Modelling Users' Preference Using Three-Staged Coarse-To-Fine Transfer To Rank For E-Commerce And Social Networking Applications

Toby Subhash[1], Dr. Smita C Thomas[2]
[1, 2] Dept of computer science and engineering
[1, 2] Mount Zion College Kadammanitta

Abstract- *Recommendation systems play a vital role in e-commerce and social networking applications. They aid in capturing user interest and increase the overall user interaction of the application. Suggesting recommendations which aligns with the user's interest increases the revenue of an e-commerce site as the user tends to buy more items if presented with choices that reflect his/her interests. When a user receives recommendations and feeds that caters to his taste, it makes him use the application more. Through this paper , we aims to explore popular recommendation systems currently in use & to dive into the use of Coarse-to-fine transfer to rank (CoFiToR) for user preference modelling*

Keywords- E-Commerce , Social networking, Recommendation system, user preference modelling.

I. INTRODUCTION

Content discovery and Information retrieval applications like E-commerce and social network, rely on personalised recommendations. They enable users to navigate through the huge content using querying and browsing. These applications use recommendation algorithms to generate recommended item lists for their user's based on user interaction data. The user interaction data includes the user's selected items, items rated by the user, items viewed by the user etc. These algorithms helps to filter and select interesting and useful information from a large volume of information

Collaborative filtering [1] is a recommendation methodology used to model user preferences by considering the user interaction with the items in form of views ratings & selection. Unlike content-based approaches, CF can recommend items without additional computational expense or copyright issues involved with the direct processing of items.

This paper explores various recommendation systems in usage & analyses a flexible model based collaborative filtering framework, coarse to fine transfer to rank (CoFiToR) [2] which uses a 3-staged transfer learning framework to model users' preferences from coarse to a fine granularity.

II. TYPES OF RECOMMENDATION TECHNIQUES

Recommendation techniques can be classified based on various parameters. Specifically, they are characterized by

(i) Background data - The information in the system before the recommendation process ,
(ii) Input data- The information collected from user in order to generate a recommendation, and
(iii) An algorithm - Combines i and ii to generate suggestions.[3]

Based on these, the recommendation systems can be characterized as below:

Table 1 : Characterization of recommendation systems

Technique	Background	Input	Process
Collaborative	Ratings from user	User rating	Identify similar users to the user under consideration, and extrapolate from their ratings.
Content-based	Features of items	User rating	Generate a classifier that fits the user's rating behavior and use it for the current item.
Demographic	Demographic information about users and their ratings of items	Demographic information about users	Identify users that are demographically similar to the current user, and extrapolate from their ratings.
Utility-based	Features of items	A utility function over the set of items that describes the user's preferences.	Apply the function to the items and determine each item's rank.
Knowledge-based	Features of items &. Knowledge on how these items meet user needs	A description of the user's needs or interests.	Infer a match between each item & user's need.

Hybrid Recommender Systems [3] : To overcome the drawbacks of any individual recommendation technique, two or more techniques can be combined to create a hybrid recommender system. Below are some of the hybrid recommender systems generally used:

Table 2 : Types of hybrid recommendation systems

Hybridization method	Description
Weighted	The scores of different recommendation techniques are combined together to produce a single recommendation.
Switching	The system dynamically switches between recommendation techniques based on the current scenario.
Mixed	Recommendations from several different recommenders are combined
Feature combination	Features from different data sources used for recommendation are combined & fed as input to a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is fed as an input feature to another.
Meta-level	The learned model of one recommender is used as input to another.

III. TOP-N RECOMMENDATION

Given n users and m items, the top-N recommendation system returns a personalised and sorted candidate list containing N items unrated by a user by exploiting the observed rating records R[4].

Classification of Top-N recommendation systems:

a. Memory-Based systems:

Memory based systems predict users' ratings for unrated items using regression, assuming like-minded users in the past usually have similar interests and tastes in the future. Types of memory based approach are:

User-based approaches - focuses on constructing neighbours of the users with similar interests Item-based approaches - construct neighbours of the items with similar properties

Hybrid methods - Combination of User-based approaches & Item-based approaches.

b. Model-Based systems

Model-based CF methods learn the latent semantic representation of users and items in order to predict a user's preference for an item. Memory-based systems are simple but have less scalability. Model-based CF methods, on the contrary, are complex but can be easily implemented.

IV. EXISTING SYSTEMS

A. Amazon.com Recommendations using Item-to-Item Collaborative Filtering

To create a recommendation algorithm that uses input about a customer's interests to generate recommended items list, amazon uses item to item collaborative filtering [5]. The main issues addressed by this approach are:

a. Large data

A large retailer might have huge amounts of data, customers and distinct catalog items.

b. Requires real time results

Results need to be generated in no more than half a second, while still producing high-quality recommendations.

c. Recommendation generation for new customer & older customers are challenging

As new customers have only a few purchases or product ratings, only limited information is available about them. Older customers can have a lot of information, based on thousands of purchases and ratings.

d. Volatile customer data

Each interaction however minute has valuable customer data, and the algorithm must respond quickly to new information.

To find the most-similar match for a given item, the algorithm builds a table of similar-items by finding items that customers tend to purchase together. Algorithm for calculating the similarity between a single product and products related to it:

For each item in product catalogue, I1
For each customer C who purchased I1
For each item I2 purchased by customer C
Record that a customer purchased I1 and I2
For each item I2

Compute the similarity between I1 and I2

The similarity can be found out using:

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

Where each vector represents an an item and the vector's M dimensions With the increase in the number of data correspond to customers who have purchased that item.

Time complexity of the similarity algorithm: $O(N^2M)$

Practically it will be O(NM) as most customers have very few Purchases.

Advantage of the proposed system:

a. The algorithm creates the expensive similar-items table offline. The algorithm's online component (looking up similar items based on the user's purchases and ratings) scales independently of the catalog size or the total number of customers; it is only dependent on the number of titles the user has purchased or rated. Thus, the algorithm is fast even if the data set is extremely large.
b. Because the algorithm recommends highly correlated similar items, recommendation quality is excellent.
c. Unlike traditional collaborative filtering, the algorithm also performs well with limited user data, producing high-quality recommendations based on as few as two or three items.

Disadvantage of the proposed system are:

a. It requires huge computational power
b. Not generalised, It is designed specifically for e-commerce, it cannot be applied to other applications such as social networking or content sharing sites.

B. The YouTube Video Recommendation System

The YouTube video recommendation system delivers personalised sets of videos to signed in users based on their previous activity on the YouTube site [6].

Challenges addressed:

a. Videos uploaded often have no or very poor metadata.
b. Volume of video is large
c. Most of the videos are small (less than 10 minutes), so user interactions are short & noisy
d. Many of the interesting videos on YouTube have a short life cycle going from upload to viral in the order of days requiring constant freshness of recommendation.

Type of data considered:

1) content data,

Includes Raw video streams , Video metadata such as title, description, etc

2) user activity data

Can be divided into

a.Explicit data

Includes Rating a video Favoriting/liking a video, Subscribing to an uploader.

b. Implicit data

Data generated when users watch and interact with videos

Ranking:

The candidate videos are scored and ranked using a variety of signals, which can be broadly categorized into three groups corresponding to three different stages of ranking:

1) video quality - signals that are used to judge the likelihood that the video will be appreciated irrespective of the user (view count, the ratings of the video, commenting, favouring and sharing activity around the video, and upload time)
2) user specificity - signals that are used to boost videos that are closely matched with a user's unique taste and preferences.(properties of the seed video in the user's watch history, such as view count and time of watch is considered)
3) diversification - Since a user generally has interest in multiple different topics at differing times, videos that are too similar to each other are removed at this stage to further increase diversity.

System Implementation is done in 3 steps:

1) data collection - The raw data signals are initially deposited into YouTube's logs. These logs are processed to extract signals and stored on a BigTable for each user.
2) recommendation generation - Recommendations are generated by MapReduces computations that

analyzes video of the user graph to generate and score recommendations

3) recommendation serving - Done by simplified read-only Bigtable servers to YouTube's web servers

V. BEHAVIOR RANKING USING TRANSFER LEARNING [7]

Behavior Ranking (BR)

Input:

Examinations E in the form of (user, item) pairs.
Ratings R (user, item, rating).

Goal:

Generate a personalized ranked list of items for each user u from the set of items that user u has not seen before, i.e., $I \backslash (R_u \cup E_u)$, where E_u and R_u denotes the set of examined items and the set of rated items of user u, respectively.

Challenges:

The sparsity challenge.

Users' rating behavior are usually few due to the fact of users' unwillingness to provide such feedback to a system, which makes the learning problem rather difficult.

The scalability challenge.

Users' examination behavior are usually of large volume as they are recorded implicitly by a deployed system without users' proactive involvement, which requires to be modeled in an efficient way.

Proposed solution:

Transfer to Rank (ToR): a global preference learning task and a local preference learning task with candidate lists of items as shared knowledge[7].

Advantages of Proposed Solution

To identify a candidate list of items, a global preference learning is conducted, which is based on the abundant examination behavior. The candidate list is refined by conducting local preference learning using the sparse rating information. Thus, the sparsity challenge is addressed in a sequential manner.

The generic learning algorithm ToR is instantiated and efficient pairwise and pointwise recommendation algorithms are chosen for modeling the examination behavior and rating behavior, respectively. In this way, the scalability issue of modeling users' examination and rating behavior is addressed separately.

Bayesian Personalised Ranking (BPR) [8]

In BPR, the examination behaviour of the user are modeled using pairwise preference assumption, (a user u prefers an examined item i to an unexamined one j)

$$\text{Pref}(u, i \mid \Phi) > \text{Pref}(u, j \mid \Phi), (u, i) \in E, (u, j) \notin E \quad (1)$$

where Pref(u, i) and Pref(u, j) denote the unobserved preference scores of user u on item i and item j, respectively, and Φ is the model parameters to be learned.

BPR is effective and efficient for positive-only feedback scenarios such as users' examinations of items BPR will be used as a base learner in the global preference learning task.

Probabilistic Matrix Factorization (PMF) [9]

In PMF, the rating behavior of users are analysed pointwise i.e., ratings are treated one at a time and the rating score is approximated by some latent variables

$$\text{Loss}(\text{Rating}(u, i) - \text{Pref}(u, i \mid \Psi)), (u, i) \in R \quad (2)$$

where Rating (u, i) and Pref (u, i | Ψ) are the observed user u's rating assigned to item i, and the predicted preference score with model parameters Ψ, respectively.

PMF is the state-of-the-art method for rating prediction in various reported studies.

PMF will be used as a base learner in our local preference learning task.

Objective Function

The behavior ranking problem with both examinations E and ratings R can be formulated as maximizing the following objective function[6],

$$\text{Prob}(E, R \mid \Theta) \quad (3)$$

where Θ denotes the model parameters that govern the generation of the two types of user behavior.

The problem is difficult to solve due to the correlation of the two types of user behavior.

Dependent Preference Assumption

Dependent preference assumption, i.e., a user's rating behavior follows a user's examination behavior.

A user's rated items are from the set of items that have been examined by the user, which means that a user's rating behavior is dependent on a user's examination behavior.

It is actually a common practice adopted by most e-commerce and entertainment sites that a user can rate an item only if he/she has examined the item.The dependency between heterogeneous user behavior can be expressed as follows,

$$\text{Prob}(\text{examination} \mid (\text{user}, \text{item})) \cdot \text{Prob}(\text{rating} \mid \text{examination}, (\text{user}, \text{item})) \quad (4)$$

the probability of an assigned rating to a (user, item) pair is not only dependent on the preference of the user to the item, but is also dependent on whether the item will be examined by the user.

The term Prob (rating | examination,(user, item)) denotes that an observed rating is not only dependent on a (user, item) pair but also on the abundant examination information - Fixes the sparsity problem.

In order to obtain an efficient preference learning solution, the preference assumption in Eq.(4) is simplified and converted to two sequential and dependent tasks inspired by a user's online behavior of rating after examination:

$$\text{Prob}(\text{examination} \mid (\text{user}, \text{item})) \rightarrow \text{Prob}(\text{rating} \mid \text{examination}, (\text{user}, \text{item})) \quad (5)$$

where the notation "→" means that there is a sequential ordering between the left part and right part.

The LHS represents a global collaborative filtering which deals with whether an item will be examined by a user or not.

The right part denotes a local collaborative filtering aiming to estimate the preference score a user will assign to an item if he/she has examined it.

The condition of examination of a user to an item in the right part of Eq.(5) denotes the connection and dependency between the global preference learning and local preference learning, which is a knowledge bridge between the two tasks of preference learning in transfer learning.For this reason, the solution is called transfer to rank (ToR).

Global Preference Learning

In order to find some candidate items from I\(Eu ∪ Ru), examined items and rated items of each user are combined.

The probability Prob(examination|(user, item)) is instantiated as:

$$\text{Prob}(E \cup ER \mid \Phi) \Rightarrow K \quad (6)$$

ER denotes the (user, item) pairs converted from the rating behavior by removing the rating scores, Φ is the set of parameters that models the generation of the amalgamated examination behavior E ∪ ER.

The symbol "⇒" indicates the generation of candidate lists of items with the learned model in the global preference learning task.

The candidate list of items is generated using both the examined items and rated items, i.e., the information from both examination behavior and rating behavior. So the task is called global preference learning.

Local Preference Learning

In order to locally refine the candidate list of items for each user u, the rating behavior R of the user is used to further estimate the preference scores of the items on the candidate list.

Prob(rating|examination,(user, item)) can be instantiated as:

$$\text{Prob}(R \mid E, \Psi) \approx \text{Prob}(R \mid K, \Psi) \quad (7)$$

where the likely-to-be-examined candidate lists of items (i.e., K) from the global preference learning task are used to replace the examination behavior E, and Ψ is the set of model parameters used to govern the generation of the rating behavior R. The candidate lists K denote the transferred knowledge from the global preference learning task to the local preference learning task in the transfer learning view.

Learning the ToR

We can instantiate the generic transfer to rank algorithm can with base learners for examination behavior

and rating behavior, respectively, i.e., BPR(ER ∪ E) and PMF(R).

BPR ($E_R \cup E$) as a background model for global preference Learning: - whether a user will examine an item.

PMF(R) as a model to refine local preference learning - how will a user like an item after examination.

The algorithm of transfer to rank (ToR) for behavior ranking (BR).

Input: Users' examination behavior E and rating behavior R.

Output: A personalized ranked list of items for each user.

Task 1. Conduct global preference learning (i.e., BPR), and recommend a candidate list.

Task 2. Refine candidate list by conducting local preference learning (PMF)

VI. COARSE TO FINE TRANSFER TO RANK FOR USER PREFERENCE MODELLING (CoFiToR)

Coarse-to-fine transfer to rank(CoFiToR) is a novel and generic transfer learning based recommendation framework that can be used to model users' behaviour by simulating users' shopping processes

The aspects of user's shopping behaviors considered are:

(i) whether user will examine an item
(ii) how an item will be scored by a user, and
(iii) whether the user will purchase the item.

CoFiToR is a model-based CF framework. CoFiToR adopts a pointwise approach and a pairwise approach in a progressive way.

In comparison with other ranking-oriented CF algorithms, CoFiToR has the merits of both pointwise CF methods and pairwise CF methods, and their complementarity by knowledge sharing and transfer

The proposed system is a three-staged framework which progressively models users' preferences from a coarse granularity to a fine granularity.

The process is composed of 3 stages

Each stage represents one step of a user's shopping process and answers one specific question.

The characteristics of sequentiality and progressiveness make the framework more adaptive and easy to be deployed and maintained in real applications

Stages:

1. E-stage

Determines whether the item will be examined by the user

2. S-stag

Determines how the item will be scored by the user.

3. P-stage

Determines whether the item will be purchased by the user.

Algorithm:

E-Stage

Input : Examinations E
Output : Candidate list : LE

1. Optimize the objective function
2. Estimate the likelihood of examination of the item with respect to the user
3. Construct LE from top NE candidate items of each user

S-Stage

Input : Scores S, candidate list LE
Output: Candidate list LS

1.Optimize the objective function
2.Estimate the score of items with respect to the user
3.Construct LS via top NS candidate items of each user

P-stage

Input: Purchases P, candidate list LS
Output: Candidate list LP

1.Optimize the objective function
2.Estimate the purchasing likelihood of item by the user
3.Construct LP via top NP candidate items of each user

VII. CONCLUSION

In this paper, we studied various recommendation systems that can be used for e-commerce websites. In particular, a novel and generic transfer learning based recommendation algorithm, i.e., coarse-to-fine transfer to rank was explored, in which The user's shopping process is reverse engineered & decomposed from examination to pre-purchase score to final purchase, to to create three sequential stages

different views of the explicit user feedback, i.e., examinations, scores and purchases.

In comparison with existing recommendation approaches, the coarse-to-fine transfer to rank approach accurately captures users' preference information in a coarse-to fine manners and efficiently models users' shopping processes. The candidate list of items retrieved from the examination stage can be transferred to subsequent stages to be progressively refined via the scores and purchases. The 3-staged solution is able to gain significantly better performance than the two-staged solution using transfer to rank (ToR) .

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992

[2] Wei Dai, Qing Zhang, Weike Pan and Zhong Ming "Transfer to Rank for Top-N Recommendation " IEEE TRANSACTIONS ON BIG DATA

[3] R. Burke, "Hybrid recommender systems: Survey and experi-ments,"User Modeling and User-Adapted Interaction, vol. 12, no. 4,pp. 331–370, 2002

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th International Conference on World Wide Web, ser. WWW '01, 2001, pp. 285–295.

[5] 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.

[6] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston et al., "The youtube video recommendation system," in Proceedings of the 4th ACM Conference on Recommender Systems, ser. RecSys '10, 2010, pp. 293–296.

[7] W. Pan, Q. Yang, Y. Duan, B. Tan, and Z. Ming, "Transfer learning for behavior ranking," ACM Transactions on Intelligent Systems and Technology, vol. 8, no. 5, pp. 65:1–65:23, 2017.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, ser. UAI '09, 2009, pp. 452–461

[9] S. Balakrishnan and S. Chopra, "Collaborative ranking," in Proceedings of the 5th ACM International Conference on Web Search and Data Mining, ser. WSDM '12, 2012, pp. 143–152

[10] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.