Elver: Recommending Facebook Pages in Cold Start Situation Without Content Features

Yusheng Xie, Zhengzhang Chen, Kunpeng Zhang, Chen Jin, Yu Cheng, Ankit Agrawal, Alok Choudhary

Northwestern University
Evanston, Illinois 60201 USA
{yxi389,zzc472,kzh980,ych133,cji970,ankitag,choudhar}@eecs.northwestern.edu

***Abstract*—Recommender systems are vital to the success of online retailers and content providers. One particular challenge in recommender systems is the "cold start" problem. The word "cold" refers to the items that are not yet rated by any user or the users who have not yet rated any items. We propose ELVER to recommend and optimize page-interest targeting on Facebook. Existing techniques for cold recommendation mostly rely on content features in the event of lacking user ratings. Since it is very hard to construct universally meaningful features for the millions of Facebook pages, ELVER makes minimal assumption of content features. ELVER employs iterative matrix completion technology and nonnegative factorization procedure to work with meagre content inklings. Experiments on Facebook data shows the effectiveness of ELVER at different levels of sparsity.**

***Index Terms*—Recommender system, Facebook, Sparse matrix, Social media, Behavioral targeting**

I. INTRODUCTION

Recommender systems automate the process of suggesting interesting items to users based on their explicit and implicit preferences. The user preferences are perceived from their and similar users' activity history [1]. For example, Netflix uses recommender systems (instead of choosing at random) to suggest movies that a user might be interested in watching. Three components are widely assumed and thought to be indispensable for recommender systems. They are users, items, and users' ratings for the items. Assuming the aforementioned three components, two well-established approaches are widely used in constructing a recommender system: collaborative filtering and content-based filtering.

Collaborative filtering makes recommendation by considering inter-user similarities [2][3]. Collaborative filtering works well at web scale. However, one common difficulty in collaborative filtering systems is the "cold start" problem. The word "cold" refers to the items that are not yet rated by any user or the users who have not yet rated any items [4]. "Cold start" refers to the problem of recommendation involving cold items or users, which is common at early stage of the system. "Cold start" problems are well studied in the past decade [2][1][4]. Despite differences in algorithms and models, most existing approaches choose to rely on content-based filtering techniques to overcome cold starts [5][6].

Content-based filtering represents a user by the descriptive contents of the items rated by the user. For example, in the case of movie recommendations, a piece of descriptive content could be the movie's leading actor or director. A major benefit of using content-based filtering in recommender systems is its effectiveness in handling so-called "cold items". Since content-based filtering relies on content features and not the rating, the lack of user ratings will not affect the system. However, purely content-based filtering could potentially destroy the diversity among the recommended items. For example, consider a user watches 5 James Bond movies. Then content-based filtering would recommend the rest 18 James Bond movies and other action movies. As a result, the recommender system fails to give diverse recommendations for the user to explore different genres.

In addition to the diversity problem, a basic underlying assumption in handling cold items with content-based filtering is that any well-rated item, as well as a cold item, must be well represented in the same high-dimensional feature space. Feature extraction from item contents has never been a problem with items such as movies because each valid movie would have dozens of actors/actresses, which are plenty to make a good, discriminative feature vector.

A cold start situation becomes more challenging when a good feature space is unavailable. Because their content features are easy to extract, movies and music are the *nice* cases of the cold start problem. On Facebook, people express their interests and preferences via entities beyond music and movies. For example, on Facebook people can form groups based on very broad range of concepts. It is not surprising that there are group pages for celebrities like Justin Bieber[1]. But many more group pages are about far less solid concepts such as *Kids Who Hid in Dep't Store Clothing Racks while their Mom Was Shopping*[2]. Since Facebook has so many group pages with unprecedented volume (more than 10 million by business owners alone) and variety (virtually unclassifiable as illustrated above), designing a recommender system that suggests interesting group pages is very challenging.

We are particularly interested in the cold start scenario for recommending Facebook group pages since it bears significant commercial interests. Suppose that a small local retailing business, called *Crown Jewel*, decides to join Facebook and to start its own page in order to engage more potential customers. Facebook allows Crown Jewel to target its ads to users based on what pages they like (i.e. behavioral targeting). It is

[1] https://www.facebook.com/JustinBieber

[2] https://www.facebook.com/pages/kids-who-hid-in-dept-store-clothing-racks-while-their-mom-was-shopping/10150138929940447

978-1-4799-1293-3/13/$31.00 ©2013 IEEE

beneficial for Crown Jewel to consult a recommender system to find out which pages to target. Admittedly, recommending Facebook pages would not be an entirely new challenge if the ad placer is Walmart or Target because they are large brands and already have many fans. Existing collaborative filtering techniques would work well in such a scenario. But, to help Crown Jewel with its goal, a recommender system has to solve the cold start problem. Existing methods for the cold start problem, to the best of our knowledge, are all targeted for and tested on movie benchmark data and therefore make heavy use of movie's rich content features.

A. Our Contribution

We propose ELVER algorithm for recommending cold items from large, sparse user-item matrices. ELVER differs from existing cold recommendation techniques in two aspects. First, ELVER recommends and optimizes page-interest targeting on Facebook instead of movies or songs. The large scale of a social network like Facebook influences ELVER's design. We find that similarity-based user stratification can both improve the speed of ELVER as well as its accuracy. Second, ELVER makes minimal assumption of content features. Existing techniques for cold recommendation mostly rely on content features in the event of lacking user ratings. Traditional items (e.g., movies or music) have rich, organized content features like actors, directors, awards, etc. Since Pages on Facebook ideologically vary so much from each other, it is very hard to construct universally meaningful features for the millions of Facebook pages. ELVER uses iterative matrix completion technology and nonnegative factorization procedures to work with meagre content inklings and avoid heavy requirement on rich content features.

We conduct experiments on real Facebook pages (22.6K) with activities from real users (27.4M). To illustrate the effectiveness of ELVER in handling cold pages, three different levels of sparsity (99.9%, 99.99%, and 100.00%) in the user activities are tested.

II. Related Work

Related work to our paper can be roughly divided into two categories: notable existing solutions to the cold start problem, and non-negative matrix factorization (NNMF) algorithms.

A. Cold Start Recommendations

[6] represents a class of classical methods for dealing with cold-start problems. Specifically dealing with movie recommendations, [6] assumes that the content of each movie is represented by the participating actors in the movie. Other later works such as [2] and [1] follow a similar path, although each work motivates very different models for the underlying user preference variable(s). A key reason for their success is the fact that actors are expressive and discriminative features for the movie contents. As a consequence, a major difficulty in applying such techniques to the problem of Facebook page cold recommendation is that the available contents from each item are either not descriptive enough or not universal enough to make the modeling work. It can be very difficult to extract features that are both descriptive and universal for Facebook pages.

B. Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NNMF) [7] is a family of techniques to factorize a matrix V into two matrices W and H. Given a nonnegative matrix $V \in \mathcal{R}^{m\times n}$ and an integer k ($0 < k < \min\{m, n\}$), the NNMF goal is to find nonnegative matrices $W \in \mathcal{R}^{m\times k}$ and $H \in \mathcal{R}^{k\times n}$ such that the objective

$$f(W, H) = \frac{1}{2}||V - W \times H||_F^2 \tag{1}$$

is minimized [7]. Most NNMF algorithms solve Equation 1 by iterative steps, which require initialization of W and H. Efforts have been made in seeding W and H with some meaningful values [7].

III. Problem Statement

Suppose that V is a n-by-m user-page matrix and $V(i, j) \in \{0, 1, 2, 3, \ldots\}$ for $1 \le i \le n$ and $1 \le j \le m$. $V(i, j)$ is determined by the number of comments left on the j-th page by the i-th user. Recall that our motivation is to perform recommendations for a cold page that has just entered the Facebook ecosystem. Without loss of generality, assume that the cold page of concern is in the 1st column, which is referred to as $V(:, 1)$. In addition to V, our formulation also includes content *inkling*. Content inkling is different from traditional content feature by its general lack of discriminative or descriptive nature. Only three pieces of content inkling are assumed for each page: 1) page likes received from fans, 2) the category assigned to the page by Facebook, and 3) the age group, with which the page is most popular.

Our goal can be expressed as recommending indices j for $1 < j \le m$ such that, for all suggested j, $V(:, j)$ is targetable for $V(:, 1)$ to acquire fans.

IV. The Elver Algorithm

ELVER works with three matrices: the user-page matrix V, the content-page matrix H, and the user-content matrix W. ELVER consists of 4 basic algorithmic steps: 1) encoding content inkling, 2) initializing the user-page matrix V into $\tilde{V}$, 3) user stratification, and 4) a stochastic Expectation Maximization algorithm that iteratively updates $\tilde{V}$ and H. Eventually, recommendations for a cold Facebook page are made from the converged $\tilde{V}$.

A. Encoding Content-page Matrix From Content Inkling

The three pieces of inkling assumed available for a Facebook page are inconvenient to work with in numeric matrix operations. Further encoding is therefore needed in H, which is our notation for the content-page matrix.

First, the number of page likes from fans, $H(1, :)$, is already in integer form. But the actual numbers can vary so greatly from page to page that they cannot be directly used. For example, pages for business like Crown Jewel in our example may have a few hundred likes while global pages such as Coca-Cola have over 50 million likes. Simply normalizing and putting such vastly different numbers in H will effectively set most entries to be 0. We use $\log(\text{number of page likes})$ instead. Second, the category $H(2,:)$, to which the page belongs, is a categorical variable. Distinct categories can be mapped to discrete integer values. There are two choices to be made here. One is which categories should get greater/less integer values. The other is whether similar categories should be assigned to closer integer values. Third, the age group, $H(3,:)$, should also be encoded into real numbers, since different age groups can be compared and ordered like integers.

Three additional statistically derived [8] (from the first three inklings above) inklings are also be part of H: $H(4,:) = H(1,:) \times H(2,:)$, $H(5,:) = H(1,:) \times H(3,:)$, and $H(6,:) = H(2,:) \times H(3,:)$. Note that H in this paper is always 6-by-m in dimension.

B. Initializing User-page Matrix

The original user-page matrix V is *incomplete* from two aspects. First, its cold column $V(:,1)$ contains few or no known entries. This aspect of incompleteness defines the cold start problem: it is our goal to provide estimates for the values in $V(:,1)$. The second aspect is the distinction between a zero entry and an unknown entry. It is well known that in the Netflix problem, some users do not rate certain movies not because they are uninterested in giving a rating after watching the movie (i.e. a zero entry), but because they are never aware of the movies' existence (i.e. an unknown entry). The same situation exists with users and pages on Facebook.

The initialization of V is the process of finding the most likely matrix $\tilde{V}$, based on nonzero entries in V [9]. For doing so, ELVER uses matrix completion technologies [10] to a version of $\tilde{V}$ such that $\tilde{V}$ is in accordance with V and is low-rank. The exact optimization goal is given below:

$$\begin{aligned} &\text{minimize } ||\tilde{V}||_* \\ &\text{subject to } \sqrt{\sum_{(i,j)\in\Omega_0} |\tilde{V}(i,j) - V(i,j)|^2} < \epsilon, \\ &\quad \text{where } \Omega_0 = \{(i,j) | 1 \le i \le n, 1 \le j \le m\} \\ &\quad\quad \setminus \{(i,1) | V(i,1) = 0\}. \end{aligned} \tag{2}$$

In Equation 2, $||\cdot||_*$ denotes matrix nuclear norm.

Completing matrix under minimal nuclear norm constraint is a member of the family of convex cone problems. Fast, first-order algorithms for solving such convex cone problems can be constructed from general templates [10].

C. User Stratification in $\tilde{V}$

User stratification is the process of dividing users into individual strata, on which independent analysis and estimation are carried out. We say that $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is a possible stratification of the users if $\cup_{i=1}^{|S|} s_i = \{1, 2, \ldots, n\}$ and $s_i \cap s_j = \emptyset$ for $i \neq j$. Since each user is represented as a unique row in $\tilde{V}$, stratification can be done by clustering the n rows of $\tilde{V}$ into $|S|$ non-overlapping clusters. We use K-means to find the clusters. In the experiments, we vary $|S|$, the number of clusters, and evaluate their performance with n. Cosine distance is used to measure the distance between $\tilde{V}(u,:)$ and $\tilde{V}(v,:)$.

Algorithm 1: User stratification and Stochastic Expectation-Maximization (EM) algorithm in ELVER

```
1  $\{\tilde{V}_1, \tilde{V}_2, \ldots, \tilde{V}_{|S|}\} \leftarrow$ kmeans($\tilde{V}$, $|S|$, "cosine distance")
2  for $\tilde{V}_i \in \{\tilde{V}_1, \tilde{V}_2, \ldots, \tilde{V}_{|S|}\}$ do
3      $W^{(0)} \leftarrow \tilde{V}_i \times H^\top \times (H \times H^\top)^{-1}$; $H^{(0)} \leftarrow H$
4      for $j \in \{1, 2, \ldots, itr_{\max}\}$ do
5          $\tilde{V_i}^{(j)} \leftarrow W^{(j-1)} \times H^{(j-1)}$
6          $W^{(j)}, H^{(j)} \leftarrow$ nnmf($\tilde{V_i}^{(j)}$) using $H^{(j-1)}$ as
           initial guess for $H^{(j)}$
7      end
8      Approximately, $\tilde{V_i}^{(\infty)} \leftarrow \tilde{V_i}^{(j)}$
9  end
10 return $\tilde{V}^{(\infty)} \leftarrow \left[\tilde{V}_1^{(\infty)}\ \tilde{V}_2^{(\infty)}\ \ldots\ \tilde{V}_{|S|}^{(\infty)}\right]^\top$
```

Suppose that non-empty clusters are discovered as user strata, we can represent $\tilde{V}$ as $\left[\tilde{V}_1\ \tilde{V}_2\ \ldots\ \tilde{V}_{|S|}\right]^\top$.

D. Stochastic EM algorithm

After encoding H, initializing $\tilde{V}$, and stratifying the $\tilde{V}$ into $\tilde{V}_i$'s, ELVER employs a stochastic Expectation-Maximization (EM) algorithm to search for $\tilde{V}^{(\infty)}$, the converged user-page matrix. Algorithm 1 describes the stochastic EM algorithm. In Figure 1, we visualize the convergence of $\tilde{V}$ over 500 iterations. The error rate at the ith iteration of the jth stratum is measured between $\tilde{V_j}^{(i-1)}$ and $\tilde{V_j}^{(i)}$. The error rate between $\tilde{V}^{(\infty)}$ and V is not shown in Figure 1 but in Figure 2.

The matrix multiplication in line 6 of Algorithm 1 is interpreted as the E(xpectation) step. The non-negative factorization in line 7 of Algorithm 1 is interpreted as the M(aximization) step. The factorization algorithm is not exact simply because $\tilde{V_i}^{(j)}$ is not guaranteed to be a product of two nonnegative matrices $W^{(j)}$ and $H^{(j)}$. By minimizing the objective function in Equation 1, the factorization maximizes the "fit" of $W^{(j)}$ and $H^{(j)}$.

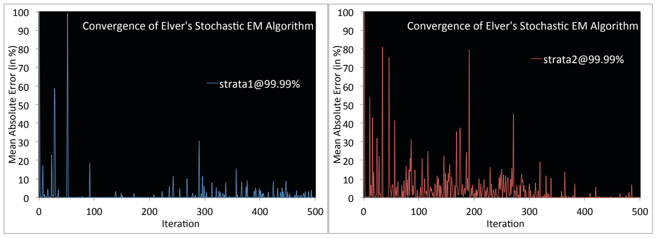
EM algorithm is designed for fitting incomplete data to the models. Similarly, the iterations in Algorithm 1 try to recover missing values in $\tilde{V}$ and H.

V. EVALUATION METRICS

A. Mean Absolute Error

"Normalized Mean Absolute Error" (NMAE) is used in [1] as a performance metric for recommender systems. NMAE is a metric that uses "Mean Absolute Error" (MAE), which is also a standard evaluation metric [11]. MAE can be calculated through *user averaging*:

$$MAE_u = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{w_i}{m \sum_{k=1}^{n} w_k} \sum_{j=1}^{m} |\tilde{V}(i,j) - V(i,j)| \right], \tag{3}$$



(a)

(b)

Fig. 1. Stratum-wise convergence of the $\tilde{V}$ user-page matrix over 500 iterations. (a) Convergence for$\tilde{V_1}^{(1 \to 500)}$ at 99.99% sparsity; (b) Convergence for$\tilde{V_2}^{(1 \to 500)}$ at 99.99% sparsity.

where w_i is a weights assigned to each user. Another important component in NMAE is MAE_b, the MAE_u score when predictions on user preferences are made by a baseline algorithm. Then NMAE is defined as $NMAE = MAE_u/MAE_b$.

Both MAE_u and NMAE are built around users because, in a traditional recommender system setting, users are the only party that the system needs to please. Recommender systems have always been user-centric. But the situation presented in our running example of Crown Jewel is very different. In our case, Facebook wants to recommend the right pages for Crown Jewel so that targeting them can convert the most users to fans of Crown Jewel given the same budget. This becomes an entirely different problem and therefore needs a better-suited metric in addition to MAE-based ones.

B. Using Kendall τ-statistics

The system's goal is to recommend an *ordered* list of pages for Crown Jewel to target. Crown Jewel would be interested in getting the pages to target in the right order, based on which the campaign manager can go down the list as far as he/she likes or his/her funds allow.

The computation of the new performance metric consists of two steps. First, it needs to assign a rank to each page column $\tilde{V}(:,i)$ for $2 \leq i \leq m$. We assign the rank based on $\tilde{V}(:,i)$'s distance from $\tilde{V}(:,1)$. Five different ways of measuring inter-column distance are applied and compared. They are L_1-*norm*, L_2-*norm*, L_{Inf}-*norm*, *Mean Square Error*, and *Frobenius Norm*.

The second step is to compare obtained rankings R_i for $1 \leq i \leq 5$ (one for each distance measure) with the optimal ranking order R_o. R_o is determined apriori and experimentally. We use Kendall's tau(τ) coefficient [12] to measure how similar R_i is to R_o. The coefficient range is $-1 \leq \tau \leq 1$, where 1 implies perfect agreement between rankings.

VI. Experiments

A. Dataset

In the experiments, the data collected over 2012 is used [13]. Table I shows more details. Simple rules are applied to filter out potential spam users: we remove a user from the experiments if he/she is active on over 100 pages or is only active on 1 page.

TABLE I
Datasets summary statistics

Statistic	Available	Used in paper
Unique pages	32K+	22,576
All user activities across all pages	10B+	226M
Unique users	740M+	27.4M

B. Masking Entries in $V(:,1)$

In order to evaluate the Elver algorithm, we place a non-sparse column at $V(:,1)$, then masking most of its entries. This scheme can simulate a cold page at $V(:,1)$ and can still reliably evaluate how well Elver performs since we know the hidden values in $V(:,1)$. Three different levels of sparsities in $V(:,1)$ are tested in our experiments: 99.9%, 99.99%, and 100.00%. These numbers are chosen because traditional "cold start" settings often choose between 95% 99.9% sparse. We emphasize that the 100.00% sparsity simply indicates that $V(:,1)$ is an empty column.

In this section, we use the notation E(x,y) and B(x,y) respectively to denote the results when Elver and Random Bot baseline algorithm are applied to V with x-level sparsity in $V(:,1)$ and y user strata.

C. Random Bot Baseline Method

A Random Bot algorithm [1] is used as the baseline approach in our experiments. The algorithm injects user activity values, generated from random variables, in $V(:,1)$. There are $|S|$ Random Bots, $X_1, X_2, \ldots, X_{|S|}$, for the $|S|$ user strata. $V_i(:,1)$ is filled by values drawn from $X_i \sim \mathcal{N}(\hat{\mu_i}, s_i^2)$, where

$$\hat{\mu_i} = \frac{1}{n}\sum_{j=1}^{n_i} x_j,\ s_i^2 = \frac{1}{n-1}\sum_{j=1}^{n_i}(x_j - \hat{\mu_i})^2,$$

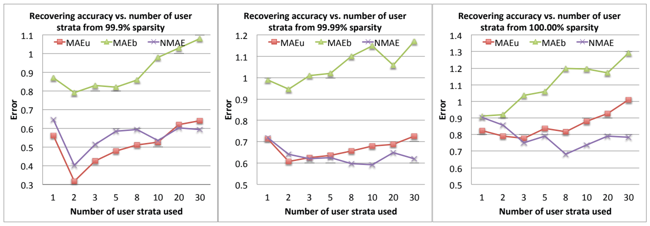
and $x_1, \ldots, x_{n_i}$ are existing activity values in the ith stratum of V.

TABLE II
Kendall τ-statistic based accuracy of recovering user-page matrix $\tilde{V}$ from 99.9% sparsity in $V(:,1)$ under various norms and using different numbers of user strata.

User Strata	L_1-norm	L_2-norm	L_{Inf}-norm	Mean Square Error	Frobenius Norm
1	0.543	0.543	0.543	0.543	0.543
2	0.608	0.608	0.608	0.556	0.608
3	0.386	0.346	0.330	0.346	0.346
5	0.438	0.438	0.438	0.438	0.438
8	0.333	0.307	0.229	0.307	0.307
10	0.281	0.111	0.111	0.111	0.111
20	0.124	0.111	0.111	0.085	0.111
30	0.163	0.085	0.085	0.085	0.085

D. Quantifying Results with Mean Absolute Error

Figure 2 shows how accurately the Elver algorithm can recover the entries in $V(:,1)$. On one hand, the error rates are consistent with the sparsity levels as the overall error rates decrease as sparsity decreases, which basically shows that the Elver algorithm can effectively make use of the sparse ratings in $V(:,1)$ when they are present. On the other



(a) (b) (c)

Fig. 2. Recovering accuracy of ELVER algorithm, compared to random baseline prediction (MAE_b) under 3 different levels of sparsity of $V(:,1)$: 99.9%, 99.99%, and 100.00%. Three mean absolute error rates are used: MAE_r, MAE_u, and $NMAE$.

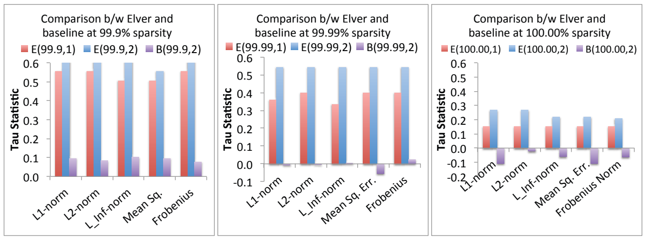
hand, the error rates vary less monotonically with the number of user strata chosen. As shown in Figure 2, 1 stratum, 2 strata, or 3 strata seem to be good choices that can find balance between capturing inter-user similarity/inter-stratum distinction and avoiding sparsity in small stratum.

E. Quantifying Results with Kendall τ-statistics

Table III provides the Kendall statistics for three configurations $E(100.00, 2)$, $E(99.99, 1)$, and baseline results for 99.99% sparsity. Both configurations of the ELVER algorithm have significantly higher τ values than the baseline, implying that our rankings are closer in agreement compared to original true rank R_o. Figure 3 presents results of τ-statistics with the ELVER algorithm applied to more configurations of different numbers of user strata, distance measure, and data sparsity. τ-statistics metric can really tell apart the performance in random bot baseline and ELVER. Stratification is also effective in improving τ-statistics for ELVER.

TABLE III
KENDALL τ RANK CORRELATION TABLE SHOWING DETAILED STATISTICS FOR RANKINGS BY E(100.00,2), E(99.99,1) AND B(99.99,2) UNDER THE L_1-NORM MEASURE.

Measure	**E(100.00,2)**	**E(99.99,1)**	**B(99.99,2)**
Kendall τ-statistic	0.2679	0.3595	-0.0075
2-sided p-value	0.1297	0.02803	0.9396
S, Kendall Score	41	59	-3
Var (S)	697	697	697
S/τ, Denominator	153	153	153



(a) (b) (c)

Fig. 3. τ-statistic-based accuracy of recovering user-page matrix $\tilde{V}$ under 3 different levels of sparsity of $V(:,1)$: 99.9%,99.99%, and 100.00%. Results are shown in L_1-norm, L_2-norm, L_{Inf}-norm, Mean Square Error, and Frobenius Norm.

VII. CONCLUSION AND FUTURE WORK

Most existing techniques for cold recommendation mostly rely on content features in the event of lacking user ratings. ELVER makes minimal assumption of content features: It is very hard to construct universally meaningful features for the millions of Facebook pages. ELVER uses iterative matrix completion technology and nonnegative factorization procedure to work with meagre content inklings and avoid the heavy requirement on rich content features. Our experiments on Facebook data have confirmed the effectiveness of ELVER at three different levels of sparsity.

In the future, we plan to improve ELVER's cold recommendation by incorporating the network information among social users, which is largely ignored by existing studies.

ACKNOWLEDGMENT

This work is supported in part by the following grants: NSF awards CCF-0833131, CNS-0830927, IIS-0905205, CCF-0938000, CCF-1029166, and OCI-1144061; DOE awards DE-FG02-08ER25848, DE-SC0001283, DE-SC0005309, DESC0005340, and DESC0007456; AFOSR award FA9550-12-1-0458.

REFERENCES

[1] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naive filterbots for robust cold-start recommendations," in *KDD '06*. ACM, 2006, pp. 699–705.

[2] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *ICDM '10*, 2010, pp. 176–185.

[3] Y. Cheng, Y. Xie, Z. Chen, A. Agrawal, and A. Choudhary, "Jobminer: A real-time system for mining job-related patterns from social media," in *KDD '13*. ACM, 2013.

[4] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *KDD '09*. ACM, 2009, pp. 397–406.

[5] A. V. Bodapati, "Recommendation systems with purchase data," *Journal of Marketing Research*, vol. 45, no. 1, pp. 77–93, 2013/06/01 2008. [Online]. Available: http://dx.doi.org/10.1509/jmkr.45.1.77

[6] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *SIGIR '02*. ACM, 2002, pp. 253–260.

[7] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," in *Computational Statistics and Data Analysis*, 2006, pp. 155–173.

[8] Y. Cheng, K. Zhang, Y. Xie, A. Agrawal, and A. Choudhary, "On active learning in hierarchical classification," in *CIKM '12*. ACM, 2012.

[9] X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *WSDM '13*, 2013.

[10] S. Becker, E. Cands, and M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, 2011.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.

[12] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. pp. 81–93, 1938. [Online]. Available: http://www.jstor.org/stable/2332226

[13] Y. Xie, Z. Chen, K. Zhang, M. M. A. Patwary, Y. Cheng, H. Liu, A. Agrawal, and A. Choudhary, "Graphical modeling of macro behavioral targeting in social networks," in *SDM 2013*. SIAM, 2013.