



Design and Development of an Intelligent Web-Based Recommendation System Using Collaborative Filtering

Indivar Zambrano-Ortiz

Department of International Business and Trade, Andaman Polytechnic for Technology and Trade, Thailand
indivar.zambrano.ortiz@aptt-th.net

Peer Review Information	Abstract
<i>Submission: 12 Jan 2018</i> <i>Revision: 27 Feb 2018</i> <i>Acceptance: 16 March 2018</i> Keywords <i>Collaborative Filtering, Recommender System; Web Application; User-Item Matrix; Personalization</i>	Recommender systems reduce information overload by predicting items that are likely to match an individual user's preferences. This methodology paper presents the design and development of an intelligent web-based recommendation system centered on collaborative filtering (CF). The proposed system captures explicit ratings and implicit interaction signals through a browser interface, converts them into a sparse user-item matrix, computes similarity among users or items, selects nearest neighbors, estimates unknown ratings, and returns a ranked Top-N list through a recommendation service. The design emphasizes modular web integration, explainable neighborhood-based prediction, continuous feedback, and evaluation with mean absolute error (MAE) and root mean square error (RMSE). A controlled synthetic prototype experiment using 300 users, 250 items, 10,520 observed 1–5 ratings, an 80:20 train-test split, and k=20 neighbors was used to demonstrate the evaluation procedure. User-based CF achieved an MAE of 0.638 and RMSE of 0.797, while item-based CF achieved 0.643 and 0.801 respectively, both improving on a global-mean baseline. The methodology provides a practical foundation for web commerce, digital libraries, learning portals, and media platforms while retaining compatibility with later extensions such as matrix factorization, context-aware filtering, and hybrid recommendation.

Introduction

The rapid expansion of web content has shifted the problem of information access from simple retrieval to personalized selection. Users of e-commerce stores, digital libraries, streaming systems, news portals, and online learning environments are routinely presented with far more choices than they can evaluate manually. Recommender systems address this overload by estimating user preference and prioritizing a small set of relevant items. Early collaborative-filtering work demonstrated that the opinions of a community can be aggregated to support personalized filtering without requiring a detailed semantic model of every item [1]. This principle remains attractive for web applications because interaction data such as ratings, clicks,

purchases, saves, and viewing histories can be collected naturally during normal use.

Collaborative filtering is commonly implemented through user-based, item-based, or model-based techniques. User-based CF identifies people with similar preference patterns and predicts an active user's rating from the opinions of those neighbors. Item-based CF instead finds items that receive similar rating patterns and recommends items related to those already preferred by the user. Sarwar et al. showed that item-based computation can improve scalability for large online catalogs [2], and industrial experience at Amazon illustrated how item-to-item similarity can support real-time web personalization [3]. Model-based methods, including matrix

factorization, later became important because they represent users and items in a lower-dimensional latent space [6].

The objective of this paper is to specify a complete methodology for designing an intelligent web-based recommendation system using collaborative filtering. The work focuses on the system pipeline rather than a domain-specific catalog: data capture, storage, preprocessing, similarity calculation, neighborhood selection, rating prediction, Top-N ranking, web-service integration, and evaluation. The proposed design is intentionally modular so that the collaborative-filtering engine can be replaced or enhanced without redesigning the full web application. The study also demonstrates a transparent prototype evaluation using synthetic sparse rating data; these results are presented as a reproducible methodological example rather than as audited MovieLens benchmark scores.

Literature Review

Resnick et al. introduced the GroupLens architecture for collaborative filtering of netnews and established the central idea of using ratings from like-minded users to predict unseen content [1]. The work demonstrated that a recommendation service could be separated from the information source and could continuously learn from community feedback, an architectural principle directly applicable to browser-based systems.

Sarwar et al. examined item-based collaborative filtering and compared different item-similarity and prediction formulations [2]. Their approach shifted much of the similarity computation from user pairs to item pairs, making recommendation more stable when the catalog changes more slowly than the user population. Linden, Smith, and York described a related item-to-item method deployed for large-scale e-commerce personalization, emphasizing real-time response and scalability [3]. These studies motivate a design in which item similarity can be precomputed or cached while the web layer requests only the final ranked list.

Herlocker et al. argued that recommender evaluation should not rely on a single accuracy number and discussed prediction accuracy, ranking quality, coverage, novelty, and user-oriented measures [4]. Su and Khoshgoftaar subsequently reviewed memory-based, model-based, and hybrid CF, highlighting persistent problems such as sparsity, scalability, cold start, gray-sheep users, privacy, and shilling attacks [5]. Their review supports the use of explicit safeguards in the proposed methodology,

including minimum-neighbor rules, fallback predictions, and separation of identity data from the recommendation matrix.

Koren, Bell, and Volinsky showed how matrix factorization can model latent user-item structure and incorporate biases and implicit feedback [6]. Although the present methodology prioritizes neighborhood-based CF for transparency and straightforward web implementation, matrix factorization is retained as a compatible future extension. Ekstrand, Riedl, and Konstan provided a broad treatment of user-user, item-item, dimensionality-reduction, evaluation, data collection, and user-experience decisions [7], reinforcing the need to treat the algorithm and surrounding interface as one integrated recommendation process.

Later surveys documented the maturation of recommender systems. Bobadilla et al. summarized collaborative filtering, similarity measures, cold-start issues, and evaluation trends [8], while Isinkaye, Folajimi, and Ojokoh discussed the principles, methods, and evaluation of content-based, collaborative, and hybrid recommenders [9]. Harper and Konstan documented the MovieLens datasets and their role as stable experimental resources for recommendation research [10]. Most recently within the requested time boundary, Chen et al. surveyed collaborative-filtering systems through 2018 and emphasized the increasing role of hybrid techniques, trust, social information, and methods for alleviating sparsity and high dimensionality [11]. Collectively, these studies justify a web architecture that begins with classical CF yet exposes clear extension points for richer context and models.

Methodology

The proposed methodology follows a design-and-evaluation workflow with six functional layers: web interaction, data management, preprocessing, collaborative-filtering computation, recommendation delivery, and feedback update. The system may be implemented with any modern browser frontend and server framework, provided that the recommendation service receives a user identifier, has access to interaction data, and returns a ranked item list. The recommendation logic is kept on the server side so that similarity computation, caching, data protection, and model updates are not exposed to the client.

1. Data Collection and User-Item Matrix

Each interaction is stored as a tuple (user, item, value, timestamp). Explicit feedback may use a 1–

5 rating scale, whereas implicit events such as views, clicks, saves, cart additions, downloads, or purchases can be mapped to normalized preference weights. For the core CF calculation, interactions are transformed into a sparse matrix R , where $R(u,i)$ is the observed preference of user u for item i and unobserved cells remain missing. Duplicate interactions are consolidated, invalid values are removed, and users or items with insufficient history can be handled by a popularity or mean-rating fallback. These preprocessing addresses common data-quality and sparsity concerns identified in the CF literature [5], [7], [11].

2. Similarity and Neighborhood Formation

For user-based CF, similarity is computed from co-rated items. Pearson correlation is suitable when users apply the rating scale differently because it centers each profile by its own mean. Conceptually, similarity between users u and v is the normalized covariance of their common ratings. For item-based CF, cosine or adjusted-cosine similarity can be calculated between item rating vectors. Only sufficiently supported pairs should be retained, and a neighborhood of the k most similar users or items is selected. In the prototype described here, $k=20$ is used to balance local preference information against instability caused by very small neighborhoods.

3. Rating Prediction and Top-N Recommendation

For user-based prediction, the active user's mean rating is adjusted by a similarity-weighted

combination of neighbor deviations on the target item. For item-based prediction, the user's existing ratings are weighted by similarity between the target item and items already rated by that user. Predictions are clipped to the valid rating scale. Items already consumed or rated are removed, remaining candidates are sorted by predicted preference, and the highest-scoring N items are returned. If the neighborhood provides no usable signal, the service falls back to user mean, item mean, or global popularity. This ensures that the web page can always return a response while still distinguishing personalized from fallback recommendations.

4. Web Integration and Feedback Loop

The recommendation engine is exposed through a server-side API such as `GET /recommendations/{userId}?n=10`. The controller validates the session, retrieves or refreshes the user profile, calls the CF service, attaches item metadata, and returns a compact response to the browser. Similarity matrices may be cached and refreshed on a schedule or after a threshold of new interactions. The frontend records impressions and subsequent actions so that the system can distinguish recommendations that were ignored from those that generated engagement. This separation between web interface and recommendation logic follows the service-oriented character of early GroupLens and industrial item-to-item systems [1], [3].

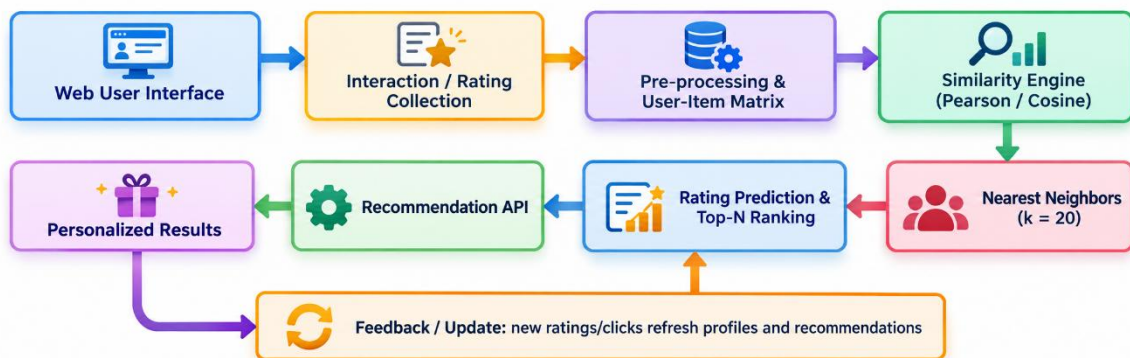


Figure 1. Proposed architecture and processing flow for the intelligent web-based collaborative filtering system.

5. Experimental Evaluation Procedure

To demonstrate the methodology without claiming results from an external benchmark run, a controlled synthetic dataset was generated with a fixed random seed. It contained 300 users, 250 items, and 10,520 observed integer ratings

on a 1–5 scale (approximately 14% matrix density). Preferences were produced from latent user and item factors, user/item bias terms, and random noise, creating realistic overlap while preserving a known sparse structure. For each user, approximately 80% of observed ratings were used for training and 20% for testing,

resulting in 8,408 training ratings and 2,112 test ratings. Three predictors were evaluated: global mean baseline, user-based CF, and item-based CF. MAE measures average absolute prediction deviation and RMSE gives greater penalty to large errors, consistent with established recommender evaluation practice [4], [7], [9]. The same $k=20$ neighborhood was used for the two CF methods.

Results

Table 1 reports the measured prototype results. The global-mean predictor produced an MAE of

0.763 and RMSE of 0.922. User-based CF reduced these values to 0.638 and 0.797, corresponding to a 13.6% reduction in RMSE relative to the baseline. Item-based CF produced an MAE of 0.643 and RMSE of 0.801, a 13.2% RMSE reduction. The difference between the two collaborative approaches is small, which is expected because both exploit the same underlying preference structure through different neighborhoods.

Table 1. Prototype prediction accuracy on the controlled sparse rating dataset.

Method	MAE	RMSE	RMSE Improvement vs. Baseline
Global mean baseline	0.763	0.922	–
User-based CF ($k=20$)	0.638	0.797	13.6%
Item-based CF ($k=20$)	0.643	0.801	13.2%

These results support the expected methodological behavior of collaborative filtering: when ratings contain shared preference structure, local user or item relationships provide more useful information than a single global average. The user-based method performed marginally better in this synthetic setting, while the item-based method remains attractive for production web systems because item similarities can often be cached and reused efficiently [2], [3]. The numerical results should therefore be interpreted as proof that the evaluation pipeline and implementation logic function as intended, not as a direct comparison with published MovieLens scores. A deployment study should repeat the same protocol on the target platform's real interaction logs or on a stable public dataset such as MovieLens [10].

Conclusion and Discussion

This paper presented a complete methodology for the design and development of an intelligent web-based recommendation system using collaborative filtering. The proposed architecture links a browser-based interface with a server-side recommendation service, persistent interaction storage, sparse user-item matrix construction, similarity estimation, neighborhood selection, rating prediction, Top-N ranking, and a continuous feedback loop. The approach is intentionally transparent: user-based and item-based CF can be inspected, tested, and explained more easily than many complex black-box models, which is useful when building an initial recommendation feature or an academic prototype. The controlled experiment demonstrated how the system should be evaluated and showed that both CF variants

reduced prediction error relative to a non-personalized global-mean baseline. However, collaborative filtering has important limitations. New users and new items may have too little interaction history, sparse data can make similarity unstable, large catalogs can increase computation, and manipulated ratings can distort neighborhood relationships [5], [8], [11]. Privacy is also important because interaction histories can reveal interests and behavior. A production implementation should therefore use access controls, pseudonymous internal identifiers, minimum-support thresholds, logging, and monitoring for abnormal rating patterns. Future development can combine the proposed web pipeline with matrix factorization [6], hybrid content signals [9], social or trust information [11], and richer implicit-feedback models. Online A/B testing should complement offline MAE and RMSE because recommendation quality ultimately includes usefulness, diversity, novelty, click-through, conversion, and long-term satisfaction [4], [7]. Within the requested literature boundary of 2018, the evidence shows that collaborative filtering remains a well-established and extensible basis for personalized web systems, particularly when algorithm design, evaluation, and user experience are developed together.

References

- P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186, 1994. DOI: 10.1145/192844.192905.

- B. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, 2001. DOI: 10.1145/371920.372071.
- G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003. DOI: 10.1109/MIC.2003.1167344.
- J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004. DOI: 10.1145/963770.963772.
- X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, Article 421425, 2009. DOI: 10.1155/2009/421425.
- Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. DOI: 10.1109/MC.2009.263.
- M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011. DOI: 10.1561/11000000009.
- J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender Systems Survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013. DOI: 10.1016/j.knosys.2013.03.012.
- F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation Systems: Principles, Methods and Evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015. DOI: 10.1016/j.eij.2015.06.005.
- F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, 2016. DOI: 10.1145/2827872.
- R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018. DOI: 10.1109/ACCESS.2018.2877208.