A Holistic Hybrid Algorithm for User Recommendation on Twitter

Sara Guimarães, Marco Túlio Ribeiro, Renato Assunção and Wagner Meira Jr.

Universidade Federal de Minas Gerais, Brazil
{sara, marcotcr, assuncao, meira}@dcc.ufmg.br

Abstract.

As Twitter grows larger and larger, finding interesting users to follow becomes an increasingly difficult task, making it a great scenario for the application of recommender systems. Previous research has shown that there is value in combining different recommendation algorithms, as each algorithm has strengths and weaknesses. However, previous works have focused on specific classes of recommendation algorithms, or on naïvely combining different algorithms. In contrast, in this work we present a holistic hybrid algorithm that simultaneously takes into account content-based, collaborative-based and user-based information. Our algorithm learns how to combine different sources of evidence (including the output from other algorithms) from the data itself, by using a Logistic Regression model. Therefore, instead of manually determining the importance of each source, or worse - weighting all the sources equally, the appropriate emphasis given to each of the sources in our model comes from the data. Our experiments on a real dataset from Twitter show that our algorithm outperforms current state-of-the-art algorithms. In addition, we propose new user representations for content-based algorithms (such as algorithms based on tf-idf and LDA) that capture the users' interests more fully, by also taking into account the content posted by the people they follow. Our experiments also show that these new representations outperform traditional content-based algorithms.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Miscellaneous

Keywords: data mining algorithms, logistic regression, social media, topic models, user recommendation

1. INTRODUCTION

With the advent of digital information systems and the Internet, the dissemination of information went through a revolution. Suddenly, anyone was allowed to publish whatever they wanted, without the need of a professional publisher, printer, distributer, etc. However, even with the Internet, there still was some cost associated with distributing information: one had to either learn or hire someone in order to build and maintain a Website, and to make the Website known to the public (with the use of marketing, SEO, etc). Also, in order to get constant updates, users had to visit the websites they liked repeatedly.

The Web resources evolved significantly in the last 20 years, making it much easier to disseminate user-generated content through blogs, comment sections, and social networks, among others. A notable example is Twitter, an online social information network launched in 2006. Twitter enabled massive content generation and dissemination by simplifying everything. It allows users to post 140-character text messages (called *tweets*) to a constantly updating *public timeline* of user messages. Users receive other users' tweets by explicitly *following* them. And, most importantly, (almost) everyone participates. As of 2013, Twitter has over 200 million active users, including the president of

This work was partially supported by CNPq, CAPES, FAPEMIG, and InWeb – National Institute of Science and Technology for the Web.

the United States[1] and most media celebrities, generating over 400 million tweets each day[2]. Since anyone can post content to it, Twitter is also revolutionizing real-time news, as regular folk tweets about current events before the traditional media even knows about it. A notable example was the news about Osama Bin Laden's death, which reached Twitter before any major news outlet reported it. Actually, a user live-tweeted about the attack even before it was over, without knowing exactly what was going on[3].

Twitter is a typical scenario where *Information Overload* takes place. No user on Twitter has the time, or the patience, to look through 200 million active users, and filter out which ones are interesting. Much has been done in terms of identifying the most "important" users [Cha et al. 2010; Weng et al. 2010], but different people have completely different interests. This is where recommender systems come in, offering recommendations that are tailor-made to particular users. The task of recommending other users on Twitter is different from the traditional scenario, where items (such as songs, movies or products) are recommended to users [Adomavicius and Tuzhilin 2005; Ricci et al. 2011]. In Twitter, the objects of the recommendation (other users) are not well-defined, and are constantly being updated, as new content is posted. The proportions are also different. The well-known Netflix dataset [Bennett et al. 2007], for example, which deals with movie recommendation, contained 17,700 movies that could be recommended. In Twitter, a recommender system has to choose from over 200 million users.

Previous research on recommending users on Twitter has focused on recommendations based on user features (non-personalized), on content-based and collaborative based approaches or on hybrids, which combine two or more strategies [Hannon et al. 2010; Armentano et al. 2011; Pennacchiotti and Gurumurthy 2011]. In this work we propose a holistic hybrid algorithm, that simultaneously takes into account content-based, collaborative-based and user-based information. Content-based information is useful for obvious reasons: Twitter is used in order to disseminate content. Collaborative-based information leverages the network structure in order to predict the importance of a user v in respect to another user u. Finally, information about the users helps the algorithms determine what kind of user is being dealt with. For example, a certain user v might post only content related to soccer - which fits nicely to what interests user u, a soccer aficionado. However, if v has very few followers, this might indicate that v's content is not considered very useful by most people, even though it is about soccer - and thus v is probably not a good recommendation for u. On the other hand, if v is followed by most of u's friends, v may be a good recommendation after all. It is clear, then, that each source of information (content, collaborative or user-based) provides different insights into what might be a good recommendation for u.

In contrast to previous work on hybrids (which aggregated strategies naïvely), our algorithm learns how to combine different sources of evidence from the data itself, by using a Logistic Regression model. Therefore, instead of manually determining the importance of each source, or worse - weighting all the sources equally, the appropriate emphasis given to each of the sources in our model comes from the data. Our hybrid is general, in the sense that it is easy to incorporate or remove other algorithms or features into the model. Using real data from Twitter, we evaluate our algorithm and show that it outperforms current state-of-the-art techniques.

In building our hybrid, we found that the traditional content-based approaches were lacking, so we also propose a new way of representing users and potential followees. Instead of representing users by what they post, we represent users as a combination of what they post and the content posted by the users they follow. Most users are passive, while other users tweet only about a subset of their interests. However, we can assume that users usually follow people that disseminate content they are interested in (by definition). For example, a certain user might be interested in soccer, politics and religion, but choose not to talk about politics and religion in order to avoid controversy - while at the

[1]https://twitter.com/BarackObama
[2]https://blog.twitter.com/2013/celebrating-twitter7
[3]http://mashable.com/2011/05/01/live-tweet-bin-laden-raid/

same time following political and religious figures. Our experiments show that this new representation results in great improvements on the accuracy of content-based algorithms, including ones based on traditional information retrieval techniques and topic models.

The remainder of this work is organized as follows: Section 2 provides a summary of the current algorithms for user recommendation on Twitter. In Section 3, we present the dataset used in our experiments, and detail the evaluation methodology. Section 4 starts by detailing the algorithms and features we combine, and proceeds to present our hybrid algorithm. Section 5 is comprised of all of our results, and the work is concluded in Section 6.

2. RELATED WORK

Recommendation on Twitter is a problem that has been studied from multiple points of view. Kywe et al. [2012] propose a taxonomy of recommendation tasks on Twitter based on the type of function being helped by the recommendation (such as following a user, retweeting a tweet or mentioning a certain user in a tweet). In this work, we deal with the task denoted *Followee Recommendation* - recommending users that are likely to be of interest to a specific user.

Much work has been done on identifying influential users on Twitter. Cha et al. [2010], for example, observe that popular users (users with many followers) are not necessarily influential in terms of spawning retweets or mentions, although there is an obvious correlation. Weng et al. [2010] tried to identify influential users in certain topics. Based on this potential correlation, Krutkam et al. [2010] tried to leverage influence-related information (such as number of followers and number of topic-related lists the user is listed in) in order to recommend followees for users interested in Thai News (their recommendations are not personalized). However, finding influential users is not the same as recommending interesting users, as a certain user might not be interested in certain profiles, regardless of their influence in general or in a certain topic. This is one motivation for Garcia and Amatriain [2010] to use the activity (number of tweets) of users, in addition to popularity, in order to make recommendations. In this work, we do not restrict ourselves only to user measures, such as influence and activity. Instead, we incorporate simple user measures (such as number of followers) with more sophisticated and personalized techniques such as collaborative and content-based methods. This contrasts sharply with several current proposals which disregard the simple user measures in favor of content and/or collaborative based techniques [Hannon et al. 2010; Pennacchiotti and Gurumurthy 2011].

Pure collaborative filtering algorithms have not been widely applied to recommending users on Twitter despite their success in other domains, such as movie recommendation. Adomavicius and Tuzhilin [2005] provide a comprehensive survey of this research area. A more recent survey is done by Ricci et al. [2011]. One notable example for the task of ranking a set of items is Matrix Factorization optimized by Bayesian Preference Ranking (BPR-MF), proposed by Rendle et al. [2009]. This algorithm is shown to outperform traditional methods based on k-nearest neighbors and matrix factorization when the task is ranking, in both movie and e-commerce recommendation, which makes it a good candidate for ranking potential followees on Twitter. Also, while most of the research in recommender systems focuses on datasets where ratings are available, BPR-MF is designed to implicit feedback scenarios - such as following a user on twitter. To the extent of our knowledge, BPR-MF has not been used for user recommendation on Twitter until this work. We show that it greatly outperforms pure content-based techniques, a class discussed next. We use BPR-MF as a representative of the state-of-the-art in collaborative filtering techniques, both as a baseline and as a component in our hybrid algorithm.

As for pure content-based methods to recommend users, Pennacchiotti and Gurumurthy [2011] investigated the use of topic models, also known as Latent Dirichlet Allocation (LDA) [Blei et al. 2003]. They compare their approach with another pure content-based method, the traditional Vector

Table I. Dataset summary.

#users	#edges	#tweets	#RT (retweets)	period
17,069,982	1,470,000,000	476,553,560	71,835,017	06/01/2009 - 12/31/2009

Space Model with tf-idf weights [Baeza-Yates and Ribeiro-Neto 1999]. Their preliminary results show that LDA outperforms tf-idf. In this work, we argue that LDA and tf-idf work well in different circumstances, and are complementary - and therefore should be used together. Further, we present a new way of representing users for both tf-idf and LDA, which we show to be superior to the previous alternatives. Finally, instead of ignoring network and global user information, we incorporate these content-based methods as components in our hybrid algorithm.

The current most successful algorithms to recommend followees is in the form of hybrid algorithms. Armentano et al. [2011] combine collaborative filtering information, popularity and number of mentions in order to make recommendations. The three features are combined naïvely - by taking either their average or their product. Hannon et al. [2010] proposed a system for followee recommendation named Twittomender. Twittomender represents users with content and/or collaborative features, using the TF-IDF weighting scheme. We discuss their user representation in relation to ours in Section 4. The way they aggregate their collaborative and content algorithms is by either combining their scores (the authors do not specify how), or by using what seems to be a variation of Borda Count (again, not fully specified). These approaches suffer from many drawbacks, especially if they are used to combine different algorithms. For example, when one of the algorithms being combined is substantially inferior to the others, it could affect the final result negatively. The authors themselves note that the hybrid versions of Twittomender did not outperform their best collaborative algorithm in terms of precision.

In summary, our algorithm is set apart from previous hybrid methods in three ways: (1) we are able to combine content (both topic-based and word-based), collaborative and user features simultaneously; (2) instead of relying on naïve ways of combining different sources of evidence, we learn how to combine in an optimal way the different features from the data; (3) adding or removing sources of evidence from our algorithm is straightforward.

3. METHODOLOGY

In this section, we begin by presenting the dataset used in our experiments. Next, we describe the evaluation methodology we used. We introduce this information here since it is used in the sections that follow.

3.1 Dataset

This work uses data crawled from Twitter[4]. The dataset we used was obtained from the previous work of Yang and Leskovec [2011] and is estimated to contain about 20-30% of all public tweets shared during the collection period. We also obtained a follower network from Twitter, obtained from a previous work by Kwak et al. [2010]. Table I shows relevant information about our dataset.

We pre-processed the dataset, keeping only tweets written in English. We also removed from the dataset all users that have less than 15 followees and less than 1 tweet. This step reduces the dataset to 188,563 users, 3,903,985 tweets and 20,994,952 edges.

3.2 Evaluation Methodology

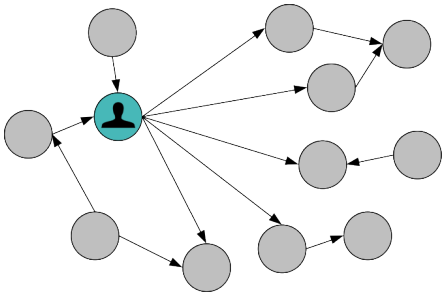
The aim of this work is to devise an algorithm to recommend users for a given user to follow (followee recommendation). The user on the receiving end of the recommendations is called a **target user**.

[4]www.twitter.com

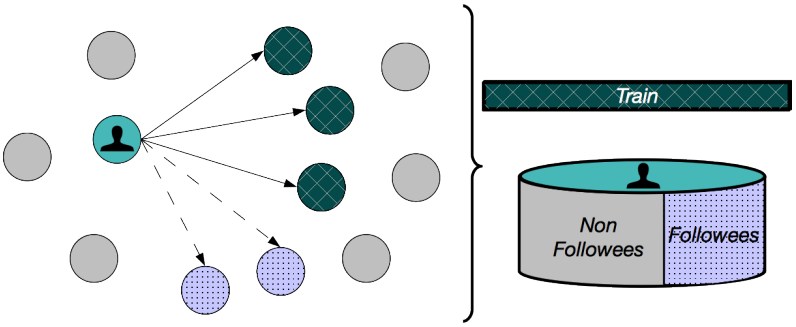
In order to evaluate our algorithms and the baselines, we selected a random sample of 1,000 target users. This set of users is our *test set*. For every user from the test set, we hide 10 followees (it is worth remembering that each user in our dataset follows at least 15 users), which we will use as a gold standard. These followees, therefore, are not considered by any of the algorithms evaluated. An illustration of this process is presented in Figure 1, where we hid only two followees instead of 10, for illustration purposes. We also constructed a validation set with the same process, with half the size of the test set, for the algorithms that need parameters tuning.

For each user u in the test set, we select as a positive set the 10 users that are followed by u (followees) and were hidden (the light purple dotted nodes in Figure 1). As a negative set, we select 10 users that are not followed by u (non-followees). We then evaluate the system on the task of ranking these positive and negative users. Ideally, a good algorithm would assign higher similarity scores to the positive set, rather than to the negative set, with respect to the target user. Figure 2 illustrates this process. The evaluation metric we used was area under the ROC curve (AUC) [Provost et al. 1997], which we calculate for each individual target user in the test set, and then average them. We also looked at the mean ROC curves themselves in order to compare different algorithms.

This evaluation methodology has been used previously by Pennacchiotti and Gurumurthy [2011]. Evaluating recommender systems on the whole universe of possible users would be troublesome for



(a) Suppose this is our dataset's network. The green node, which is highlighted by an user inside, is our target user.



(b) We select a random sample of our target user's followees and hide them (light purple dotted users). These users will be part of the test set, and will not be visible (as followees) to any of the algorithms used. The remaining followees, in dark green and with crossed lines, will be visible to the algorithms, and therefore belong to the training set. All the other users (in plain gray and no edges) are not followed by the target user, so they are part of this target user's set of non-followees.

Fig. 1. Sampling target user's followees for training and test sets.

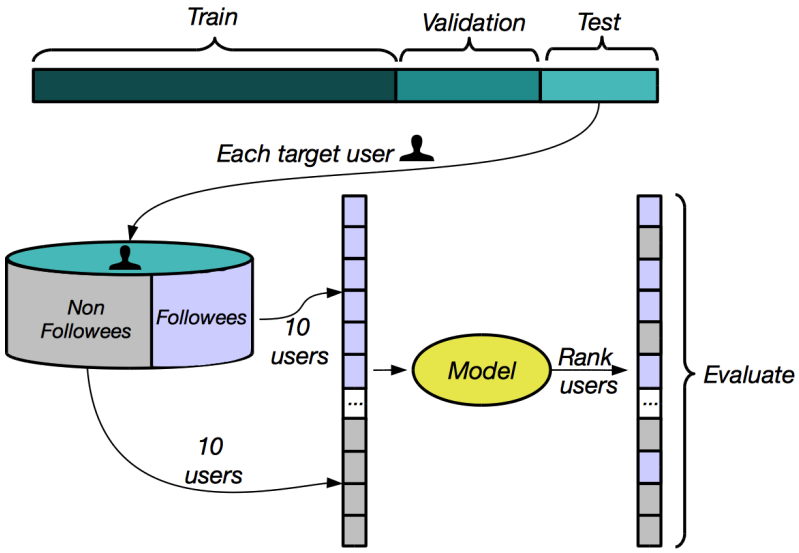


Fig. 2. An illustration of the evaluation methodology.

evaluation, due to the high sparsity of the following relationship. This methodology is an attempt to mitigate problems of this kind. Selecting a gold standard (the positive users set) and measuring the system's ability to rank them higher than users that are not followed (and thus assumed to be negative) is a way to evaluate top-n recommendations that is being accepted as standard by the recommender systems community in other domains [Koren 2008; Cremonesi et al. 2010].

4. ALGORITHMS FOR USER RECOMMENDATION

In this section we begin by describing different algorithms or features from three classes: content-based, collaborative-based and user-based. Our hybrid algorithm makes use of all of them. Then, we present our hybrid algorithm combining some of the algorithms and features presented.

4.1 Content-Based Algorithms

4.1.1 *Pure text.* In Twitter, each user posts his own content, and automatically receives the content posted by his followees. It is intuitive that users post content related to their interests. Following someone is an active choice made by the user, so it is reasonable to assume that the tweets of a user's followees provide insights into this user's interests. Although Hannon et al. [2010] represented users using four different content-based strategies, we do not agree that it makes sense to represent a user by the content posted by his followers, as the user has no control whatsoever on who follows him. When using pure text, we decided to represent each user in two different ways:

(1) Each user is represented by the content (the words) of his own tweets.
(2) Each user is represented by the combination of the content of his own tweets and the content posted by his followees.

To give some intuition for having both, we made a word cloud using Wordle[5] for each of the two user representations discussed above. The results can be seen in Figure 3. We selected a specific user (whose account name is *metacosm*), and presented a word cloud for the content posted by him, and

[5]www.wordle.com

a word cloud for the content posted by him and his followees. It is hard to determine what *metacosm* is interested in by looking at the word cloud formed by his own tweets, in Figure 3(a). However, by looking at the content posted by him and his followees in Figure 3(b), his interests become more clear. Words like groovy (a programming language), grails (a groovy framework) and maven (a build manager for Java projects) indicate that he is interested in programming. Words like broncos (a reference to the Denver Broncos) and Marshall (a reference to Brandon Marshall, a football player) are evidence that he is also interested in American Football.

When representing the content in each of the two options, we used the TF-IDF weighting scheme, where the weight of each term t for a certain user u is proportional to the frequency $tf_{t,u}$ of the usage of t (either by the user u or by his followees, depending on the representation) and inversely proportional to t's occurrence throughout the dataset (idf_t), as shown in Equation 1. The intuition is that the term t should receive a large weight if it is common in the text of user u (hence, characterizing

(a) User is represented by his own content

(b) User is represented as a combination of his own content and his followees'

Fig. 3. Word Cloud of the user *metacosm*, for each user representation.

u) and, at the same time, it is not common in the dataset (hence, discriminating who uses it). This weighting scheme has been successfully applied in a variety of contexts, including traditional Information Retrieval [Baeza-Yates and Ribeiro-Neto 1999]. Finally, we discarded stop words and words appearing less than 5 times in the dataset.

$$\begin{aligned} tf\_idf_{t,u} &= tf_{t,u} * idf_t \\ tf_{t,u} &= \begin{cases} 1 + log(freq_{t,u}), & \text{if } freq_{t,u} > 0 \\ 0, & \text{if } freq_{t,u} = 0 \end{cases} \\ idf_t &= log(N/n_t) \end{aligned} \tag{1}$$

A straightforward algorithm for recommending users on Twitter is to calculate the similarity between a target user u and every other user, and then to recommend the users that are most similar to u. The similarity score between two users, u_1 and u_2, represented as vectors of length n with entries w_u^i, is calculated using the cosine similarity, as shown in Equation 2. The tf-idf weights depend on the representation being used.

$$sim(u_1, u_2) = \frac{\sum_{i=1}^{n} w_{u_1}^i \times w_{u_2}^i}{\sqrt{\sum_{i=1}^{n} (w_{u_1}^i)^2} \times \sqrt{\sum_{i=1}^{n} (w_{u_2}^i)^2}} \tag{2}$$

Hannon et al. [2010] always used the same representation for both the target users and all the other users. We use this strategy, with the representation 1 (users are represented by their own tweeted words) as a baseline. We name this algorithm **tfidf**, and it is the same as S1 by Hannon et al. [2010]. However, as we argued before, it makes sense to represent the user by the combination of the tweets he posted and the tweets posted by his followees (representation 2), in order to discover his interests more fully - while representing all the other users with their own tweets (representation 1), since the target user can only benefit from the content they post, and not by the content they receive. We name this new algorithm **tfidf+f**, and we show in Section 5 that it greatly outperforms **tfidf**.

4.1.2 *LDA.* Instead of using word counts to represent content, we could divide the content up into topics (such as news, sports, etc) and represent users' content as the degree to which the content is represented by each topic. A user who is interested in soccer, golf and basketball, for example, would have a higher measure of the topic "sports" than a user who is interested in other types of news stories. A solution to this problem is using Latent Dirichlet Allocation (LDA), which is a generative model used in order to extract topics from documents in a non-supervised way [Blei et al. 2003]. Each document in a collection is seen as a mixture of different topics or themes. The proportion of each topic changes from document to document and the document-specific composition gives the signature of each document in the collection. What is striking in the LDA algorithm is that the topics are generated automatically, without requiring the pre-specification of themes or the labeling of the documents with topics or keywords. The topics do not come out with labels such as "sports", but rather as a list of most frequently words from which a label can be assigned. Table II shows the most representative words of a few topics that were discovered using LDA on Twitter in the way explained below. It is clear that, from the words, distinct topics can be easily identified.

In the traditional use of LDA, each document $u \in U$ is represented by a multinomial distribution θ_u over each one of the K topics. This distribution is traced from a Dirichlet prior, using α as parameter. The topics are also represented as multinomial distributions β_k derived from another Dirichlet prior with parameter γ. The generative model states that each word position n in a document stream is assigned a topic $z_{u,n}$ drawn from θ_u, and that the word in that position $w_{u,n}$ is drawn from the distribution $\beta_{z_{u,n}}$. Figure 4 is a representation of how this model works.

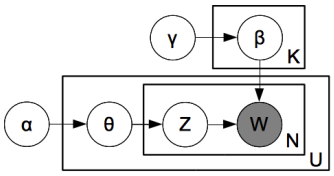


Fig. 4. LDA model.

Pennacchiotti and Gurumurthy [2011] applied LDA to recommend users on twitter. For each user u, they concatenated all tweets sent by u creating a single document containing all the words tweeted. LDA is then applied to the collection of documents and each user is represented by the distribution of topics of his tweets. The algorithm for recommending users becomes analogous to **tfidf**. Users that are similar to a target user u by means of the cosine similarity are recommended to u.

In an analogous way to what we proposed with **tfidf+f**, we argue that it makes more sense to represent a target user u as the distribution of topics of his own tweets and also of the tweets of his followees, while representing potential followees as the distribution of the topics of their own tweets. This will capture the target user's broad interests, even those that he does not tweet about. Hence, we propose a new algorithm for recommending users using LDA, which we name **lda+f**, and is analogous to **tfidf+f**. We show in Section 5 that **lda+f** greatly outperforms **lda**.

It is worth mentioning that we used PLDA, a parallel implementation of LDA proposed by Wang et al. [2009] in order to train the LDA model and infer the users' topics efficiently.

4.2 Collaborative-Based Algorithms

As we mentioned in Section 2, we applied BPR-MF [Rendle et al. 2009] to the task of recommending followees on Twitter. The application is straightforward: when a user u follows another user v, we add the relation $u - v$ as feedback. **BPR-MF** is shown to be an efficient recommendation algorithm in Section 5. We used an open source implementation of BPR-MF, by Gantner et al. [2011][6].

[6]http://www.mymedialite.net/

Table II. Some LDA topics discovered on Twitter.

# Topic	Topic Theme	Top Words
2	Tragedy News	news, police, bulletin, breaking, plane, dead, reports, found, crash, killed
6	Mobile Phones	mobile, blackberry, android, phone, nokia, app, iphone
9	Healthy Life	food, health, healthy, weight, eat, diet, more, body, foods, fat, exercise
12	Children	street, disney, show, sesame, mad, one, muppets, watching, men, marvel
17	Volunteering	relief, help, need, philippines, victims, donate, donations, red, typhoon, cross, pls, volunteers
18	Baseball	game, yankees, baseball, red, sox, now, mlb, phillies, cubs, reds, dodgers
28	Music	new, album, itunes, myspace, song, single, music, tour, check, video
39	Teen Movies	new, moon, twilight, rob, robert, harry, eclipse, more, taylor, pics, pattinson, kristen
43	Photography	photo, art, photography, photos, flickr, camera, photographer, project, images
50	Ecology	green, energy, water, eco, solar, save, new, power, home, wind, friendly
68	Pets	dog, dogs, cat, animal, pet, help, animals, cats, pets, puppy, shelter
73	News	news, times, new, journalism, nyt, journalists, story, york, bill, clinton
76	Football	game, nfl, football, season, team, favre, sports, vick, coach, play, espn, fans
81	Computers	windows, Microsoft, mac, snow, apple, leopard, os, pc, new, office, hd, google
98	Christianity	god, jesus, lord, church, pray, christ, faith, love, prayer, life, heart, bible
120	Food	food, eat, ice, coffee, chocolate, cream, chicken, cheese, bacon, day, recipe, pizza
159	U.S. Politics	health, care, obama, bill, house, senate, reform, president, sen, white, vote, public
162	Books	book, writing, read, story, write, writers, books, writer, new, fiction, novel
187	Video Games	game, games, ps, xbox, wii, play, video, live, new, gaming, nintendo
188	Tourism	travel, hotel, hotels, beach, thailand, best, world, flight, resort, island

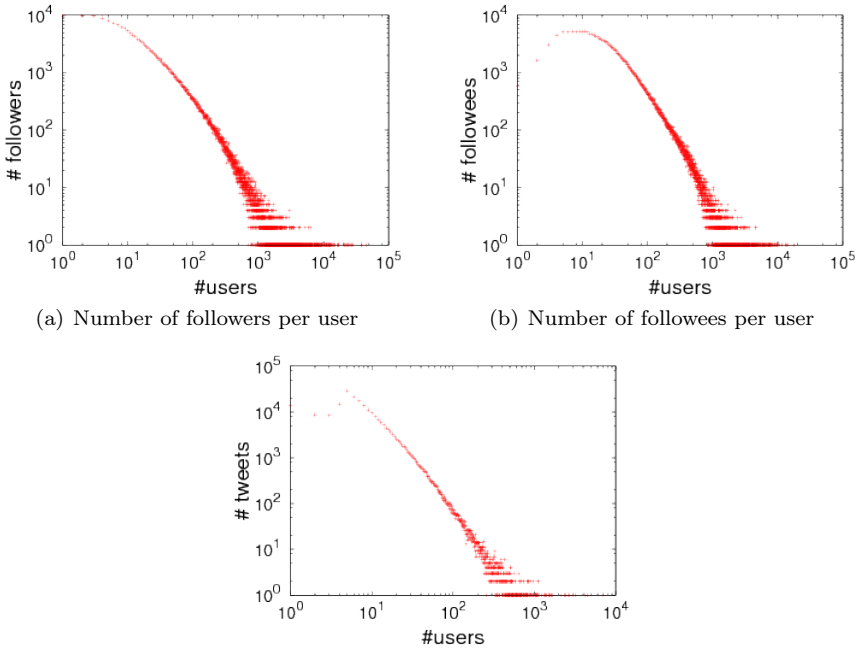
Hannon et al. [2010] proposed many content, collaborative-filtering and hybrid algorithms. Out of all of them, the one with the highest precision was a pure collaborative filtering algorithm (named S6, for strategy 6), which we have replicated here. This algorithm works as follows: instead of using their own words, users are represented by the IDs of their followers. Then, TF-IDF weights are applied (according to Equation 1), and user recommendation is performed using cosine similarity (according to Equation 2). We named this algorithm **TM-S6** (Twittomender Strategy 6).

4.3 User-Based Features

Each user in Twitter has a set of features, such as stated location, number of tweets, number of followers, number of followees, etc. Some of these features are useful in measuring the global influence, or importance of a user. In this work, we considered three user-based features: number of tweets, number of followers and number of followees. The intuition behind using these features for recommendation is simple. Users who post a lot of content (high number of tweets) are more likely to be good users to follow, since they have a lot to share. Users who have a lot of followers have already demonstrated that they are considered interesting by a large number of people, and therefore are likely to be interesting to other people as well. Finally, there is some reciprocity on Twitter, so the number of followees a user has also provides valuable information about the user.

Figure 5 presents the distribution of these features over the users of our dataset. As the figure shows, these distributions are similar to power law distributions, indicating that most of the users rarely tweet, follow and are followed by few users. In contrast, a small number of users tweet very intensively, and follow and are followed by a large number of users.

It is worth noticing that these features are "global", in the sense that they do not measure the importance of a certain user v with respect to a specific user u (personalization), which is generally more useful in recommendation. As mentioned in Section 2, previous work [Armentano et al. 2011] has shown that these features, when used by themselves, do not provide good recommendations. However,



(a) Number of followers per user

(b) Number of followees per user

(c) Number of tweets per user

Fig. 5. Dataset distributions for number of followers, followees and tweets for each user

in this work we show that they are useful when combined with other more sophisticated, personalized features.

4.4 Logistic Regression

As the results in Section 5 will show, some of the proposed algorithms from the previous sections are complementary, in the sense that they work well in different scenarios, and do not generate rankings that are too similar. Also, the rankings produced by the several different algorithms are very diverse, which motivates finding a way of aggregating them.

As we mentioned in Section 2, previous attempts of combining different algorithms or sources of evidence relied on naïve approaches. Hannon et al. [2010], for example, either use a variation of borda count or a simple combination of scores. This may work relatively well when the number of algorithms being combined is small, and on the same scale (although their hybrids did not perform better than pure collaborative-based algorithms). A better approach, however, is learning how the different algorithms and features impact user recommendation from the data itself.

What we would expect from a hybrid algorithm or user recommendation on Twitter is that it would output, for each pair of users u, v, and a vector of given features X, an estimate of $P(u \rightarrow v|X)$, where $u \rightarrow v$ denotes the event that u follows v. In this case, if u was our target user, we would rank every other user v by this estimate, producing a final recommendation list. Since the following relationship is binary (either u follows v or not), we use Binomial Logistic Regression in order to estimate $P(u \rightarrow v|X)$. We now proceed to explain how we model the hybridization problem using Logistic Regression.

The conditional probability modeled by Logistic Regression is shown in Equation 3. X is a vector of features, which may include features that relate u and v (such as the similarity between the two, given by any of the aforementioned algorithms) or features that are specifically about u or specifically about v (such as the number of followers that v has). Since the final objective is ranking every other user with respect to u, for recommendation it does not make sense to include features that are specifically about the target user u, since u will remain fixed (and therefore this kind of feature will have the same value every time). It is common to add a synthetic constant feature to X, called intercept. In the equation, ρ is a vector of weights that has $|X|$ dimensions, such that ρ^k corresponds to the weight given to the feature X^k.

$$P(u \rightarrow v|X) = \frac{1}{1 + e^{-\rho^T X}} \tag{3}$$

Let there be a training set $\{u_i, v_i, X_i, y_i\}_{i=1}^{n}$ of size n, containing tuples u_i, v_i, X_i, y_i where $y_i = 1$ if $u_i \rightarrow v_i$ and $y_i = 0$ otherwise, and X_i is the set of features related to u_i and v_i. We then apply an optimization method in order to get the optimal vector of weights ρ with respect to this training set. In this case, we used a Dual Coordinate Descent method, with L2 regularization, proposed by Yu et al. [2011][7]. In our experiments, we used the validation set referred in Section 3 as the training set for logistic regression. It is worth noticing again that no users in the validation set are present in the test set.

As for the feature vector X_i, we used a combination of the previously mentioned content-based and collaborative-based algorithms and user-based features. Therefore, let there be two users, u_i and v_i. The vector X_i would be composed of a subset, or all of the following:

(1) The similarity score between u_i and v_i, given by **tfidf+f**

[7]We used an implementation available at http://scikit-learn.org/

(2) The similarity score between u_i and v_i, given by **lda+f**
(3) The similarity score between u_i and v_i, given by **BPR-MF**
(4) The similarity score between u_i and v_i, given by **TM-S6**
(5) The number of followers of v_i.
(6) The number of followees of v_i.
(7) The number of tweets posted by v_i.

Features 1-2 are **content-based features**, Features 3-4 are **collaborative-based features** and Features 5-7 are **user-based features**. Note that the scores given by the previously mentioned algorithms were turned into features for logistic regression (Features 1-4). Since the range of possible values for features 1-4 is very different than for features 5-7, we performed a standard feature scaling, centering the data and scaling it to unit standard deviation. It is clear, then, that adding or removing algorithms and user-based features in our hybrid is straightforward: it just means adding or removing features. The seven features we listed here are just a particular instantiation of this generic hybrid algorithm, one that we show to outperform all of the baselines. Another advantage of Logistic Regression is that the weights are discovered from the data, meaning that if low-performing algorithms are added as features, the final result is not compromised, since they will have low (or even negative) weights associated to them.

It is worth noticing that our method has some connection to the stacking ensemble method in classification [Jahrer et al. 2010; Segrera and Moreno 2006] where a held-out validation dataset is used to tune a combination of alternative models rather than being used to select a single best model. In particular, Bao et al. [2009] has a strategy similar to ours, by mixing methods and features in the context of rating prediction for movie recommendation. Our method is different as it is applied to social networks, and particularly to a different task: top-n recommendation, instead of rating prediction.

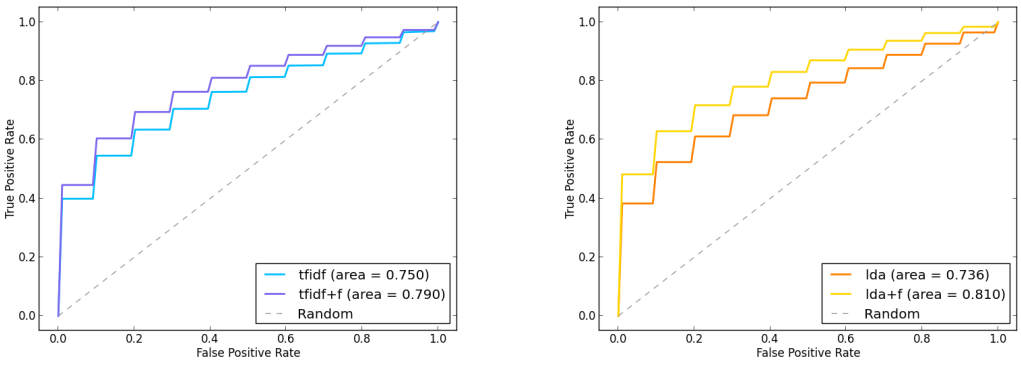
5. EXPERIMENTAL EVALUATION

In this section we present an empirical evaluation of our hybrid algorithm. We begin by showing that our content-based algorithms are more efficient than the traditional baselines. Then we proceed by presenting results that indicate that the algorithms proposed in Section 4 are complementary. Lastly, we present the results obtained by our hybrid algorithm, showing that it outperforms all the baselines.

5.1 Content-Based Algorithms

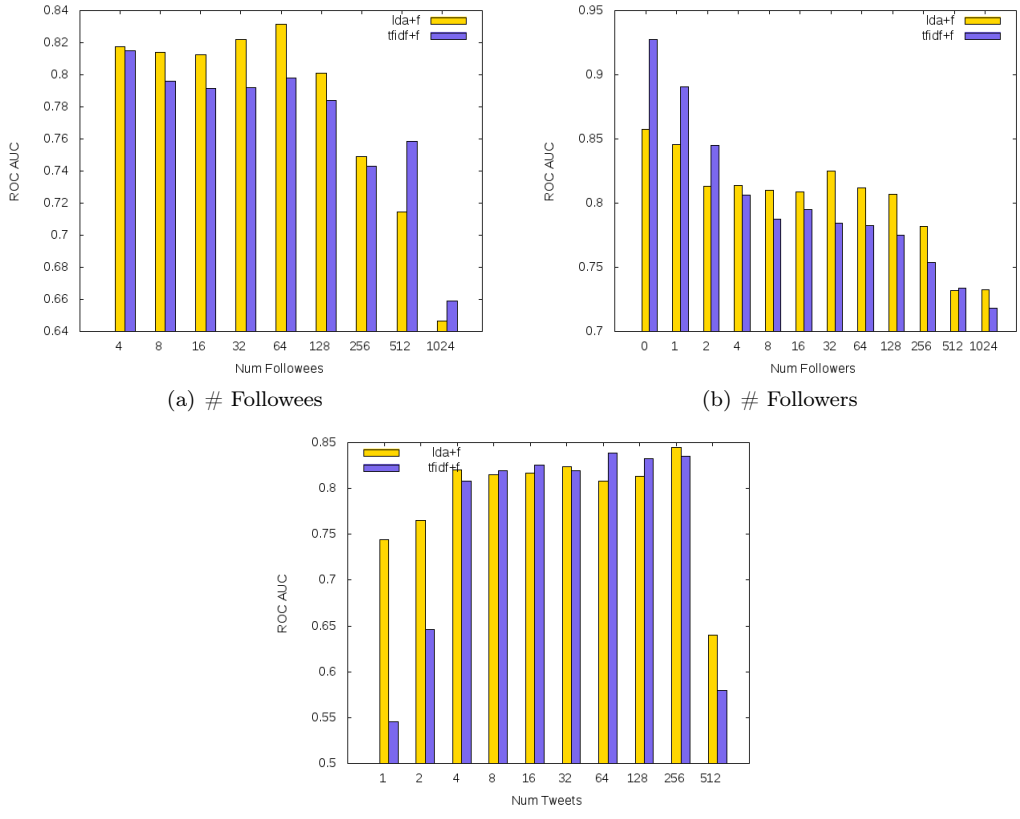
Using the evaluation methodology described in Section 3, we compared the results using the rankings generated by the content-based algorithms. We also used a baseline random algorithm (named **Random**), which always ranks the 20 selected users for each target user in a random order. Figure 6 shows a comparison of the mean ROC curves generated from the two user representations presented in Section 4.1 using both TF-IDF and LDA based algorithms. The parameters we used for LDA were: *iterations:* 150, *topics:* 200. For space reasons, we do not show the results for other parameter configurations. The results indicate that our intuition about using two representations (one for the target user and one for all the other users) was correct, both for pure text and for LDA, as **tfidf+f** outperforms **tfidf** and **lda+f** outperforms **lda**, in all threshold settings, and in AUC. From now on, we will not show results for **tfidf** and **lda**.

As for **tfidf+f** and **lda+f**, in Figure 7, we split the target users into bins related to the user features, and then present the AUC results for each algorithm in each bin. Each bin contains the interval between its value on the x axis and the next value. So, for example, the AUC results for the first bin in Figure 7(a) are for all of the target users that have between 4 and 7 followees. The next bin includes all of the users that have between 8 and 15 followees, and so on. The purpose of



(a) TF-IDF based Algorithms (b) LDA based Algorithms

Fig. 6. Mean ROC curve for different user representations, using TF-IDF and LDA based algorithms.



(a) # Followees (b) # Followers

(c) # Tweets

Fig. 7. **tfidf+f** vs **lda+f** for different users.

this analysis is to show that algorithms based on pure text and topic models behave very differently depending on which set of users they are applied to. For example, Figure 7(b) indicates that **tfidf+f** is much better than **lda+f** for users who have a small number of followers, and much worse for users who have many followers. This type of pattern motivates a combination of the two algorithms.

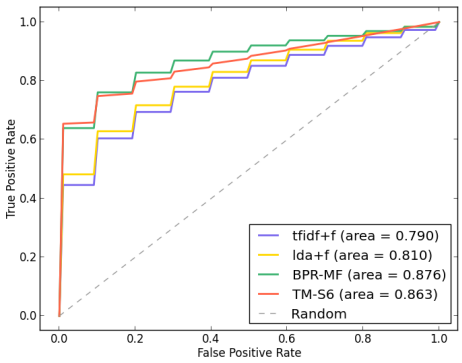


Fig. 8. Mean ROC curve for different algorithms.

Table III. Pairwise rank correlations among metrics: **lda+f**, **tfidf+f**, **BPR-MF** and **TM-S6** scores. The correlations were calculated using Kendall-τ rank correlation.

	tfidf+f	**BPR-MF**	**TM-S6**
lda+f	0.38	0.29	0.26
tfidf+f	-	0.36	0.24
BPR-MF	-	-	0.31

5.2 All Non-Hybrid Algorithms

Figure 8 presents the mean ROC curves for all of the non-hybrid algorithms. It is clear that the collaborative-based algorithms greatly outperform the content-based algorithms. This is coherent with previous research [Hannon et al. 2010]. **BPR-MF** slightly outperforms **TM-S6**, specially in the latter threshold settings and in AUC.

In order to compare the level of agreement between the rankings given by each algorithm, we computed the pairwise rank correlations for the rankings given to each target user, using the Kendall-τ rank correlation metric. The Kendall-τ coefficient ranges from -1 to 1. Values close to 1 indicate strong agreement, while values close to -1 indicate strong disagreement. Table III shows the pairwise correlations between the following algorithms: (1) **lda+f**, (2) **tfidf+f**, (3) **BPR-MF**, (4) **TW-S6**. The results presented in Table III are the averages over the target users. The results show that although the rankings produced by the algorithms have some positive correlation, the pairwise correlations are never very high. Therefore, the four algorithms presented have some disagreement level on rankings, which further motivates a combination between them.

5.3 Hybrid Algorithms

Although Hannon et al. [2010] found that none of their hybrid strategies did any better than their best collaborative-based algorithm (**TM-S6**), we decided to use one of their hybrids as a baseline for our hybrid Logistic Regression algorithm. We implemented a hybrid strategy equivalent to what they call Strategy 8, which is a combination (we used a simple sum) of the scores of the content-based algorithm **tf-idf** and the collaborative-based algorithm **TM-S6**. We name this baseline **TM-S8** (Twittomender Strategy 8). We named our Logistic Regression algorithm **logistic**.

Figure 9 shows a comparison of the mean ROC curves for the two hybrid algorithms presented here. Table IV presents the AUC results for the individual algorithms, as well as for the hybrid algorithms, for comparison. It is clear that the hybrid **TM-S8** outperforms every non-hybrid algorithm. However, **logistic** greatly outperforms every baseline, including **TM-S8**, both in every threshold setting and in

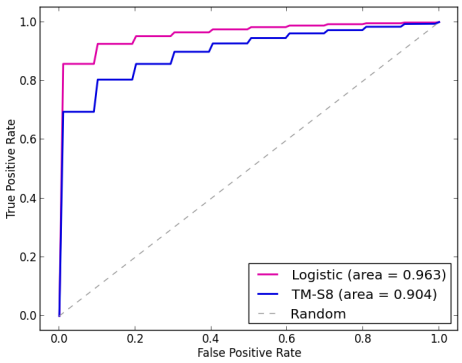


Fig. 9. Mean ROC curve for hybrid algorithms.

Table IV. AUC results for all algorithms. All results are t-significant at $p < 0.01$

Group	Algorithm	ROC AUC
Baseline	**Random**	0.496
Content	**tfidf+f**	0.790
	lda+f	0.810
Collaborative	**BPR-MF**	0.876
	TM-S6	0.863
Hybrids	**TM-S8**	0.904
	logistic	**0.963**

the AUC result. This indicates that our hybrid algorithm is indeed effective for user recommendation.

6. CONCLUSION

In this work, we presented an effective algorithm for combining multiple sources of evidence for user (followee) recommendation on Twitter, based on a logistic regression model. We argued and provided evidence for the need to combine different and complementary sources of information. We presented new user representations for content-based algorithms, which seem to better capture the users' interests, and outperform current content-based state-of-the-art algorithms. Our algorithm is holistic, in the sense that it combines content-based, collaborative-based, and user-based information simultaneously. In our algorithm, we trained a logistic regression function to evaluate potential followees by inputting features based on their individual characteristics as well as similarity scores between pairs of Tweet users. Our offline experiments, based on real-user data, suggest that our algorithm is more effective than current state-of-the-art content-based, collaborative-based and hybrid algorithms. As for future work, some directions we see as worthwhile are: (i) Investigate the differences between the content posted by users and the content posted by the same users' followees, and try to leverage that information; (ii) Evaluate the incorporation of more elaborate user influence measures, such as those based on propagation; and (iii) Evaluate the generation of personalized logistic regression models - that is, learning the weights for each feature for each specific user, instead of learning a single set of weights for the whole dataset.

REFERENCES

Adomavicius, G. and Tuzhilin, A. Towards the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–749, 2005.

Armentano, M., Godoy, D., and Amandi, A. A Topology-Based Approach for Followees Recommendation in Twitter. In *Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*. Barcelona, Spain, pp. 22–29, 2011.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, Boston, MA, USA, 1999.

Bao, X., Bergman, L., and Thompson, R. Stacking Recommendation Engines with Additional Meta-Features. In *Proceedings of the ACM Conference on Recommender Systems*. New York, NY, USA, pp. 109–116, 2009.

Bennett, J., Lanning, S., and Netflix, N. The Netflix Prize. In *Knowledge Discovery and Data Mining Cup and Workshop*. NY, USA, pp. 3–7, 2007.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research* vol. 3, pp. 993–1022, 2003.

Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. P. Measuring User Influence in Twitter: the million follower fallacy. In *Proceedings of the International ACM Conference on Web Search and Data Mining*. Washington, DC, pp. 10–17, 2010.

Cremonesi, P., Koren, Y., and Turrin, R. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the ACM Conference on Recommender Systems*. Barcelona, Spain, pp. 39–46, 2010.

Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. MyMediaLite: a free recommender system library. In *Proceedings of the ACM Conference on Recommender Systems*. NY, USA, pp. 305–308, 2011.

Garcia, R. and Amatriain, X. Weighted Content Based Methods for Recommending Connections in Online Social Networks. In *Proceedings of the ACM Workshop on Recommender Systems and the Social Web*. Barcelona, Spain, pp. 68–71, 2010.

Hannon, J., Bennett, M., and Smyth, B. Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In *Proceedings of the ACM Conference on Recommender Systems*. Barcelona, Spain, pp. 199–206, 2010.

Jahrer, M., Töscher, A., and Legenstein, R. Combining Predictions for Accurate Recommender Systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, DC, USA, pp. 693–702, 2010.

Koren, Y. Factorization Meets the Neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, Nevada, USA, pp. 426–434, 2008.

Krutkam, W., Saikeaw, K., and Chaosakul, A. Twitter Accounts Recommendation Based on Followers and Lists. In *Proceedings of the Joint International Information and Communication Technology*. Luangeprabang, Lao, pp. 1–6, 2010.

Kwak, H., Lee, C., Park, H., and Moon, S. What is Twitter, a Social Network or a News Media? In *Proceedings of the International World Wide Web Conferences*. Raleigh, North Carolina, USA, pp. 591–600, 2010.

Kywe, S. M., Lim, E.-P., and Zhu, F. A Survey of Recommender Systems in Twitter. In *Proceedings of the International Conference on Social Informatics*. Lausanne, Switzerland, pp. 420–433, 2012.

Pennacchiotti, M. and Gurumurthy, S. Investigating Topic Models for Social Media User Recommendation. In *Proceedings of the International World Wide Web Conferences, Poster session*. Hyderabad, India, pp. 101–102, 2011.

Provost, F., Fawcett, T., and Kohavi, R. The Case Against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the International Conference on Machine Learning*. Madison, Wisconsin, USA, pp. 445–453, 1997.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. BPR: bayesian personalized ranking from implicit feedback. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence*. Montreal, Quebec, Canada, pp. 452–461, 2009.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender Systems Handbook*. Springer, 2011.

Segrera, S. and Moreno, M. N. An Experimental Comparative Study of Web Mining Methods for Recommender Systems. In *Proceedings of the International Conference on Distance Learning and Web Engineering*. Lisbon, Portugal, pp. 56–61, 2006.

Wang, Y., Bai, H., Stanton, M., Chen, W.-Y., and Chang, E. Y. PLDA: parallel latent dirichlet allocation for large-scale applications. In *Proceedings of the International Conference on Algorithmic Aspects in Information and Management*. San Francisco, CA, USA, pp. 301–314, 2009.

Weng, J., Lim, E.-P., Jiang, J., and He, Q. TwitterRank: finding topic-sensitive influential twitterers. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. New York, NY, USA, pp. 261–270, 2010.

Yang, J. and Leskovec, J. Patterns of Temporal Variation in Online Media. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. Hong Kong, China, pp. 177–186, 2011.

Yu, H.-F., Huang, F.-L., and Lin, C.-J. Dual Coordinate Descent Methods for Logistic Regression and Maximum Entropy Models. *Machine Learning* 85 (1-2): 41–75, 2011.