World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:3, 2021

Hybrid Collaborative-Context Based Recommendations for Civil Affairs Operations

Patrick Cummings, Laura Cassani, Deirdre Kelliher

***Abstract*—**In this paper we present findings from a research effort to apply a hybrid collaborative-context approach for a system focused on Marine Corps civil affairs data collection, aggregation, and analysis called the Marine Civil Information Management System (MARCIMS). The goal of this effort is to provide operators with information to make sense of the interconnectedness of entities and relationships in their area of operation and discover existing data to support civil military operations. Our approach to build a recommendation engine was designed to overcome several technical challenges, including 1) ensuring models were robust to the relatively small amount of data collected by the Marine Corps civil affairs community; 2) finding methods to recommend novel data for which there are no interactions captured; and 3) overcoming confirmation bias by ensuring content was recommended that was relevant for the mission despite being obscure or less well known. We solve this by implementing a combination of collective matrix factorization (CMF) and graph-based random walks to provide recommendations to civil military operations users. We also present a method to resolve the challenge of computation complexity inherent from highly connected nodes through a precomputed process.

***Keywords*—**Recommendation engine, collaborative filtering, context based recommendation, graph analysis, coverage, civil affairs operations, Marine Corps.

I. Introduction

ADVANCES in recommendation engine technology have led to the rapid implementation of capabilities across the commercial world, as evidenced in activities such as browsing items online for purchase, selecting music playlists, and choosing what television show or movie to watch. Collaborative recommendation approaches, such as matrix factorization, use information about interactions between users and entities to extract latent representations of each. Those representations can then be combined (e.g. with cosine similarity) to provide recommendations, matching users to specific entities. Current state-of-the-art recommendation engines used by companies such as Netflix, Hulu, and Pandora have shown promising results in predicting user ratings [1], [2] using approaches based on collaborative filtering. However, despite these advances, there has been limited penetration of recommendation engine technology in the military domain, where often a relatively small number of users require access to relevant information in context to specific mission parameters. These systems often lack the large amount of interaction information that power many existing recommendation approaches. This is particularly true for the MARCIMS. It is a Marine Corps civil affairs data collection, aggregation, and analysis tool. In this application, users conducting operations or exercises add information about the civil environment (e.g., bridges, hospitals, villages) to a knowledge base. These data are semi-structured, with a combination of free text and predefined fields capturing data about the civil entities. Pages aggregate knowledge and query data to generate various maps, charts, graphs, and tables to support situational awareness and decision-making tasks. A common problem however is that rotation of personnel creates gaps in knowledge, with individuals often unaware of what information is available or what has been done previously in an area of operations. For example, a team deploying to support a humanitarian operation in an area of operations may be unaware of available information about what has been previously assessed in that area. In this application it is important to support operators in discovering the full context of relevant information that may exist in the knowledge base to enable mission activities.

We present here a hybrid collaborative-context based approach for recommending content on MARCIMS. This approach was designed with three specific outcomes in mind. First, models must be constructed using a limited amount of interaction data. Second, the cold start problem must be frequently overcome to assist with frequent new documents. Finally, since much of the content is obscure and entered by others, we needed to avoid confirmation bias by recommending with coverage in mind rather than solely relevance.

P. Cummings, L. Cassani, and D. Kelliher are with Aptima Inc, Woburn, MA 01801 USA (e-mail: pcummings@aptima.com, lcassani@aptima.com, dkelliher@aptima.com)

II. Related Work

While a collaborative filtering approach provides several benefits in many application domains, in the military context there are several challenges to address. The first and most critical drawback of collaborative filtering is the amount of data necessary to train the models for recommendations. Collaborative approaches, particularly those using deep learning [3] require vast amounts of interaction data between users and items to be able to develop patterns between them. In our case, the target application had a relatively small user base, and lacked the amount of training data necessary to ensure our models would be robust with only collaborative filtering. The second critical drawback of collaborative filtering is the cold-start problem. When a new item is generated and enters the system (e.g., a new movie in Netflix), end-user interaction data do not yet exist for that content, so collaborative filtering fails. While that scenario may be rarer for media streaming (Netfilx, Hulu) or shopping (Amazon),

the Marine Corps application is frequently generating new page items as Marines are engaging with the population during operations and exercises. We solve this problem by employing a hybrid approach using both collaborative filtering and a context-based approach—an approach that uses features within the pages and those related to the user to make recommendations [4], [5]. Finally, the third drawback is the tendency for collaborative filtering to promote confirmation bias [6]. The credibility of recommendation systems that rely on collaborative filtering suffers when those systems fail to accommodate alternative or conflicting viewpoints. Many current recommendation systems create a positive feedback loop because people are drawn to highly recommended topics that the system then recommends to others in turn. Analysts need information distributed across supporting and refuting data sources and topics. We prevent confirmation bias by weighting pages to promote diversity and coverage.

Context-based recommendation approaches tackle the recommendation problem from a completely different angle. Rather than viewing the interactions between users and items, context-based approaches evaluate and gather information about each page and user in order to compare based on their content. Many methods have been created that do this in an automated fashion by tracking the user or extracting features from the items [7], but they lack the patterns found by looking at the item-user interactions together [8].

By utilizing a hybrid collaborative-context based approach we can leverage advantages and mitigate weaknesses of the disparate approaches described above. We present here a hybrid approach, entitled Civil Military Operations Recommendation Engine (CMORE), which uses a combination of CMF and graph-based random walks to provide collaborative-context based recommendations to civil military operations users.

III. Methods

CMORE's recommendation process is split into two phases. The first phase builds a model adapting a CMF approach [9]. Similar to what has been done in [5], [10], and [4], we build off of that CMF approach by adding context to the classifier. By extracting latent features from CMF and using other context features, a graph is built and used to provide recommendations through a random walk. Of particular note is that using this combined approach allows us to provide recommendations based solely on a user's current page, the users themselves, or a combination of both (as well as potentially incorporating recently visited pages). We represent these page and user data with matrix structures. Formally, we define the set of documents as $D = \{d_i\}_{i=1}^{n_d}$, users as $\mathrm{U} = \{\mathrm{u_i}\}_{\mathrm{i=1}}^{\mathrm{n_u}}$ and document features as $f_d \in F^d$ where the features for each document is some subset of F^d. Our framework also allows the encoding of user features, but that is excluded from this analysis. We then track the interactions between users and documents via $E_{u,d}$, the interactions between documents and other documents via $E_{d,d}$, and the document features via E_{d,f^d}, where each matrix element is defined via the relationship between the two items (e.g. $e_{u,d}^{jk} = 1$ if user u_j modified document d_k). With this information, we seek to build latent representations of user-document pairs (u,d), document-document pairs (d,d), and document-document feature pairs (d,f^d). To do so, we build latent representations of their components. Precisely, we define latent representations ℓ of the user, document, and document features that when combined via cosine similarity give us latent representations between them. Those latent features are generally combined with the dot product and mapped by some recommendation function, g, (commonly the identity or sigmoid function) to form a score $R(u,z,d)$ for each user-document pair.

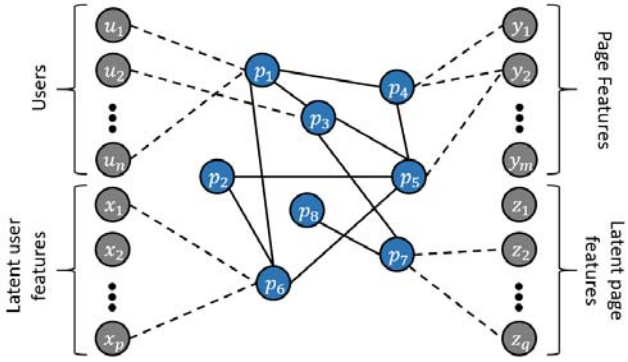

Fig. 1 CMORE architecture graph structure

The output of the first phase of CMORE is then vectors representing the pages, the users, and the page features. These latent features capture relationships between pages that are not found within the explicit features. These features are combined with the explicit features for each. Explicit features on the user side is a binary "on/off" for whether a user modified a page; explicit on the page side is a binary "on/off" for if that page has the feature. This combination of latent and explicit features form the basis for the second phase of the CMORE recommendation approach via a graph representation of their relationships as shown in Fig. 1. In the center each page is represented as a node and page-to-page links represented as edges connecting those nodes. On the left and right sides the explicit and latent feature nodes, respectively, connect to pages that have that feature. Formally, this graph is defined as $G = (V,E)$, where $V = \{1, \ldots, |V|\}$ is a set of nodes and $E = \{(i,j) \mid i,j \in V\}$ is a set of links between them. The node can represent users, pages, page features, and latent features of both the pages and the users. The edges represent actual links between pages for page-to-page edges, and the presence of that feature for the others. With this graph we can then use random walk with restart to provide a ranking of recommended pages. Random walk with restart exploits a random surfer to produce page rankings. Given an adjacency matrix A_G for the graph $\boldsymbol{G}$ we compute the rankings for page i, defined in the vector x_i, via

$$x_i^{n+1} = (1-c)A_G x_i^n + c \cdot 1_i \quad (1)$$

where 1_i is the vector with one in the i^{th} position and zero elsewhere. Starting with $x_i^0 = 1_i$ we iterate (1) until convergence.

A. Highly Connected Nodes

A difficulty inherent in using latent features as nodes in a graph-based approach is that those nodes are highly connected to the items those features describe. Iteration of (1) in real-time systems depend on the sparsity of the graph, and therefore the sparsity of x_i, in order to compute in near real-time. These highly-connected nodes remove this sparsity, and so computational complexity becomes intractable for real-time systems. Nevertheless, we can solve this issue through the utilization of the Theorem:

Theorem. Let A be an $N \times X$ adjacency matrix and e^i the unit vector with $e^i_j = 1$ if $i = j$ and 0 otherwise. We define the restart probability c and $r^i =$ rankings for node i given by steady state of

$$r^i = (1 - c)Ar^i + ce^i$$

Let $X = \{x_i\} \subset \{1, \dots, N\}$ be a subset of nodes and $\widetilde{A}$ be an updated adjacency defined by

$$\mathrm{col}_i(\widetilde{A}) = \begin{cases} \mathrm{col}_i(A) \text{ if } i \notin X \\ e^i \quad \text{if } i \in X \end{cases}$$

Then if $\tilde{r}$ is defined as the steady state of

$$\tilde{r}^i = (1 - c)\widetilde{A}\tilde{r}^i + ce^i, \tag{2}$$

then

$$r^i = \sum_{k=0}^{N} \chi_X^k \cdot \tilde{r}_k^i \cdot r^k + \left(1 - \chi_X^k\right)\tilde{r}_k^i. \tag{3}$$

In other words, by precomputing the steady state scores for the highly-connected nodes X, the graph A can be adjusted such that each of those nodes are sinks. This sink structure ensures the sparsity of $\tilde{r}_k^i$ in (3). Then the solution to (1) can be computed with a combination of (2) and (3) which are much more tractable.

IV. Advantages

CMORE's recommendation process has four features that are particularly advantageous to the application at hand. Those are: reduction of bias, smooth handling of cold-start problem, transparency to end users, and composition of recommendations. We discuss each in detail below.

Bias reduction: Frequently, recommendation approaches can be bogged down by popular features or items that are linked to many objects. This results in the most popular items being recommended more, even when that might not be the most useful or beneficial recommendation. In our context, this was a challenge given that the application had several pages that were aggregations of data for multiple operations and served as a hub of relevant knowledge for users and therefore were highly frequented. This led to a large amount of training data biasing the recommendation engine toward those pages. In the case of many recommendation engines, such as movies or articles, recommending this popular item might be helpful, but in our application we want to recommend useful information that users would not normally see otherwise—we do not want to recommend a page of which everyone is already aware. Similarly, if many pages contain a particular feature, we may want to lessen the weight of that feature. Not only do these bias terms reduce the tendency to frequently recommend most popular pages, but they also mitigate the model overfitting to those pages and features. Formally, the method to introduce these bias terms is via weighting the edges within the linking graph such that pages or features are linked with a weight inverse to the frequency they appear within the data. Then as the "random walker" within RWR traverses the graph, the probability of the walker following an edge to a less frequent feature or page will be greater than that of a more prevalent feature or page. In (1) this lower probability is translated to a smaller coefficient in A_G.

Cold-Start: As mentioned above, a challenge for this application is that the page set we are working with is constantly being updated and added to. This means the cold-start problem is a critical component that must be addressed. Collaborative filtering approaches fail at the cold-start problem because they do not have the user data to draw latent relationships between pages. Our graph-based addition to the collaborative matrix factorization allows for the addition of pages and their explicit features to the graph upon creation. Therefore, even with only those explicit features we can provide relevant recommendations.

Transparency: Since the RWR algorithm works by recommending nodes that the "random walker" visited, we are provided with an explicit score for that node. By viewing the weights of r_i from (3) we can put significant weight on greater nodes, and the converse is also true. By viewing the features, pages, and users that contributed to the recommendation it would be possible for the user to track why each page was recommended. This could help both internally to determine whether the recommendation engine is generating recommendations appropriately, but also externally as a resource for end users to understand why pages are being recommended. Furthermore, the graph-based approach allows us to see which nodes connect (or are around) the base and recommended nodes. These provide insight into the shared attributes and features that lead to the recommendations. The visualization of the recommendation explanation is a feature we aim to focus on in future phases of this research.

Recommendation composition: By updating the query to include multiple nodes, we can include both the user node (which is linked to features specific to that user) as well as recently viewed pages (which provide context to the user's current interest) in the query to recommend pages specific to the user's current interests. This approach allows for quick computation to recommend content based on more than one page or based on user by generating the recommendation vector r_i for each page and user. Therefore, making recommendations based on multiple pages (e.g., the last few

pages visited) or the user viewing the page requires only a (potentially weighted) sum of vectors.

V. Validation

A. Data

The CMORE recommendation approach uses data across three dimensions: user activity logs, page-to-page links, and page features.

User activity logs consist of user edits, logins, logouts, creations, deletions, and other actions (but not simple viewing). For this initial experiment, we filtered the logs to only include the creation and edit actions, implying that a user is interacting with a page. Other cleaning steps included removing test/admin users and grouping edits by the hour, to avoid skewing the results caused by multiple edits in a single editing session. These user activity logs form the matrix $E_{u,d}$ of user-document relationships.

Page-to-page links are hyperlinks between pages. Due to the structure of the information, we utilized the wealth of links that occur between pages. We make the assumption that pages that link to each other are related to each other and worth recommending. This results in a symmetric matrix $E_{d,d}$, or equivalently biderectional edges between pages in the graph.

The pages in the dataset have two types of features. The first is explicit features filled into tables. These properties are consistent across groups of pages providing a repeatable and easily extractable structure. For this experiment we include only those features that are found on at least N pages (for some suitably chosen N). The second type of feature comes from entities extracted from the text. We use off-the-shelf extraction tools to extract features in text [11] and again only included those features found on at least N pages. These relationships form the matrix E_{d,f^d}.

B. Methods

To evaluate the validity of our recommendation engine, we compare its performance to two alternative methods. The first is a pure matrix factorization approach, which is, as stated, the leading approach for collaborative filtering. The second alternative is a standard baseline that always recommends the most popular or highly connected pages. This method, while simple, has the advantage of always recommending pages that are generally relevant to most users, even if they are not relevant in all contexts [12]. In addition to analyzing general relevance of outputs, we compare these three systems using two advanced metrics: diversity and coverage. In recommender systems, higher performance on these metrics has been linked with higher user satisfaction with recommendations [13]. We define these metrics as follows: (1) Diversity: the average distance between recommendations within a set (i.e. recommendations on one page), (2) Coverage: the percentage of all eligible items that are recommended at least once across the whole system [13], [14].

Higher levels of diversity from a recommendation method would indicate richer sets of recommendations being presented for users to choose from, offering them to a wider array of options [13]. In turn, higher coverage would indicate increased overall page exposure for users as they navigate through the system, allowing them a greater understanding of and familiarity with the system content. For our application, these are both desirable outcomes as they encourage users to become more familiar with the website as a whole and allow access to less popular but potentially relevant information they otherwise may not find. Using mathematical analysis, we measure each algorithm's performance on these metrics, looking for high performance that does not compromise relevance.

For a singular page with one set, R, of recommendations, diversity can be measured as the average pairwise distance between recommendations in that set:

$$Diversity(R) = \frac{\sum_{i \epsilon R} \sum_{j \epsilon R \setminus \{i\}} dist(i,j)}{|R|(|R|-1)}$$

This can then be extended to calculate the diversity of the whole system as the average diversity of all sets. For our purposes the distance function, $dist(i,j)$, compares qualitative features of pages i and j to measure their similarity. Specifically, our analytic looks for matches across three genres relevant to our application: page types, associated operations, and publishing year. Similarity on any of these features leads to a lower distance score. Next, coverage can be measured as

$$Coverage(P_e) = \frac{P_r}{P_e}$$

where P_e is the set of all recommendable pages and P_r is the subset of P_e pages that are recommended at least once. To calculate coverage, we developed an analytic that catalogs all recommendations and divides the number of unique recommendations by the total number of eligible pages. Both the diversity and coverage analytics are run across the entire dataset three times, once for each of the recommendation conditions.

C. Results

Using the recommendable pages from the dataset, we evaluated the CMORE recommendation architecture against the two baselines. After filtering for testing and incomplete pages, the evaluation used 4,361 recommendable pages and six recommendations per page. The results of those experiments are shown in Table I.

TABLE I
Validation Results

Analytic	Coverage	Diversity
CMORE	0.48	0.26
CMF	0.03	0.21
Popularity	0.001	0.35

As shown, the CMORE architecture outperforms matrix factorization and the popular pages baseline on the coverage measure. With the primary goal of exposing critical and buried information to civil affairs operators, this coverage score is

very promising for our approach. The diversity score was comparable to the competing approaches. We note, however, that the diversity of the popular pages approach is somewhat deceptive. Typically, the diversity measure is the calculation of the diversity of the six recommended pages averaged across all pages. However, for this baseline, all pages have the same six recommendations. So this tells us that the most popular six pages are very diverse, but as you go to multiple pages the diversity does not increase.

VI. Conclusion

In this paper we presented a hybrid context and collaborative filtering approach to generate recommendations for a system focused on civil military operations data. This architecture has four distinct advantages—bias reduction, cold-start handling, transparency, and composition. With a strong focus on removing bias and exposing operators to a greater breadth of content, this architecture excels at providing operators with coverage of relevant information to support their mission and will be deployed as part of the system baseline going forward.

Acknowledgment

This research was funded by the Office of Naval Research under the guidance and leadership of Martin Kruger and Gary Kollmorgen (contract N00014-16-C-1039). Additionally, we would like to thank our colleagues on the MARCIMS project team at Marine Corps Systems Command, Army Geospatial Center, and the Marine Corps Civil Military Operations Schoolhouse, for generously providing their time and expertise in helping to shape this research effort to ensure that is relevant and useful for the Marine Corps civil affairs community.

References

[1] Y. Zheng, B. Tang, W. Ding, and H. Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (ICML'16). JMLR.org, 764–773. http://dl.acm.org/citation.cfm?id=3045390.3045472

[2] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. 2008. Large-Scale Parallel Collaborative Filtering for the Netflix Prize. *In Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management* (AAIM '08). Springer-Verlag, Berlin, Heidelberg, 337–348. https://doi.org/10.1007/978-3-540-68880-8_32

[3] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook, Recommender Systems Handbook*, Springer, 2011, pp. 1-35

[4] M. Liu, X. Xie, and H. Zhou. 2018. Content-based Video Relevance Prediction Challenge: Data, Protocol, and Baseline. CoRR abs/1806.00737 (2018). arXiv:1806.00737 http://arxiv.org/abs/1806.00737

[5] J. Wilson, S. Chaudhury, and B. Lall. 2014. Improving Collaborative Filtering Based Recommenders Using Topic Modelling. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* - Volume 01 (WI-IAT '14). IEEE Computer Society, Washington, DC, USA, 340–346. https://doi.org/10.1109/WI-IAT.2014.54

[6] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis. 2017. Reducing Controversy by Connecting Opposing Views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (WSDM '17). ACM, New York, NY, USA, 81–90. DOI:http://dx.doi.org/10.1145/3018661.3018703

[7] R. V. Meteren and M. V. Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pages 47–56, 2000.

[8] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

[9] A. P. Singh and G. J. Gordon. 2008. Relational Learning via Collective Matrix Factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '08). ACM, New York, NY, USA, 650–658. https://doi.org/10.1145/1401890.1401969

[10] M. Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems* (RecSys 2015), Vienna, Austria, September 16-20, 2015. (CEUR Workshop Proceedings), Toine Bogers and Marijn Koolen (Eds.), Vol. 1448. CEUR-WS.org, 14–21. http://ceur-ws.org/Vol-1448/paper4.pdf

[11] M. Honniba and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[12] G. Adomavicius and Y. Kwon, "Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques," *in IEEE Transactions on Knowledge and Data Engineering,* vol. 24, no. 5, pp. 896-911, May 2012, doi: 10.1109/TKDE.2011.15.

[13] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in *recommender systems,"ACM Trans. Interact. Intell. Syst*., vol. 7, pp. 2:1–2:42,Dec. 2016.

[14] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: Evaluating recommender systems by coverage and serendipity," in *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, (NewYork, NY, USA), pp. 257–260, ACM, 2010.