Clustering Enhancement for a Token-Based Recommender

Ville Ollikainen
VTT Technical Research Centre of Finland
02150 Espoo, Finland
ville.ollikainen@vtt.fi

Abstract

This paper introduces a clustering enhancement to an established token-based collaborative recommendation method ("upcv"). The method creates privacy-protecting abstractions for users and items by exchanging and collecting randomly generated N-bit values, "tokens", in user-item transactions. The novel enhancement considers users' random value spaces as hyperspaces in which the tokens are N-dimensionally clustered. Instead of selecting exchanged tokens at random, as in the baseline upcv, tokens are now selected from a cluster, which has the best match with item's token collection. Recommendation quality is evaluated with the same 3.5% density data set as in a previous publication. The quantitative analysis indicates overall improvement in recommendation quality while learning time decreased without exception, up to one-third. There was improvement even when the number of exchanged tokes was exactly one, instead of over 100 in the baseline upcv. The performance improvement may be explained by the clustering enhancement inherently recognizing versatility of each individuals' interests. The paper also presents a study with news data set, where the improvement was in coverage.

1 Introduction

This paper is based on a collaborative token-based recommendation method[Oll13, Oll17], which creates privacy-protecting abstractions for users and items by exchanging and collecting randomly generated N-bit values, "tokens", in user-item transactions. This enables distributed recommendations in a multi-player environment, based on bilateral communications between a user and a service. As such, while also providing easy scalability, the approach relates to multi-domain collaborative filtering proposed by [Zha12] and cross-domain recommendations proposed by [Gao13], overcoming sparsity problems that are often experienced in single domain collaborative recommenders. Privacy properties of the method have been presented in [Oll16].

The upcv recommender has been in public use in Helsinki Metropolitan area libraries since 2014. Available online, it has currently 600,000 patrons and actively covers 300,000 book titles.

Some approaches, such as [Hua15], acknowledge that users can hold multiple interests and items may belong to multiple categories. The same topic from context point of view is addressed in [Sap16], suggesting that it is also important to incorporate the contextual information into the recommendation process. [Bin12] in turn introduced multi-class co-clustering for the purpose. Indeed, human life is versatile, and recommenders should respect this: Although one person may be interested in cooking and motorcycles and another interested in cooking and gardening, the recommender should not end up associating motorcycles with gardening.

The rest of this paper is structured as follows: Section 2 provides preliminaries about the token-based recommender. In Section 3, we introduce unsupervised N-dimensional clustering of tokens. Results are presented in Section 4 and concluded with discussions in Section 5.

2 Preliminaries

2.1 A Token-Based Recommender ("UPCV")

The method associates both users and items with collections of tokens, each token carrying a random value. Interaction between user and item triggers selected tokens to be copied from the token collection of the user to the token collection of the item, and vice versa. In previous papers[Oll13, Oll16], the maximum size of collections has been 1024 tokens, from which up to 15% have been randomly selected for the exchange.

When the same user interacts with several items, or the same item is involved in interactions with several users, tokens are spread around, resulting in statistical similarities among different token collections in the system. Since tokens are copied in user-item interactions only, it is likely that similarities between two token collections originate from similar user behavior. The method is collaborative by nature and requires no content analysis.

Collections are dynamic by nature and, **unlike cookies and other tracking means, tokens have no persistent associations with real world**. In particular, they do not have any association with persons: tokens are mere random values that, over time, will be copied to and deleted from collections.

The following example[Oll17] in Figure 1 explains how tokens are able to provide recommendations.

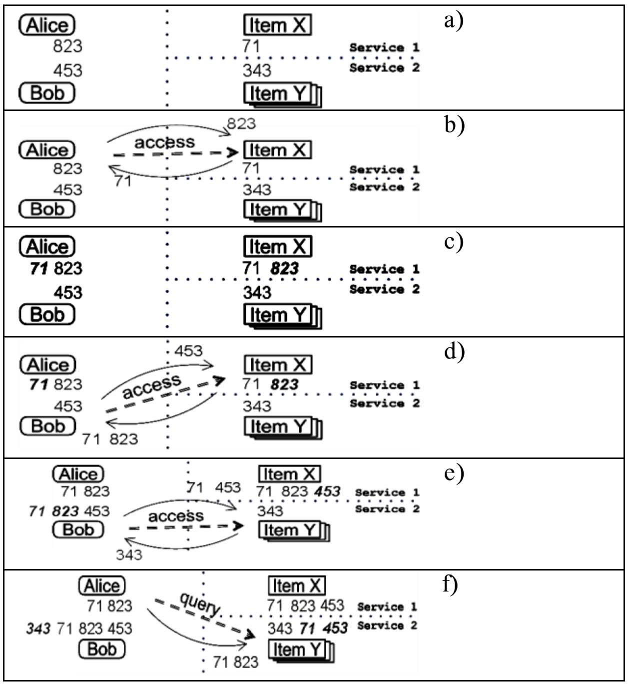

Figure 1: Example of Two Users and Two Services; Recommendation for Alice at Her First Visit in Service 2

The step a) in Figure 1 illustrates two independent services on the top, the upper service ("Service 1") having just a single item ("Item X") and the lower service ("Service 2") having a plurality of items, including "Item Y". Two users, "Alice" and "Bob", are presented for simplicity. In the beginning, each user and item has a single random number, "token", in their collections. Tokens are illustrated as 823, 463, 71 and 343.

In step b), Alice is accessing Item X, triggering a token exchange procedure; copying tokens from the user to the item and vice versa. In this step, both Alice and Item X have only one token; these are the exchanged tokens resulting step c), in which both Alice and Item X do have common tokens. (In the figure bold & italics font highlights the most recently acquired tokens.)

Next, Bob is accessing the same item X in the Service 1, and once again, a couple of tokens are copied over (step d). It should be noted that this time Item X is able to provide more than one token. After Bob's action, it should be noted that also Bob and Alice have similar tokens, as can be seen in step e).

Still in e): Bob is accessing Item Y in the Service 2. A couple of tokens are requested for exchange; since Bob has more than one token to give, some tokens are picked *randomly* from his collection.

As the last action in this example, Alice requests a recommendation for herself from ***the Service 2, which she is now visiting the very first time*** (step f). Finally, Service 2 goes through all available items and compares their tokens with the provided tokens. It is likely that Item Y will be in the recommendation list, since there is a token in common. For a similarity measure in general, previous papers [Oll13, Oll16, Oll17] suggest using Jaccard index.

2.2 Book Club Data Set and a Previous Study

The recommendation method was first introduced in [Oll13], using a data set collected from book club members of Bonnier Books Finland in 2013. This publication presents some details of the data set. Concisely, book club members (users) were asked, without limitation, which books (items) they had read and liked from a collection of 1041 books. 1575 members responded, of which 1532 selecting at least one book, providing 55434 individual selections total. Hence, density was 3.5%. 18 users selected exactly one book.

The selections were converted into *user-item pairs* ("transactions"), *shuffled into random order and divided into two subsets*, each consisting of 27717 user-item pairs in random order. The first subset was used as training data, while the second subset remained for validation. In the baseline study[Oll13], exchanged tokens were selected randomly while a maximum of 15% of token collections were exchanged in each transaction. When a token collection reached its maximum size (1024 tokens), tokens were deleted at random to make space.

The quantitative assessment was focusing on a practical question: how long should a recommendation list be, in order to have at least one successful recommendation, i.e. a book that a user has selected exists in the validation subset.

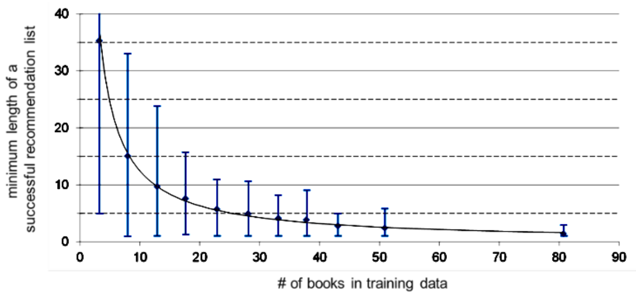

Figure 2: Median Length of a Successful Recommendation List (vertical axis, with 10%- 90% intervals) for Users With Specific Number of Books in Training Data (horizontal axis) [Oll13]

Figure 2 (above) from the baseline study illustrates that shorter recommendation lists were adequate for users who had a higher number of books selected in the training data [Oll13]. For example, half of the users having ~28 books in the training data got *at least one* successful recommendation in a list with five books. The figure also illustrates intervals that exclude the upper and lower 10% of the users in each group, and a best matching trend line.

2.3 News Portal Data Set Studies

"Ilkka" news portal dataset consists of 2123 users, 2439 news articles and 35891 news clicks in chronological order over a period of 30 days. Success was evaluated by checking if a user eventually clicked a recommended news article. In recommendations, each article had a 24-hour active lifespan from its first click. Training period was 7 days individually, beginning at each user's first click. Recommendations were created and evaluated at each click, excluding articles already clicked by the user. With these parameters, recommendation lists with at least one success were possible in 14552 clicks (hereinafter: 100% '*Click Coverage*'). Maximum token collection size was 256; otherwise, the setup was the same as in Chapter 2.2, with 15% token exchange (i.e. max. 38).

3 Methodology

3.1 Clustering of Token Collections

The motivation behind clustering is, that a recommender should not mix multiple interests, such as "cooking", "motorcycles" and "gardening", as mentioned in the Introduction.

As a novelty, we introduce now a conceptual view of tokens as points in a hyperspace by converting token's numeric value values into N-bit representation (e.g. N=24), resulting an N-dimensional binary hypercube. Each token represents one corner in this hypercube, each bit value (0 or 1) defining its projection in the respective dimension. A generic illustration of a hypercube is presented in Figure 3: An N-dimensional hypercube can be created by adding a copy of an N-1 dimensional hypercube with edges connecting respective corners. In the illustration, the corners are numbered in such a way that each dimension adds one most significant bit, in the original hypercube the bit value being 0 and in the copy 1.

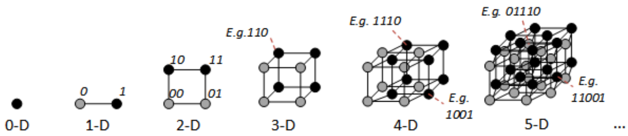

Figure 3: N-dimensional Hypercube Is a Doubled Copy of an N-1-Dimensional Hypercube with an Edge Between the Original (grey) and the Copied (black) Corner

In the novel concept, each token represents one corner in the hypercube, while a token collection is a set of corners, respectively.

Tokens *in each individual collection (separately)* were then clustered with bisecting K-means, recursively splitting a hypercube into clusters and sub clusters each time by applying a cutting hyperplane through a previous cluster, until a pre-defined maximum number of clusters was reached or the remaining sub clusters were dense enough.

For each cluster, we pick a best representing token, a "center token". When tokens are exchanged, the "item" discloses its center token and the "user" activates the cluster, which is closest to it. Only the tokens in the active cluster take part in token exchange. If the collection reached its maximum size, those tokens that are farthest away from any cluster were deleted first.

If an already existing token was received, a new token was created as a randomly selected neighboring corner (out of 24), making the particular cluster more dense.

A 24-dimensional hypercube has $2^{24} \approx 16$ million corners, being capable to accommodate a relatively high number of clusters reflecting a multitude of our interests.

3.2 New Similarity Metric

Instead of Jaccard [Oll13], a new similarity metric (1) is introduced in this paper to take into account cluster densities, while not requiring exact matches as in Jaccard. It sums up, how well each token in collection A matches the other collection B:

$$s(A,B) = \sum_{x \in A} 2^{-\min(\{ d^{HAD}(x,y) \mid y \in B \})} \quad (1)$$

where A and B are token collections and d^{HAD} is Hamming distance (i.e. number of differentiating bits). The new similarity metric improves coverage, since it does not require exact matches between token collections. As a drawback, it is computationally heavier.

3.3 Experiment

The experiment was carried out as in [Oll13], with the same data and other parameters when applicable, in order to compare the effectiveness of clustering. The maximum number of clusters for any single user was set to four, after some preliminary experiments. Items had a single cluster (i.e. no clustering). The number of exchanged tokens was reduced to 1, 3 and 10 tokens, however with an additional condition that the number of received tokens was never allowed to exceed the number of already existing tokens. Tokens that were closest to cluster centers were selected for exchange, instead of random picks used in the baseline method.

In the book experiment, the quantitative assessment was similar as in [Oll13]. Recommendations were given only to those users that were in the training data, containing 1523 users; some users having only one book selected had that book not in training but in validation data.

A second experiment was carried out with the news portal data set, comparing results of non-clustered and clustered methods with recommending random (with Monte Carlo evaluation) and most popular articles. The clustering method with 3 token exchange was compared to the baseline method as described in Chapter 2.3.

4 Results

As Figure 4 illustrates, clustering reduced learning time (# of books in training data) to up to one-third, while the overall quality was invariably improved. As found in [Oll13], the method seems to be not critical of the number of tokens exchanged: there was notable improvement *even when exchanging only one token* at a time.

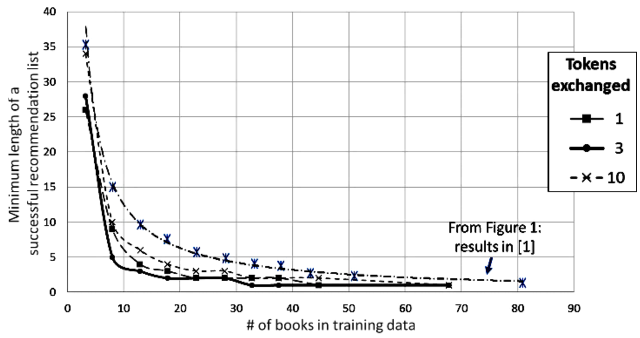

Figure 4: Median Length of a Successful Recommendation List With Clustering Token Collections, Exchanging Max 1, 3 and 10 Tokens in Each Transaction; all Outperform the Baseline

Exchanging 3 tokens provided fastest learning: For instance, half of the users having ~8 books in training data got a successful recommendation in a list of mere five books; the previous study indicated similar performance for users with ~25 books in training.

Table 1 presents respective recommendation coverage and number of successful recommendations (out of 1523 users) in the book club experiment. It is notable that recommendations were most diverse when only 1 token was exchanged: 31% of all books were in top-5 recommendations at least once. The number of unsuccessful recommendations were ~10 (i.e. 99.3% success): failures typically related to users with one rare book in training data. Only two exceptions were observed: users with two and three such classic books (at 1 and 3 token exchange, respectively) in training data that could be classified as bestsellers.

Table 1: Recommendation Coverage and Number of Successful Recommendations in the Book Club Experiment

# exchanged	# different books in			# average user		# successful
tokens	Top-1	Top-3	Top-5	tokens	clusters	recommendations
1	181	278	327	19	2.4	1509
3	136	220	257	37	2.7	1512
10	159	224	247	63	2.7	1513

As can be expected, the average number of tokens in each users' collections varied according to the number of exchanged tokens: e.g., when exchanging exactly 1 token at a time, a collection also grows by one at most. While the collections had different number of tokens, the number of clusters did not substantially vary.

Figure 5 below illustrates proportions of users that would get successful recommendations with given length of a recommendation list, when exchanging 3 tokens.

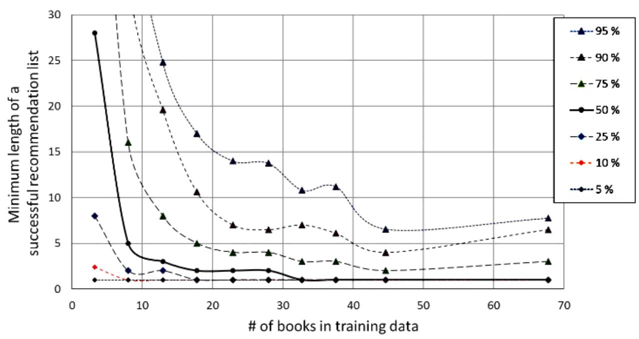

Figure 5: Length of a Recommendation List When Exchanging 3 Tokens in Each Transaction; the Proportion of Users Getting Successful Recommendations at Given Length of a Recommendation List

Finally, Table 2 resents quantitative results of the news recommendation experiment. The baseline method outperformed 'most popular' recommendations, but had a reduced click coverage. Compared to the baseline, the clustering method fell behind in recommendation quality but in turn provided full click coverage (c.f. 2.3). 3 tokens were exchanged, versus max. 38 in the baseline.

Table 2: Results of the News Recommendation Experiment

	Popularity	Random	Baseline	3 token exch.
Click coverage	100 %	100 %	88 %	**100 %**
Average minimum lenght of successful rec. lists	3.7	10.0	3.2	**4.6**

5 Conclusions and Discussion

This paper was focusing on comparing the results with a previous baseline study[Oll13], with the same data set and evaluation metrics. As a novelty, unsupervised N-dimensional (N=24) clustering was applied to each users' token collection individually, and only one cluster at a time was selected for token exchange. Each user had a maximum on four clusters.

In book club study this enhancement improved recommendation quality, while it at the same time reduced learning time. A previous publication[Oll13], reported that the method seems stable while varying the percentage of exchanged tokens. Clustering makes no difference: The results remained better in all cases, when the number of exchanged tokens was reduced from maximum of over 100 (15% out of 1024) to 1, 3 and 10.

Recommendations were diverse: About one-fourth of the entire 1041 book collection were presented in Top-5 recommendations. The evaluation suggested the best results with 3 token exchange. It is in further studies to optimize token exchange with different data sets.

Tokens provide privacy-protecting abstraction. In certain arrangements, a single token exchange will enable privacy, since it solves the remaining returning-user privacy issue introduced in [Oll16].

In the news recommendation experiment, quality improvement was less clear. Although the recommendation quality was close to popularity even with full coverage, a tradeoff between coverage and recommendation quality may exist: also worst cases get recommendations with full click coverage, adversely affecting quality evaluation. Further studies could quantify this phenomenon.

While the N-dimensional clusters can be assumed to reflect multiple interests of users, the true relation between clusters and multiple interests, perhaps utilizing newspaper sections, would be worth studying, together with optimizing the number of clusters.

References

[Bin12] Bin Xu, Jiajun Bu, Chun Chen, and Deng Cai. 2012. An exploration of improving collaborative recommender systems via user-item subgroups. In Proceedings of the 21st International Conference on World Wide Web. ACM, 21–30.

[Gao13] Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., & Guo, J. (2013). Cross-domain recommendation via cluster-level latent factor model. Lecture Notes in Computer Science, 8189 LNAI(PART 2), 161–176. doi:10.1007/978-3-642-40991-2_11

[Hua15] Huang, S., Ma, J., Cheng, P., Wang, S. (2015) A Hybrid Multigroup Coclustering Recommendation Framework Based on Information Fusion. ACM Trans. Intell. Syst. Technol. 6, 2, Article 27 (March 2015), 22 pages. DOI=http://dx.doi.org/10.1145/2700465

[Oll13] 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 available at www.vtt.fi/inf/julkaisut/muut/2013/OA_JPMTR_1314_Ollikainen.pdf

[Oll16] 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).

[Oll17] Ollikainen, V., Niemi, V. (2017) Evaluating the Performance and Privacy of a Token-Based Collaborative Recommender. In Proceedings of WI '17, Leipzig, Germany, August 23-26, 2017, 5 pages. http://dx.doi.org/10.1145/3106426.3109434

[Sap16] Sappelli, M., Verberne, S., & Kraaij, W. (2016). Evaluation of context-aware recommendation systems for information re-finding. Journal of the Association for Information Science and Technology, 68(4), 895-910. doi:10.1002/asi.23717

[Zha12] Zhang, Y., Cao, B., & Yeung, D.-Y. (2012). Multi-domain collaborative filtering. arXiv Preprint arXiv:1203.3535.