Personalized Retweet Prediction in Twitter

Liangjie Hong Aziz Doumith Brian D. Davison
Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA USA
{lih307, asd309, davison}@cse.lehigh.edu

1. INTRODUCTION

In social network services like Twitter, LinkedIn and Facebook, users are informed instantly via rich multimedia content from their social connections. However, when facing a large amount of content from their social connections, users are simply unable to consume them in an effective and efficient way, leading to the problem of information overload. On the other hand, information for a user is usually limited in scope to the user's social connections. Thus, it is difficult for a user to obtain information distributed outside of their circle, even though it might match their interests, leading to a problem of information shortage. In both cases, users may spend a significant amount of time to filter and search relevant information in social media; thus, it is also very important to understand how users interact with these systems. Interactions between users and social media occur through a variety of actions such as posting, re-posting, replying and commenting. Ideally, social media services would be able to filter and recommend content to users based on their history of previous interactions and interests. This area has attracted the attention of academic and industrial research communities.

The task of understanding users' behaviors and their interests has a number of challenges. First, although the number of items (updates, tweets, etc.) generated by users in services is huge, few of them are interacted by users, making the interaction data is sparse. Second, new users and new content items flow into the system continuously. Thus, the "cold start" problem tends to be severe in these social platforms, compared to traditional information systems. In addition, a tremendous amount of content is rich yet noisy. Simple information retrieval or topical modeling techniques may not be sufficient to capture users' interests.

The problem tackled in this work has strong links to research in recommender systems and collaborative filtering. However, social content systems are much more dynamic than traditional recommender systems: many new items are pushed into the system every second. Therefore, recommender systems should be adjusted to this novel situation. Traditional successful collaborative filtering models are based on latent factor models (LFM), partially due to their superior performance in the Netflix competition. However, the basic assumption for standard LFM is to exploit a user-item interaction matrix and it cannot handle arbitrary features easily. Although some of newly proposed frameworks, based on LFM, can consider features, fundamental modeling assumptions prevent them from handling high-order interaction data (e.g., tensor). In addition, current extensions to LFM that incorporate rich text information are usually cumbersome, requiring complicated inference algorithms that cannot scale to large datasets. Moreover, researchers in collaborative filtering are realizing that pointwise-based measurements may no longer be appropriate, and so a handful of ranking-based metrics are proposed. However, no work to date has compared them systematically on real world datasets.

In this work, we study the problem of modeling users' behaviors by focusing on one particular decision—retweets—in Twitter and try to understand users' interests. Our method can be easily extended to model multiple types of users' decisions as well. We use a state-of-the-art recommendation model, Factorization Machines (FM) [2], to model user decisions and user-generated content simultaneously. Our contributions can be summarized as follows:

- We propose Co-Factorization Machines (CoFM), which deal with two (multiple) aspects of the dataset where each aspect is a separate FM. This type of model can easily predict user decisions while modeling user interests through content at the same time.
- We apply Factorization Machines to text data with constraints. Thus, the resulting method can mimic state-of-the-art topic models and yet benefit from the efficiency of a simpler form of modeling.
- For personalized retweets prediction, we introduce the newly proposed WARP loss [3], which has been successfully applied in text and image retrieval tasks, into the context of recommendation.

- We apply our proposed methods to the problem of modeling personal decision making in Twitter and explore a wide range of features, revealing which types of features contribute to the predictive modeling and how content information can help with the prediction.

We next formalize CoFM with different strategies of shared latent features and describe features used in our model. In Section 3, we report the experiments with the discussions of datasets and baselines used. We conclude the paper in Section 4.

2. OUR METHOD

Two tasks are addressed in our work. First, we wish to uncover the kind of tweet that users prefer to retweet and what features contribute to the mechanism that cause certain pieces of information to be shared across social connections. Second, content is of great importance in Twitter and thus it is vital to discover topics in which users are interested and how these topics influence users' decisions.

We utilize FM, one of the state-of-the-art predictive models for recommender systems, and extend it in several aspects. One important aspect of FM is that the model can mimic the structure of many state-of-the-art models like matrix factorization, pairwise interaction tensor factorization, SVD++ and neighborhood models in one unified framework, as demonstrated by Rendle [2]. For the first task, we focus on a binary response for each tweet: whether the tweet will be retweeted by a target user. If only tweet id indicators and user id indicators are used as features, we can easily recover the heavily used matrix factorization with biases as a function for our responses, while learning latent factors for users and tweets. For the second task, we will model terms in each tweet while treating raw word counts of terms in a tweet as responses. Simply using FM for term counts will result in a decomposition of the term-document matrix without any constraints, obtaining latent factors for each term and latent factors for each tweet, playing as topical codes. While it is similar to conventional topic models, the difference is that latent factors for words as well as for tweets are not constrained to rest on the simplex, yielding difficulties for interpretation. In this paper, we introduce a method to perform non-negative decomposition of the term-tweet matrix and obtain a topic matrix, compatible to standard topic models. This is similar to sparse topical coding [4], but under FM. Thus, we can have use FM to model two tasks separately.

To link them together, we propose CoFM with three paradigms to utilize two separate FM:

- **Shared feature paradigm**: One natural approach to link two latent representations of the same tweet is to treat one type of latent representation as features and feed it into another modeling process. Thus, for each separate FM, this is essentially a latent factor re-weighting scheme with additional parameters to be estimated.
- **Shared latent space paradigm**: A simpler approach is to assume that the latent factor learned from user modeling is exactly same as the one learned from content modeling. Therefore, some parts of the latent factors of the same tweet are shared across different aspects. This formalism shares the idea of matrix co-factorization used in relational learning scenarios.
- **Regularized latent space paradigm**: One can regularize two such representations such that they do not reside too far away from each other. A simple approach is to impose an l2 regularizer. Under this assumption, we can also view that one latent factor is drawn from the multivariate normal distribution with the mean as another latent factor.

The discrepancy between the estimation by FM/CoFM and the true value can be measured by a loss function. For modeling user decisions, we formalize the problem as an optimization problem by using Weighted Approximately Ranked Pairwise loss (WARP). This loss, proposed in Usunier et al. [3], has been successfully applied in image retrieval tasks and IR tasks. By using the precision at k measure, one can weigh the pairwise violations depending on their position in the ranked list. Different parameterizations of the loss can lead to optimize different ranking metrics. Although it is difficult to directly optimize WARP due to the discrete nature of indicator functions, efficient sampling methods are shown to be effective.

Putting things together, a CoFM framework for learning a model for user decisions and content understanding is to optimize the joint objective function for user decisions and content modeling. By choosing different coupling strategies, the model can effectively perform predictive modeling and maximum likelihood estimation of content at the same time. We adopt a hybrid of stochastic gradient descent and coordinate descent to optimize the objective function.

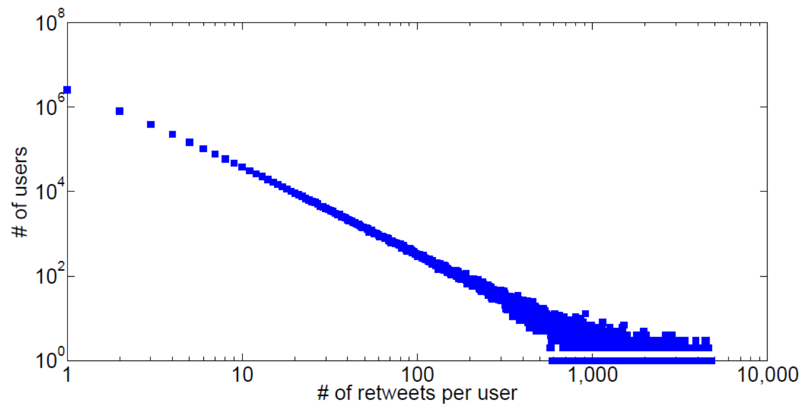

Figure 1: The sparsity of retweets per user.

The core part of FM/CoFM is to utilize a wide range of features. Here, we discuss the features used in our models. All of these features try to capture users' interests. Features are divided into five groups:

- **Categorical features**: The key idea of FM/CoFM is to use both indicator features and explicit features together to obtain competitive performance in predictive tasks. For modeling user decisions, we use three categorical features: 1) target user id, 2) neighbor user id, and 3) the tweet id. For content modeling, we use the term id and the tweet id as features.
- **Content features**: We utilize features to characterize what users have posted and what their friends have posted by building three different types of content profiles for each user: content profile, neighborhood profile and retweet profile. These profile features will capture the interests of users at a fine-grained level.
- **Relevance features**: We measure the relevance between an incoming tweet versus three types of profiles by using dot product while many other IR relevance scores could also apply.
- **Meta features**: Many meta information also might be useful, including length of tweet, hash tag count, hash tag history, URL count, URL domain history and retweet count.
- **Local graph & user features**: These features potentially characterize how popular and how well connected a user is. Intuitively, a popular user who has many friends and followers can be actively passing information by retweeting messages. We have mention count, friend count, follower count, and status count and account age as features.
- **User relationship features**: Relation features refer to those features that represent the relationship between a target user and his/her friends. We have co-friends similarity score, co-follow similarity score, mention score, retweet score and mutual friend as features. The similarity measure used is the Jaccard coefficient.
- **Temporal features**: We estimate a user's activity level in a time period, which is calculated by the average number of tweets he/she published in a periodical time slot, e.g., every Monday. With the estimated response time, the number of accumulated tweets can be calculated by a simple integration. We calculate both activities using period of a day and a week.

All features are pre-calculated through a Hadoop cluster and can be processed efficiently.

3. EXPERIMENTS

We gathered 0.7M target users with approximately 11M tweets from the Twitter public API. For all these target users, we obtained 4.3M neighbor users with 27M tweets. For each target user, we treat all tweets from his/her neighbor users as incoming tweets. If a tweet *d* from incoming tweets is retweeted by user *u*, *d* is treated as a positive instance. Otherwise, it is a negative instance. We plot the unnormalized distribution of number of retweets per user in Figure 1, demonstrating that a great number of users only retweet a limited number of times while some users retweet thousands of times. We borrow Mean Average Precision (MAP) from the IR community and treat retweets as "relevance" labels. We split all incoming tweets into 5 consecutive time periods with equal number of tweets in each time period and train models on one time period while testing them on the next. We compare several aspects of proposed methods and other state-of-the-art baselines: 1) Matrix factorization (MF), 2) Matrix

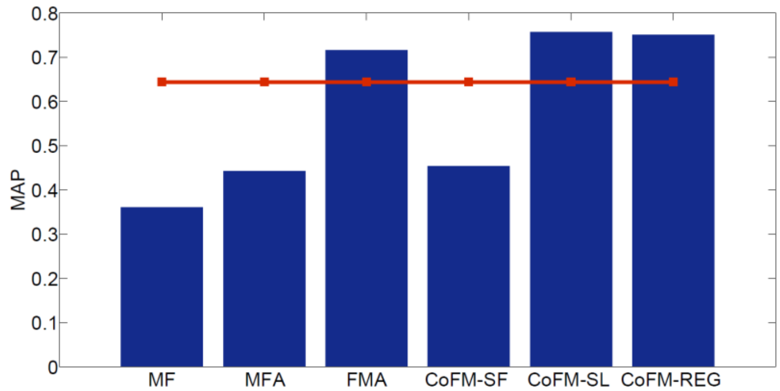

Figure 2: The results on retweet prediction. The red line is the baseline `CPTR`**.**

factorization with attributes (MFA), 3) Collaborative personalized tweet recommendation (CPTR [1]), 4) Factorization machines with attributes (FMA), 5) CoFM with shared features (CoFM-SF), 6) CoFM with shared latent spaces (CoFM-SL) and 7) CoFM with latent space regularizations (CoFM-REG).

The overall performance of retweet prediction is shown in Figure 2 where the loss function of WARP is chosen as it is best performed. The red line across all bars is the baseline CPTR. The first observation is that the performance of MF that only uses user-item interactions is significantly worse than the ones utilizing explicit features. The second observation is that CoFM-SL and CoFM-REG are noticeably better than all other methods. This validates our discussion before that these two paradigms can be viewed as variants of many successful co-factorization models where the predictive aspect can benefit from the content modeling aspect. On the other hand, CoFM-SF performs poorly and even cannot match the performance of FMA. We also observe that FMA performs better than MFA, indicating that ternary interactions “target user-item neighbor user” can indeed capture the dynamics between users on Twitter, compared to “target user-item” binary interactions. We also noted that FMA, CoFM-SL and CoFM-REG are much better than the others, where all three are above CPTR significantly. In addition, CoFM-SL and CoFM-REG are consistently 3%−4% better than FMA in absolute MAP scores across 5 split of data.

We explore how topics are learned through the modeling as the topic matrix can be interpreted as the one obtained in standard PLSA/LDA. Thus, we can describe topics as in other topic models by ranking terms in probabilities. This is superior to CPTR [1] where term factors are not in simplex. We show some example topics in Table 1. We can easily see that these topics are easily recognized and have the benefit of normal topic models while using a simpler model. Note, however, that content modeling is not only for explanatory analysis—it is indeed helpful for prediction tasks. From Figure 2, we can see that CoFM that utilizes content modeling has better performance in general, and especially for CoFM-SL and CoFM-REG which can outperform state-of-the-art methods significantly.

Table 1 Examples of topics are shown. The terms are top ranked terms in each topic. The topic names in bold are given by the authors.

Entertainment
album music lady artist video listen itunes apple produced movies #bieber bieber new songs
Finance
percent billion bank financial debt banks euro crisis rates greece bailout spain economy
Politics
party election budget tax president million obama money pay bill federal increase cuts

Last, we study how different types of features contribute to the predictive power of the model. Instead of using methods like 2 to calculate the correlation between feature values with respect to classification labels, we adopt two simpler methods. First, we start from a base model CoFM-SL using WARP without any explicit features, which is the best model from previous experiments, and then add one group of features consecutively. This method, denoted

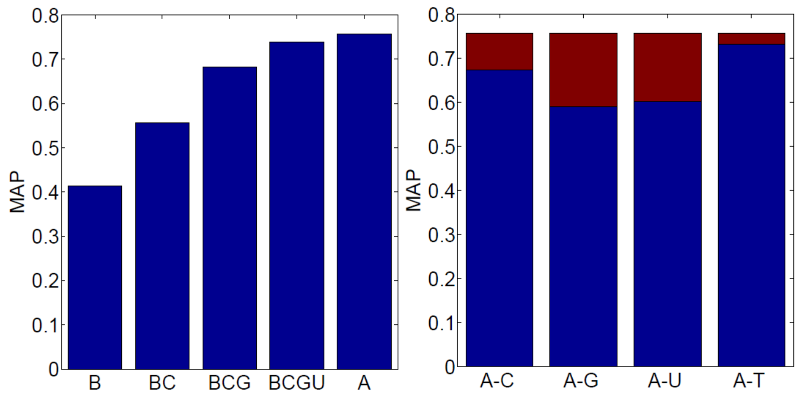

Figure 3: The impact of different groups of features. The effect of "add on" is shown on the left and the effect of "take out" is on the right. For both figures, "A", "B", "C", "G", "U" and "T" stand for "All", "Base model", "Content feature", "Graph feature", "User feature" and "Temporal feature" respectively.

as "add on", would show the contribution of each group of features as it adds into the model. The second method, denoted as "take out", starts with a complete model and removes one group of features to see how performance drops accordingly. The results for "add on" and "take out" are shown in Figure 3. For "add on", it is clear that each group of features contributes to the final performance of the model and "Temporal" features have the least marginal gain. The most gains come from "Content" features and "Local Graph" features. For "take out", again, "Temporal" features have the least impact on performance while removal of "Local Graph" features hurts performance much more than that of "Content" features. From both "add on" and "take out", it seems that "Local Graph" plays an important role in the performance, followed by "Content" features. This may indicate that social connections are more important in determining retweets as well as content factors.

4. CONCLUSION

Users of social media services are often simultaneously overwhelmed with the amount of information delivered via their social connections and miss out on content that they might have liked to see. Both issues serve as difficulties to the users and drawbacks to the services. Social media service providers can benefit from understanding user interests and how they interact with the service, potentially predicting their behaviors in the future. We propose Co-Factorization Machines (CoFM) to address the problem of simultaneously predicting user decisions and modeling content in social media by analyzing rich information gathered from Twitter. The task differs from conventional recommender systems as the cold-start problem is ubiquitous, and rich features, including textual content, need to be considered. Additionally, we discuss and compare ranking-based loss functions in the context of recommender systems, shedding light on how they vary from each other and perform in real tasks, providing the first work in this direction. We explore a large number of features and conduct experiments on a real-world dataset, concluding that CoFM with ranking-based loss functions is superior to state-of-the-art methods and yields interpretable latent factors.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation under Grant Number IIS-0803605.

REFERENCES

[1] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In SIGIR, pages 661–670, 2012..

[2] S. Rendle. Factorization machines with libFM. ACM TIST, 3(3):57:1–57:22, May 2012.

[3] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In ICML, pages 1057–1064, 2009.

[4] J. Zhu and E. P. Xing. Sparse topical coding. In UAI, pages 831–838, 2011.