data (e.g., social tags). A number of mature algorithms have already been disseminated and a selection of them will be integrated into the reference framework (WP2). The algorithms build on state-of-the-art methods for collaborative filtering, in particular, Factorization Machines. During the next iteration the algorithms will be extended and new algorithms developed, informed by the requirements for real-world recommender systems established in WP2.

co-funded by the European Union

History

Version	Date	Reason	Revised by
Ver 0.1	04-03-2014	Initiation	Martha Larson
Ver 0.2	18-04-2014	Basic descriptions of algorithms	Paolo Cremonesi
Ver 03	23-06-2014	Additional algorithms/information	Roberto Turrin
Ver 1.0	21-07-2014	Finalization	Martha Larson

Author list

Organization	Name	Contact Information
MOV	Roberto Turrin	roberto.turrin@moviri.com
TUD	Martha Larson	m.a.larson@tudelft.nl
GRA	Hidasi Balazs	
MOV	Paolo Cremonesi	
TUB	Frank Hopfgartner	
TID	Alexandros Karatzoglou	
TUB	Andreas Lommatzsch	
TUD	Babak Loni	
MOV	Davide Malagoli	
GRA	Domonkos Tikk	
TUD	Alan Said	
TUD	Yue Shi	

Executive Summary

This deliverable describes the results achieved in **WP3 “Stream Recommendation algorithms**” during its **first iteration** (M1-M6), of a total of three iterations. The overall goal of WP3 is to support **first objective** of the CrowdRec project: developing algorithms able to provide **stream recommendations**, i.e., able to integrate real-time feedback from user interactions and user context (e.g., time, location) and to adapt when new media or other information is available within the social community. A selection of the algorithms developed in this WP will be implemented and tested in the reference framework (WP2).

The algorithms developed in WP3 are classified based on the specific challenge they address: contextual information, social and heterogeneous content, user interactions, real-time requirements. Within CrowdRec these challenges are summarized with the acronym RICHeS: Real-time, Interaction, Context-aware, Heterogenous, and Social. The following table summarizes the algorithms presented in this deliverable, and lists the particular challenge or challenge that each algorithm addresses.

Algorithm	**Challenges**
Continuous context in the matrix factorization framework.	Context
Context Aware Recommendation by Learning Context Representation.	Context
Context-aware Gaussian Process Factorization Machines	Context
Exploiting collaborative expertise in communities for question recommendation	Social
Cross-Domain Collaborative Filtering with Factorization Machines	Heterogeneous
Tag-induced cross-domain collaborative filtering	Social Heterogeneous
Semantic analysis on social data	Social Heterogeneous
“Free Lunch” Enhancement for Collaborative Filtering with Factorization Machines	Interaction
Real-Time Recommendations using Context-aware Ensembles	Context Real-time
Real time recommendations using time series forecasting	Context Real-time Stream

The first iteration has laid the foundation for future iterations by establishing the basic breeds of algorithms and strategies that will be used during the project. Specifically, Factorization Machines have emerged as a favorite choice, with strong future potential for joint-work between consortium partners. Also, ensembles, graph-based approaches, and learning-to-rank approaches are important. The first iteration of WP3, took place in parallel with the first iteration of WP2 “Requirements and Reference Framework”. Communication between consortium partners ensured that the two were well aligned. However, the challenge of the next iteration of WP3 is to use the requirements specified in D2.1 “First Iteration Requirements Plus Evaluation Specifications” to guide the further development of stream recommendation algorithms. The next iteration will also see the implementation of selected WP3 algorithms into the Reference Framework, where they are publically released and can be evaluated against new developments in the state of the art.

Table of Contents

1 1. Introduction

This deliverable is the first main result of work package **WP3 “Stream Recommendation algorithms**”. WP3 is strictly related to the **first objective** of the project: developing algorithms able to provide **stream recommendations**, i.e., able to provide a continuous feed of information tightly fitted to user needs. Stream algorithms must be able to integrate real-time feedback from user interactions and user context (e.g., time, location) and to adapt when new media or other information become available within the social community. Stream algorithms should be able to combine in **real-time** four different streams of input data:

- **Interaction** with the user and relevance feedback,
- **Context** information about users, items, and interactions,
- **Heterogeneous** digital media from multiple sources.
- **Social** information from social networks and user communities.
-

The ability to integrate the four streams, together with the real time requirements, constitute **five RICHeS challenges** (**R**eal-time, **I**nteraction, **C**ontext, **H**eterogeneous, **S**ocial) of stream algorithms.

In the CrowdRec DoW the challenges of Stream Recommendation is represented with the following figure. These challenges have been identified as the key challenges that recommender systems must face in order to push forward beyond the state of the art.

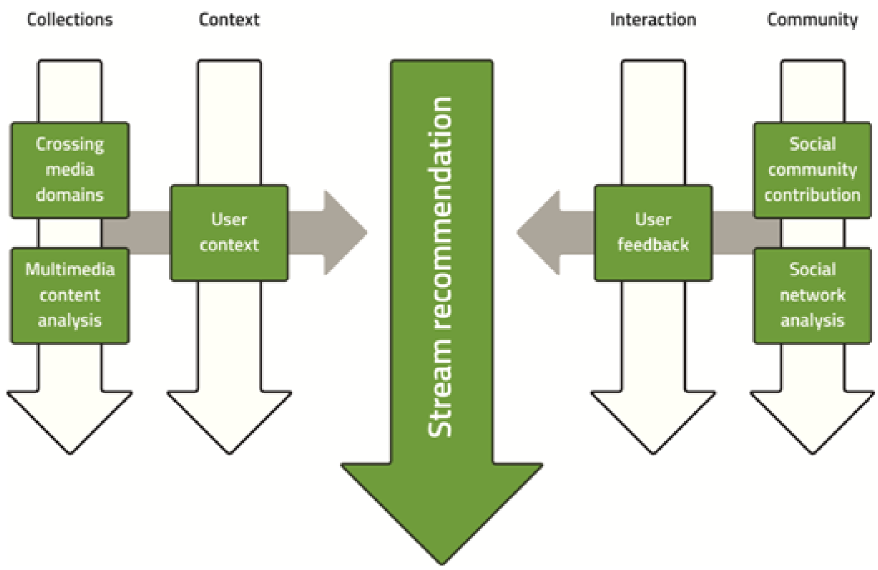

Figure 1. Stream recommendations (from the DoW)

In this figure, the challenge of Heterogeneous content (here, called “Collections”) is broken down into the challenge of exploiting information drawn from two or more different domains (i.e., cross-domain recommendation), and the challenge of multimedia content. Multimedia content analysis, which generates labels that can be exploited in the same way as user-contributed tags, is specifically addressed in WP4, rather than in this deliverable.

2 2. Structure and Strategy

In this section, we discuss how WP3 “Stream Recommendation Algorithms” is structured, as well as the overall strategy it uses to plan and generate results. This includes how WP3 is related to the other workpackages in the CrowdRec project.

The work package is organized in four tasks, as shown in the excerpt of the CrowdRec Gantt chart displayed in Figure 3.

	Year 1												Year 2											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Milestones												MS1						MS2						MS3
WP 3	Stream Recommendation algorithms																							
Task 3.1	User Context stream analysis																							
Task 3.2	Social and Community stream analysis																							
Task 3.3	User interaction stream analysis																							
Task 3.4							Real-time stream recommendation																	
Deliverables						▲												▲						▲

Figure 2: Timing of workpackage WP3

Each tasks is focused on the development of algorithms related to the five RICHeS challenges. The following mapping from tasks to challenges is useful to understand the structure of the project.

- Task 3.1: **Context**: Contextual information
- Task 3.2: **Social** and **Heterogeneous**: Social information and combination of multiple resources across domains
- Task 3.3: **Interactions**: Patterns of interactions of users.
- Task 3.4: **Real-time** requirements

In general, Tasks 3.1-3.3 are focused on functional requirements (quality of recommendations) while Task 3.4 is focused on non-functional requirements (performance). Since real-time requirements partially depend on the algorithmic choices, Task 3.4 starts only in Month 7. However, this deliverable already reports initial results working in the area of real-time recommendation.

All work packages in the project are organized according to three main iteration steps.
In turn, each iteration is composed of four phases (requirements, development, testing, and feedback) and constitutes one milestone in the project. **Requirements** are elicited from the business partners (the platform providers and also the recommender system vendors). On

the basis of these requirements, algorithms are **developed**. A selection of these is implemented, and **tested** in the Reference Framework. Finally a further selection of the best algorithms is made and deployed for live testing and **feedback** collection. This specific deliverable is related to the **first iteration**. This same work package will produce two more deliverables corresponding to the end of the second and third iterations.

The following table summarizes the status of the algorithms covered in this deliverable. The challenge addressed by each algorithm is stated.

Table 1 Stream algorithms in the First Iteration of CrowdRec

Algorithm	Challenges	Status	Partners
Continuous context in the matrix factorization framework.	Context	*First publication (disseminated)*	GRA
Context Aware Recommendation by Learning Context Representation.	Context	*Consolidated results (ongoing)*	TUD/TID
Context-aware Gaussian Process Factorization Machines	Context	*First publication (disseminated)*	TID
Exploiting collaborative expertise in communities for question recommendation	Social	*Consolidated results (ongoing)*	TID
Cross-Domain Collaborative Filtering with Factorization Machines	Heterogeneous	*First publication (disseminated)*	TUD
Tag-induced cross-domain collaborative filtering	Social Heterogeneous	*Consolidated results (ongoing)*	TUD
Semantic analysis on social data	Social Heterogeneous	Seminal idea	MOV
“Free Lunch” Enhancement for Collaborative Filtering with Factorization Machines	Interaction	Preliminary results (ongoing)	TUD
Real-Time Recommendations using Context-aware Ensembles	Context Real-time	*First publication (disseminated)*	TUB
Real time recommendations using time series forecasting	Context Real-time Stream	Preliminary results (ongoing)	MOV

Each algorithm is labeled with its current status, according to its maturity level as expressed by the following scale:

1. **Seminal idea**: there is a first design of the algorithm, not yet mature to produce results.
2. **Preliminary results**: there is a draft report not yet ready for submission.
3. **Consolidated results**: a paper has been submitted or is ready for submission.
4. **First publication**: a paper has been accepted for publication, results are disseminated.

In the rest of this deliverable, we cover each of the challenges in turn. We discuss each of the algorithms in the table in more detail. The final section of the deliverable is a conclusion and an outlook to the work in the next iterations.

3 Algorithms

3.1 User Context Stream Analysis (Task 3.1)

This task inspects the use of contextual information that affects the needs of users. According to [6], context can be described as “any information or conditions that can influence the perception of the usefulness of an item for a user” [6]. In particular, context can be classified as (i) physical context (e.g., time, location, weather, temperature, user activity,...), (ii) social context (e.g., latest trend, presence of other users,...), (iii) interaction-media context (e.g., device), and (iv) modal context (e.g., mood, user intent,...).

In this task, we go beyond the dualistic concept of modeling recommender systems exclusively in terms of users and items modeling concept. Instead, we consider additional contextual information that may influence user preferences for recommendations. The integration of a continuous flow of contextual information into the model improves the capacity of a recommender system to rapidly adapt its predictions to the ever-changing social network inputs. Recent contributions to Context-Aware Recommender Systems (CARS) have built upon the foundation of latent factor models, which have been shown to be among the most effective approaches for conventional collaborative filtering recommender systems. In a typical latent factor model, the relevance of a user/item/context combination is modeled by a three-mode inner product of the latent factors of the user, the item and the context. However, three main issues remain unsolved with this approach:

First, traditional CARSs assume that the context dimension is categorical. As traditional factorization methods are not able to cope with continuous dimensions. However, continuous context dimensions are frequent in contextual modeling. For example time-based information, such as seasonality, is one of the most dominantly used context dimensions. Second, the latent space of the context can hardly be consistent with that of the user and item. For example, if 10 latent factors are used to represent a user and an item, then the latent factors of the item can be interpreted as 10 latent properties possessed by the item and those of the user can be interpreted as the interest of the user to these 10 properties. However, such an interpretation may not be adequate for the latent factors of a context. Third, it is unclear how to interpret the relevance between a user and a context (or an item and a context). For example, it is not intuitive that a user is relevant to “Friday” rather than “Monday". Similarly, it is also not intuitive if a movie is better watched in ”Rome" rather than in “Venice".

3.1.1 Continuous context in the matrix factorization framework

***Status**: First publication (disseminated)* [1]

In order to cope with continuous context, the work proposes two approximate modeling approaches that can model the continuous context and can be embedded into standard factorization frameworks. The method is fully compatible with most factorization algorithms in the sense that those can be easily adapted to cope with continuous dimensions using the proposed modeling approaches.

Two problems are identified with the current modeling of continuous dimensions:

1. context-state rigidness (i.e., the boundaries of the context-states are rigid), and
2. context-state ordering (i.e., the context-states are independent from each other, but there should be a gradual change with the change of the context of the continuous dimension).

Two approaches have been proposed: (i) fuzzy event modeling and (ii) fuzzy context modeling. *Fuzzy event modeling* addresses the first problem (i.e., rigidness) by considering the uncertainty of the context-state boundaries and by duplicating events that are close to the boundaries.

Fuzzy context modeling addresses both issues (i.e., rigidness and ordering) by allowing context-states to overlap, and replacing the context-state feature vector by a combination of feature vectors of two neighboring context-states. This forces the methods to learn context-state feature vectors simultaneously and thus making neighboring context-states similar.
The described techniques have been incorporated into the state-of-the-art context-aware algorithm iTALS [7]. Experiments were executed on five implicit feedback based datasets (three public, two proprietary) with results indicating that (i) *fuzzy event modeling* improves recommendation accuracy (measured in the terms of recall@20) (usually) by a small amount, and (ii) *fuzzy context modeling* improves recommendation accuracy significantly.

3.1.2 Context Aware Recommendation by Learning Context Representation

***Status**: consolidated results (ongoing)* [5]

This work addresses two specific issues of context modeling: the consistency between latent features and interpretation of relevance between a user and a context (or an item and a context).

The approach consists of on a novel formulation of latent factor models based on context-aware representations of users and items. In this model, the context is defined in its own latent space, which can be independent of that of the user and item. In addition, instead of modeling the relevance (which is unreasonable as explained earlier at the beginning of Task-3.1 section) between the user/item and context, the proposed approach includes an additional layer of latent space for the user given a context, and an additional layer of latent space for the item given a context. The latent factors representing the user/item in the new latent space are introduced as the context-aware representation of the user/item.

Figure 3 illustrates the proposed context-aware model. The cubes represent the tensors that project the user (or item) latent features to another latent space that reflects the interaction between a user u_i (or item v_j) and context k. The tensors encode the learnt relationships between the two latent layers, i.e., the user and item latent space and the context latent space.

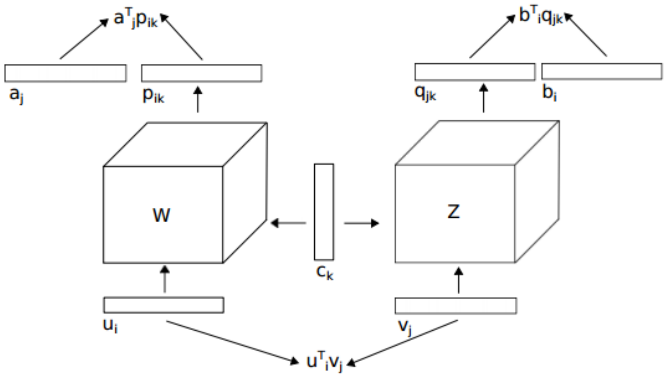

Figure 3: Illustration of the context-aware model in the "Learning Context Representation" algorithm

The context-aware representations can be interpreted much more clearly than the relevance between the user/item and the context. The context-aware user representation can be interpreted as the hidden properties (e.g., mood) that may influence the user under the context (e.g., Friday). Similarly, the context-aware item representation can be interpreted as the hidden properties (e.g., suitability for families) of the item under the context (e.g., Sunday afternoon).

Based on the proposed context-aware representations, ongoing activities are meant to develop new latent factor models for both the rating prediction problem in the case of explicit feedback data, and for the top-N recommendation problem in the case of implicit feedback data.

3.1.3 Gaussian Process Factorization Machines for Context-aware Recommendations

***Status**: disseminated results (completed)* [8]

Context-aware recommendation (CAR) can lead to significant improvements in the relevance of the recommended items by modeling the nuanced ways in which context influences preferences. The dominant approach in context-aware recommendation has been the multidimensional latent factors approach in which users, items, and context variables are represented as latent features in a low-dimensional space. An interaction between a user, item, and a context variable is typically modeled as some linear combination of their latent features. However, given the many possible types of interactions between user, items and contextual variables, it may seem unrealistic to restrict the interactions among them to linearity.

To address this limitation, we develop a novel and powerful non-linear probabilistic algorithm for context-aware recommendation using Gaussian processes. The method that we call Gaussian Process Factorization Machines (GPFM) is applicable to both the explicit feedback setting (e.g., numerical ratings as in the Netflix dataset) and the implicit feedback setting (i.e., purchases, clicks). We derive stochastic gradient descent optimization to allow scalability of the model. We test GPFM on five different benchmark contextual datasets. Experimental results demonstrate that GPFM outperforms state-of-the-art context-aware recommendation methods.

3.2 Social and Community Stream Analysis (Task 3.2)

Here, we explore the adoption of social information mined from the communities of users. Such information includes simple data such as users and comments, but also complex data such as textual descriptions (e.g., comments). This kind of social information can be used to enrich both the items (e.g., creating new tags and attributes) and the relationship network (e.g., inferring connections among users, links among items, or user-to-item ratings). Social information also includes the structure of the social network itself, such as friendship information.

Interesting is the exploitation of interconnections among multiple domains - referred to as cross-domain recommender systems (CDRS)—where the different individual communities are used together. While two individual domains may be sparse and it may be difficult to infer recommendations within each single domain, missing user preferences could be inferred based on tags that are used in both domains. Social tags used with CDRS also have the potential to enrich the interaction between the users and the recommender system. For instance, CDRSs could help better explain the recommendations to the users in the target domain, and especially to new users who have no previous preference data in the system. This potential stems from the additional information transferred from the auxiliary domain to generate recommendation in the target domain. This additional information helps new users to get familiar with the target domain and improves the experience of interacting with the content there. From the business perspective, CDRSs can be useful since cross-domain makes it possible to “cross-sell”, in other words, motivate users to buy or use items in domains in which they have not yet been active.

3.2.1 Exploiting collaborative expertise in communities for question recommendation

Status*: consolidated results (ongoing)*

Collaborative question answering (CQA) communities rely on user participation for their success. This paper presents a supervised Bayesian approach to model expertise in on-line

CQA communities with application to question recommendation, aimed at reducing waiting times for responses and avoiding question starvation. We propose a novel algorithm called RankSLDA that extends the supervised Latent Dirichlet Allocation (sLDA) model by considering a learning-to-rank paradigm. This allows us to exploit the inherent collaborative effects that are present in CQA communities, where users tend to answer questions in their topics of expertise. Users can thus be modeled on the basis of the topics in which they demonstrate expertise. In the supervised stage of the method, we model the pairwise order of expertise of users on a given question. We compare RankSLDA against several alternative methods on data from the Cross Validate community, part of the Stack Exchange CQA net- work. RankSLDA outperforms all alternative methods by a significant margin.

3.2.2 Cross-Domain Collaborative Filtering with Factorization Machines

Status: *First publication (disseminated)* [2]

In order to explore the potential of social tags in cross-domain recommender systems, we have developed a first family of cross-domain algorithms using factorization machines (FM), FMs are more flexible than the tensor representation adopted in classical cross-domain recommender systems. Moreover, FMs are polynomial in the number of domains, making them computationally less expensive than tensor factorization models. FMs allow us to encode any auxiliary knowledge in the form of real-valued feature vectors. In this algorithm, we exploit the advantage of FMs to extend the basic FM representations with additional features from auxiliary domains. This algorithm is suitable for the so-called *joint* domains, that is, the domains in which the users are the same or can be mapped one-to-one. The algorithm represents each user-item interaction in the target domain as a binary feature vector. It then matches the users in target domain to the users in auxiliary domains and extends the feature vectors by additional real-valued features that are inferred based on the ratings that the matched users gave in auxiliary domains. Experimental results show that with the extended feature vectors better models can be learnt by FMs and better recommendations can be generated.

3.2.3 Tag-induced cross-domain collaborative filtering

Status: *consolidated results (ongoing)*

We are also developing a Generalized Tag-induced Cross-Domain Collaborative Filtering (GTagCDCF) approach that builds on the concept of collective matrix factorization (CMF). We choose CMF as the basis of this new approach because it has proven to be one of the most effective ways of building recommender systems and it can elegantly and effectively be expanded to a multi-domain case.

The GTagCDCF approach extends the CMF method by simultaneously factorizing the user-item rating matrices from K different domains. In this model, a set of latent features are learnt for social tags. These features act as a bridge that connects K domains. In fact the latent features of users in one domain influence those of other domains via the common tags. The latent features of users and items are learnt is such a way that the loss of information from the simultaneous factorization of users, items and tags is minimized. As a result, the unknown latent features of users and items are learnt more accurately and better recommendations can be generated.

3.2.4 Semantic analysis on social data

Status: *seminal (ongoing)*

The algorithm starts from a social data source, in particular a graph dataset with social data. Example of possible input data are: (i) social network graph with user-to-user relations, (ii) user comments about items (e.g., the comments of users about songs or part of songs). Processing of such data occurs by means of NLP and semantic tools. The output of the algorithm is a set of tagging to support (i) content enrichment, (ii) adding new discovered links (e.g., new user-to-user connections).

3.3 User Interaction Stream Analysis (Task 3.3)

In many application domains the user profile required to generate recommendations is based on the user's current interaction with the system (implicit elicitation). This is mainly motivated to support users who have no rating history (the cold-start problem) and to improve the quality of recommendation models when data are very sparse (the sparsity problem). Interactions can be used to estimate a numeric rating and apply canonical recommendation techniques. For instance, [3] addresses the problem by using users' interactions with multiple components of online applications (e.g., tags, links, maps, pictures). Whenever a user interacts with an object on the GUI (graphical user interface), a signal is assigned to that object. Signals are used to estimate the user ratings for the current browsing session.

3.3.1 "Free Lunch" Enhancement for Collaborative Filtering with Factorization Machines

Status*: preliminary results (ongoing)*

The 'Free Lunch' enhancement algorithm exploits information inherent in the user-item matrix that reflects the patterns of interaction between users and items. Rating patterns are discovered by clustering users (or items) with same rating behavior into a same user (or item) cluster. The users and items are represented by histograms encoding their rating patterns. For each pair of user and item, the 'Free Lunch' algorithm encodes the user and item cluster as an additional feature into the FM training model. The algorithm then trains a model based on the extended feature vectors. This algorithm is also able to exploit rating pattern from auxiliary collaborative filtering domains.

3.4 Real-time Stream Recommendation (Task 3.4)

In many emerging application domains, the main challenge of a recommender system is the need to process a continuous flow of information that affect users, items, and context. New items are continuously ingested into the system (e.g., photos, news, TV programs) or existing items are updated (e.g., with tags). Users continuously interact with the system (e.g., zapping TV channels, watching online videos, chatting on social networks, tagging photos) or change their context (e.g., they are alone or with friends, they use a TV, a smartphone or a tablet, or again, they move from their office to home).

Recommender algorithms must be able to process all the newly arrived information in real-time in order to be able to provide timely and up-to-date recommendations. The risk of failing in doing so may lead to unrealistic or poor recommendations. All of the algorithms developed in the previous tasks have this requirement in mind and are designed to be easily scalable.
Real-time recommendation has two facets: updating capabilities and response time constraints.

First, real-time refers to the ability of a recommender to update its recommendation model with new data, and to update the recommendation results for its users in such a short time, that users perceive recommendation to be “live”. Several approaches can be used in real-world environments, such as the implementation of highly scalable/parallel algorithms, the use of algorithms able to be incrementally trained (instead of being updated by a full re-training with all data) a suitable data filtering (e.g., to lighten the amount of data to process), and optimize timing (e.g., learning during off-peak time).

Second, real-time refers to the ability of a recommender to grant certain response time (e.g., in accordance to a Service Level Agreement), independently of the amount of data. In fact, the generation of recommendations typically requires that the system reply to users within a predefined response time in order to make the user perceive a responsive system. Contrarily, slow response time can affect the perceived quality.

In this task we focus our effort on the design of algorithms that are specifically tuned for the real-time recommendations of short-lived items (e.g., news, events, TV programs).

3.4.1 Real-Time Recommendations using Context-aware Ensembles

Status*: First paper (disseminated)* [4]

The performance of recommender algorithms often strongly depends on the particular context. For example, users often expect different types of recommendations dependent on the time of the day or the day of the week. In addition, recommender algorithms may show a different scaling behavior. This means the recommender algorithm's performance depends on scenario specific properties such as the number of users or the number of requested recommendation per minute if the computational resources are limited.

In order to address the context-dependent requirements we developed a recommender ensemble that integrates different recommender algorithms. A management component continuously evaluates the performance of the different recommender algorithms in the ensemble. Based on the measured performance of each recommender algorithm and the properties of the context (e.g., time or current system load) the management component learns a delegation strategy (e.g., on a decision tree or a naive Bayes classifier) that predicts the best-suited recommender algorithm for a new request. The developed framework is open to new recommender algorithms allowing us to test new parameter configurations as well as new algorithms.

3.4.2 Real-Time Recommendations using Time series Forecasting

***Status**: Preliminary results (ongoing)*

In many application domains (e.g., TV broadcasting, Twitter) the arrival rate of new input data does not allow for the real time updating of a global predictive model on the whole dataset. In order to overcame this problem we aim at designing predictive recommender algorithms leveraging ideas from time series analysis and forecasting. Time series forecasting allow the model to take advantage of the natural temporal ordering of observations. This is beneficial in many stream-based recommender systems, where time is a primary contextual attribute. Different from traditional time-based contextual recommender systems, time series forecasting uses time and ordering of events as the main model parameters, while other information (e.g., users and item attributes) is considered as second-order effects.

4 Conclusion and outlook

This deliverable has presented the stream recommendation algorithms developed during the first iteration of CrowdRec WP3 “Stream Recommendation Algorithms”. The algorithms address the RICHeS challenges (i.e., Real-time, Interaction, Context, Heterogenous, and Social). We briefly summarize the results that have been produced.

Context (T3.1) has been modeled in collaborative filtering by extending state-of-the-art techniques based on matrix factorization, latent factor models, and factorization machines. While canonical solutions consider context as a discrete dimension, a fuzzy approach has been proposed to model continuous context, such as time.

Social (T3.2) have been explored in the settings of cross-domain recommender systems, where **heterogeneous** information, in the form of additional knowledge coming from the external domain has been used to enhance the recommendation quality of matrix factorization approaches. Furthermore, social data of Q&A (Question and Answer) communities have been analyzed to exploit the participation patterns of users.

Factorization machines have been also used to process the **interaction** of users (T3.3), clustering users and items according to their rating patterns.

Finally, **real-time** feedback (T3.4) has been included in an ensemble algorithm that predicts the best strategy to apply on the basis of the current, real-time context. In addition, user real-time data (e.g., TV viewing activity) have been used to forecast the user consumption patterns.

Some algorithms are still at a seminal status (i.e., there is a first design of the proposed solution), others are at a more mature stage (i.e., there is a paper ready to be submitted), and a number of activities have been already disseminated (i.e., a first paper is accepted for publication).

The first iteration of WP3 has contributed to the overall project, by laying the groundwork for the types of algorithms that will be pursued in the project. Specifically, we see convergence of consortium members on the choice of Factorization Machines. Factorization Machines are more flexible than tensor representations. They are particularly suited to the type of algorithms that we are developing in CrowdRec because they allow for easy incorporation of additional knowledge sources. In the future, the work of partners on Factorization Machine approaches will become more tightly integrated. Other types of algorithms that have potential to be very productive in the project include ensembles, graph-based approaches, and learning-to-rank approaches.

The first iteration of WP2 “Requirements and Reference Framework” is now complete. Communication between consortium partners kept WP3 well aligned with both the process of eliciting requirements from business partners and establishing the Reference Framework architecture. However, moving forward the integration will be even tighter. Specifically, the requirements specified in D2.1 “First Iteration Requirements Plus Evaluation Specifications” will directly inform and guide the further development of stream recommendation algorithms.

We will then select the most promising of the stream recommendation algorithms to be implemented in the Reference Framework. We determine “most promising” by evaluating the algorithm within the 3D benchmarking paradigm, which takes requirements along three dimensions, user requirements, technical constraints and business requirements (cf. D2.1). Algorithms integrated in to the Reference Framework are publically released, and they can be benchmarked against the full range of existing alternative approaches, as well as compared with new developments in the state of the art.

5 References

[1] Balázs Hidasi and Domonkos Tikk, "*Approximate modeling of continuous context in factorization algorithms",* 4th Workshop on Context-awareness in Retrieval and Recommendation (ECIR 2014), Amsterdam.

[2] Babak Loni, Yue Shi, Martha Larson and Alan Hanjalic, "Cross-Domain Collaborative Filtering with Factorization Machines", ECIR 2014, Amsterdam.

[3] Paolo Cremonesi. Franca Garzotto. Roberto Pagano. Massimo Quadrana, "Recommending without Short Head", In proceedings of WWW 2014.

[4] Andreas Lommatzsch, Sahin Albayrak, "Real-Time News Recommendations using Context-aware Ensembles", ECIR 2014, Amsterdam

[5] Yue Shi, Alex Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic. "CARS2: Learning Context-aware Representations for Context-aware Recommendations", Under review.

[6] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. 2010. "A survey of context modelling and reasoning techniques". Pervasive Mob. Comput. 6, 2 (April 2010), 161-180.

[7] Balázs Hidasi and Domonkos Tikk. 2012. "Fast ALS-Based tensor factorization for context-aware recommendation from implicit feedback". In Proceedings of the 2012 European conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II (ECML PKDD'12), Peter A. Flach, Tijl Bie, and Nello Cristianini (Eds.), Vol. Part II. Springer-Verlag, Berlin, Heidelberg, 67-82.

[8] Trung V. Nguyen, Alexandros Karatzoglou, Linas Baltrunas "Gaussian Process Factorization Machines for Context-aware Recommendations" ACM SIGIR 2014, Gold Coast, Australia.