

The Million Domain Challenge: Broadcast Email Prioritization by Cross-domain Recommendation*

Beidou Wang^{#†}, Martin Ester[†], Yikang Liao[‡], Jiajun Bu[#], Yu Zhu[‡], Ziyu Guan^{*}, Deng Cai[‡],
[#]Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, China
[†] School of Computing Science, Simon Fraser University, Canada
^{*}School of Information and Technology, Northwest University of China
[‡]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
[†]{beidou,ester}@sfu.ca, ^{#‡}{ykliao,bjj,zhuyu_cad,dcai}@zju.edu.cn, ^{*}{ziyuguan}@nwu.edu.cn

ABSTRACT

With email overload becoming a billion-level drag on the economy, personalized email prioritization is of urgent need to help predict the importance level of an email. Despite lots of previous effort on the topic, broadcast email, an important type of emails with its unique challenges and intriguing opportunities, has been overlooked. The most salient opportunity lies in that effective collaborative filtering can be exploited due to thousands of receivers of a typical broadcast email. However, every broadcast email is completely “cold” and it is very costly to obtain users’ preference feedback. Fortunately, there exist up to million-level broadcast mailing lists in a real life email system. Similar mailing lists can provide useful extra information for broadcast email prioritization in a target mailing list. How to mine such useful extra information is a challenging problem that has never been touched. In this work, we propose the first broadcast email prioritization framework considering large numbers of mailing lists by formulating this problem as a cross domain recommendation problem. An optimization framework is proposed to select the optimal set of source domains considering multiple criteria including overlap of users, feedback pattern similarity and coverage of users. Our method is thoroughly evaluated on a real world industrial dataset from Samsung Electronics and is proved highly effective and outperforms all the baselines.

CCS Concepts

•Information systems → Collaborative filtering;

Keywords

Email Prioritization, Cross Domain Recommendation

*This work was partly done when the first author worked as an intern at Samsung Research Canada

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

© 2016 ACM. ISBN 978-1-4503-4232-2/16/08...\$15.00

DOI: <http://dx.doi.org/10.1145/2939672.2939863>

1. INTRODUCTION

Despite 23 years history, email still remains as one of the most important communication tools nowadays, with 2.6 billion users worldwide and over 205 billion emails sent or received everyday[26]. However, together with the blessing comes a curse. Email overload, “a \$650 Billion Drag on the Economy” described by New York Times[20], is causing serious troubles for email users. Based on previous research, 58% of emails are irrelevant or unimportant and a person on average has to waste at least one hour per day to handle them[5][14].

The serious situation of email overload leads to a thriving research field, *personalized email prioritization*, in which importance labels for non-spam emails are predicted and various literature[41][32] have been working on it. One of the most notable applications of personalized email prioritization comes from Gmail. An email importance ranking algorithm[1] is proposed by Google and in the inbox of Gmail, every important email is high-lighted with a yellow marker.

However, *broadcast email*, an important type of email, has been overlooked in the previous personalized email prioritization literature. A *broadcast email* is an email message that is sent to a group of receivers, usually by organizations, companies and web services[38] and each group of receivers is called a *mailing list*. Everyday huge numbers of broadcast emails are sent to millions of mailing lists, often for group notification (e.g. mails from a university graduate student list) and email marketing (e.g. promo mails from an e-commerce website list). Handling these broadcast emails can be both overwhelming and time consuming and the really important and interesting broadcast emails can easily get swamped. The following characteristics of broadcast emails make methods for prioritization of personal emails inapplicable:

The Same Sender In prioritization of personal emails[1][41], one important type of features comes from the interaction between the sender and the receiver. For example, if the receiver has read a large percentage of a sender’s emails, the sender will be labeled as an important contact and his subsequent emails will be predicted as important to the receiver. However, for a broadcast mailing list, there usually only exists one sender (e.g. mailing list admin) and a receiver may get hundreds of different emails with various importance level from the same broadcast sender.

Limited Types of User Feedback Various types of user feedback are exploited by personal email prioritization methods. However, the types of user feedback for broadcast emails are limited. For instance, we usually will not reply to, forward or cc a broadcast email; we will not manually label an irrelevant broadcast mail as spam either, because we still wish to receive other mails from the mailing list. The common user feedback for broadcast emails is viewing.

Despite the above mentioned challenges, broadcast emails also bring a new opportunity. Since each broadcast email is sent to all users of a mailing list, other users' feedback (view or not) can be very helpful in predicting the priority for a target user. In other words, broadcast email prioritization can be formulated as a *collaborative filtering* task: for a user, if other users with similar interest have considered the email important (i.e. viewed it), he should likely also consider it as an important email. However, there exists one key challenge. Each email waiting for priority prediction is completely *cold*. That is to say no viewing action has been observed, since the email has not yet been sent to any users, which makes it impossible to perform collaborative filtering directly [38].

In our recent work [38], an active learning framework was proposed to solve the above mentioned cold start problem of collaborative filtering for broadcast email prioritization. However, the obtained user feedback from active learning is still limited for well addressing the prioritization problem. For example, it cannot well handle new users of a mailing list and new mailing lists, which are very common in real systems and have limited historical data for collaborative filtering. In our previous model from [38], only one mailing list was considered, i.e., each mailing list is modeled independently without considering the existence of other mailing lists. In a mailing system like Gmail, there exist up to millions of various mailing lists, covering different topics varying from political campaigns to e-commerce promos. The size of the mailing lists is typically large, commonly containing thousands or even millions of receivers. A user may be a member of dozens of mailing lists and mailing lists can have large numbers of shared users. The viewing information accumulated in similar mailing lists can be very useful to enrich the collaborative filtering evidences of a target mailing list. By resorting to these similar mailing lists, the aforementioned new user and new mailing list issues could be significantly alleviated.

In this paper, we propose a new cross domain recommendation framework to solve the problem of broadcast email prioritization for many mailing lists. *Cross domain recommendation* systems adopt different techniques to transfer learning from source domains (e.g. book) to target domains (e.g. movie) in order to alleviate the sparsity problem and improve the accuracy of recommendations. The intuition of our approach is that each broadcast mailing list can be regarded as a domain in a cross domain recommendation system. The problem of predicting the priority of emails with the help of extra information from other related mailing lists can thus be formulated as the problem of improving the quality of recommendations in the target domain by incorporating information accumulated from source domains, as in cross-domain recommendation. However, due to the unique characteristics of the broadcast email prioritization task, several challenges exist and make the traditional cross domain recommendation methods fail.

Million Domain Challenge Most previous cross domain recommendation works focused on a relatively small set of domains, like two or three domains[29]. And the selection of source domains is usually done manually based on expert intuition. For instance, book intuitively makes a good source domain for movie because they can share similar sets of genre. However, with millions of domains in an email system, there is no way to rely on intuition or expert to select the optimal set of source domains for each target domain. What to select, how many to select and what makes a good set of source domains are challenging questions that can only be answered by a carefully designed algorithm.

Multi-criteria Source Domain Selection To select the optimal set of source domains, multiple criteria need to be considered and none of previous cross domain recommendation models has worked on it. Moreover, signals from these criteria come in different formats and can be contradictory with each other, which makes it challenging to support them all at the same time.

A Dynamic Source Domain Set Size In the cross-domain recommendation literature, there exists an underlying assumption that the number of source domains used in a cross-domain recommendation is given or fixed. However, this is not the case in our task. For instance, if a mailing list happens to have many similar mailing lists with lots of shared users and similar user feedback patterns, more source mailing lists should be included so that more useful extra information can be included. The algorithm should be able to dynamically decide the number of source domains to be selected.

To address the above mentioned challenges, we formulate the selection of the set of source domains as an optimization problem considering criteria including overlap of users, feedback pattern similarity and coverage of users. Two methods are then proposed to solve the optimization problem efficiently. A weight regularized matrix factorization method is used to make predictions based on information from both the selected source domains and the target domain.

Our main contributions are as follows:

1. We present the first in-depth discussion of personalized prioritization for broadcast emails considering large numbers of mailing lists.
2. We propose the first cross domain recommendation framework that can select the optimal set of source domains from large numbers of domains.
3. Our method is thoroughly evaluated on a real life dataset, and is demonstrated to be highly effective compared to baseline methods.

2. RELATED WORK

2.1 Prioritization for Emails

Email prioritization focuses on making a personalized prediction of the importance label of non-spam emails [41, 15, 11].

Douglas et. al [1] propose a simple linear logistic regression model to do prioritization for gmail, in which the final prediction is the sum of the global model and the user

model log odds. Four categories of features are considered in the model, including social features, content features, thread features and label features. In [41], the authors use personal social networks to capture user groups and to obtain rich features that represent the social roles from the viewpoint of a particular user. They also developed a semi-supervised (transductive) learning algorithm that propagates importance labels from training examples to test examples through a personal email network. In [32], the authors proposed multiple classification and semi-supervised clustering methods for spam detection and email categorization tasks. A social clustering approach is proposed in [15] to predict the email priority based on the relations between its sender and induced social clusters. [24] defines metrics for measuring the social importance of users based on the email elements: from, to and cc, and user actions of replying and reading, which can potentially be used for measuring email prioritization. Horvitz et al. [11] regard the email prioritization prediction task as a classification problem and use Support Vector Machines to predict whether the utility of newly arrived emails is high or low. However, due to the previously described characteristics of broadcast emails, i.e. The Same Sender and Limited Types of feedback, these traditional methods designed for normal emails cannot be effectively applied to the broadcast email prioritization task.

Broadcast email prioritization has been overlooked for a long time. Our recent paper [38] is the first work handling this task and we are the first to use collaborative filtering to handle it. As each email waiting for prioritization is completely cold, we propose an active learning framework. The intuition is that a new email is first sent to a small portion of users from the mailing list. Then based on their feedback obtained during a short waiting period, the priority of the email is predicted for the remaining users. Similar cold-start, unbalanced or insufficient data problems have been widely studied in settings like recommendation [36][27] and classification [34][4][33][35]. Since it is not the main contribution of the paper, to make it easier to follow, instead of previous active learning method proposed in [38], in this paper we use a much simpler strategy. It is also worth noting in our previous work [38], each mailing list is treated independently, disregarding the existence of large numbers of other mailing lists, which limits the performance of the method.

2.2 Cross Domain Recommendation

The basic idea of cross domain recommendation is to improve the quality of recommendations in one domain (known as target domain) by exploiting auxiliary knowledge in other domain (known as source domain)[8]. This technique is particular useful to address the cold-start problem[28] and mitigating the sparsity problem[31]. Also, [39] claims that cross domain recommendations can make the result more diverse which is very important in real world applications[2].

Recent works on cross domain recommendation can roughly be divided into two categories, content-based approaches [3, 7, 17, 30, 31] and collaborative filtering approaches[9, 16, 12, 28]. For the content-based approach, an early attempt is presented in[3], in which authors come up with a general framework for enhancing the accuracy of user modeling by incorporating data collected from other domains. [17] uses location data in a particular kind of context aware recommendation task to improve recommendation qualities. Deep learning is also used in the content-based cross domain ap-

proach, Elkahky et al. propose a method which can map users and items to a latent space where the similarity between users and their preferred items is maximized[7]. [31] proposes a Bayesian hierarchical approach based on Latent Dirichlet Allocation (LDA) to transfer user interest cross domains or media.

The other major category of cross domain recommendation method is collaborative filtering based ones. [18, 19, 9] all try to exploit cluster level preference patterns which are similar across domains to improve the recommendation performance. Hu et al. propose a generalized Cross Domain Triadic Factorization model which leverages both explicit and implicit feedback [12]. [21] show that using Factorization Machines in cross domain collaborative filtering can significantly improve the recommendation quality in cold start context. Joshi et al. propose an extension to two multi-domain learning techniques and enable them to simultaneously learn from several metadata attributes[16].

Domain selection problem for cross domain recommendation is an emerging area.[29] proposes the first work considering a large number of domain pairs in cross-domain recommender systems by using a Canonical Correlation Analysis approach. In [40] and [22], to cope with the source domain selection burden, authors design algorithms to turn to some online information sources such as social networks or the Wikipedia for help. However, none of the previous works addresses the problem of selecting the optimal set of source domains for a target domain.

3. PROBLEM DEFINITION

The task of this paper is personalized prioritization for broadcast emails. That is to say we want to predict whether a broadcast email is important or not for a given user. There are large numbers of mailing lists in an e-mail system. For simplicity, we assume each broadcast email is only sent to one mailing list and a user can enroll in multiple mailing lists. For a broadcast email waiting for prioritization prediction, we define the mailing list which it is sent to as the target mailing list (equivalent to the target domain in cross-domain recommendation) and all the remaining mailing lists as source mailing lists (equivalent to source domains in cross-domain recommendation). In this paper, *domain* and *mailing list* are used as synonyms. The broadcast email prioritization problem can be divided into the following three *sub problems*.

1. Sample the feedback from a small portion of users to solve the cold start problem, since each broadcast email waiting for prioritization is completely cold with no user interaction.
2. Find the optimal set of source mailing lists whose extra information can help the most with priority prediction.
3. Predict the priority of the broadcast email with the help of the feedback from the sampled users and extra information from the source mailing lists.

For user set \mathbf{U} and email set \mathbf{E} , we define a binary email importance matrix \mathbf{I} based on users' feedback on emails. I.e., for user $u \in \mathbf{U}$ and email $e \in \mathbf{E}$,

$$I_{u,e} = \begin{cases} 1 & \text{if } u \text{ has viewed } e \\ 0 & \text{if } u \text{ hasn't viewed } e \end{cases} \quad (1)$$

We define a mailing list as a set of users $M_i \subset U$ and assume that there are n mailing lists in total, denoted by $M = \{M_1, \dots, M_n\}$. We define the email set E_i as the set of emails sent to M_i .

For each email e , we record its sender and receiver. For each user u , we record user features like country and time-zone. Given a new email e_{new} to be sent to mailing list M_t , we define the three sub problems mentioned above as:

Sampling Users for Feedback Given U, E, I, e_{new}, M_t and time interval $T_{feedback}$ in which we collect users' feedback, select a subset S of users from M_t and collect their feedback, $I_{S, e_{new}}$, within time window $T_{feedback}$.

Choose Source Mailing Lists Given U, E, I, M_t, E_t , select the optimal set of source mailing lists $M' \subseteq M$ whose extra information maximizes the prediction accuracy of email priority for e_{new} . The choice of mailing lists is independent of e_{new} .

Prediction for Remaining Users Given $U, E, I, M, M_t, M', E, e_{new}, I_{S, e_{new}}$, predict the priority for $I_{M_t - S, e_{new}}$.

4. THE CBEP FRAMEWORK

In this section, we introduce our Cross-domain Broadcast Email Prioritization (CBEP) framework to solve the three sub problems of broadcast email prioritization: user feedback sampling, optimal source domain set selection and priority prediction. Optimal source domain set selection is the major contribution of this paper and will only be briefly described in this section with details presented in section 5.

4.1 User Feedback Sampling

In a broadcast email prioritization task, each email waiting for priority prediction is completely *cold*. That's to say no view-email action has been observed, since the email has not yet been sent to any users, which makes it impossible to perform collaborative filtering directly. Thus, we need to first sample the feedback from a small portion of users to solve the cold start problem. *Feedback* in this paper refers to user's view-email action. There are two types of possible feedback. *Positive feedback* which means the user has viewed the email, is a relatively clear evidence that this email is important and *negative feedback* which means the user fails to view the email, is a mixture of cases where the user is unaware of the email or the user thinks the email is unimportant.

We propose a simple strategy to collect the initial feedback. For a new email, we send it to all the users without priority labels and we wait for a short period of time. We then predict the priority based on the initial feedback collected within this period of time. The only challenge of this strategy is to determine how long we should wait to achieve the best trade-off between gathering enough feedback for accurate priority prediction and making predictions as soon as possible. We employ cross validation to determine the optimal length of the waiting time

4.2 Optimal Source Domain Set Selection

Each mailing list can be regarded as a domain and the viewing information accumulated in one domain can be used to improve the quality of recommendations in another domain, which is especially helpful if the user has few or no

views (e.g. a new user to a mailing list) or the target mailing list has little of information (e.g. a mailing list with limited number of users or items). With up to millions of mailing lists in a mailing system, how to select the optimal set of source domains to best improve the prediction accuracy creates a million domain challenge.

To solve the million domain challenge, multiple criteria need to be considered. Signals from these criteria come in different format and can be contradictory with each other. Thus we propose to formulate this as an optimization problem. Basically we will search for a binary assignment to each candidate source domain that optimizes the prediction accuracy. Details will be introduced in section 5.

4.3 Priority Prediction

The feedback from the target domain, the selected source domains and the feedback of sampled users will be used for priority prediction. Formally, we define the feedback set used for priority prediction as $I' = \{I_{M_t, E_t}, I_{M', E_{M'}}, I_{S, e_{new}}\}$.

We use a weighted low-rank approximation method for the priority prediction. The intuition behind the proposed method is to assign different weights to the feedback based on the source of the information. Given the target domain M_t and a selected source domain M_i , we assign larger weight to feedback from the target domain (I_{M_t, E_t}) and the weight of feedback from the source domain (I_{M_i, E_i}) will be determined based on the similarity of the feedback patterns between the target domain and source domain, $Sim_i(t)$, (defined in section 5.1.2). For feedback from source domains, the feedback from the shared users between M_i and M_t will be given larger weights than those from the non-shared users. Details of the weighting scheme are summarized in Table 1. W_{pos} and W_{neg} is the weight for positive feedback and negative feedback which are set to be 5 and 1. δ is a constant used for smooth and ι is a decay factor which are set to be 0.1 and 0.9 respectively. All the constant parameters are tuned by cross validation.

Our objective is to minimize the following loss function:

$$\mathcal{L}(P, Q) = \sum_{ij} W_{ij} (I'_{ij} - P_i \cdot Q_j^T) + \lambda (\|P\|_F^2 + \|Q\|_F^2) \quad (2)$$

in which $P \in \mathcal{R}^{(|M' \cup M_t| \times d)}$ and $Q \in \mathcal{R}^{(|E_{M'} \cup E_t| + 1) \times d}$ stand for the latent vectors for users $\{M', M_t\}$ and items $\{E_{M'}, E_t, e_{new}\}$. W_{ij} is a non-negative weight for u_i and e_j and the weighting scheme of non-negative weight matrix W is summarized in Table 1.

Alternating Least Squares (ALS) is used to solve the optimization problem by fixing P and Q alternatively while optimizing the other parameter.

When fixing Q and solving $\frac{\partial \mathcal{L}(P, Q)}{\partial P_i}$

$$P_i = I'_i \widetilde{W}_i \cdot Q (Q^T \widetilde{W}_i \cdot Q + \lambda (\sum_j W_{ij} ID))^{-1} \quad (3)$$

where $\widetilde{W}_i \in \mathcal{R}^{(|E_{M'} \cup E_t| + 1) \times (|E_{M'} \cup E_t| + 1)}$ is a diagonal matrix with the elements of W_i on the diagonal and $ID \in \mathcal{R}^{d \times d}$ is an identity matrix.

Similarly, when fixing P and solving $\frac{\partial \mathcal{L}(P, Q)}{\partial Q_j}$

$$Q_j = I'_j{}^T \widetilde{W}_j P (P^T \widetilde{W}_j P + \lambda (\sum_i W_{ij} ID))^{-1} \quad (4)$$

where $\widetilde{W}_j \in \mathcal{R}^{(|M' \cup M_t|) \times (|M' \cup M_t|)}$ is a diagonal matrix

Table 1: Weighting Schemes

Source of Feedback	Type	Weight
\mathbf{I}_{M_t, E_t}	Positive	W_{pos}
\mathbf{I}_{M_t, E_t}	Negative	W_{neg}
$\mathbf{I}_{M_i \cap M_t, E_i}$	Positive	$(2 * Sim_i(t) + \delta) * W_{pos}$
$\mathbf{I}_{M_i \cap M_t, E_i}$	Negative	$(2 * Sim_i(t) + \delta) * W_{neg}$
$\mathbf{I}_{M_i - M_i \cap M_t, E_i}$	Positive	$(2 * Sim_i(t) + \delta) * W_{pos} * \iota$
$\mathbf{I}_{M_i - M_i \cap M_t, E_i}$	Negative	$(2 * Sim_i(t) + \delta) * W_{neg} * \iota$

with the elements of \mathbf{W}_j on the diagonal. Details of using ALS to solve matrix factorization problems are discussed in [13].

For each remaining user $u_i \in (\mathbf{M}_t - \mathbf{S})$, the priority to e_{new} is predicted as

$$\mathbf{I}_{i, e_{new}} = \mathbf{P}_i \mathbf{Q}_{e_{new}}^T \quad (5)$$

After estimating $\mathbf{I}_{i, e_{new}}$ for all the remaining users, it can be used as a feature in a classification model (e.g. a logistic regression model as proposed in [1]) and additional features like content feature can also be easily added to the classification model. It is worth noting above mentioned method is just one of the many matrix factorization methods which can be applied to the priority prediction task. The model needs to be re-trained after each new email comes in, which may be infeasible in real life scenarios. However, there are already several methods [37][23] for fast incremental matrix factorization for recommendation with positive-only feedback, which can be easily applied to the priority prediction task. Since it is not the major contribution of this paper, readers can refer to [37][23] for further information.

For simplicity, the estimated importance feedback $\mathbf{I}_{i, e_{new}}$ will be the only feature considered in priority classification of this paper. The intuition of our method for priority classification is that for each email a certain percentage of users will consider it as important, but the percentage varies among different emails since they are with different topics, written quality etc. An important email will result in more views during the time we wait for user feedback. Thus we can infer the percentage of important emails by the number of views we observed in the sampling phase. We define the percentage of users considering email e_{new} important as:

$$H(e_{new}) = \frac{pos(e_{new})}{pos_{avg}(\mathbf{M}_t)} \frac{tr(\mathbf{I}_{M_t}^T \mathbf{I}_{M_t})}{|\mathbf{M}_t| * |\mathbf{E}_t|} \quad (6)$$

$pos(e_{new})$ is the total number of view-email behaviors observed in the waiting time window for e_{new} and $pos_{avg}(\mathbf{M}_t)$ is the average number of view-email behaviors observed in the waiting time window for all the emails from \mathbf{M}_t . The second term of (6) stands for the average percentage of important emails for \mathbf{M}_t .

For the top $H(e_{new})$ percent of users according to $y_{i, e_{new}}$, we predict e_{new} as important while for others as unimportant.

5. OPTIMAL SOURCE DOMAIN SELECTION

To solve the optimal source domain set selection problem, we formulate it as an optimization problem. In section 5.1, we first discuss the optimal solution including all the selection criteria. However, since the optimization problem is

difficult to solve directly, we propose two approximate solutions in section 5.2. In section 5.3, some additional measures are proposed to improve the efficiency.

5.1 Optimal Solution

Formally, given the target mailing list \mathbf{M}_t , we define a binary vector $\boldsymbol{\alpha}$ of size n where each entry α_i indicates whether the source mailing list \mathbf{M}_i is selected or not. If a source mailing list \mathbf{M}_i is selected, the corresponding entry α_i is 1 else 0. Thus our problem reduces to finding $\boldsymbol{\alpha}$ that maximizes the objective function introduced in the following sections.

5.1.1 Overlap of Users

A user can enroll in any number of mailing lists and thus mailing lists can have shared users, i.e. users enrolled in multiple email lists. We prefer source mailing list with a larger number of shared users with the target mailing list for the following reasons.

1. Recent work[6] has confirmed that knowledge between two domains can only be transferred if they are linked by shared users or items. In our case, only the shared users can be used to transfer information between domains and the higher the percentage of shared users, the easier to transfer extra knowledge from the source domain.
2. Since similar mailing lists are more likely to attract the same set of users, the overlap of users is also an indication of the similarity of two mailing lists and similar mailing lists result in users making similar choices in them. For instance, the mailing lists of e-commerce fashion brands A&F and Hollister may have a large number of shared users because they are targeting similar group of users with similar products and user's preference of promotions can be transferred from one to the other.

We define overlap percentage between source mailing list \mathbf{M}_i and target mailing list \mathbf{M}_t as:

$$overlap_i(t) = \frac{|\mathbf{M}_i \cap \mathbf{M}_t|}{|\mathbf{M}_t|} \quad (7)$$

The larger $overlap_i(t)$ is, the larger the number of shared users between the source mailing list.

5.1.2 Similar Feedback Pattern

Users' feedback patterns vary across domains. For example, users who share similar feedback in an e-commerce mailing list may share completely different preference in the mailing list of the university. Only source domains with similar feedback patterns to the target domain will be helpful in cross domain recommendation. Or else, the source domain may introduce noise to the system and jeopardize the recommendation performance. In this paper, we model the similarity of the feedback patterns between two mailing lists as the average similarity difference of each pair of shared users between the two mailing lists. Intuitively, two mailing lists have similar feedback patterns means that for shared users between the two mailing lists, two users with similar feedback patterns in one mailing list also share similar feedback patterns in the other and vice versa.

Formally, for each mailing list \mathbf{M}_i , user u can be represented as a binary vector $\mathbf{v}_{i, u}$ of size $|\mathbf{E}_i|$ with each entry

indicating whether u has read the corresponding email from \mathbf{E}_i or not. We define the shared user set between two mailing lists i and j as $\mathbf{C}_{i,j}$, and their feedback pattern similarity as:

$$sim(i, j) = 1 - \frac{1}{2|\mathbf{C}_{i,j}|^2} \sum_{u,w \in \mathbf{C}_{i,j}} |\cos(\mathbf{v}_{i,u}, \mathbf{v}_{i,w}) - \cos(\mathbf{v}_{j,u}, \mathbf{v}_{j,w})| \quad (8)$$

Larger $sim(i, j)$ value indicates larger feedback pattern similarity. Specifically, we denote the feedback pattern similarity of a target mailing list \mathbf{M}_t and a source mailing list \mathbf{M}_i as $sim_i(t) = sim(i, t)$ and $sim_i(t)$ will be used as a constraint in our objective function.

5.1.3 Coverage of Users

We aim to select the optimal set of source mailing lists \mathbf{M}' and each $\mathbf{M}_i \in \mathbf{M}'$ has a set of shared users $\mathbf{C}_{i,t}$ with target domain \mathbf{M}_t . Intuitively, we want the number of shared users between \mathbf{M}' and \mathbf{M}_t to be as large as possible so that we can cover and transfer extra information for more users in the target mailing list. That's to say we want to choose a size- k mailing list set \mathbf{M}' so that the number of shared users between these source mailing lists in \mathbf{M}' and the target mailing list \mathbf{M}_t is maximized:

$$max |\cup_{\mathbf{M}_i \subseteq \mathbf{M}'} \mathbf{C}_{i,t}| \quad (9)$$

This is actually an unweighted maximum coverage problem, which is NP-hard. Instead of modeling this criteria directly, we propose a constraint related with this criteria. We define the overlap percentage between source mailing lists $\mathbf{M}_i, \mathbf{M}_j$ and target mailing list \mathbf{M}_t as

$$overlap_{i,j}(t) = \frac{|\mathbf{M}_i \cap \mathbf{M}_j \cap \mathbf{M}_t|}{|\mathbf{M}_t|} \quad (10)$$

By introducing the triple-domain overlap $overlap_{i,j}(t)$ as an constraint in the objective function, we expect the user sets shared with the target mailing list from different source mailing lists to be as diverse as possible and not to concentrate on the same set of users.

5.1.4 Objective Function

Combining all the criteria mentioned above, we propose the following objective function:

$$\begin{aligned} \arg \max_{\alpha} & (\lambda_{overlap} \sum_{i=1}^n overlap_i(t) \alpha_i + \lambda_{sim} \sum_{i=1}^n sim_i(t) \alpha_i \\ & - \frac{\lambda_{cov}}{n} \sum_{i=1}^n \sum_{j=1}^n overlap_{i,j}(t) \alpha_i \alpha_j) \frac{1}{\sum_{i=1}^n \alpha_i + \delta} \end{aligned} \quad (11)$$

subject to:

$$\alpha : \alpha_i \in \{0, 1\} \text{ for } i = 1 \dots n$$

The first term in equation (11) ensures the selected mailing lists to have large percentage of overlap users with the target mailing list. The second term ensures the selected mailing lists to have similar feedback patterns with the target mailing list. The third term is subject to favoring lists of mailing lists whose shared user sets with the target mailing list have little overlap. Combining the third factor with

the first one, we prefer mailing lists that have large overlap with the target mailing list but small overlap with each other, which provides good coverage of users. It is worth noting that we do not specify the number of source mailing lists to be selected, because we think it should be a dynamic number related with the target domain and should be chosen automatically by the algorithm. We add the fourth term $\frac{1}{\sum_{i=1}^n \alpha_i + \delta}$ as a normalizer to the objective function to prevent it from selecting too many source mailing lists. Selecting too many source mailing lists will not only introduce noise but also increase the computational burden of the system. $\lambda_{overlap}$, λ_{sim} , and λ_{cov} are constant weights for the first three terms which can be learned by cross-validation. δ is also a constant, which influences the number of source mailing lists selected. The larger δ is, the more source mailing lists will be picked.

5.2 Approximate Solutions

The optimization problem (11) is an integer programming problem with both quadratic term and fraction in its objective function, which makes it extremely difficult, if not possible, to be solved directly. So we propose two approximate solutions to solve the problem. Both of them first approximate the original optimization problem as a quadratic integer programming problem and then further relax the constraints to make it a quadratic linear programming problem, which can be solved in polynomial time.

For the first approximate solution, we transform the denominator of the objective function (11) into a term in the numerator.

$$\begin{aligned} \arg \max_{\alpha} & (\lambda_{overlap} \sum_{i=1}^n overlap_i(t) \alpha_i + \lambda_{sim} \sum_{i=1}^n sim_i(t) \alpha_i \\ & - \frac{\lambda_{cov}}{n-1} \sum_{i=1}^n \sum_{j=1}^n overlap_{i,j}(t) \alpha_i \alpha_j - \lambda_{pen} \sum_{i=1}^n \alpha_i \end{aligned} \quad (12)$$

The newly added fourth term serves as a penalty to prevent the objective function from selecting too many source mailing lists. λ_{pen} is a constant parameter which can be learned by cross-validation.

The second approximate solution is based on the intuition that since selecting too many source domains will not only increase the computational burden of the system but also introduce noise, we do not want to choose too many source domains. That is to say we can set an upper bound z_{max} as the maximum number of source domains that can be used in a cross domain recommendation. Our optimization problem then can be approximately formulated as

$$\begin{aligned} \arg \max_{\alpha} & \lambda_{overlap} \sum_{i=1}^n overlap_i(t) \alpha_i + \lambda_{sim} \sum_{i=1}^n sim_i(t) \alpha_i \\ & - \frac{\lambda_{cov}}{n-1} \sum_{i=1}^n \sum_{j=1}^n overlap_{i,j}(t) \alpha_i \alpha_j \\ \text{s.t. } & \alpha_i \in \{0, 1\}, \quad \sum_{i=1}^n \alpha_i = z_k \end{aligned} \quad (13)$$

We solve the optimization problem (13) z_{max} times for

$z_k \in \{1, 2, \dots, z_{max}\}$ and for every z_k , a set of source domains α_k is obtained. We compute the value of the original objective function (11) for each α_k and select α_k with the highest value. The corresponding z_k is the number of source mailing lists to be selected.

Both approximate solutions transform the original optimization problem into an integer quadratic programming problem, which is still NP-hard. We relax the constraints $\alpha_i \in \{0, 1\}$ to $0 \leq \alpha_i \leq 1$, which makes it a continuous optimization problem that can be solved in polynomial time. For the first approximate solution, a threshold γ will be learned by cross validation and the source domains with $\alpha_i > \gamma$ are selected. For the second approximate solution, the optimal number of source domains can be automatically determined by the solution. We use the 'quadprog' function in MATLAB to solve this quadratic programming problem.

5.3 Efficiency Improvement

Directly calculating the coefficients $sim_i(t)$ and $overlap_{i,j}(t)$ in objective function (11) may be time consuming. Given target domain M_t , the time complexity of calculating $sim_i(t)$ and $overlap_{i,j}(t)$ is $O(n * |C_{i,t}|^2)$ and $O(n^2 * |U|)$, which may be unacceptable in real mailing system because the number of mailing lists and the number of shared users between mailing lists can be very large. The following measures are proposed to improve the efficiency.

1. Only consider source mailing lists with a certain number of shared users with the target mailing list. That is to say we set up a minimum overlap threshold κ and only consider M_i with $|C_{i,t}| \geq \kappa$. In this way, we will dramatically reduce the number of candidate mailing lists.
2. Randomly sample user pairs for the feedback pattern similarity calculation. The number of pairs of shared users can be extremely large, if there are lots of shared users between M_i and M_t . We only need a sample of the pairs to approximate $sim_i(t)$ defined in (8) to obtain a relatively accurate approximate value.
3. Optimal source mailing list selection can be performed offline, since it does not depend on e_{new} waiting for email prioritization but depends only on the mailing list M_t to which e_{new} is sent. Therefore, the optimal set of source domains can be pre-calculated offline and updated periodically (e.g. weekly), which further relieves the computational burden of the system.

6. EXPERIMENTAL EVALUATION

6.1 Dataset

In the experiments, we used a real life dataset from Samsung Electronics. We collected emails and their view logs from a large business mailing list within Samsung. The mailing list sends notifications related to a large internal forum and employees from all around the world receive emails with various topics, like win notices of deals, meeting agendas of customers, business objectives, news and technical issues. When a new thread is posted, a notification will be emailed to all the users. We split the mailing list into 490 sub mailing lists based on sections of the forum. That is to say a user belongs to a sub mailing list iff he has interacted with notification emails of threads published in that section. We

treat each sub mailing list as a real mailing list in our experiments. The dataset contains 6506 broadcasting emails sent to 2433 Samsung employees, generating 333,979 view records. We split the data set into training set (containing 5475 emails and their view logs) and testing set (1031 emails and their view logs) based on a certain time point.

We only record users' email viewing data and we assume an email is important to a user if the user has viewed it. It is worth noting that the data set is relatively small with only view-email behavior. Accuracy could be better if we can incorporate information like deletions of emails, flagging emails as important and skipping an email. However, due to the privacy concerns, there is no public dataset containing importance judgments for broadcast emails [41].

All data was analyzed and stored in accordance with Samsung's privacy policy. Only the view logs of the broadcast emails were extracted and all the users are Samsung Employees. The dataset was completely anonymized by mapping user ids and email ids to integer indices before any analysis.

6.2 Evaluation Metrics

Since our task is a classification task, we use precision, recall and f-score as the main evaluation metrics. Based on the predicted label from the algorithm and ground truth label from the data set, a prediction is either true positive (tp), true negative (tn), false positive (fp), or false negative (fn). The metrics are defined as

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$F - score = \frac{2 * precision * recall}{precision + recall}$$

In the experiments, we evaluate the precision, recall and f-score at two levels:

Mail Level The *mail level* precision, recall and fscore of an algorithm are defined as the average precision, recall and f-score of all the emails in test set.

Mailing List Level The *mailing list level* precision, recall and fscore of an algorithm are defined as the average precision, recall and f-score of all the mailing lists in test set. At the mailing list level, the evaluation metrics implicitly gives more weights to cold mailing lists since all mailing lists are treated equally disregarding how many emails or users they contain.

6.3 Baselines

It is worth noting in our previous work[38], experiments have already confirmed that by using collaborative filtering, our method can outperform various existing email prioritization methods, including the content-based methods[1][38]. So in this paper, the experimental design focuses on testing how different source domain selection strategies affect the performance of email prioritization and evaluating the performance of our major contribution, a cross domain recommendation framework to select an optimal set of source domains from a large numbers of candidate source domains.

Since we propose two approximate methods for CBEP, we refer to the first approximate method as *CBEP-A1* (corresponding to objective function 12) and the second method

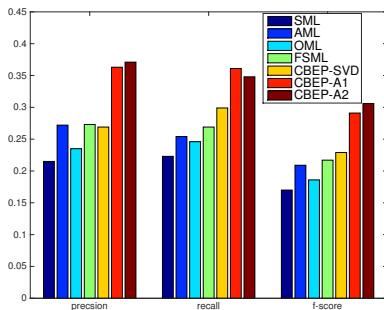


Figure 1: Baseline Comparison at Mailing List Level

as *CBEP-A2* (corresponding to objective function 13). We adapt the following methods for comparison. The first four baselines are four different source domain set selection strategies, which we use to replace the optimal source domain selection part of CBEP and they all use the same user feedback sampling strategy and classification strategy as described in section 4.1 and 4.3. The weighted matrix factorization method designed for implicit feedback from [25][13] [10] is used for these four baselines. The last baseline is a variation of CBEP by eliminating the weighting scheme in the weighted low-rank approximation process. We set the waiting time for gathering user feedback as 45 min for all algorithms and set $z_{max} = 15$ for CBEP-A2.

Single Mailing List (SML) SML only considers information from the target mailing list, disregarding information from other mailing lists. SML is similar to the email prioritization method proposed in [38].

All Mailing Lists (AML) AML considers all the source mailing lists and combines the information from every mailing list with the target mailing list.

Overlapping Mailing Lists (OML) OML selects the top k mailing lists with the largest percentage of overlap with the target domain as the source domains. Overlap is defined in section 5.1.1 and we choose $k = 5$ for the experiments.

Feedback Similar Mailing Lists (FSML) FSML selects the top k mailing lists with the highest feedback similarity with the target domain as the source domains. Feedback pattern similarity is defined in section 5.1.2 and we choose $k = 5$ for the experiments.

CBEP Without Weight (CBEP-SVD) In order to test how the weighted low-rank approximation impacts the result, we propose another baseline called *CBEP-SVD*, which is the same as our algorithm except that we eliminate the whole weighting scheme mentioned in section 4.3 and just use a basic SVD model.

6.4 Results and Analysis

6.4.1 Comparison with Baselines

In this section, we compare the two approximate solutions to all baselines in terms of precision, recall and f-score at both the mail and mailing list level.

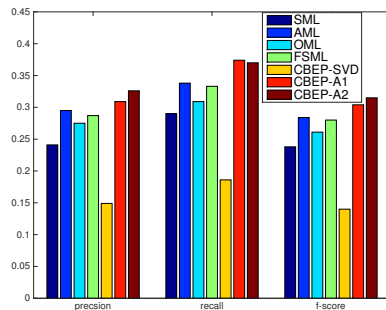


Figure 2: Baseline Comparison at Mail Level

As shown in figure 1 and 2, CBEP-A1 and CBEP-A2 significantly outperform all the baselines on all the evaluation metrics. SML performs worst, which makes sense since it only considers information from the target domain and disregards all the additional information from other mailing lists. Moreover, new users (user who newly enrolled in a mailing list) and cold mailing lists (mailing list with limited number of emails) are common, and SML cannot handle these cold start problems. AML performs significantly better than SML since it includes the information from all the other mailing lists. However, considering all the mailing lists deteriorates the prioritization precision due to the noise introduced by the unrelated mailing lists.

OML and FSML choose the set of source domains based on overlap of users and feedback pattern similarity. They both outperform SML by incorporating information from similar domains, and FSML performs better than OML. By selecting a limited number of source domains, FSML can already achieve similar performance as AML. The performance of CBEP-SVD is not good compared with CBEP-A1 and CBEP-A2, which further confirms our weighting scheme is useful. Both of our methods CBEP-A1 and CBEP-A2 perform significantly better than the best baseline FSML. Compared with FSML, CBEP-A1 improves the f-score by 40% at mailing list level and by 12% at mail level. The improvement comes from three aspects. First of all CBEP can choose an optimal set of source domains based on multiple criteria. Secondly it is able to dynamically determine the number of source domains to be selected. Last but not least, CBEP uses a weighted matrix factorization method and gives different weights to different source domains.

6.4.2 Criteria for Optimal Source Domain Selection

As mentioned in section 5.1, we consider three different factors to select the optimal source domains, namely, overlap of users, feedback pattern similarity and coverage of users. In this section, we remove our three criteria considered in our objective function (11) one at a time by eliminating the corresponding term from (11) and optimizing the objective function based on the remaining terms. In this way, we evaluate how each criteria individually affects the prediction precision. Due to the page limit, we only show the mailing list level results for CBEP-A1 and note that the other results show similar trends.

From the results displayed in Figure 4, we can see all these three factors are useful in boosting the email prioritization performance. The feedback pattern similarity criterion

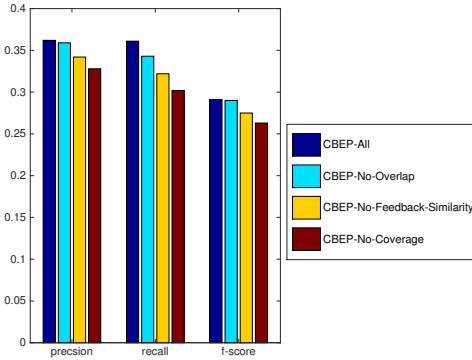


Figure 3: Optimazation Criteria Analysis

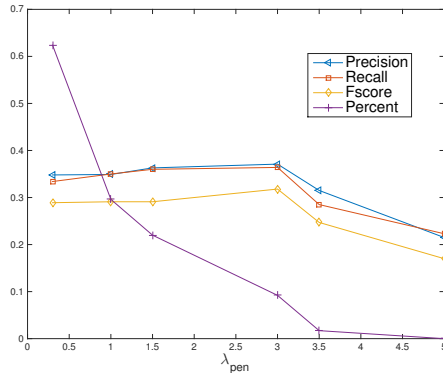


Figure 4: Number of Selected Source Domains and Prediction Performance vs. λ_{pen}

shows more importance than the overlap of users criterion, which is in accordance with our observation of the results of OML and FSML in figure 1. The coverage of users criterion turns out to be the most important one by allowing the set of source domains to cover more users in the target domain.

6.4.3 Number of Selected Source Domains

One of the advantages of the CBEP framework is its ability to dynamically determine the number of source domains to be selected. That is to say for each target domain, we can get an optimal source domain set size. In this section, we take our first approximation method CBEP-A1 as an example to analyze how the parameter settings affect the number of selected source domains and the precision, recall and f-score of the prioritization algorithm.

In CBEP-A1, the penalty term weight λ_{pen} affects the number of the source domains to be selected. The larger λ_{pen} is, the less source domains are selected. In figure 5, we show how the average percentage of source domains selected (denoted as percent in figure 4), precision, recall and f-score vary for different settings of λ_{pen} . It is worth noting that choosing too many or too few source domains can both jeopardize the prediction performance.

6.4.4 Performance on Cold Domains

Cold mailing lists with limited number of items, users or user feedback are common in real systems. One of the ad-

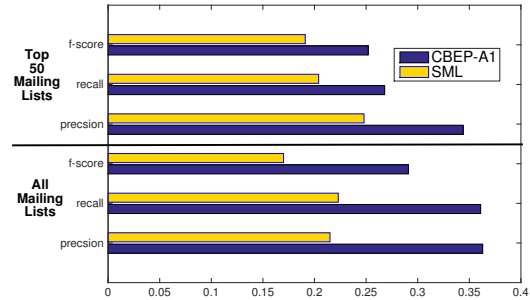


Figure 5: All Domains Vs. Hot Domains

vantages of cross domain recommendation is its ability to solve this kind of cold start problem by incorporating information from other domains. This can already be verified by the results in figures 1 and 2, in which CBEP-A1 and CBEP-A2 gain more improvement at the mailing list level than at the mail level, since mailing list level implicitly gives more weight to cold mailing lists.

We further verify the good performance of CBEP on cold mailing lists by evaluating CBEP-A1 and SML on the top 50 hottest (in terms of the number of user feedback) mailing lists and comparing these results to the results on all the mailing lists. See Figure 5. We expect precision, recall and f-score should be higher on the top 50 dataset, since these mailing lists have more abundant training data. Surprisingly, CBEP-A1 performs even better on all mailing lists. We attribute this to two reasons. On one hand, CBEP indeed performs well on cold mailing lists by incorporating information from other source mailing lists. On the other hand, for some of the most popular mailing lists, there may already be abundant data so that the marginal utility of the extra information introduced by CBEP diminishes.

7. CONCLUSION

In this paper, we introduce the problem of personalized broadcast email prioritization considering large numbers of mailing lists and propose a novel cross domain recommendation framework CBEP to solve the problem. To select the optimal set of source domains from the large number of domains, we propose an optimization model that considers multiple selection criteria including the overlap of users, feedback pattern similarity and coverage of users. A weighted low-rank approximation method is proposed to make predictions based on information from both the target domain and the selected source domains. Comprehensive experiments are conducted on a real life dataset from Samsung Electronics. The results show that our method CBEP outperforms all the baselines and the various optimization criteria considered indeed all help to improve the prediction performance.

Since this is the first work on cross domain recommendation for broadcast email prioritization, there is still much room for future research. First, more criteria can be considered in the optimal source domain selection. For example, since cross domain recommendation works better on

cold mailing lists, we may consider to use the number of existed user feedback of the target domain as a criterion. That is to say for a mailing list short of information we choose to select more source domains and for a mailing list already with abundant information we choose to select fewer source domains to avoid noise. Secondly, we use a relatively small real-life dataset in our experiments due to the lack of other dataset. Further experiments should be conducted on a larger dataset with more diverse user feedback to better evaluate our method.

8. ACKNOWLEDGMENTS

This work was supported by the National Basic Research Program of China(973 Program) under Grant 2013CB336500, NSERC Discovery Grant, National Science Foundation of China (Grant No. 61373118, 61522206, 61173186), National Key Technology R&D Program (2014BAK15B02).

9. REFERENCES

- [1] D. Aberdeen, O. Pacovsky, and A. Slater. The learning behind gmail priority inbox. In *LCCC: NIPS 2010*, 2010.
- [2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.
- [3] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *UMUAI*, 18(3):245–286, 2008.
- [4] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye. Batch mode active sampling based on marginal probability distribution matching. In *SIGKDD*, pages 741–749. ACM, 2012.
- [5] M. Chui. *The social economy: Unlocking value and productivity through social technologies*. McKinsey, 2013.
- [6] P. Cremonesi and M. Quadrana. Cross-domain recommendations without overlapping data: myth or reality? In *Recsys*, pages 297–300. ACM, 2014.
- [7] A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*, pages 278–288, 2015.
- [8] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. Cross-domain recommender systems: A survey of the state of the art. In *SCIR*, 2012.
- [9] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo. Cross-domain recommendation via cluster-level latent factor model. In *SIGKDD*, pages 161–176. Springer, 2013.
- [10] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith. Librec: A java library for recommender systems. In *UMAP*, 2015.
- [11] E. Horvitz, A. Jacobs, and D. Hovel. Attention-sensitive alerting. In *UAI*, pages 305–313, 1999.
- [12] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu. Personalized recommendation via cross-domain triadic factorization. In *WWW*, pages 595–606, 2013.
- [13] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. Ieee, 2008.
- [14] T. Jackson, R. Dawson, and D. Wilson. Case study: evaluating the effect of email interruptions within the workplace. © EASE, 2002.
- [15] L. Johansen, M. Rowell, K. R. Butler, and P. D. McDaniel. Email communities of interest. In *CEAS*, 2007.
- [16] M. Joshi, M. Dredze, W. W. Cohen, and C. P. Rosé. What’s in a domain? multi-domain learning for multi-attribute data. In *HLT-NAACL*, pages 685–690, 2013.
- [17] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In *UMAP*, pages 183–194. Springer, 2011.
- [18] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI*, volume 9, pages 2052–2057, 2009.
- [19] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *ICML*, pages 617–624. ACM, 2009.
- [20] S. Lohr. Is Information Overload a \$650 Billion Drag on the Economy? . *New York Times*, 2007.
- [21] B. Loni, Y. Shi, M. Larson, and A. Hanjalic. Cross-domain collaborative filtering with factorization machines. In *AIR*, pages 656–661. Springer, 2014.
- [22] Z. Lu, Y. Zhu, S. J. Pan, E. W. Xiang, Y. Wang, and Q. Yang. Source free transfer learning for text classification. In *AAAI*, pages 122–128, 2014.
- [23] X. Luo, Y. Xia, and Q. Zhu. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27:271–280, 2012.
- [24] C. Neustaedter, A. B. Brush, M. A. Smith, and D. Fisher. The social network and relationship finder: Social sorting for email triage. In *CEAS*, 2005.
- [25] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE, 2008.
- [26] S. Radicati, P. Analyst, and J. Levenstein. Email Statistics Report , 2015-2019. 44, 2015.
- [27] N. Rubens, R. Tomioka, and M. Sugiyama. Output divergence criterion for active learning in collaborative settings. *IPSJ*, 2(3):87–96.
- [28] S. Sahebi and P. Brusilovsky. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *UMAP*, pages 289–295. Springer, 2013.
- [29] S. Sahebi and P. Brusilovsky. It takes two to tango: An exploration of domain pairs for cross-domain collaborative filtering. In *Recsys*, pages 131–138. ACM, 2015.
- [30] S. Sahebi and T. Walker. Content-based cross-domain recommendations using segmented models. In *RecSys*, pages 57–64, 2014.
- [31] S. Tan, J. Bu, X. Qin, C. Chen, and D. Cai. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 127:124–134, 2014.
- [32] G. Tang, J. Pei, and W. S. Luk. Email mining: tasks, common techniques, and tools. *KIS*, pages 1–31, 2013.
- [33] D. Tao, X. Li, X. Wu, and S. J. Maybank. General tensor discriminant analysis and gabor features for gait recognition. *PAMI*, 29(10):1700–1715, 2007.
- [34] D. Tao, X. Li, X. Wu, and S. J. Maybank. Geometric mean for subspace selection. *PAMI*, 31(2):260–274, 2009.
- [35] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *PAMI*, 28(7):1088–1099, 2006.
- [36] N. Tintarev and J. Masthoff. *Active Learning in Recommender Systems*, volume 54. 2011.
- [37] J. Vinagre, A. M. Jorge, and J. Gama. Fast incremental matrix factorization for recommendation with positive-only feedback. In *User Modeling, Adaptation, and Personalization*, pages 459–470. Springer, 2014.
- [38] B. Wang, M. Ester, J. Bu, Y. Zhu, Z. Guan, and D. Cai. Which to view: Personalized prioritization for broadcast emails. In *WWW16*, pages 1181–1190, 2016.
- [39] P. Winoto and T. Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26(3):209–225, 2008.
- [40] E. W. Xiang, S. J. Pan, W. Pan, J. Su, and Q. Yang. Source-selection-free transfer learning. In *IJCAI*, volume 22, page 2355, 2011.
- [41] S. Yoo, Y. Yang, F. Lin, and I.-C. Moon. Mining Social Networks for Personalized Email Prioritization. *SIGKDD*, page 967, 2009.