# 'Free Lunch' Enhancement for Collaborative Filtering with Factorization Machines

Babak Loni, Alan Said, Martha Larson, Alan Hanjalic
Multimedia Computing Group
Delft University of Technology, Netherlands
{b.loni, a.said, m.a.larson, a.hanjalic}@tudelft.nl,

## ABSTRACT

The advantage of Factorization Machines over other factorization models is their ability to easily integrate and efficiently exploit auxiliary information to improve Collaborative Filtering. Until now, this auxiliary information has been drawn from external knowledge sources beyond the user-item matrix. In this paper, we demonstrate that Factorization Machines can exploit additional representations of information inherent in the user-item matrix to improve recommendation performance. We refer to our approach as 'Free Lunch' enhancement since it leverages clusters that are based on information that is present in the user-item matrix, but not otherwise directly exploited during matrix factorization. Borrowing clustering concepts from codebook sharing, our approach can also make use of 'Free Lunch' information inherent in a user-item matrix from an auxiliary domain that is different from the target domain of the recommender. Our approach improves performance both in the joint case, in which the auxiliary and target domains share users, and in the disjoint case, in which they do not. Although the 'Free Lunch' enhancement does not apply equally well to any given domain or domain combination, our overall conclusion is that Factorization Machines present an opportunity to exploit information that is ubiquitously present, but commonly under-appreciated by Collaborative Filtering algorithms.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

## General Terms

Algorithms, Performance, Experimentation

## 1. INTRODUCTION

Factorization Machines (FMs) [8] are general models that factorize user-item collaborative data into real-valued feature vectors. FMs have recently attracted the attention of the recommender system community because of the ease and effectiveness with which they can integrate information from external sources, for example, context information [9]. However, until now, focus of FM approaches in the recommender system community has been on integrating new sources of information. In this paper, we investigate the potential of FMs to help us make the most of information that we already have. Because our approach makes use of information that is already present in the user-item matrix, it has the feel of delivering 'something for nothing.' In this spirit, we call our approach 'Free Lunch' enhancement of Factorization Machines.

The specific 'Free Lunch' effect that we focus on in this paper arises from known, but under-appreciated information inherent in the user-item matrix. In general, conventional Collaborative Filtering (CF) approaches, including memory-based and model-based methods, exploit similarities that are based on sets of rated items. The information in user-item matrixes can, however, be repackaged to create other representations of items and users. Specifically, here, we investigate one such repackaging that views users and items in terms of their overall rating patterns. Overall rating patterns are expressed as rating histograms, which are 'trivial' in the sense that they require simple aggregation of information inherent in the user-item matrix. However, clustering these histograms creates user and item categories that constitute new representations of information that is not otherwise exploited by matrix factorization. In this paper, we introduce an approach that uses Factorization Machines to integrate these category labels into a Collaborative Filtering algorithms capable of improving recommendation performance. The approach reveals its full potential when used to exploit information not only from the user-item matrix of the target domain, but also information from the user-item matrix of auxiliary domains. In this respect it constitutes a simple, yet effective, method for Cross-Domain Collaborative Filtering.

## 2. BACKGROUND AND MOTIVATION

The 'Free Lunch' approach does not literally achieve something for nothing. Rather, as mentioned above, it takes advantage of an underexploited representation of users and items as rating histograms. A rating histogram encodes the normalized frequency with which a user assigns a certain rating or an item is assigned a certain rating. Upon first consideration, rating histograms appear to destroy the very connections between users and items that provide the basis for CF. Worse, they capture bias in user rating habits, e.g., the general tendency of individual users to assign very high or very low scores. It is exactly this bias that many similarity metrics seek to avoid. However, seen from a different perspective, rating habits have the ability to capture something about the 'rating style' of individual users or the 'style' of users who rate specific items. Ultimately, the success of our approach attests to the benefit of taking this perspective.

The perspective is supported by evidence from the literature that similarity of rating style is connected to similarity in taste in populations of consumers. For example, Tan and Neetessine [10] presents a study of the Netflix Prize data set that reveals patterns such as a


*RecSys'14,* October 6–10, 2014, Foster City, Silicon Valley, CA, USA.

ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645771.

tendency of people who give higher ratings to watch mainstream movies. In other words, users who are similarly "generous" or "stingy" with high ratings when rating movies, may actually have a tendency to prefer the same kind of movies. The basic insight of our approach is that such relationships already exist implicitly in the user item matrix, but are not exploited by conventional matrix factorization approaches. Note that we are not making a claim that rating histograms are optimally suitable for comparing users. Our 'Free Lunch' enhancement still stands to benefit from a weak indicator of similarity. Our position is if we can stand to benefit from this information, we should seize the opportunity. Note that using normalized rating histograms to represent users means that it is possible to calculate a similarity between two users who have not necessarily rated the same items. In cases where recommender systems must confront extremely sparse data conditions, weak indicators of similarity could prove to be particularly useful.

Use of clustering techniques in CF stretches back into early history of recommender systems, and includes many variants. In the earliest work, clustering was used to reduce dimensionality, focusing on standard representations of users as vectors of rated items. Also, clustering has been used as a way to integrate content into CF to create hybrid recommendation algorithms. For example, in Li and Kim's approach [5], content-based information about items is used to create item clusters, which are then represented in a cluster-based matrix to which CF approaches are applied on. The clustering in our work differs from previous uses of clustering for CF in that it relies on neither standard representations, nor or additional information about items or users, but rather uses new representations of existing information in order to create clusters.

Clustering has proven particularly useful in Cross-Domain Collaborative Filtering (CDCF) algorithms, algorithms that exploit one domain to make recommendations in another. Li et al. [4] proposed a clustering approach known as *codebook transfer* for CDCF. Their model tries to find similar rating patterns in different domains, and then transfer cluster-level patterns to improve recommendation in a target domain. The approach requires additional factorizations on auxiliary domains which makes it computationally more complex compared to a typical matrix factorization model. Our 'Free Lunch' enhancement approach also uses domain transfer based on clusters. However, where Li et al. [4] use conventional representations of users and items and co-clustering, our approach represents users and items in terms of rating histograms. Our approach incurs little computational burden with clustering. Rather, in 'Free Lunch' Enhancement, clustering is a pre-processing step, since it is performed off-line on individual domains and is not part of building the model.

Our previous work [7], established that FMs achieve state-of-the-art performance in CDCF, and can be straightforwardly applied in the case of joint domains, i.e., cases in which the target domain and the auxiliary domain do not share the same users. That work did not use clusters, but rather exploited individual ratings. In this work, we take FM performance as our baseline, and investigate how it can be improved by exploiting user and item clusters. Our proposed approach also applies to disjoint domains, i.e., domains that do not share users.

# 3. ENHANCEMENT APPROACH FOR FMS

This section provides the necessary background on FMs and explains the details of our 'Free Lunch' enhancement approach, that creates clusters using rating histograms for integration into FMs.

## 3.1 Factorization Machines

In contrast to typical Factorization techniques in which the interaction of users and items are represented by a matrix, in Factorization Machines [8] the interaction of a user and an item (i.e., when a user rates an item) is represented by a feature vector. To understand how FMs work, let us assume that the data of a rating prediction problem is represented by a set $S$ of tuples $(\mathbf{x}, y)$ where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is an $n$-dimensional feature vector representing user-item interaction and $y$ is the rating value. Factorization machines model all interactions between features using factorized interaction parameters. The interactions can be between a pair of features, or it can even be between larger number of features. In this work, we adopted an FM model with order 2 where only the interactions between pairs of features are taken into account. This model can be represented as follows:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{j=1}^{n} w_j x_j + \sum_{j=1}^{n} \sum_{j'=j+1}^{n} w_{j,j'} x_j x_{j'} \qquad (1)$$

where $w_j$ are model parameters and $w_{j,j'}$ are factorized interaction parameters and are defined as $w_{j,j'} = \mathbf{v}_j.\mathbf{v}_{j'}$ where $\mathbf{v}_j$ is $k$-dimensional factorized vector for feature $j$. For an FM with $n$ as the dimensionality of feature vectors and $k$ as the dimensionality of factorization, the model parameters that need to be learnt are $\Theta = \{w_0, w_1, \ldots, w_n, v_{1,1}, \ldots, v_{n,k}\}$.

The parameters of the model can be learnt by three different learning algorithms [8]: Stochastic Gradient Descent (SGD), Alternating Least-Squares (ALS) and Markov Chain Monte Carlo (MCMC) method. MCMC learning method proved fastest in our exploratory work and were adopted for our experiments.

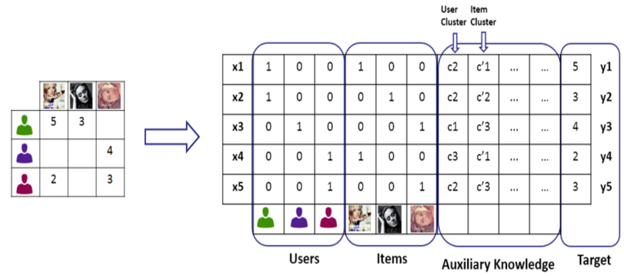


**Figure 1: Encoding user and item clusters into user-item feature vectors.**

## 3.2 Cluster Encoding

Factorization machines require the user-item interactions to be represented by a feature vector. This characteristic allows us to incorporate any additional knowledge in terms of real-valued features. We take this advantage of FMs into account and extend the user-item feature vectors with cluster level features which potentially can improve the rating prediction task.

In FMs, a standard rating prediction problem is represented by a target function $y : U \times I \to \mathbb{R}$. We represent each user-item interaction $(u, i) \in U \times I$ with a feature vector $\mathbf{x} \in \mathbb{R}^{|U|+|I|}$ with binary variables indicating which user rated which item. In other words, if user $u$ rated item $i$ the feature vector $\mathbf{x}$ is represented as:

$$\mathbf{x} = (\underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|U|}, \underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{|I|}) \qquad (2)$$

where non-zero elements correspond to user $u$ and item $i$. The feature vector $\mathbf{x}$ can also be represented by its sparse representation $\mathbf{x}(u, i) = \{(u, 1), (i, 1)\}$.

Our cluster encoding algorithm extends the above feature vector by user and item clusters. Therefore, the target function would be

$y : U \times I \times C_u \times C_i \to \mathbb{R}$ where $C_u$ and $C_i$ are user and item cluster spaces. User and item clusters can also be drawn from other, auxiliary domains. In other words, the clusters that users and items belong to in auxiliary domains can also be identified and added to the feature vector. More specifically, we can represent the sparse form of a cluster-enhanced feature vector as follows:

$$\mathbf{x}(u,i) = \{(u,1),(i,1),(c_j(u),1),(c_j(i),1) : j = 1 \ldots m\} \tag{3}$$

where $c_j(u)$ and $c_j(i)$ are user and item cluster ids in domain $j$ and $m$ is the total number of domains.

Figure 1 illustrates our proposed approach to building feature vectors from a user-item matrix. For each rating, a binary feature vector is created in which the corresponding user and item are indicated by 1s and the remaining features are 0s. This feature vector is then extended by additional binary features which specify the user and item cluster in different domains. The rating values are specified as the output of each vector. Using FM we try to predict the output of the test samples.

### 3.3 Cluster Construction

Our clustering approach considers users with similar rating patterns to be similar, assigning them to the same cluster. In the context of our work, rating patterns are based on the histogram of rating values. More specifically a user $u_j$ is represented by the feature vector $u_j = (u_{j1}, \ldots, u_{jp})$ where $p$ is the upper bound of rating values in the dataset domain and $u_{jk}$ is the number of items which are rated with value $k$ by user $u_j$. Similarly, an item $i_j$ can be represented by $i_j = (i_{j1}, \ldots, i_{jp})$. Given the above representations, we calculate user and item clusters using the K-means clustering algorithm. The distance between vectors is calculated based on Euclidean distance. We choose K-means due to its simplicity and efficiency, which distinguishes it from alternative clustering algorithms [1].

For cross-domain scenarios, we create clusters of users and items in auxiliary domains, and for each user and item in the target domain the most similar clusters are identified by measuring the distance of a user or an item to the center of clusters. The most similar clusters in the auxiliary domain are then used to extend the user-item feature vector in Equation 3.

## 4. EXPERIMENTS

In this section we explain the datasets and frameworks that we used and describe the experiments that we did with their results. We further analyze and discuss the results to find out whether the improvements obtained by our method is significant or not.

### 4.1 Datasets and Framework

We conducted our experiments on a dataset from Amazon [3] consisting of four joint domains, Movielens and Epinions datasets, each having a different set of users and items. The four domains in the Amazon dataset are books, music CDs, DVDs and video tapes. The Amazon dataset contains 7,593,243 ratings on a 1-5 scale provided by 1,555,170 users over 548,552 different products including 393,558 books, 103,144 music CDs, 19,828 DVDs and 26,132 VHS tapes. We use the Movielens 1 million dataset, which consists of 1,000,209 ratings on a 1-5 scale, 6,040 users and 3,883 items. The Epinions dataset contains 664,865 ratings also on a 1-5 scale, 49,289 users and 139,737 items. The data is split into a training set (75%) and a test set (25%).

We implemented our approach within our recommendation framework [6] using C#, which is built on top of two open source libraries for recommender systems: MyMediaLite [2], which implements the most common CF approaches, and LibFM [8], which implements FMs algorithms.

### 4.2 Results

We evaluated our 'Free Lunch' enhanced FM approach with respect to three different enhancement scenarios, and a range of different settings for the number of clusters, which were created with K-means clustering. Figures 2 and 3 present the performance of our approach on the Amazon Books and and the Amazon Music datasets in terms of Root Mean Square Error (RMSE). Each graph shows the performance of the dataset under our three scenarios: 1) Target domain clusters: user and item clusters from the same domain (i.e., a single target domain) are used to extend the feature vectors (i.e., Equation (3) with $j = 1$). 2) Target and joint auxiliary domain clusters: user and item clusters from a joint auxiliary domain are used (i.e., Equation (3) with $j > 1$). 3) Target and disjoint auxiliary domain clusters: user and item clusters from a disjoint auxiliary domain are used (i.e., users do not overlap). To show the effectiveness of our approach, we compare the performance of each scenario with a baseline in which no user or item clusters are used. In other words, for our baseline, the user-item interactions are simply represented by feature vectors as described in Equation (2). The results in Figure 2 and 3 demonstrate that our method outperforms the baseline on both datasets and in all three scenarios. The best performance is achieved on joint domains, which is not unexpected since the domains share users, and the auxiliary domain brings additional information about items rated by these users into play. What is more surprising is that clustering applied in the single domain case, or applied in the case of joint domains is able to come very close to the performance in the joint domains. Figures 2 and 3 also reveal that for all three scenarios, the performance also improves as the number of clusters is increased, until it reaches a point where there is no improvement, at $k = 10$.

We investigate the results achieved in the joint domain scenario with a statistical analysis of the results (in terms of RMSE). This analysis reveals that the improvements obtained for the Amazon Books dataset (Figure 2) when extended with clusters from Amazon Music are statistically significant ($p < 0.05$) regardless of the number of clusters used when compared to the results from the baseline (no clusters). However, a similar analysis of the results obtained for the Amazon Music dataset enriched with clusters from Amazon Book (Figure 3) does not point to significance ($p > 0.2$). We note that the performance of the joint and disjoint scenarios on each data set track the performance of the single domain scenario. This fact suggests that a strongly performing target domain is critical for the approach to benefit from joint domains. Strong performance of the target domain could be related to size, here, 400 thousand books vs.100 thousand music CDs.

Next we turn to investigate the 'target domain cluster' scenario involving a single domain in more detail. Table 1, summarizes the results from the Amazon Books and Amazon Music data set, and also reports further results calculated on the Movielens and Epinions datasets (both RMSE and MAE evaluation metrics). The performance on the different data sets is reported for $k$=10 clusters, the best condition determined in the previous experiment, and compared with the no-clustering baseline. These results confirm that the performance improvement delivered by 'Free Lunch' enhancement is not limited to Amazon data sets, but is achieved on other data sets as well. The relative improvement on the Amazon Book dataset (12% in terms of RMSE) is the largest. The relative greater improvement on the Amazon data set may be attributable to the fact that it is sparser than the Movielens or Epinions sets.

As the results in Table 1 show, the best improvement is achieved on the Amazon Book dataset (12% in terms of RMSE), while there

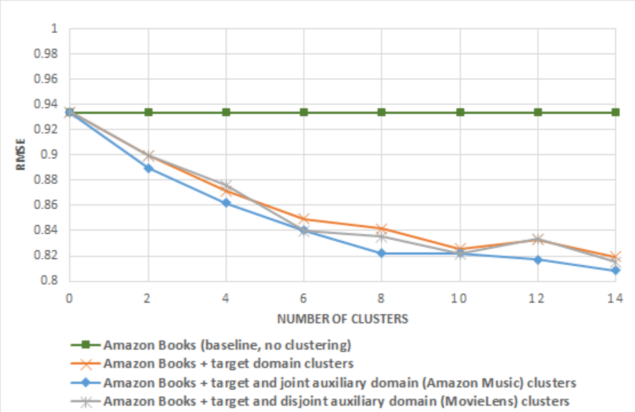


**Figure 2: Performance of our proposed method on Amazon Books dataset based on different number of clusters and different enhancement scenarios. The baseline (no clusters) is represented by the straight line.**

are less dramatic improvements on the Movielens and Epinions datasets. This can be due to the fact these datasets are denser compared to the Amazon dataset, implying that the underlying similarities between users and items are already well captured, and clustering information is of less importance. 'Free Lunch' enhancement can apparently contribute more to the rating prediction task in cases where datasets are sparse, meaning that additional representations of the information in the user-item matrix have more to contribute.

**Table 1: Comparison of our cluster-enhanced approach with the no-cluster baseline**

| | No clusters | | 10 clusters | |
|---|---|---|---|---|
| **Dataset** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| Amazon Books | 0.933 | 0.841 | 0.825 | 0.778 |
| Amazon Music | 0.984 | 0.861 | 0.945 | 0.832 |
| Movielens | 0.936 | 0.862 | 0.890 | 0.838 |
| Epinions | 1.082 | 0.920 | 1.024 | 0.891 |

## 5. CONCLUSION AND OUTLOOK

We have presented a 'Free Lunch' enhancement approach that makes use of the ability of Factorization Machines to easily and effectively integrate additional information. Our approach demonstrates that it is not necessary to turn to outside resources to find additional information useful for improving Factorization Machines, but instead we can make better use of information already at hand. The larger message is that FM approaches that do not first attempt to maximize the benefit they can derive from information at hand, i.e., 'Free Lunch', are missing opportunities.

In this paper, we have focused on clusters formed using rating histograms. We show that this information can be used to improve rating prediction in cases where only a single, target domain is available, or in cases where an auxiliary domain (joint or disjoint with the target domain) is also available. Our future work will also investigate other possible types of 'Free Lunch' information. The list of sources that we are interested in ranges from popularity information to random assignment of users to user-groups. Further, we will work to make more detailed understanding of what makes a domain particularly a suitable target domain for 'Free Lunch' enhancement, and how to best identify useful auxiliary domains.

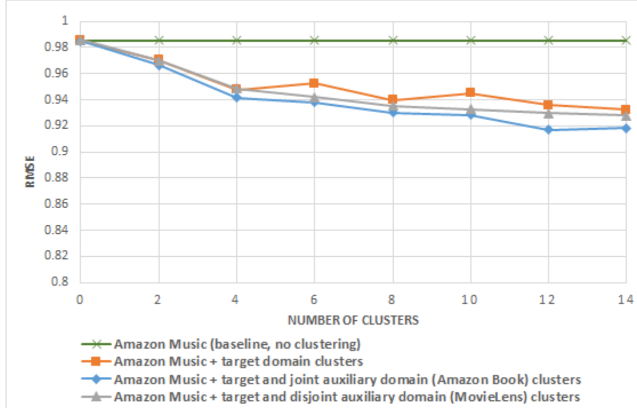


**Figure 3: Performance of our proposed method on Amazon Music dataset based on different number of clusters and different enhancement scenarios. The baseline (no clusters) is represented by the straight line.**

## Acknowledgment

This research is supported by funding from two European Commission's 7th Framework Program projects under grant agreements no. 610594 (CrowdRec) and no. 601166 (PHENICX).

## 6. REFERENCES

[1] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol. Data Mining Methods for Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 2, pages 39–71. Springer US, 2011.

[2] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: a free recommender system library. RecSys '11. ACM, 2011.

[3] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[4] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction. IJCAI'09, 2009.

[5] Q. Li and B. M. Kim. Clustering approach for hybrid recommender system. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 33–38, Oct 2003.

[6] B. Loni and A. Said. Wraprec: An easy extension of recommender system libraries. In *Proceedings of 8th ACM International Conference of Recommender Systems*, RecSys '14, 2014.

[7] B. Loni, Y. Shi, M. Larson, and A. Hanjalic. Cross-domain collaborative filtering with factorization machines. In *Proceedings of the 36th European Conference on Information Retrieval*, ECIR '14, 2014.

[8] S. Rendle. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3), May 2012.

[9] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. SIGIR '11. ACM, 2011.

[10] T. Tan and S. Neetessine. Is Tom Cruise threatend? using Netflix Prize data to examine the long tail of electronic commerce. Technical report, University of Pennsylvania, Wharton Business School, 2009.