# A hybrid recommender combining user, item and interaction data

Jens Grivolla, Toni Badia
*Universitat Pompeu Fabra
Department of Translation and Language Sciences
Barcelona, Spain
Email: {jens.grivolla, toni.badia}@upf.edu

Diego Campo, Miquel Sonsona, Jose-Miguel Pulido
†Novaventus
Barcelona, Spain
Email: {dcampo, msonsona, jpulido}@nova-ventus.com

***Abstract*—While collaborative filtering often yields very good recommendation results, in many real-world recommendation scenarios cold-start and data sparseness remain important problems. This paper presents a hybrid recommender system that integrates user demographics and item characteristics, around a collaborative filtering core based on user-item interactions.**

**The recommender system is evaluated on Movielens data (including genre information and user data) as well as real-world data from a discount coupon provider. We show that the inclusion of additional item and user information can have great impact on recommendation quality, especially in settings where little interaction data is available.**

***Keywords*—Recommender systems, Natural language processing, Machine learning applications, Information mining and applications**

## I. Introduction

Recommender systems are nowadays used in a large variety of application setting, ranging from online stores, music and movie recommendation, to social media recommender and many more. Each of these applications has its particular characteristics, with greatly differing temporal dynamics or volatility, amounts of available data, use of explicit or implicit indicators, etc.

The domain of online discount offers is particular in that it has much greater turnover than traditional stores, as new offers appear daily and old offers expire fast. It also suffers from great turnover in users, with few users making a large amount of repeat purchases, as opposed to e.g. music recommendation where usually a large amount of preference indication is available for each user.

Due to these characteristics, coupon (or discount offer) recommendation is particularly sensitive to cold start problems and needs to be able to recommend new offers with no or very little previous interaction data, and be able to connect with new users as soon as they start using the system. It is therefore a prime candidate for recommender systems that go beyond collaborative filtering and try to use all available inputs to make the best possible recommendations.

## II. A hybrid ALS-WR recommender

Our core recommender system combines collaborative filtering (that uses the coupon usage history) with additional data sources in order to improve recommendation quality, especially for new users and new offers.

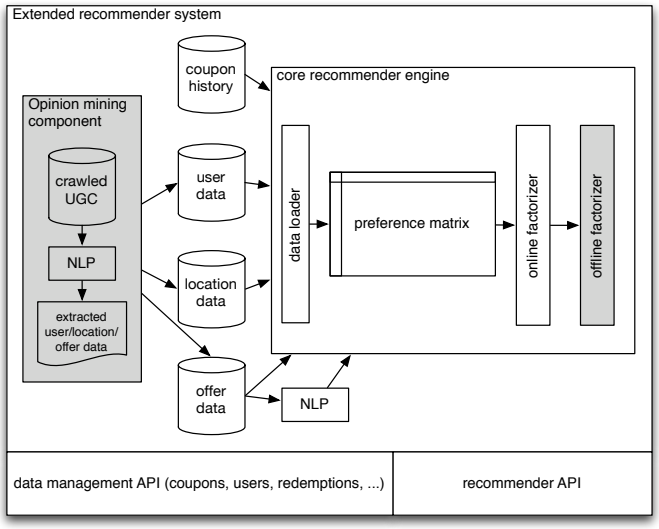


Fig. 1. recommender system architecture

It follows an integrated approach, treating user demographics, offer information, and usage history at the same level. The overall architecture can be seen in Figure 1.

### A. Background

The recommender uses a matrix factorization approach to predict unknown preference associations from a sparse (incomplete) matrix of known associations.

A matrix $R$ can be decomposed, e.g. via Singular Value Decomposition (SVD), into matrices $U$, $S$, and $V$, such that the product of those matrices is equal to the original matrix. It can also be decomposed into smaller (truncated) matrices $U_k$, $S_k$ and $V_k$, such that the product of these is a good approximation of the original matrix.

By optimizing the decomposition such that the reconstructed matrix $R_k$ is a good approximation of $R$ for those elements that have known values while ignoring the blank values, this method has been successfully applied for recommender systems, and has gained great popularity in the context of the Netflix prize [1]. Unknown values are then estimated by

reconstructing the full matrix from its truncated decomposition, thus filling in the blanks with values that best fit with a good approximation of the previously known parts of the matrix.

We use an implementation of ALS-WR provided by Myrrix [2], which is designed to deal with implicit feedback data, as opposed to explicit ratings. As stated on the official Myrrix page:

> Myrrix uses a modified version of an Alternating Least Squares algorithm to factor matrices. The essential elements of this approach are explained in, among other resources:
>
> - ”Collaborative Filtering for Implicit Feedback Datasets” by Hu, Koren and Volinsky [3]
> - ”Large-scale Parallel Collaborative Filtering for the Netflix Prize” by Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan [4]
> - Alex Smola’s Scalable Machine Learning course notes, Lecture 8, Section 2 from slide 34 in particular. [5]

However, like all collaborative filtering methods, such an approach using only user-item preference data has major downsides when dealing with new users or items that are not sufficiently connected in the known data (cold start problem).

### B. Hybrid recommendation

The cold start problem arises when a user or a content item does not have sufficient historical data known to the system (or none at all), which makes it impossible to recommend content for new users or to recommend new offers. A common approach is to use content based or hybrid recommendation, combining content based recommendation with collaborative filtering. However, the hybrid approaches most commonly only integrate information about the recommended items, which can be modelled as a similarity metric between items and thus integrates easily in an item-based recommender model. Those approaches are therefore useful to deal with cold start for new content, but do not help with recommendations for new users. Other recommenders use user neighbourhoods (user based recommendation) and thus can easily integrate demographic information, but few systems can include both user and item information in combination with interaction data.

While the Cold Start problem is a major problem with collaborative filtering in general, it is a particularly important issue in a discount offer setting due to high turnover in users consuming coupons and the constant introduction of new offers. Even for existing and registered users, usage data is often very limited (dormant users) and being able to provide recommendations for those users is a strong business case. Our system therefore integrates additional data about users and offers, in order to match new items to existing ones without relying purely on preference data.

Additional information about users and offers is used to match them to existing users or offers. This includes structured metadata (e.g. demographics, geo-location, or product category) as well as unstructured textual data (e.g. offer descriptions, etc.). Natural language processing techniques are used to extract relevant information from the unstructured text.

| | item1 | item2 | item2 | lives in BCN? | age 20-25? |
|---|---|---|---|---|---|
| user1 | 1 | ? | 1 | 1 | 1 |
| user2 | ? | 1 | ? | 1 | 0 |
| user3 | ? | 1 | 1 | 0 | 1 |
| “cosmetics” | 0 | 1 | 1 | | |
| “facial cream” | 0 | 1 | 0 | | |
| “beer” | 1 | 0 | 0 | | |
| “diapers” | 0 | 0 | 0 | | |

Fig. 2. Enriched preference matrix

An interesting approach that allows to integrate both user and item information is the use of machine learning to learn a projection from user- or item-features directly into the feature space used by an SVD-recommender as proposed by Almosallam and Alkanhal [6]. We opted for a simpler alternative, also using a matrix factorization (or SVD) recommender, that maps demographics, content features, and other information directly onto user or item preference vectors, making that information an integral part in the entire transformation process. We thus extended the user-item preference matrix by integrating demographic data as well as product / item information, as seen in Figure 2.

The recommender uses user-item associations from the coupon redemption history. Different levels of user-item association strength can be defined based on different types of interaction (viewing, acquisition, redemption, etc.). Additionally, demographic data about registered users, or similar meta-data about locations in the case of recommendations for public displays are incorporated. Information about offers is also used, including textual offer descriptions as well as more specific information extracted automatically from those texts using Natural Language Processing techniques and semantic analysis.

The matrix elements are thus not anymore considered strictly ”preferences” but are rather ”associations” that allow to obtain a richer profile for users as well as items. It is important to note that our approach is made possible by this recent framework that defines the user–item matrix not as a rating matrix but rather as a matrix of associations, where each user is defined through their associations with content items, and inversely each item is defined through its association with users. It is thus sound to extend users’ and items’ representations by incorporating additional associations, which would not be possible when seen as a matrix of ratings.

### C. Implementation

Taking into account the necessity of scalability and ease of integration with existing platforms, the recommender system leverages highly scalable frameworks for recommendation, indexing, search, and NLP. While initially built on Mahout [7], the direct usage of Mahout has been substituted by its derivative Myrrix. Myrrix is a recent development, launched in April 2012 by Sean Owen, the original developer of the recommender framework (taste) within Mahout. It is based on matrix factorization and optimized for fast online recommendations,

while providing the possibility to offload the computationally expensive calculations to an offline task that is based on the distributed computing framework Hadoop. While the online aspects of Myrrix are fully Open Source, the offline part is available as pay-per-use on Amazon's Elastic MapReduce.

The natural language processing and semantic analysis component is based on Apache UIMA [8] and uses low-level linguistic processing such as sentence splitting, tokenization, part-of-speech tagging, and lemmatization , as well as higher-level techniques such as noun phrase chunking, Named Entity Recognition, or specific information extraction techniques. Available open-source analysis modules, such as those provided by the OpenNLP project [9], as well as proprietary modules, are used in conjunction with open or proprietary language resources, and adapted to the application's needs.

The semantic analysis component provides information (e.g. additional metadata extracted from plain text) that is used as input to the recommendation process. UIMA is called directly from Solr [10] using SolrUIMA when adding documents (i.e. user or offer information) to the database and the added information is thus available as input to the recommender matrix as well as for explicit filtering using Solr queries.

All domain specific aspects are completely isolated from the matrix factorisation component, going through a layer that converts user and item information into association data that can be processes by Myrrix's domain-agnostic ALS-WR implementation.

### D. Feature generation and natural language processing

One key aspect in adding rich features to better describe users and particularly items is the automatic extraction of relevant labels from unstructured text.

The straightforward approach would be to consider all words of the item descriptions as input features for the recommender (in a bag-of-word fashion), however this would introduce much noise through spurious associations between items based on non-descriptive words used e.g. in the offer descriptions. One main objective of the natural language processing is thus select the most relevant features from the large and noisy space of possible labels. Another use is to facilitate association between different content items that may not share the same vocabulary in their description, by introducing higher-level features that assign semantically meaningful categories to the items.

The main approaches we used to reach these objectives are the use of noun phrases instead of single words, and the automatic tagging of items with domain-specific categories.

*1) Noun phrase detection:* While many of the words used in an offer description are not representative of the product that is offered, much of the relevant information is found in so-called *noun phrases*, which are basically adjective-noun combinations. Using noun phrases to represent the offer instead of using all words directly thus eliminates much of the noise. At the same time noun phrases such as "facial cream" or "solar protection" are more specific than just using individual nouns or adjectives.

*2) Automatic tagging:* In order to better relate items that may not be presented using the same words in the offer description, and in particular to obtain a similarity measure that best reflects domain-specific aspects of importance, we also developed a system for automatically tagging items with domain-specific categories, in the case of the coupon recommendation setting parapharmaceutical product categories.

We used a collection of 2416 product descriptions, categorised into 97 coarse and 207 fine-grained categories. For each noun phrase, adjective, or noun encountered in the collection we calculated the point-wise mutual information (PMI) between the occurrence of that term and each of the categories.

$$pmi(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)}$$

where $x$ means "the given term occurs in the document" and $y$ means "the document pertains to the given category".

We thus obtained a dictionary of term-category associations that we used to label new incoming items with the categories (and their weights) that were most strongly related to the terms found in the description.

## III. Datasets

### A. Coupon recommendation

The main dataset of interest is the actual production data from a discount offer distribution website, representing the real-world domain data that is the focus of this work.

This data includes user profiles, offer descriptions and coupon redemption data gathered over the last few months:

- 79035 users
- 1843 offers
- 156935 user-offer pairs (coupon acquisition and/or redemption)
- 922174 additional user-offer interactions (clicks) from 877549 distinct user-item pairs

As can be seen from these numbers, the data is very sparse, with a very low number of actual coupon acquisitions/redemptions per user.

Most users have very few (or no) interactions with items (offers), making it very difficult for a pure collaborative filtering system to make good recommendations, or even make any recommendations at all for a large number of users. Less than 40% of users have made at least one purchase, and only 50% of those have made more than 2 purchases. Around 20% of active users made more than 5 purchases (not counting dormant users that have made no purchase at all), or less than 10% of total users.

### B. Movielens

For better comparability with other existing recommender systems, it is useful to evaluate performance on a widely accepted established dataset. The Movielens dataset is one of

the most widely used for evaluation of recommender system performance, and one of the few that besides user-item interactions for collaborative filtering also provides content features and demographic data.

The dataset of interest is the "MovieLens 1M" dataset, consisting of 1 million ratings from 6040 users on 3883 movies. This is the largest available dataset that also includes demographics (the 10M dataset doesn't). In order to avoid issues of how to interpret low ratings, and to avoid problems when selecting a holdout "gold-standard" set and resampling data, we limited this work to only the 4 and 5 star ratings. These can be seen as unambiguously positive, and account for 575,281 ratings, or almost 58% of the total.

The characteristics of this dataset are quite different from our domain data, with a much higher density of user-item interactions. Our main challenge is recommendation in a context of extreme interaction sparsity, and as such we proceeded to produce variations of the Movielens dataset with varying degrees of sparsity (eliminating user-item interactions from the training data).

Another difference is the fact that Movielens uses explicit feedback in the form of ratings, whereas in many use cases such as ours the available feedback is implicit and not on a rating scale. Therefore evaluation is done using precision-recall measures, treating highly rated items as relevant for recommendation.

## IV. Experiments and results

Given the boolean nature of preference indicators (redeemed coupon or not), typical evaluation measures such as RMSE are not applicable to evaluate the quality of recommendation results. Automatic evaluation is therefore currently limited to information retrieval based measures, using precision and recall.

To that end, for each user a subset of preferences are eliminated from the data set and a recommendation model is trained on the reduced data. Recommendation results are then compared to the formerly extracted preferences to see if the corresponding offers appear in the recommendation list. Considering those offers as equivalent to relevant documents in the information retrieval sense, it is then possible to calculate metrics such as "precision @N" or mean average precision.

Automatic evaluation is done using a holdout strategy by which a number of "known good" recommendations, i.e. items associated positively in the original dataset, are removed from that dataset and the model is trained on the remaining data. Recommendations provided by the system are then compared to the list of "known good" items for each user, thus obtaining precision and recall measures.

It is a known limitation of this approach that the user's preference for unrated items is not actually known for evaluation, but those items are considered "not relevant" when calculating the precision. As such, the precision measured in the evaluation is typically much below the actual precision of the system, as many unknown items may in fact be good recommendations. This limitation applies even more to the calculation of recall as it is impossible to even estimate the number of potential good recommendations in the dataset.

While from a practical standpoint the most relevant measure of quality is the precision of the (few) top items returned by the system, the number of items considered for evaluation needs to be augmented significantly due to the aforementioned limitations.

### A. Protocol

For each dataset, a random subset of 1000 users having at least 20 item interactions (ratings or purchases) was selected. For each of these users, a random subset of 20 ratings or purchases was removed from the dataset and put in the holdout set used as reference for evaluation (in the case of Movielens only 4 or 5 star ratings were considered).

In the case of Movielens we additionally created randomly subsampled data sets based on the remaining data, creating four differently sized sets:

- the full 1 million ratings (1M)
- the 4 and 5 star ratings (500k)
- a 50k subsample of the 500k set (50k)
- a 10k subsample of the 500k set (10k)

Different configurations of the recommender system were then queried for each of the 1000 users and the holdout set compared to the top 20 or 50 recommendations, calculating the intersection (precision@20 and precision@50).

### B. Coupon recommendation

The different configurations used on the discount offer data set were combinations of the following:

- using click data or only purchases
- using item information (manually supplied categories and features extracted using NLP from offer descriptions)
- user information (gender and age)

When using click data, it was included in the model with a lower weight than purchase data. User gender was used as a simple binary association with each gender, and age was converted to binary associations with five age ranges surrounding the actual age (sliding window). Offer data was included in the form of one category and one subcategory per offer (although around 10% of the offers had no categories), as well as noun phrases extracted from the offer description.

The results are presented in table I. Precision numbers are given as average size of intersection (with a theoretical maximum of 20, the size of the holdout set) rather than percentages.

### C. Movielens

Using the Movielens data, for each of the datasets (subsamples), we compared using only the interaction (rating) data vs. including user and movie information. User information included:

- age range

| clicks | item | user | P@20 | P@50 |
|---|---|---|---|---|
| x | x | x | 3.121 | 5.61 |
| x | x | | 3.173 | 5.618 |
| x | | x | 3.167 | 5.641 |
| x | | | 3.125 | 5.577 |
| | x | x | 2.469 | 4.343 |
| | x | | 2.426 | 4.158 |
| | | x | 2.623 | 4.514 |
| | | | 2.414 | 4.267 |

TABLE I. COUPON RECOMMENDATION RESULTS

| | P@20 | | P@50 | |
|---|---|---|---|---|
| dataset | CF | hybrid | CF | hybrid |
| 1M | 5.814 | 5.849 | 9.417 | 9.508 |
| 500k | 4.834 | 4.968 | 8.005 | 8.205 |
| 50k | 1.12 | 1.295 | 2.221 | 2.85 |
| 10k | 0.493 | 0.86 | 1.033 | 2.192 |

TABLE II. MOVIELENS RECOMMENDATION RESULTS

- gender
- occupation

And movie information included:

- year
- genres

Each movie could have several assigned genres, and the year was converted to overlapping year ranges (as done with user ages for coupon recommendation) to model proximity, avoiding arbitrary boundaries such as using a movie's decade of production. Users' ages were already discretised into ranges in the original data, and occupation was codified into 21 categories. As for the coupon recommendation, gender had two possible values, 'M' and 'F'.

The results are presented in table II. Precision numbers are again given as average size of intersection (with a theoretical maximum of 20, the size of the holdout set) rather than percentages.

## V. CONCLUSIONS AND FUTURE WORK

The results, particularly on the Movielens dataset, show that user and item information can have an important positive impact on recommender performance. This effect can be very strong when little interaction data is available, with diminishing returns when greater amounts of collaborative filtering data are available. On the 10k Movielens dataset, using user and content information can double the recommender precision, whereas the effect is almost negligible on the full 1M data. It is interesting that including the low ratings in the recommender model improves results considerably over using only the high ratings. Note that no ratings were considered "negative", but rather all treated as positive associations (of varying weight).

On the discount offer data the use of click data in addition to purchase data has a very important positive impact on recommendation quality. The inclusion of additional user or offer information, however, has little impact. The inclusion of user information appears to have a slight positive impact, but adding offer information does not appear to contribute positively. More work is needed to compare different sets of content based features to determine which information can be valuable while avoiding introducing noise through uninformative features.

The positive results of using a hybrid system on the very sparse Movielens datasets (10k and 50k) suggest that such a system can be helpful when faced with lack of interaction data for new users or content. We are currently evaluating our system in a production environment through A/B tests to determine the real-world impact that is difficult to fully assess through artificial evaluations.

Building on these results, the EUMSSI project (FP7-ICT-2013.4.1, n° 611057) will focus on developing a content aggregation and recommendation system for journalists and media consumers, incorporating automatic analysis of different media modalities (text, audio, video) with social feedback information.

## ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme managed by REA-Research Executive Agency http://ec.europa.eu/research/rea ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement n° [262451], as part of the dico(re)$^2$s project, and was conducted at Barcelona Media.

## REFERENCES

[1] J. Bennett and S. Lanning, "The netflix prize," in *Proceedings of KDD Cup and Workshop*, vol. 2007, 2007, p. 35. [Online]. Available: http://reference.kfupm.edu.sa/content/n/e/the_netflix_prize_67297.pdf

[2] S. Owen, "Myrrix | a complete, real-time, scalable recommender system, built on apache mahout," http://myrrix.com/, 2013. [Online]. Available: http://myrrix.com/

[3] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 263–272. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4781121

[4] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," *Algorithmic Aspects in Information and Management*, pp. 337–348, 2008. [Online]. Available: http://www.springerlink.com/index/j1076u0h14586183.pdf

[5] A. Smola, "SML: recommender systems," http://alex.smola.org/teaching/berkeley2012/recommender.html, 2012. [Online]. Available: http://alex.smola.org/teaching/berkeley2012/recommender.html

[6] I. Almosallam and M. Alkanhal, "Speech rating system through space mapping," in *2011 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, vol. 1, 2011, pp. 31–35.

[7] ASF, "Apache Mahout: Scalable machine learning and data mining," http://mahout.apache.org/, 2013. [Online]. Available: http://mahout.apache.org/

[8] ——, "Apache UIMA," http://uima.apache.org/, 2013. [Online]. Available: http://uima.apache.org/

[9] ——, "Apache OpenNLP," http://opennlp.apache.org/, 2013. [Online]. Available: http://opennlp.apache.org/

[10] ——, "Apache Lucene - Apache Solr," http://lucene.apache.org/solr/, 2013. [Online]. Available: http://lucene.apache.org/solr/