

Unifying Explicit and Implicit Feedback for Collaborative Filtering

Nathan N. Liu¹ Evan W. Xiang¹ Min Zhao² Qiang Yang¹

¹Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong, China
{nliu,wxiang,qyang}@cse.ust.hk

²NEC Labs China
Beijing, China
zhaomin@research.nec.com.cn

ABSTRACT

Most collaborative filtering algorithms are based on certain statistical models of user interests built from either explicit feedback (eg: ratings, votes) or implicit feedback (eg: clicks, purchases). Explicit feedbacks are more precise but more difficult to collect from users while implicit feedbacks are much easier to collect though less accurate in reflecting user preferences. In the existing literature, separate models have been developed for either of these two forms of user feedbacks due to their heterogeneous representation. However in most real world recommender systems both explicit and implicit user feedback are abundant and could potentially complement each other. It is desirable to be able to unify these two heterogeneous forms of user feedback in order to generate more accurate recommendations. In this work, we developed matrix factorization models that can be trained from explicit and implicit feedback simultaneously. Experimental results on multiple datasets showed that our algorithm could effectively combine these two forms of heterogeneous user feedback to improve recommendation quality.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Collaborative Filtering, Ranking, Latent Class Model

1. INTRODUCTION

Collaborative filtering(CF) is a popular approach for building recommender systems. It works by collecting *user feedback*, which could generally refer to any form of user actions on items that may convey relevance judgements on the items. User feedback data

generally fall into two categories: explicit feedback and implicit feedback. Explicit feedback is more widely used and are often in the form of numeric ratings input by users to express their preference over the items. For example, Amazon let users rate books on a 1-5 star scale. Explicit feedback, like its name suggests, could capture user preferences in a direct and granular way. However, it requires a user to perform an extra "rating" action, which may lead to inconveniences. Given the huge volume of digital content that people normally interact with on a daily basis, it would be rather burdensome if one had to enter a rating for every item he liked or hated. Therefore, in recent years, CF researchers have started looking into inferring user preference from the much more easily available *implicit feedback*, which consists of user behaviors that indirectly express user interests. For example, an online store may record what products a user has purchased and an internet radio could track which songs a user has listened to. Other forms of implicit feedback include search history, browse history, mouse movement and even eye tracking. Implicit feedback mostly involves user actions that would naturally occur as a result of using a particular system and thus could be more easily collected. However, implicit feedback is much less accurate and granular than explicit feedback. For example, a user may rent a movie and only realizes it is a bad movie after watching it, so just knowing a user has rented a movie does not necessarily imply that he likes the movie.

We can see that there is clearly a trade off between quantity and quality when comparing explicit and implicit feedbacks. Explicit feedback is higher in quality but lower in quantity while implicit feedback is lower in quality but more abundant in quantity, so the two forms of feedback are indeed complementary to each other. However, the vast majority of existing works have solely considered the use of explicit feedback whereas only recently had several algorithms specifically designed for implicit feedback been developed. To the best of our knowledge, very few works have studied the simultaneous use of both explicit and implicit feedback, although in most real world recommender systems, both explicit and implicit feedback can be collected.

In this paper, we develop collaborative filtering models that can be simultaneously learnt from both explicit and implicit feedback regarding the same set of users and items. The key challenge in unifying explicit and implicit feedback for collaborative lies in coping with the heterogeneous representations of these two forms of user feedback. Explicit feedback data are mostly numeric responses whereas implicit feedback data mostly consists of binary responses. As a result, the accuracy and granularity of these two forms of user feedback are highly distinct. Explicit feedback can more accurately reflect user preference with its magnitude precisely indicating the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...\$10.00.

degree to which a user prefers an item. On the other hand, implicit feedback are much less accurate and granular. To unify implicit and explicit feedback, we have developed a matrix factorization model called *co-rating* that can effectively cope with the heterogeneity between the two forms of feedback. Experimental results on multiple datasets demonstrated the effectiveness of the proposed approach.

2. EXPLICIT AND IMPLICIT FEEDBACK

2.1 Formalization

The notational framework is defined as follows. We use $U = \{u_1, u_2, \dots, u_m\}$ to denote the set of m users in the system and $I = \{i_1, i_2, \dots, i_n\}$ to denote the set of n items. Both explicit and implicit feedback are observable interactions between U and I that could reflect a user's preferences on the items.

Explicit feedback is usually in the form of numeric ratings assigned to items by the users, the scale of which indicate the how much the user likes a particular item. We use $x_{ui} \in \mathbb{R}$ to denote user u 's rating on item i and let $S^* \subseteq U \times I$ denote the set of user-item pairs for which explicit feedback are available. We use $X \in \mathbb{R}^{m \times n}$ to denote the matrix of observed ratings.

Types of implicit feedback typically include various actions that users performed on items that are automatically tracked by the systems. Examples of implicit feedback include purchases at an online store, clicks on a news Website, or plays on an online radio. In the most general case, implicit feedback can be captured binary variable $y_{ui} \in \{-1, +1\}$ such that if an user u had acted upon an item i (purchased a book) then y_{ui} is equal to +1 and -1 otherwise. We let $Y \in \{-1, +1\}^{m \times n}$ to denote the matrix of observed implicit feedback and use $S^+ \subseteq U \times I$ and $S^- \subseteq U \times I$ denote the set of user-item pairs for which y_{ui} are equal to -1 and +1 respectively. Unlike explicit feedback, y_{ui} 's are weak and ambiguous indicators of user preferences. For example, a user may not really like a song or a news story he just accidentally played or read and similarly, it would be possible for him to actually like a song or a news story he has not played or read because he is not aware of such items. Without further information, it would be very hard to discern such ambiguities in implicit feedbacks and most existing works[3, 6] are based on the assumption that items for which $y_{ui} = +1$ are in general more relevant to the user than those items for which $y_{ui} = -1$.

There are two important differences between explicit and implicit feedbacks:

1. Given explicit feedbacks, x_{ui} are only known for those $(u, i) \in R$ whereas given implicit feedbacks, the values of y_{ui} are known for all $(u, i) \in U \times I$.
2. Explicit feedbacks are numeric in the nature and the magnitude of the ratings x_{ui} are correlated with user preferences whereas for implicit feedbacks, the variables y_{ui} are indeed categorical responses.

For convenience, we use the I_u^* to refer to the set of items i on which user u has explicit feedback (i.e. $(u, i) \in S^*$). Similar, we define the I_u^+ and I_u^- to be the sets of items for which $y_{ui} = +1$ and -1 respectively. Symmetrically, we may define the user sets U_i^* , U_i^+ and U_i^- that are associated with a particular item i .

2.2 The Co-Rating Model

2.2.1 Matrix Factorization

Most collaborative filtering algorithms are based on designing either a parametric or a nonparametric form of a scoring function

$f(u, i)$ that determines each item's relevance to each user. The rating oriented approaches is essentially regression modeling in that the parameters of the function $f(u, i)$ are fitted so that the predicted scores $\hat{r}_{ui} = f(u, i)$ would match the observed user feedback x_{ui} 's and y_{ui} 's as closely as possible. Given the scores \hat{r}_{ui} , one can then rank items by the scores to produce recommendations. Such a rating oriented approach has been studied for collaborative filtering using explicit feedback[7, 5, 8, 2] and has recently been extended to cope with implicit feedback as well[6, 3].

In recent years, matrix factorization [5] models has gained popularity due to its widely publicized success on the Netflix Challenge. In addition to modeling ratings data, its effectiveness has also been demonstrated on implicit feedback datasets[6, 3]. With matrix factorization models, a matrix $X: m \times n$ of observations is approximated by the product of two low-rank matrices $U \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{k \times n}$:

$$X \approx U^T \cdot V \quad (1)$$

Under this model, each user u is modeled by a latent feature vector \mathbf{v}_u , the u -th column of V and similarly each item i is represented by \mathbf{u}_i , the i -th column of U . The prediction formula for the scores \hat{r}_{ui} thus have the following form:

$$\hat{r}_{ui} = \mathbf{u}_i^T \cdot \mathbf{v}_u = \sum_{f=1}^k u_{if} \cdot v_{uf} \quad (2)$$

Typically, the factors matrices U and V are found via solving:

$$\arg \min_{U, V} \mathcal{L}(U, V) + \lambda \cdot \mathcal{R}(U, V) \quad (3)$$

The loss function $\mathcal{L}(\cdot, \cdot)$ quantifies the how well the approximation is, the regularization term $\mathcal{R}(\cdot, \cdot)$ penalizes model complexity in order to avoid overfitting and the parameter λ controls the trade off between these two terms.

For explicit feedbacks, we want the factorization model to approximate well the observed ratings in S^R . This leads to the widely used squared error loss form for \mathcal{L}_X :

$$\mathcal{L}_E(U, V) = \sum_{(u, i) \in S^*} (x_{ui} - \mathbf{u}_i^T \mathbf{v}_u)^2 \quad (4)$$

where the squared loss are only computed over the observed ratings in S^* in contrast to standard SVD formulation which would treat the unobserved ratings as 0 and approximate these zero entries along with the observed ratings. Other forms of loss functions such as hinge loss that are more suitable for the ordinal ratings have also been considered[8].

A commonly used regularization term is the Frobenius norm:

$$\mathcal{R}(U, V) = \|U\|_F^2 + \|V\|_F^2 \quad (5)$$

which is adopted by many matrix factorization based collaborative filtering algorithm[5, 8] due to its smooth differentiable property.

A similar squared error based loss for implicit feedback had also been proposed [6, 3] by approximating the two types of feedback -1 and +1 as numerical values:

$$\mathcal{L}_I(U, V) = \sum_{u=1}^m \sum_{i=1}^n (y_{ui} - \mathbf{u}_i^T \mathbf{v}_u)^2 \quad (6)$$

where the major difference from (6) is that rather than just summing over entries in S^* , there is now a square loss over all the $m \times n$ entries. This raises a major computational challenge in terms of efficiency. Fortunately, it is shown that by exploiting special structure of the hessian matrix in the Newton's algorithm, efficient algorithms with complexity per iteration in the order of $O(|S^+|)$ rather than $O(m \times n)$ is achievable[3].

2.2.2 Unifying Explicit and Implicit Feedback

When both explicit and implicit feedbacks are available with respect to the same set of users and items, we would like the predicted scores \hat{r}_{ui} to reflect user preferences expressed in both explicit and implicit feedbacks. A natural idea is to apply a common factorization model $U^T \cdot V$ for both the explicit feedback matrix X and the implicit feedback matrix Y so that the same \hat{r}_{ui} is used to fit both x_{ui} and y_{ui} . However, there are two complications: firstly, the numeric scales of x_{ui} and y_{ui} can be very different, for example, x_{ui} may range from 1 to 10 while y_{ui} only takes on either -1 or +1. secondly, the accuracy of explicit and implicit feedback are highly variant with implicit feedbacks being much less precise. To address these two problems, we normalize the original x_{ui} and y_{ui} value to a common scale ranging from 0 to 1 and assign different importance to explicit and implicit feedback. This leads to the following co-rating model:

$$\arg \min_{U, V} \mathcal{L}_E(U, V) + \eta \cdot \mathcal{L}_I(U, V) + \lambda \cdot \mathcal{R}(U, V) \quad (7)$$

where the square losses due to explicit and implicit feedbacks are added together and an additional parameter η is introduced to weigh the relative importance of explicit and implicit feedbacks. The formulation (7) can be viewed as a special case of the more general collective matrix factorization model[9], that involves two relations about two entities.

In this paper, we extend the alternating least squares (ALS) algorithm for factorizing a single matrix[10, 3] for learning the co-rating model. To this end, we first fix U and set the gradient of the objective function in (7) with respect to \mathbf{v}_u to zero, which results in the following expression for the \mathbf{v}_u that minimizes the loss:

$$\mathbf{v}_u = (U_{I_u^*}^T U_{I_u^+}^T + \eta \cdot U^T U + \lambda I)^{-1} (U_{I_u^*}^T \cdot \mathbf{x}_{:u} + \eta \cdot U_{I_u^+}^T \cdot \mathbf{y}_{:u}^+) \quad (8)$$

where $U_{I_u^*}$ and $U_{I_u^+}$ denote sub-matrices of U formed by columns of U in I_u^* and I_u^+ respectively, $\mathbf{x}_{:u}$ is a $|I_u^*|$ dimensional vector formed by ratings of u on items in I_u^* , and $\mathbf{y}_{:u}^+$ is a $|I_u^+|$ dimensional vector of 1's. Symmetrically, we can derive the update equation for item factors by fixing V and minimize with respect to each \mathbf{u}_i , which is omitted due to space limitation.

2.3 Related Work

The idea of complementing explicit feedback with implicit feedback was first proposed in [1], where the author consider explicit feedback as how the users rated the movies and implicit feedback as what movies are rated by the users. The two forms of feedback are combined via a factorized neighborhood model, an extension of traditional nearest item-based model in which the item-item similarity matrix is approximated via low rank factorization. In particular, the ratings are predicted via the following formula:

$$\hat{r}_{ui} = a_u + b_i + \mathbf{u}_i^T \left(|I_u^*|^{-1} \sum_{j \in I_u^*} (x_{uj} - a_u - b_j) \mathbf{p}_j + |I_u^+|^{-1} \sum_{j \in I_u^+} \mathbf{q}_j \right) \quad (9)$$

where a_u, b_i are user and item rating biases and each item i is represented by 3 sets of latent factors $\mathbf{u}_i, \mathbf{p}_i, \mathbf{q}_i$. Like traditional matrix factorization, the model parameters are learnt by minimizing prediction errors on the set of observed ratings.

3. EXPERIMENTS

3.1 Data Description

Our goal is to study the if recommendations can be made more accurate when both explicit and implicit feedback are present. To the best of our knowledge, there is no publicly available datasets

containing both forms of feedback such as movie ratings data plus movie rentals records. Here we follow the idea proposed in [1] to create implicit feedback from explicit ratings data by considering "whether" a movie is rated by a user in addition to "how" a movie is rated by a user. Thus, for each $(u, i) \in S^*$, we would have the corresponding $y_{ui} = +1$, and the set S^* would correspond exactly to the set S^+ . So the implicit feedback can be regarded as the log of the "rating" action indicting which user rated which movies. It should be noted that the implicit feedback created this way are rather noisy as high ratings and low ratings become indistinguishable and that ratings lower than 3 actually accounted for nearly 30% ratings in both the Netflix and EachMovie datasets. As presence of noise is also an inherent nature of most other forms of implicit feedback such as click logs and browse logs, so we believe the findings on the implicit feedback transformed from ratings should be transferrable to other forms of implicit feedback.

In this work, we conducted the evaluation on movie ratings datasets including both Netflix¹ and MovieLens². On both datasets, we utilize 80% of the ratings in training and evaluate performances on the remaining 20% of the ratings. When doing the training/testing split, we ensured the consistency between explicit and implicit feedback by making $y_{ui} = -1$ for all the x_{ui} that were used for testing.

3.2 Evaluation Metric

We evaluate recommendation quality based on both rating and ranking quality, the former of which is measured by Residual Mean Squared Error (RMSE) while the later is measured by Normalized Discounted Cumulative Gain (NDCG). Both RMSE and NDCG measures are computed on a subset \tilde{S}^* of the observed ratings. In choosing this kind of evaluation, we are essentially assuming that we already know which items a user would choose to rate and are only interested in evaluating how these items would rated. In real world recommender systems, the top- k recommendations need to be selected from a pool consisting of all items not in the training set. So it is desirable for the top- k recommendations to have more items with high ratings and contain less items that are rated low or not rated at all.

Toward this end, we designed an evaluation framework that more realistically simulate the top- k recommendation scenario. For each user in the test set, both rated and unrated items are ranked altogether to produce the top- k list. Note some of the randomly selected unrated items might actually be of interest to the user, though it is safe to assume most of the unrated items are probably of no interest to the user at all. We still use NDCG to measure the quality of the ranked list except that the utilities of rated and unrated items in the list are defined differently: (1) utilities of items rated 4 or above are equal to their ratings. (2) utilities of both unrated items and items rated lower than 4 are equal to 0. We refer to this version of NDCG that could account for both rated and unrated items as NDCG+.

3.3 Effect of Combining Explicit and Implicit Feedback

We compared the co-rating and co-ranking models with the following baselines. The first baseline is the matrix factorization model(MF) designed for explicit feedback[10]. The second baseline is the binary matrix factorization (BINMF) designed for implicit feedback[3]. The third baseline is the factorized neighborhood model (FN)[4] mentioned in section 2.3, which, to the best of our knowl-

¹<http://www.netflixprize.com>

²<http://www.grouplens.org/node/73>

Table 1: Performance comparisons between algorithms when 30% explicit feedback

| | Netflix | | | | | MovieLens | | | | |
|-----------|---------|--------|--------|---------|---------|-----------|--------|--------|---------|---------|
| | RMSE | NDCG10 | NDCG20 | NDCG+10 | NDCG+20 | RMSE | NDCG10 | NDCG20 | NDCG+10 | NDCG+20 |
| MF | 0.9062 | 0.8015 | 0.8384 | 0.7019 | 0.7497 | 0.8759 | 0.7769 | 0.8271 | 0.6606 | 0.7254 |
| BINMF | - | 0.7171 | 0.7703 | 0.6864 | 0.7387 | - | 0.6994 | 0.7640 | 0.6744 | 0.7382 |
| FN | 0.9032 | 0.8022 | 0.8391 | 0.7034 | 0.7507 | 0.8697 | 0.7779 | 0.8281 | 0.6657 | 0.7293 |
| Co-Rating | 0.9048 | 0.7950 | 0.8335 | 0.7378 | 0.7691 | 0.8810 | 0.7727 | 0.8237 | 0.8256 | 0.8687 |

Table 2: Performance comparisons between algorithms with 100% explicit feedback

| | Netflix | | | | | MovieLens | | | | |
|-----------|---------|--------|--------|---------|---------|-----------|--------|--------|---------|---------|
| | RMSE | NDCG10 | NDCG20 | NDCG+10 | NDCG+20 | RMSE | NDCG10 | NDCG20 | NDCG+10 | NDCG+20 |
| MF | 0.8724 | 0.8183 | 0.8521 | 0.7645 | 0.8010 | 0.8340 | 0.7973 | 0.8427 | 0.7290 | 0.7797 |
| BINMF | - | 0.7170 | 0.7701 | 0.6864 | 0.7387 | - | 0.6995 | 0.7640 | 0.6744 | 0.7382 |
| FN | 0.8678 | 0.8196 | 0.8531 | 0.7770 | 0.8114 | 0.8170 | 0.8022 | 0.8467 | 0.7432 | 0.7902 |
| Co-Rating | 0.8503 | 0.8241 | 0.8570 | 0.7332 | 0.7656 | 0.8152 | 0.8062 | 0.8498 | 0.7435 | 0.7853 |

edge, is the only existing model that combines explicit and implicit feedback.

We consider the following two setting when comparing different models. The first setting is when only 30% of the total ratings in the training set are used as explicit feedback during training. This corresponds to the scenario when explicit feedback is sparse while implicit feedback is much more abundant. In the second setting, all of the ratings in the training set are used as explicit feedback.

From the results summarized in Table 1 and 2, we can make the following observations:

- The Co-Rating model could generally led to significant improvement over NDCG+, although its performance in terms of RMSE and NDCG measured only on the observed ratings were not always better than the baselines.
- The factorized neighborhood model (FN) outperformed the MF models in most cases, which proved the benefit of exploiting implicit feedback. However, as we have argued in section 2.3, the FN model only attempt to approximate the observed ratings and does not model preferences expressed in implicit feedback. As a result, its performance on the NDCG+ measure was much worse than Co-Rating.
- Recommending items based on just implicit feedback using BINMF tended to produce inferior performance in all cases, which could be due to the inherent noisy and ambiguous nature of implicit feedback. On the other hand, co-ranking can significantly outperform BINMF, which confirmed our belief that implicit feedback could indeed complement explicit feedback.

4. CONCLUSION AND FUTURE WORK

In this paper, we have developed collaborative filtering models that can be learnt from explicit and implicit feedback simultaneously, namely the co-rating and the co-ranking models. In particular, we have shown that the co-ranking model based on pairwise preferences provide a highly effective framework for unifying explicit and implicit feedback. Experiments on both the Netflix and MovieLens datasets demonstrated consistent improvements over baseline methods. In the future, we would like to conduct further evaluation of our model on more diverse forms of user feedbacks such as click logs, browse history etc., in order to study the effectiveness of the co-ranking in a broader application context.

5. REFERENCES

- [1] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM*, pages 43–52, 2007.
- [2] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.
- [3] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM'08*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.
- [4] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1):1–24, 2010.
- [5] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [6] R. Pan and M. Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676, New York, NY, USA, 2009. ACM.
- [7] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In *Proc. of UAI*, pages 473–480, 2000.
- [8] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML '05*, pages 713–719, New York, NY, USA, 2005. ACM.
- [9] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658, New York, NY, USA, 2008. ACM.
- [10] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *AAIM '08: Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management*, pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.