NETWORKED COLLABORATIVE RECOMMENDATION ARCHITECTURE

V. Ollikainen

VTT Technical Research Centre of Finland, Finland

ABSTRACT

For any individual user, the amount of available content is exploding. Recommendations have become an integral part of digital business, helping people to find content and services, while at the same time enabling carefully targeted advertising.

A major challenge in recommendation systems is that they are either domain specific or need a substantial amount of data. This favours global data-driven platforms, available from few organisations, notalbly the GAFA group (Google, Amazon, Facebook and Apple).

The technology presented in this paper enables a recommendation engine *network*, in which all parties own and are able to administer their own data, challenging centralized models of today. The approach is based on exchangeable anonymous tokens. This enables a de-centralized recommendation architecture in which different recommendation engines can be located at the edges of networks and linked together, while respecting the ownership of data.

This paper introduces architectural models for the technology and a conceptual view of an ecosystem based on them.

INTRODUCTION

In general, recommendations are used to estimate a user's response to new items based on historical information stored in the system, and suggesting novel and original items for which the predicted response for that user is high, as defined by Desrosiers and Karypis (1). These items can be pieces of content, services or goods. Therefore, advertising is a self-evident application area for recommendations.

Recommenders are commonly classified into two basic categories: content-based and collaborative. Content-based recommenders are based on *representing the items with a set of attributes*, and using these attributes to find the most relevant content for a particular user. As an example, Agatha Christie is known to write detective stories. If a user has been reading her novels, other detective novels are recommended for him.

Collaborative recommendations, on the other hand, learn from the behaviour of users as a whole, *without any need to define properties of individual items*. For instance, if users A and B have had similar behaviour in the past, and A has found item X preferable, this item is likely to be recommended for B as well. Being solely behavioural, collaborative recommendations can easily span different domains, unlike content-based recommendations that are limited to each domain with mandatory domain-specific knowledge (such as genres of individual novelists in the example).

However, when it comes to privacy, there are challenges in traditional collaborative recommenders: Since the recommendations are based on historical behaviour of large user groups, their history has to be recorded. Several studies in the past have been addressing this problem, such as Canny (2) who introduced ‘talliers’, which compute public aggregates on behalf of communities of users. This approach requires individuals to trust these talliers, who are acting as intermediaries. In another approach, Yakut and Polat (3) addressed a case in which multiple vendors (typically companies) share at least partially the same user pool, and the vendors are responsible for sharing no personal information about their customers. In this approach, users and items are arbitrarily interleaved into different partitions, and no vendor learns anything about the individual behaviour and items held by another vendor. While the method can be considered privacy-protecting, also from vendor perspective, the implementation is centralized and all parties must trust whoever operates it.

A different approach has been taken by Ollikainen et al (4), introducing a de-centralized collaborative recommendation technology that is primarily designed to protect end users’ privacy. Unlike in any other collaborative recommendation method, in this approach user data gets aggregated as a collection of random values, ‘tokens’, which under certain conditions can be exchanged without exposing users' identities or their preferences.

While the method makes fundamentally no difference, whether user or item tokens are processed, it protects item-related and user-related data equally well. This enables sharing business-related data, making co-operation between competitors possible.

The technology has been in public use since 2014 in Helsinki Metropolitan area libraries. Available online, it has currently 600,000 patrons in its databases and it actively covers 300,000 book titles. This service is running on a single virtual server, implementing a centralized model, while the method itself is topology agnostic: it can equally enable distributed, even edge-computed architectures; models discussed later in this paper.

This paper is organized as follows: The following chapter presents the principle of the method and the basis for privacy, followed by a chapter presenting different architecture models. These models introduce how token collections and recommendation engines (‘recommenders’) may be arranged. The paper is summarized and ecosystems are discussed in the last chapter.

OPERATING PRINCIPLE OF THE TOKEN-BASED RECOMMENDER

Tokens and token collections

The method associates both users and items with *collections* of tokens, each token carrying a random value. Individual tokens are typically a 24-bit numbers (from zero to about 16 million) and, unlike cookies or identifiers, when alone they are *not* associated with anything.

Interactions and token exchange

Most collaborative recommenders are based on ratings a user has given to an item. The presented method, in turn, is based on user actions, such as accessing a piece of content, loaning a book or visiting a place. These are triggers for their token collections to interact, making the whole process an ‘interaction’:

When a user (on the right in Figure 1) interacts with any other entity (user or item; illustrated as a box on the left), some of tokens are copied between the user’s collection and the collection of the other entity. This process is hereinafter referred as ‘token exchange’.

Although only a few tokens at most are exchanged at a time, users with similar behaviour have a tendency to get the same tokens quickly. This phenomenon happens, because items they have accessed have collected these tokens and delivered to alike users.

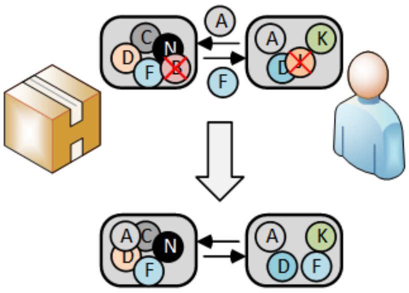

Figure 1 – In a user interaction, tokens are copied over bilaterally, making the collections resemble each other. Some old tokens are deleted.

Calculating recommendations for a given collection

These accumulating similarities are the basis for recommendations: Token collections are compared with a *given* collection, and the items with most similar collections are recommended.

When a user requests personal (user-item) recommendations, the given collection is his/her collection. In search of ‘similar items’ for an item Y (item-item recommendations), the given collection is Y’s collection with other items’ collections. Users can be searched as well (user-user or item-user recommendations), provided that users have made their collections available for external comparison.

Practical considerations

Since not all tokens are copied in an interaction, there must be an algorithm to select the particular tokens. While the collections typically are limited in size, typically 256 tokens, it also is necessary to have an algorithm for selecting equal number of old tokens to be deleted. Furthermore, comparing similarities of collections needs an algorithm. The most recent selection algorithms are based on hypercube clustering by Ollikainen (5).

These algorithms can also be implemented in low-end devices, regarding both complexity in computing and memory footprint. For instance, each 24-bit token fits well into 4 bytes of memory, and a token collection consisting of 256 tokens requires only one kilobyte.

Privacy

As explained, tokens are be copied over and over again. Consequently, numerous identical tokens are likely to exist in the recommendation system, with no information where they have originated or even where they have come from. This is the basis for privacy: disclosing any single token discloses nothing from the user.

Token collections are in constant change. Eventually, all tokens of a collection are likely to be changed, since some tokens must be deleted from a collection in order to make space for incoming tokens.

On top of this, Ollikainen and Niemi (6) present two quite different privacy scenarios: When an *anonymous user* discloses *more than one token*, either in token exchange or for recommendations, an evil-doing (cf. GDPR) vendor may detect returning customers with a certain confidence, if he memorizes token traffic. Therefore, it is advantageous for anonymity to provide different tokens each time.

However, for *registered users* the situation is the opposite: Disclosing different tokens might enable an evil-doing vendor to reconstruct part of users' token collections. Consequently, registered users should provide same tokens for a specific vendor.

Another guideline is, that the number of tokens exchanged should not depend on how many tokens the parties possess but should rather be kept constant [6]. Since the hypercube clustering in [5] minimizes token exchange, these requirements can be met, preserving privacy in the recommender system.

ARCHITECTURAL MODELS

Implementations of the presented token-based recommendations are versatile, varying in terms of

- where token collections are located and
- additional token exchange.

The models differ from each other in following respects:

1. who administers token collections for items
    - A. vendors who do not synchronize token collections of similar items
    - B. vendors who synchronize token collections of similar items
    - C. suppliers
2. who administers token collections of users
    - A. users who do not exchange tokens with each other
    - B. users who exchange tokens with each other
    - C. vendors (nothing is required on user side; easy to deploy)
3. where recommendation engines are located
    - A. on vendor side
    - B. on user side (provided that token collections are be administered by users)

While this classification leads to 15 possible permutations, the most viable models will be presented as examples, with some discussion.

It should be noted that while the token exchange remains unchanged, these *models are interoperable and may co-exist* to create a recommendation ecosystem. Each entity can choose the model that suits their operations best.

Centralized (closed) model (1A, 2C, 3A)

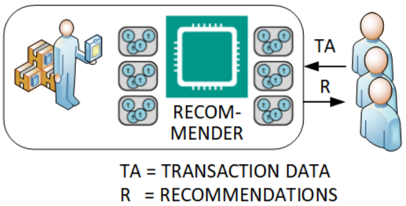

Figure 2 – Centralized model, having both user and item tokens in a server with the recommender.

The most straightforward topology consists of a single centralized server. This model is also used in the above-mentioned library recommender. Figure 2 illustrates users on the right, having their tokens in the same server with item tokens and the recommender. On the left there is a single vendor (such as the library) which defines the item set (book collection, respectively). Transactions, marked as 'TA', consisting of a user identifier and an item identifier (cf. book loan data; patron id and book id), are used as the input. A user gets recommendations 'R', either by using his/her token collection (user-item recommendations), or token collection of a selected item (item-item recommendations) as the given collection.

Centralized model with multiple vendors (1B, 2C, 3A)

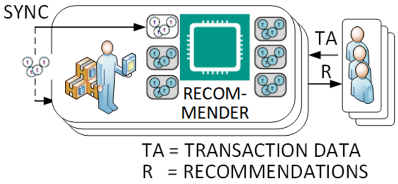

Figure 3 – Centralized model with multiple vendors, administering token collections of their users.

As a variant of the centralized model, there may be several vendors with independent user pools but at least partially same items (Figure 3). In this model, each vendor administers token collections of its users. For the same items, the respective token collections are synchronized.

This model has been especially considered for network recommenders of independent libraries: while they are not typically allowed to disclose anything related to their patrons, their recommendations would greatly benefit from each other's loan data. From the recommendation quality point of view, the networked recommender is equivalent to a single large recommender.

Independent vendors (1A, 2A, 3A)

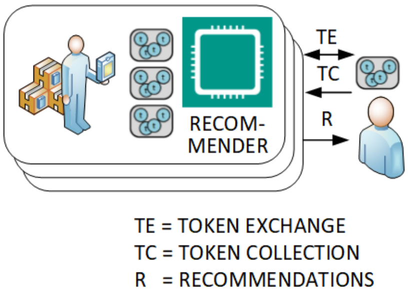

Figure 4 – Model with multiple independent vendors; users administer their token collections

Figure 4 illustrates an item-side recommendation model, in which users administer their token collections and are able to use them in different services. These services constitute an ecosystem, in which each vendor operates their recommendation engines independently, but gain advantage from a more holistic view of users.

A user exchanges his/her tokens (TE) with the recommendation engine of the vendor, with whom the transaction is taking place. While only a few tokens are disclosed in each transaction, for recommendations users have to disclose a substantial part of their token collection.

Since the tokens do not carry any history, no usage data is transferred from a vendor to another, protecting business-sensitive information. In a library domain, an example would be a user accessing movie and book recommenders with the same token collection for both.

Co-operating vendors (1B, 2A, 3A)

Multiple vendors may have the same items available, which would make it beneficial to use the same token collections for the same item in different token collections. In these cases, vendors can synchronize their related token collections, as illustrated in Figure 5.

An example of this co-operating vendor model would be libraries with same books in their collections. Since the recommenders operate independently, this model also enables mirrored-server design with fail-safe redundancy.

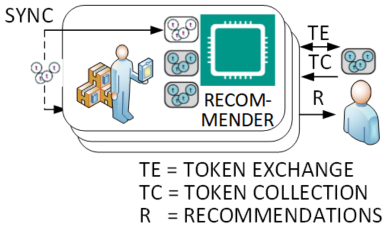

Figure 5 – Vendors co-operating in an ecosystem.

Supplier-supported vendors (1C, 2A, 3A)

As a specific case of the co-operating vendor model (Figure 6) these vendors also may use shared token collections, made available for their recommenders. A convenient location for the token collections would be in the logistics tier preceding the vendors. For instance, a content producer can maintain token collections related to their productions, at the disposal of media companies licensing the content. A technical challenge in this model relates to latency: item collections must be available and updateable in real time, requiring careful engineering.

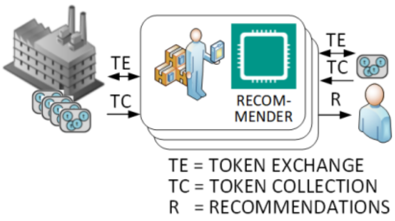

Figure 6 – Vendor tokens hosted by suppliers

Socially networked users (any, 2B, any)

Some users may prefer group recommendations or recommendations resulting in some similarities. This could be a case for instance for couples wishing to have overlapping movie recommendations. Figure 7 illustrates token exchange between two users, which makes their token collections more alike. This operation can be applied to practically any model, as long as users can administer their collections.

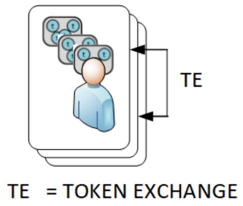

Figure 7 – Sharing tokens between networked users

Social media activity could trigger token exchange between its users. This option will be addressed in a peer-to-peer social media platform project HELIOS (7), which is adopting the technology as its background.

Edge-computed recommendations (1A, 2A, 3B)

In the previous models, vendors provide recommendations. However, the method enables users to calculate recommendations themselves, for a couple of reasons: Firstly, privacy benefits from disclosing as few tokens as possible. Secondly, it would distribute

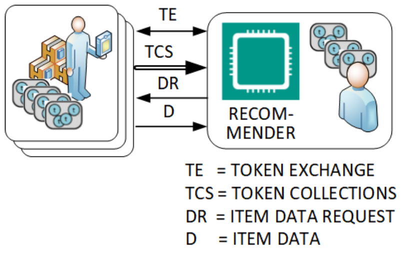

Figure 8 – Edge-computed recommendations are calculated on user side

recommendation engines to network edges, removing computational bottlenecks. User terminals today have substantial computing capacity, far beyond what is needed for calculating recommendations.

Figure 8 also illustrates a conceptual workflow: Token exchange (TE) is as in all other models. For recommendations, the process is different: First, the vendor sends token collections of the available items (TCS), without disclosing the actual items. Second, the user calculates similarities between received collections and his/her own token collection. Third, he/she requests the item data (DR) of most similar collections to be disclosed. Last, the vendor returns item data (D), to be presented to the user as recommendations.

From an engineering point of view, let us consider a vendor with a selection of 1000 items. If each token collection consists of one kilobyte, the corresponding token collections fit into one megabyte. Even if the user terminal is connected over a 3.5G mobile data link, say 5 Mbps, the transfer time would take few seconds, while calculating the actual recommendations would add only a fraction of second more. With an upcoming 5G, the total processing time will be negligible.

CONCLUSIONS AND DISCUSSION

Different models were presented and discussed in the previous chapter. The presented recommendation technology is capable of finding similarities between users and items, based on user behaviour only.

A major challenge in recommendation systems is that they are either domain specific or need a substantial amount of data. The more data available, the better and more tailor-made the services can be, as stated by the European Political Strategy Centre (8). Indeed, quality recommendations may be beneficial for all parties, especially when it comes to targeted advertising, since users easily find irrelevant ads irritating. The pursuit for performance leads to few data-driven platforms, such as the GAFA ecosystems (Google, Amazon, Facebook and Apple). From a technical perspective, they are operating gigantic recommendation engines.

In order to make their ecosystems efficient, these platforms ingest masses of both private and business related data into their platform hubs. However, performance comes at the price of losing control of data on both user and business side. Each platform owns the customer relationship of its ecosystem. Figure 9 illustrates a simplified GAFA model with a platform hub essentially disconnecting users from vendors.

In contrast and as an alternative to the current practise, the presented collaborative approach is based on a *network without a hub*. Figure 10 illustrates a conceptual

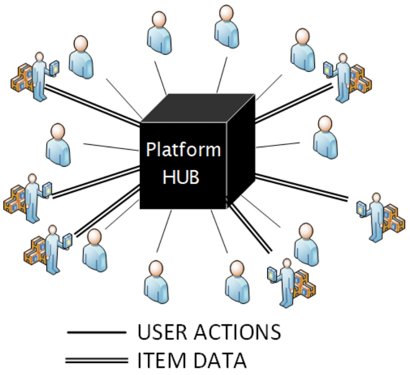

Figure 9 – Simplified GAFA ecosystem model

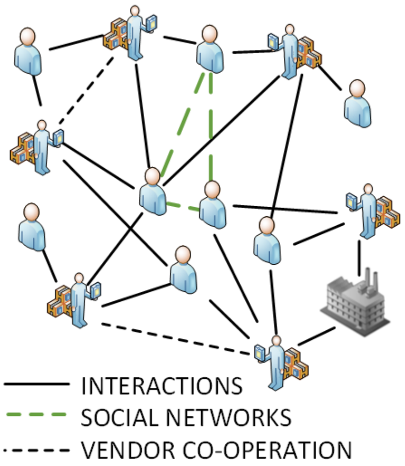

Figure 10 – Conceptual networked ecosystem architecture

ecosystem, integrating the discussed models into a single architectural view. The presented models are interoperable, giving each entity freedom to choose the model that suits their operations best.

The method enables edge-computed recommendation engines, to the extent of embedding them into user terminals. 5G transfer speed has been discussed. Related to other upcoming technologies, it would be worthwhile to study relations to the Multi-access Edge Computing (MEC) environment, summarized by Talib et al (9), especially personal clouds related to it.

REFERENCES

1. Desrosiers, C., Karypis, G. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor, Recommender Systems Handbook (pp. 107-144). Boston: Springer.

2. Canny, J. 2002. Collaborative Filtering with Privacy. Proceedings of the 2002 IEEE Symposium on Security and Privacy (p. 45). Washington, DC, USA: IEEE Computer Society.

3. Yakut, I., Polat, H. 2012. Arbitrarily distributed data-based recommendations with privacy. Data and Knowledge Engineering, 72, 239-256. doi:10.1016/j.datak.2011.11.002

4. Ollikainen, V., Mensonen, A., Tavakolifard, M. 2013. UPCV Distributed recommendation system based on token exchange. Journal of Print and Media Technology Research, Vol. 2, No. 3, pp. 195-201, 2013. ISSN 2414-6250 (Online) Available at http://www.vtt.fi/inf/julkaisut/muut/2013/OA_JPMTR_1314_Ollikainen.pdf

5. Ollikainen, V. 2018. Clustering Enhancement for a Token-Based Recommender. 6th International Workshop on News Recommendation and Analytics (INRA 2018), 22 October 2018, Turin, Italy. Publishing in progress.

6. Ollikainen, V., Niemi, V. 2016. Privacy Analysis of a Networked Collaborative Recommendation System. International Journal of Humanities and Management Sciences (IJHMS) Volume 4, Issue 4 (2016) ISSN 2320–4044 (Online). Accessed in May 2019 at https://helda.helsinki.fi//bitstream/handle/10138/232738/ED516221.pdf

7. HELIOS project, 2019-2021. A Context-aware Distributed Social Networking Framework. Funding from the European Union's Horizon 2020 research and innovation programme, grant agreement 825585. Homepage http://www.helios-social.eu

8. European Political Strategy Centre, 2017. Enter the Data Economy. EPSC Strategic Notes, Issue 21, 11 January 2017. Accessed in May 2019 at https://ec.europa.eu/epsc/sites/epsc/files/strategic_note_issue_21.pdf

9. Talib, T. et al., 2017. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657-1681. doi:10.1109/COMST.2017.2705720