

# Spotting Trends: The Wisdom of the Few

Xiaolan Sha  
EURECOM  
Sophia Antipolis  
France  
xiaolan.sha@eurecom.fr

Pietro Michiardi  
EURECOM  
Sophia Antipolis  
France  
pietro.michiardi@eurecom.fr

Daniele Quercia  
The Computer Laboratory  
University of Cambridge  
United Kingdom  
daniele.quercia@cl.cam.ac.uk

Matteo Dell’Amico  
EURECOM  
Sophia Antipolis  
France  
matteo.dell-  
amico@eurecom.fr

## ABSTRACT

Social media sites have used recommender systems to suggest items users might like but are not already familiar with. These items are typically movies, books, pictures, or songs. Here we consider an alternative class of items - pictures posted by design-conscious individuals. We do so in the context of a mobile application in which users find “cool” items in the real world, take pictures of them, and share those pictures online. In this context, temporal dynamics matter, and users would greatly profit from ways of identifying the latest design *trends*. We propose a new way of recommending trending pictures to users, which unfolds in three steps. First, two types of users are identified - those who are good at uploading trends (trend makers) and those who are experienced in discovering trends (trend spotters). Second, based on what those “special few” have uploaded and rated, trends are identified early on. Third, trends are recommended using existing algorithms. Upon the *complete* longitudinal dataset of the mobile application, we compare our approach’s performance to a traditional recommender system’s.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

## General Terms

Experimentation

## Keywords

Mobile, Social Media, Trend Detection

## 1. INTRODUCTION

Good user experience is what makes online services attractive. To be outstanding, many services not only try to provide easy access to the content of what users are looking for, but also attempt

to help them discover new information which they might be interested in. Mostly, they use recommender systems to give personalized suggestions to each user. Depending on the context of the applications, recommended items are of all kinds [2, 6, 9, 10, 20, 21, 25, 32, 33, 34].

We here consider a social mobile application called iCoolhunt<sup>1</sup>. Its users are encouraged to take pictures of objects they think “cool” and share these pictures with friends online. The idea of “hunting cool” items has attracted a very special community - users are mainly technology-savvy and design-conscious individuals. In such a context, helping users to discover trends becomes important. As we shall see in Section 2, in the literature, there have been studies on what trends are and who creates them. But no study has been yet conducted to show how such knowledge can be leveraged to make personalized recommendation on trends.

To fill the gap, we propose a trend-aware recommender system (Section 4). We build upon the insight offered by Amatriain *et al.* [2]: that is, the recommendation process should not always rely on many (crowd) ratings but might also benefit from *few* expert ratings. By exploiting this insight, we make two main contributions:

- We propose a method that detects trends by paying attention to the activities of two types of users: trend makers (those who upload items that become trends) and trend spotters (those who rate items before they become trends).
- We construct a preference matrix based on the identified trends, and test the extent to which a state-of-the-art matrix factorization algorithm (*Implicit SVD* [14]) effectively recommends trends.

We evaluate the effectiveness of our trend-aware recommender system at helping users discover trends. To this end, we compare our approach to a traditional item-based recommender system (Section 5). We then conclude by discussing how such a trends-aware recommender approach could be applied to a real-life working system (Section 6).

## 2. RELATED WORK

This paper builds upon existing studies in two main research areas: personalized recommendations, and analyses of trends in social networks.

---

<sup>1</sup>[www.icoolhunt.com](http://www.icoolhunt.com)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
*RecSys '12*, September 9–13, 2012, Dublin, Ireland.  
Copyright 2012 ACM 978-1-4503-1270-7/12/09...\$15.00.

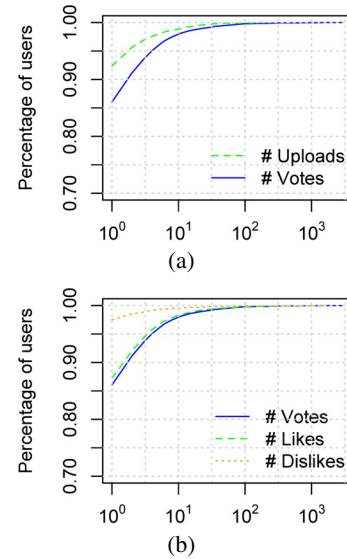
*Personalized Recommendations.* Recommender systems are used in different online services. Traditionally, studies have been focused on recommending books, CDs [21], movies [2, 34], songs [18, 32], news [9], and videos [10]. With the advent of mobile applications, many applications have been able to know where users are, and some services have thus started to recommend location-based events [25], activities, and *POIs* (Points of Interests) [33]. Adding the users’ social connections to their geographic information has been found to improve the quality of recommendations [15, 19]. Also, new social connections have themselves become “items to recommend” [6, 24]. There has been a lot of work on algorithms over the last few years (a useful categorization of them can be found in Adamic *et al.*’s work [1]), and effective techniques such as matrix factorization have emerged [14, 15, 33].

*Analyses of trends in social networks.* “Trends” are items (e.g., pictures, videos) that receive abrupt attention and are of two types: endogenous and exogenous depending on the sources that triggered them. Endogenous trends are triggered by collective activities *within* the user community, while exogenous ones result from activities *outside* the community. In Youtube, researchers have been able to distinguish between endogenous and exogenous trends by simply looking at temporal patterns (more specifically, at the user response rates) [8]. In Twitter, instead, researchers identified trends based on a richer set of features - content, user interactions and social networks [23]. In addition to identify trends, researchers have also looked at the human beings behind trends and have tried to answer the question of who creates them. In his popular book “The Tipping Point”, Malcolm Gladwell argued that the creators belong to the “special few”. Those are often called “influentials” [12] and are believed to be characterized by large social networks [17] and disproportionately high social standing [27, 29, 31]. By contrast, Watts argues that trends are created by accidental activities, that is, by adopting the right item at the right time [28]. More recently, researchers are coming round to the idea that trends in a networked environment are actually generated by a combination of accidental activities and presence of influential individuals [3, 13].

These two lines of research (i.e., recommender systems and trend analysis) are here brought together to study whether one could build simple ways of facilitating discovery and recommendation of trends. We experimentally do so in the context of a mobile social-networking application with which users discover, organize and share pictures of “cool” design items.

### 3. APPLICATION

iCoolhunt users are encouraged to take pictures of objects that they think ‘cool’, upload them and share them with friends online. If their devices are GPS-enabled, pictures are automatically tagged with (*longitude*, *latitude*) points corresponding to where pictures are uploaded. Uploaders are asked to tag each of their pictures with one of the predefined categories, which include technology, lifestyle, music, design and fashion. They must also add a brief textual description to each of their pictures. By following each other, users can then vote others’ pictures using ‘like’ and ‘dislike’ buttons and can comment on them. Users are automatically assigned to different levels of expertise depending on the quality of their uploads and votes, and the number of followers they attract. Our anonymized dataset is complete, in that, it covers all user activities from February, 2010 (its launch) to August 2010 (before the time when iCoolhunt launched their web application).



**Figure 1: Empirical CDF of 1(a) #uploads and #votes per user; 1(b) #likes, #dislikes and #votes per user.**

### 3.1 Dataset Analysis

Our dataset includes 9,316 iCoolhunt mobile application users, 6,395 photos, and 21,252 votes. To filter away inactive users and lurkers (which, based on our data, are indistinguishable), we focus on users who have uploaded or voted at least once.

**Uploads and Votes.** In the mobile application, uploading and voting pictures are two main user activities. From the distributions of uploads and votes per user in Figure 1(a), we see that, as one expects, most of the users (92%) have uploaded only once, or voted (85%) only once. The remaining active minority have contributed 83% of pictures and 94% of votes. This results in data sparsity: the fraction of non-zero values in the user-item matrix is around 1%.

Users can vote others’ pictures using ‘like’ and ‘dislike’ buttons. Figure 1(b) shows the distribution of the number of ‘likes’, ‘dislikes’ and the total number of votes per user. One can see that the distribution of number of ‘likes’ is very similar to the total number of votes per user, suggesting that users tend to frequently express their opinions.

**Social Connectivities.** Users can also create social contacts and follow each other. We depict the distribution of the number of followers/followees per user in Figure 2, and find that only a very small portion of users follow others, suggesting that using social ties to recommend content would not work in this specific application.

From this brief analysis, two main points emerge: 1) there is a core group of users who contributed most of the content; and 2) relying only on the social graph to recommend trends would not be beneficial. For these two reasons, we propose a way of recommending trends based on the idea of identifying the “special few”.

## 4. TREND-AWARE RECOMMENDATION

To recommend trends, we perform three steps (Figure 3):

1. Identify trend makers and trend spotters (Section 4.1);
2. Identify trends (Section 4.2);

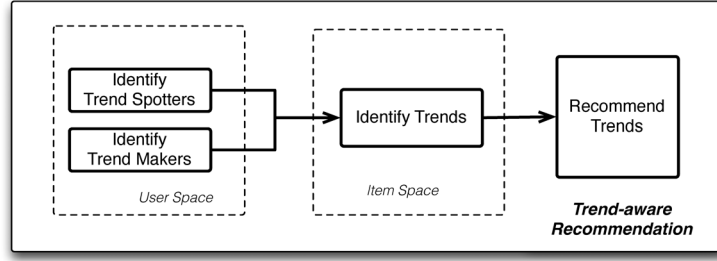


Figure 3: Trend-aware Recommender System

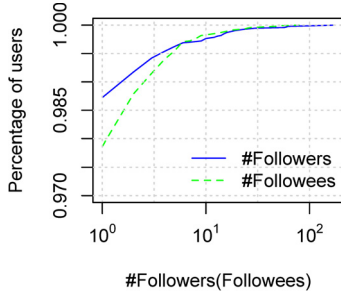


Figure 2: Empirical CDF of #followers and #followees per user.

3. Recommend the previously identified trends (Section 4.3).

In Section 6 on “Discussion”, we will explain why it is not a good idea to identify trends directly and, instead, it is beneficial to identify trend spotters (makers) first and then trends later on.

#### 4.1 Trend makers and trend spotters

In every social application, there are large behavioral differences among users [22]: some are able to identify trends early on, and some are leisure laggards. To identify the former type, we focus on two user categories – trend makers and trend spotters.

Trend makers are those who tend to *upload* items that then become trends, and trend spotters are those who tend to *vote* items that then become trends early on. To quantitatively identify those users, we consider the following features:

**Activity.** The main activities on the application are two and are voting and uploading. From them, we compute three activity features: *daily votes*, *daily uploads* and *lifetime*. This last feature reflects whether users are early adopters (i.e., are those who shape social norms [4]) or not [11].

**Content.** Users vary in how diverse their interests are: one could have a wide variety of interests, while another one could “focus” on very specific and limited set of interests. In Twitter, for example, it has been shown that influential users focus on very specific topics [5, 29]. To differentiate users based on their interest diversity, we consider two measures of content diversity. Both use the Shannon Index and are called *upload diversity* and *vote diversity*.

**Social Network.** Since information might partly propagate along social connections, we also account for how well a user is connected by considering *number of followers*, *number of followees*, and the user’s *clustering coefficient* (computed on the social graph in which each node represents a user, and an edge links two users with at least one following relationship [28]).

**Geography.** We finally consider: 1) how much and how often one user is *wandering* in the real world by using the radius of gyration [7]; and 2) a user’s *geographical span of followers* computed as the average distance between where the user is and where his/her followers are.

Having these features, we now perform three steps. For each user, we:

**Step 1** Compute the user’s spotter score and maker score;

**Step 2** Discretize the user’s scores.

**Step 3** Predict the user’s discretized scores on input of the previous features of activity, content, social network, and geography.

Let us now spell out each of the steps.

**Step 1.** To begin with, we introduce two metrics that reflect the extent to which a trend maker (spotter)  $u$  is successfully uploading (spotting) trends. User  $u$ ’s *makerScore*( $u$ ) is:

$$\text{makerScore}(u) = \frac{\sum_{i \in \mathcal{I}_u} I(i \text{ is a trend})}{|\mathcal{I}_u|}, \quad (1)$$

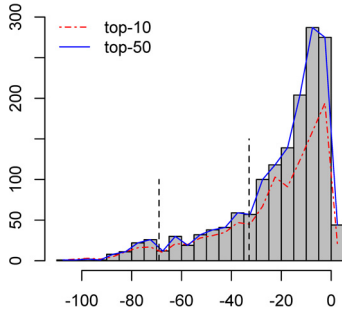
where  $\mathcal{I}_u$  is the set of trends that  $u$  has uploaded, and  $I(i \text{ is a trend})$  is an indication function which equals to 1, if  $i$  is a trend; otherwise, it is 0. To establish whether an item is a trend or not, we use a metric similar to the one proposed in [23]. That is, for each time unit  $t$ , each item  $i$  is assigned with a *trendScore*( $i, t$ ) computed as:

$$\text{trendScore}(i, t) = \frac{|v_{i,t}| - \mu_i}{\sigma_i}, \quad (2)$$

where  $|v_{i,t}|$  is the number of votes item  $i$  has received within time unit  $t$ ,  $\mu_i$  is the mean number of votes it received per time unit, and  $\sigma_i$  is its standard deviation. A high trend score tells that the item have received more attention than expected within the time unit. In each time unit, items are sorted according to their trend scores, and top- $N$  items are extracted and identified as trends. From our analysis, we found that the temporal resolution (one week or two weeks) and the length of the recommended list do not significantly change the scores. In Figure 4, one observes that the trend spotter score does not change as the list length (top-10 vs. top-50) changes.

To add the spotter score to the maker score, we observe that the ability of spotting trends is largely determined by three factors – how many, how early, and how popular one’s spotted trends become. To incorporate the factor of how early and how popular trends become, for each trend  $i$  that  $u$  has spotted (voted), we compute the following gain  $g_{u,i}$  score:

$$g_{u,i} = v_i \times \alpha^{-p_{u,i}}, \quad (3)$$



**Figure 4: Trend spotter score (log).** We split trend spotters into three classes using a proportional 3-interval discretization, as the two vertical lines show.

in which  $v_i$  is the total votes  $i$  received,  $p_{u,i}$  captures that  $u$  is the  $p^{th}$  user who spotted  $i$  (the lower  $p$ , the better), and  $\alpha$  is a decay factor. Combining a user's gains all together, we obtain a cumulative *spotterScore* for user  $u$  (which is normalized by user  $u$ 's total number of votes):

$$\text{spotterScore}(u) = \frac{\sum_{i \in \mathcal{I}_u} g_{u,i}}{v_u}, \quad (4)$$

**Step 2.** Based on their maker scores and trend scores, we are able to cluster users into  $K$  classes, which indicate their ability of uploading (spotting) trends. To do so, we apply a proportional  $k$ -interval discretization [30] over the whole range of maker (spotter) scores and assign each user to one of the three classes (with  $k = 3$ ). We chose three because the distribution of trend spotter(maker) scores in Figure 4 shows three distinct increments in score and thus lends itself to the identification of three classes of users.

**Step 3.** Based on a user's (activity, content, social network, and geographic) features, machine learning models predict to which class the user belongs. We tried a variety of models - Naive Bayesian, linear regression, and *SVM* - and found that *SVM* works best (Section 5).

## 4.2 Identify Trends

Trend makers and trend spotters are the source of trends, but not all items uploaded and voted by those users become trends - there is a certain probability that they will be so. More generally, an item is likely to become a trend depending on: the extent to which the item's uploader is a trend maker; and the extent to which the item's voters are trend spotters. We model these insights in the following logistic regression:

$$\Pr(y_i = 1) = \text{logit}^{-1}(X_i \beta) \quad (5)$$

where  $\Pr(y_i = 1)$  is the probability of an item  $y_i$  is a trend ( $\Pr(y_i = 1)$ ), and  $X_i$  are a set of predictors, which are the uploader's trend maker class and the voters' trend spotter classes. Results are again presented in Section 5.

## 4.3 Recommend Trends

Having identified items that are likely to be trends, we are now able to build a trend-aware preference matrix  $\mathcal{P}'$ , in which  $p'_{u,t}$  is 1 or 0 depending on whether  $u$  liked item  $t$  that has then become a trend. Upon this matrix, we apply two existing recommender systems algorithms: *Implicit SVD* [14] and item-based collaborative filtering [21]. We compare how these algorithms perform on input of the trend-aware matrix and on input of a traditional preference

matrix  $\mathcal{P}$ , in which  $p_{u,t}$  is 1 or 0 depending, again, on whether  $u$  voted on item  $t$  that has then become a trend. The difference between the two preference matrices is that the trend-aware one is less sparse because, at the columns, it does not have all items but only those that we have predicted to be trends.

## 5. EVALUATION

We now evaluate the effectiveness of each of the three steps we have just introduced.

**Classifying users into trend spotter(maker) classes.** We first evaluate the extent to which *SVM* is able to classify each user into one of the three maker/spotter classes on input of the user's features (introduced in Section 4.1). To this end, upon our dataset that has 209 unique trends, 50 trend makers, and 531 trend spotters, we run a 10-fold cross validation and test three algorithms: linear regression, Naive Bayesian, and *SVM*. The best accuracies are returned by *SVM*: 83.80% of trend spotters and 60.7% of trend makers are correctly identified.

**Determining whether an item is a trend or not.** After ascertaining that *SVM* is able to reasonably identify trend spotters and trend makers, we now need to test whether the logistic regression in Section 4.2 is able to identify trends based on information about uploaders and voters. The regression predicts whether an item is a trend or not based on four features: the uploader's trend maker class (first feature), and the number of votes from users who belong to: the low spotter class (second feature), the medium spotter class (third feature), and high spotter class (fourth feature). To test the logistic regression, we build a balanced dataset that contains our 209 trends plus 209 (randomly extracted) non-trends and obtain the results in Table 1. The statistically significant coefficients suggest that an item is more likely to become a trend, if its uploader is a good trend maker and its voters are in the upper (trend spotter) class.

To avoid overfitting in Equation 5, we add a Tikhonov regularization term. The problem of learning  $\beta$  now translates into the following optimization problem:

$$\beta' = \arg \min_{\beta} \sum_i \log(1 + \exp(-y_i \beta X_i)) + \lambda \|\beta\|_2^2 \quad (6)$$

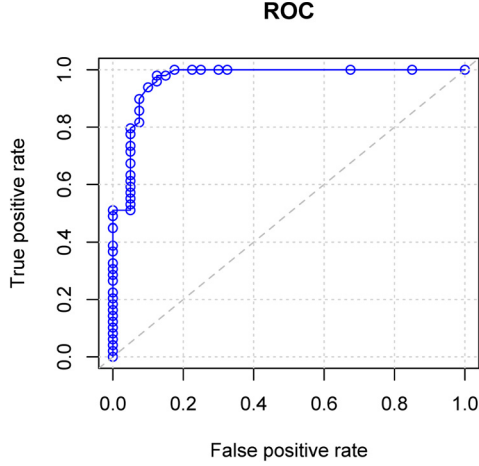
We split the dataset of trends into two subsets: the first subset consists of 80% of the entire dataset and is used to *train* the model, while the remaining 20% is used to *test* the model. With a 10-fold cross validation, we first fix the value of  $\lambda$  and then fit the model with the training set. To measure the accuracy of the regularized logistical regression model, we apply the trained model to the test set. We obtain the ROC curve plot that reflects both the model's TPR (true positive rate) and FPR (false positive rate). In Figure 5, one sees that the regression returns accurate results, whereby the point (0,1) corresponds to the best classification performance (the diagonal reflects the baseline of random guess).

**Recommending trends.** At this point, we have ascertained our ability to identify trends. Now the question is: if we were to build a *user-by-trend* matrix out of the predicted trends, what would be the performance of an existing collaborative filtering algorithm? To answer that question, we need to establish three things:

1. We need to select which existing algorithm to use. For now, we pick a simple item-based collaborative filtering algorithm [26]. Later, we will see whether we can improve performance with *Implicit SVD* [14].

Predictors	Coefficient
Uploader's trend maker class	6.21 ***
#Voters from low trend spotter class	-1.30
#Voters from medium trend spotter class	-1.17 *
#Voters from high trend spotter class	0.64 ***

**Table 1: Coefficients of the logistic regression (a correlation coefficient within 2 standard errors is statistically significant. The significance levels are marked with \*'s:  $p < 0.001$ (\*\*\*),  $p < 0.01$ (\*\*),  $p < 0.05$ (\*)**



**Figure 5: ROC curve for the logistic regression that predicts whether an item is a trend or not.**

2. We need to determine which metrics reflect performance. To be in line with the literature, we compute precision and recall [14]:

$$recall(N) = \frac{\#hits}{|T|} \quad (7)$$

$$precision(N) = \frac{\#hits}{N * |T|} \quad (8)$$

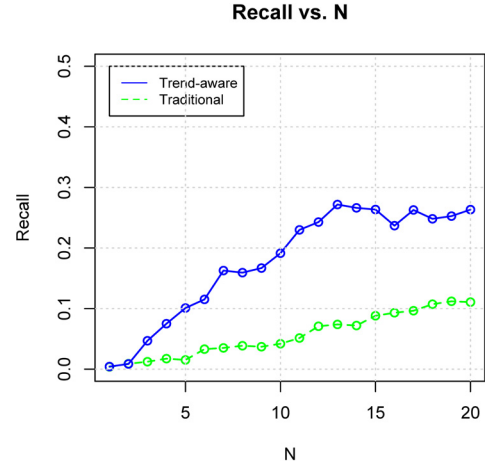
where  $T$  is the test set, and  $N$  is number of items to recommend.

3. Finally, we need to determine the baseline against which our trend-aware approach will be compared. To ease interpretability of the results, we again select item-based collaborative filtering but, this time, the algorithm would take in input the original user-item preference matrix  $\mathcal{P}$ , in which:

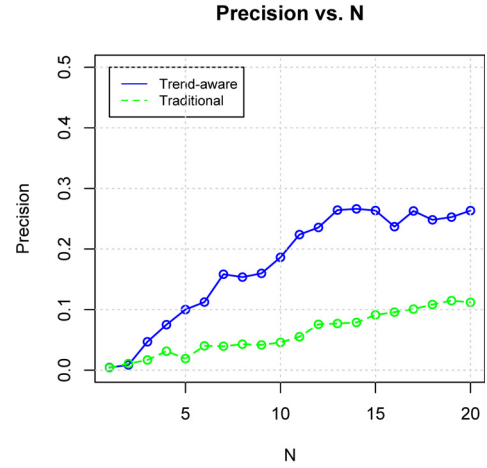
$$p_{u,i} = \begin{cases} 1 & \text{if } u \text{ likes } i \text{ \& } i \text{ is a trend} \\ 0 & \text{otherwise} \end{cases}$$

Figure 6 shows precision and recall for the traditional and trend-aware item-based collaborative filtering as a function of the recommended list size (top- $N$  recommendations). For both systems, precision and recall improve linearly as  $N$  increases, but the trend-aware's increase is faster, suggesting that a traditional item-based recommender system would not be able to recommend trends (precision/recall is only 0.05 for top-10 recommendations), while a trend-aware system would (precision/recall is 0.2). These results also suggest that, in the presence of data sparsity, relying on few expert ratings is an effective way of recommending trends.

So far, we have analyzed how an item-based collaborative filtering algorithm would perform. Next, we test whether a matrix



(a) Recall vs. top- $N$



(b) Precision vs. top- $N$

**Figure 6: Precision and Recall. Results for trend-aware recommender vs. item-based recommender. The size of the recommended list is  $N$ .**

factorization approach - *Implicit SVD* - would improve the performance. Figure 7 shows this to be the case. As the size of the recommended list increases, *Implicit SVD* consistently returns better precision and recall than item-based's.

Interestingly, if one were to recommend only popular trends, precision and recall would be worst. This suggests that even for trends - items that one expects to be non-long tail - personalization makes sense. But up to a point - precision and recall results are limited, and that is largely because of the very nature of trends.

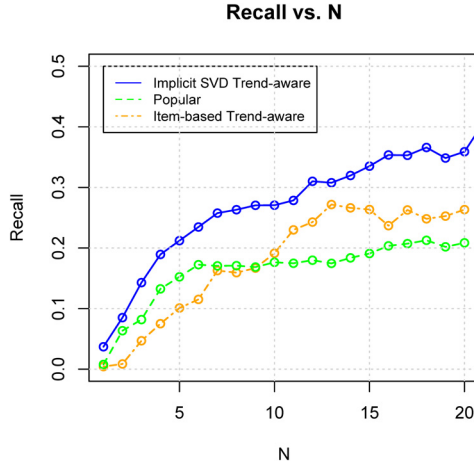
To sum up, the results on precision and recall should be interpreted in comparative terms, while the other results answer a more fundamental question - whether trend detection helps the recommendation process; and the answer is a definite 'Yes'.

## 6. DISCUSSION

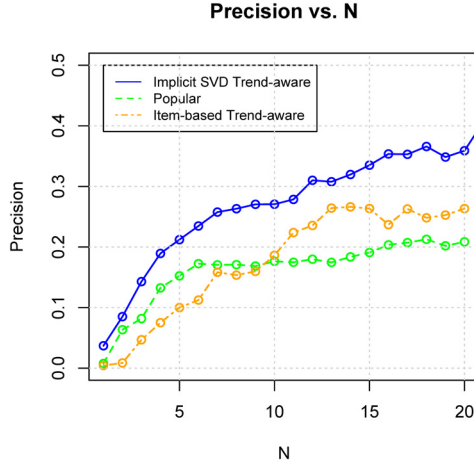
We now discuss some open questions.

**Why not detecting activity bursts directly?** Since a burst detection algorithm could be easily applied to identify trends (in a way similar to expression (2)), one might wonder why we are going to the trouble of having the intermediate step of identifying trend





(a) Recall vs. top- $N$



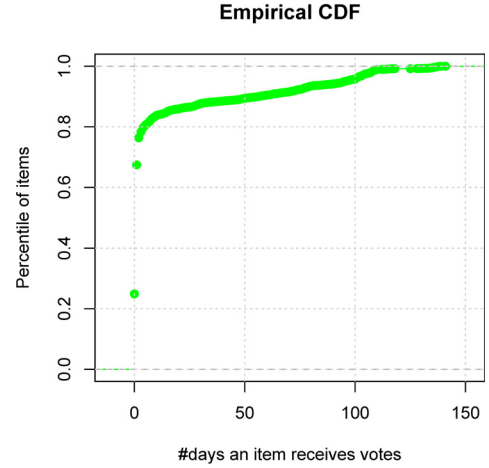
(b) Precision vs. top- $N$

**Figure 7: Precision and Recall. Results for two trend-aware recommenders (item-based and Implicit SVD) and for recommendations of most popular trends.**

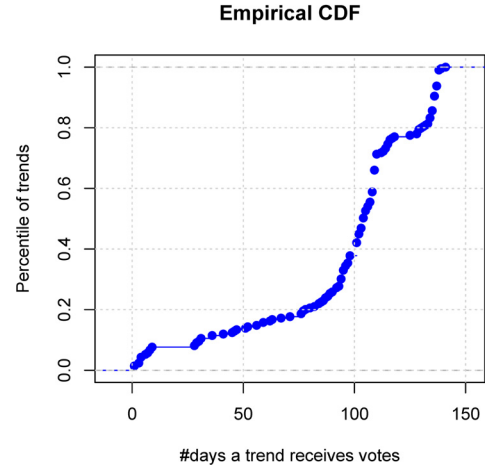
spotters (makers) and not, instead, identifying trends directly. The main reasons for this choice are efficiency and time:

1. *Efficiency.* Monitoring a limited number of users who are trend spotters and trend makers is more efficient than requiring the complete and up-to-date view of the system: a typical burst detection algorithm would require to monitor *all* votes on *all* items.
2. *Time.* This is the most important reason and comes from the temporal dynamics typical of trends. The average item is generally short-lived and dies off the first or second day (Figure 8(a)); by contrast, a trend persists for a longer period of time (as one would expect based on preferential attachment<sup>2</sup>), yet it also takes off only after two weeks or so (Figure 8(b)). As such, burst detection would miss trends for a long period of time, while monitoring key individuals - trend spotters (makers) - can be done quickly and efficiently. One contribution of this paper has been to show that monitoring

<sup>2</sup>[http://en.wikipedia.org/wiki/Preferential\\_attachment](http://en.wikipedia.org/wiki/Preferential_attachment)



(a) Item Lifetime



(b) Trend Lifetime

**Figure 8: Number of days an item (a) vs. a trend (b) receives votes for.**

key individuals not only is quick and efficient but is also an accurate way of identifying trends.

**Online Updating.** In our analysis, we have not registered the frequent emergence of new trend spotters and trend makers. However, in a system with a larger user base, that might be the case, and ways of updating the pool of key users - trend spotters (makers) - would be needed. To decide when and how to run such updates, we are currently exploring the use of controllers that automatically and accurately estimate frequency of updates. These techniques have been recently introduced with the idea of ensuring stable and high-quality recommendations in dynamically evolving environments [16].

## 7. CONCLUSION

We have shown that, upon activity, network, and geographic attributes, a machine learning approach can identify key users - trend spotters and trend makers. A simple logistic regression can then reliably infer whether an item (a picture in this case) will be a trend or not based on whether the item has been uploaded by a successful trend maker and voted by trend spotters. We have then seen that existing recommender systems can profit from this ability of

identifying these special users. One promising future research direction is to simultaneously model the two processes here treated separately - collaborative filtering and trend detection. This could be done by, for example, combining Amatriain *et al.*'s work [2] with existing models of temporal dynamics.

## 8. ACKNOWLEDGEMENTS

We thank Neal Lathia for his comments on earlier drafts of the paper. The work was funded by the French National Research Agency (ANR) under the PROSE project (ANR-09-VERS-007), by an industrial scholarship from PlayAdz<sup>3</sup>, and by RCUK through the Horizon Digital Economy Research grant (EP/G065802/1).

## 9. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005.
- [2] X. Amatriain, N. Lathia, J. Pujol, H. Kwak, and N. Oliver. The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web. In *Proceedings of the 32<sup>nd</sup> ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 2009.
- [3] S. Aral and D. Walker. Identifying Influential and Susceptible Members of Social Networks. *Science*, June 2012.
- [4] D. Boyd. The Future of Privacy: How Privacy Norms Can Inform Regulation. Invited Talk at the 32<sup>nd</sup> International Conference of Data Protection and Privacy Commissioners, October 2010.
- [5] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Proceedings of the 4<sup>th</sup> International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.
- [6] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make New Friends, But Keep the Old: Recommending People on Social Networking Sites. In *Proceedings of the 27<sup>th</sup> ACM International Conference on Human Factors in Computing Systems*. ACM, 2009.
- [7] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring Millions of Footprints in Location Sharing Services. In *Proceedings of the 5<sup>th</sup> International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.
- [8] R. Crane and D. Sornette. Robust Dynamic Classes Revealed by Measuring the Response Function of a Social System. In *Proc. of PNAS Vol.105*, 2008.
- [9] A. Das, M. Datar, A. Garg, and S. Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *Proceedings of the 16<sup>th</sup> ACM International Conference on World Wide Web*, 2007.
- [10] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In *Proceedings of the 4<sup>th</sup> ACM Conference on Recommender Systems*, 2010.
- [11] C. Droge, M. Stanko, and W. Pollitte. Lead Users and Early Adopters on the Web: The Role of New Technology Product Blogs. *Journal of Product Innovation Management Vol.27*, 2010.
- [12] M. Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. Little, Brown and Company, 2000.
- [13] S. Gonzalez-Bailon, J. Borge-Holthoefer, A. Rivero, and Y. Moreno. The Dynamics of Protest Recruitment through an Online Network. December 2011.
- [14] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8<sup>th</sup> IEEE International Conference on Data Mining*, 2008.
- [15] M. Jamali and M. Ester. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. In *Proceedings of the 4<sup>th</sup> ACM conference on Recommender systems*, 2010.
- [16] T. Jambor, J. Wang, and N. Lathia. Using Control Theory for Stable and Efficient Recommender Systems. In *Proceedings of the 21<sup>st</sup> International Conference on World Wide Web (WWW)*, 2012.
- [17] D. Kempe, J. Kleinberg, and E. Tardos. Influential Nodes in a Diffusion Model for Social Networks. In *Automata, Languages and Programming Vol.3580*. Springer Berlin / Heidelberg, 2005.
- [18] N. Koenigstein, G. Dror, and Y. Koren. Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy. In *Proceedings of the 5<sup>th</sup> ACM conference on Recommender systems (RecSys)*. ACM, 2011.
- [19] I. Konstas, V. Stathopoulos, and J. Jose. On Social Networks and Collaborative Recommendation. In *Proceedings of the 32<sup>nd</sup> International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.
- [20] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Chang. Pfp: Parallel FP-growth for Query Recommendation. In *Proceedings of the 2<sup>nd</sup> ACM Conference on Recommender Systems*, 2008.
- [21] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing*, 2003.
- [22] M. Muller, N. S. Shami, D. R. Millen, and J. Feinberg. We are all Lurkers: Consuming Behaviors among Authors and Readers in an Enterprise File-Sharing Service. In *Proceedings of the 16<sup>th</sup> ACM International Conference on Supporting Group Work (GROUP)*, 2010.
- [23] M. Naaman, H. Becker, and L. Gravano. Hip and Trendy: Characterizing Emerging Trends on Twitter. *Journal of the American Society for Information Science and Technology Vol.65*, 2011.
- [24] D. Quercia and L. Capra. FriendSensing: Recommending Friends Using Mobile Phones. In *ACM RecSys*, 2009.
- [25] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. Recommending Social Events from Mobile Phone Location Data. In *Proceedings of the 10<sup>th</sup> IEEE International Conference on Data Mining (ICDM)*, 2010.
- [26] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10<sup>th</sup> International ACM Conference on World Wide Web (WWW)*, 2001.
- [27] J. Tang, J. Sun, C. Wang, and Z. Yang. Social Influence Analysis in Large-scale Networks. In *Proceedings of the 15<sup>th</sup> International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.
- [28] D. Watts and S. Strogatz. Collective Dynamics of Small-world Networks. *Nature Vol.393*, 1998.
- [29] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank:

<sup>3</sup><http://www.playadz.com>

- Finding Topic-sensitive Influential Twitterers. In *Proc. of the 3<sup>rd</sup> ACM WSDM*, 2010.
- [30] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [31] L. Yu, S. Asur, and B. A. Huberman. What Trends in Chinese Social Media. *The 5<sup>th</sup> SNA-KDD Workshop*, 2011.
- [32] Y. Zhang, D. Séaghdha, D. Quercia, and T. Jambor. Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the 5<sup>th</sup> ACM International Conference on Web Search and Data Mining*. ACM, 2012.
- [33] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative Location and Activity Recommendations with GPS History Data. In *Proceedings of the 19<sup>th</sup> ACM International Conference on World Wide Web*, 2010.
- [34] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale Parallel Collaborative Filtering for the Netflix Prize. *Algorithmic Aspects in Information and Management*, 2008.