Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Sparse Multi-Task Learning for Detecting Influential Nodes in an Implicit Diffusion Network

Yingze Wang[1] and **Guang Xiang**[2] and **Shi-Kuo Chang**[1]
1. Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA
2. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

How to identify influential nodes is a central research topic in information diffusion analysis. Many existing methods rely on the assumption that the network structure is completely known by the model. However, in many applications, such a network is either unavailable or insufficient to explain the underlying information diffusion phenomena. To address this challenge, we develop a multi-task sparse linear influence model (MSLIM), which can simultaneously predict the volume for each contagion and automatically identify sets of the most influential nodes for different contagions. Our method is based on the linear influence model with two main advantages: 1) it does not require the network structure; 2) it can detect different sets of the most influential nodes for different contagions. To solve the corresponding convex optimization problem for learning the model, we adopt the accelerated gradient method (AGM) framework and show that there is an exact closed-form solution for the proximal mapping. Therefore, the optimization procedure achieves the optimal first-order convergence rate and can be scaled to very large datasets. The proposed model is validated on a set of 2.6 millions tweets from 1000 users of Twitter. We show that MSLIM can efficiently select the most influential users for specific contagions. We also present several interesting patterns of the selected influential users.

Introduction

The problem of modeling the diffusion of information arises in a wide spectrum of domains, including social media analysis, infectious disease spread and viral marketing. One important research question is to detect the most influential nodes in an information diffusion network. The successful identification of influential nodes is crucial in many applications. For example, in viral marketing, the company may give free samples of the product to those influential customers to trigger a large cascade of recommendations. In preventing the spread of infectious disease, once the sources of the infection have been detected, we could limit their interactions with the outside world.

The problem of identifying influential nodes has been investigated in many social network analysis literature over the past ten years (Richardson and Domingos 2002; Kempe, Kleinberg, and Éva Tardos 2003; 2005; Ma et al. 2008; Weng et al. 2010; Chen, Wang, and Wang 2010; Cha et al. 2010; Biessmann et al. 2012). However, most of these works suffer from the following two problems:

1. Many existing work makes the assumption that the complete network structure is given as *a priori* and information can only spread over the edges of the underlying diffusion network. However, in many scenarios, the diffusion network is implicit or unknown. Moreover, in many situations, even though the network structure is available (e.g., friendship/followers in social networks), the network itself cannot explain how a node gets infected by the contagion. For example, in viral marketing, a customer, who discovered and liked a new product, could have been influenced by many different types of information, e.g., recommendation from friends in real life, media sites, blogs and forums. Therefore, when analyzing social network data, it is not proper to directly model the fact that "a node gets infected" as a result of influence from its neighbors.
2. Many existing work strives to detect influential nodes for either only one contagion (i.e., information, topic, disease) or across all contagions. When there are multiple related contagions in the network, we should detect different sets of influential nodes for different contagions, yet utilizing the similarities among contagions. In other words, for two similar contagions, the estimated sets of influential nodes should also be close.

In this paper, we address the above issues by developing a new method for detecting the most influential nodes for multiple contagions, which does not require explicit knowledge of the network structure. Our method is built on the recently proposed *linear influence model (LIM)* by Yang and Leskovec (2010), which is one of the state-of-the-art methods for modeling the global influence of a node through an implicit network. However, LIM only addresses the problem of predicting the volume of each contagion but is unable to detect influential nodes for different contagions. In addition, LIM assumes a global influence function for each node without modeling the different levels of influence for different contagions. In this work, we extend LIM by introducing a contagion-sensitive influence function for each node; and formulate the problem into a *convex optimiza-*

tion problem, which jointly minimizes the prediction loss and a sparsity-inducing penalty function. Since our sparsity-inducing penalty is designed by extending the penalty in multi-task sparse Lasso (Obozinski, Wainwright, and Jordan 2009) into its tensor form, we name our method as *multi-task sparse linear influential model (MSLIM)*.

Due to the non-smoothness and high-complexity of the penalty, the optimization problem for MSLIM becomes quite challenging. To solve this optimization problem, we adopt the accelerated gradient method (AGM) (Nesterov 2003; Beck and Teboulle 2009). We prove that for the proximal mapping, which is the key step in AGM, there is an exact closed-form solution. Therefore, the corresponding AGM achieves the optimal convergence rate in $O(1/\sqrt{\delta})$, where δ is the desired optimization accuracy.

In summary, we propose a new method called MSLIM for detecting the most influential nodes in an implicit information diffusion network, which has the following three main advantages over state-of-the-art work:

1. MSLIM does not require the prior knowledge of the network structure.
2. MSLIM is a contagion-sensitive node selection method, which can detect different sets of influential nodes for different contagions.
3. MSLIM simultaneously conducts the diffusion prediction and influential node detection in a unified framework.

We demonstrate both the superior prediction accuracy and better interpretability of the selected nodes on a real Twitter dataset.

Related Works

Identifying influential nodes in social network analysis is first formulated into a discrete optimization task by Kempe, Kleinberg, and Éva Tardos (2003; 2005). After that, many work has been devoted to this research topic. Due to space limitation, we cannot survey all of them. Instead, we compare a few highly relevant work to MSLIM in Table 1 according to the following three metrics:

1. WO-NS (without network structure): whether the method allows the absence of the network structure.
2. Multi-C (multiple contagions): whether the method can select different influential nodes for different contagions and utilize the relatedness of contagions to guide the selection.
3. Temporal: whether the method incorporates the temporal information into the modeling process or treats each timestamp independently.

As seen from Table 1, all of these work requires the network structure to guide the selection of influential nodes. One recent work was proposed to predict the canonical trend in an implicit network based on a canonical correlation analysis (Biessmann et al. 2012). However, this work mainly focuses on detecting a *single* trend setter (i.e., earliest source of a canonical trend); while our method strives to detect a set of the most influential nodes.

Method	WO-NS	Multi-C	Temporal
(Kempe, Kleinberg, and Éva Tardos 2003)	✖	✖	✔
(Ma et al. 2008)	✖	✖	✔
(Weng et al. 2010)	✖	✔	✖
(Cha et al. 2010)	✖	✖	✔
(Chen, Wang, and Wang 2010)	✖	✔	✔
(Bakshy et al. 2011)	✖	✖	✔
MSLIM	✔	✔	✔

Table 1: Summary of some representative work for influential node detection.

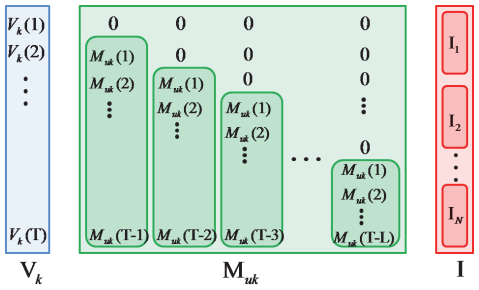

Figure 1: Volume vector $\mathbf{V}_k \in \mathbb{R}^{T\times 1}$, lower-triangular matrix $\mathbf{M}_{u,k} \in \mathbb{R}^{T\times L}$ and influence vector $\mathbf{I} \in \mathbb{R}^{L\cdot N\times 1}$. This figure is adopted from (Yang and Leskovec 2010).

Preliminary

In this section, we introduce the *linear influence model (LIM)* proposed by (Yang and Leskovec 2010), based on which we develop our MSLIM. Assume that there are N nodes and K different contagions diffused among these nodes over time, where each contagion can infect a subset of nodes at any time. We discretize the entire time span into T units: $\{0, 1, \ldots, T\}$. Let $V_k(t)$ be the total *volume* of the contagion k between $t-1$ and t, i.e., the total times that a node gets infected by the contagion k between $t-1$ and t; and $M_{uk}(t)$ be the number of times that the node u gets infected by the contagion k in $[t-1, t]$. The LIM assumes a linear model between $V_k(t)$ and $M_{uk}(t)$:

$$V_k(t+1) = \sum_{u=1}^{N}\sum_{l=0}^{L-1} M_{uk}(t-l)I_u(l+1) + \epsilon_k(t+1), \quad (1)$$

where $\mathbf{I}_u = (I_u(1), \ldots, I_u(L)) \in \mathbb{R}^{L\times 1}$ is the *non-negative influence function* for the node u, the term we want to estimate. The length L of the vector $\mathbf{I}_u$ means that the influence of a node is assumed to drop to zero after L time units. $\epsilon_k(t)$ is the i.i.d. zero-mean Gaussian noise. Following (Yang and Leskovec 2010), $V_k(t)$ and $M_{uk}(t)$ can be organized into the matrix form as shown in Figure 1. We further define the node influence function $\mathbf{I} \in \mathbb{R}^{L\cdot N\times 1}$ to be the concatenation of $\mathbf{I}_1, \ldots, \mathbf{I}_N$ and matrix $\mathbf{M}_k = (\mathbf{M}_{1k}, \ldots, \mathbf{M}_{Nk}) \in \mathbb{R}^{T\times L\cdot N}$. Given the matrix form of $\mathbf{V}_k$, $\mathbf{M}_k$ and $\mathbf{I}$, Eq. (1) can be written into a more compact form as follows:

$$\mathbf{V}_k = \mathbf{M}_k\mathbf{I} + \boldsymbol{\epsilon}. \quad (2)$$

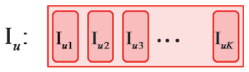

Figure 2: Influence function $\mathbf{I}_u \in \mathbb{R}^{L\times K}$ for the node u.

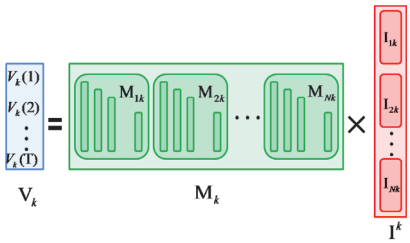

Figure 3: Linear relationship between $\mathbf{V}_k$ and $\mathbf{M}_k$.

Based on (2), LIM estimates the non-negative influence function by solving a non-negative least square problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\textstyle\sum_{k=1}^{K} \|\mathbf{V}_k - \mathbf{M}_k \cdot \mathbf{I}\|_2^2, \\ \text{s.t.} \quad & \mathbf{I} \geq 0 \end{aligned} \tag{3}$$

where $\|\cdot\|_2$ is the vector l_2-norm. By plugging the estimated influence function into (1), one can predict the total volume at the future time $T+1$ for each contagion k.

Proposed Method: Multi-task Sparse LIM

Although LIM can model the influence for each node and has been proven to be effective for predicting the future volume for each contagion, it cannot detect the most influential nodes in an implicit network. In addition, LIM uses a single influence function $\mathbf{I}_u$ for each node u as in (2), which is based on an underlying assumption that each node has the same influence across all the contagions. Clearly, this assumption could be too restrictive for many applications. For different types of contagions, the set of the most influential nodes could be very different. To achieve contagion-sensitive node selection in an implicit network, we extend the LIM model to the multitask sparse learning setting and propose the *multitask sparse linear influential model (MSLIM)*.

We define the influence function by extending $\mathbf{I}_u$ in LIM into the so-called contagion-sensitive $\mathbf{I}_{uk} \in \mathbb{R}^{L\times 1}$, which is a L-length vector representing the influence of the node u on the contagion k. For each contagion k, let $\mathbf{I}^k \in \mathbb{R}^{L\cdot N\times 1}$ be the vector obtained by concatenating $\mathbf{I}_{1k}, \ldots, \mathbf{I}_{Nk}$, which corresponds to $\mathbf{I}$ in LIM. For each node u, we further define the influence matrix for the node u: $\mathbf{I}_u = (\mathbf{I}_{u1}, \ldots, \mathbf{I}_{uK}) \in \mathbb{R}^{L\times K}$, which is shown in Figure 2. Given the contagion-sensitive influence function, we assume a linear relationship between $\mathbf{V}_k$ and $\mathbf{M}_k$ as in (2) but replace $\mathbf{I}$ in (2) by $\mathbf{I}_k$ (see Figure 3):

$$\mathbf{V}_k = \mathbf{M}_k \mathbf{I}^k + \boldsymbol{\epsilon}. \tag{4}$$

We formulate the problem of contagion-sensitive influential node selection into a convex optimization problem, which jointly minimize the square loss and a non-smooth sparsity-inducing penalty inspired by multi-task sparse learning (Obozinski, Wainwright, and Jordan 2009). In particular, we estimate the non-negative vector $\{\mathbf{I}_{uk}\}$ for all nodes and contagions by:

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{V}_k - \mathbf{M}_k \cdot \mathbf{I}^k\|_2^2 \\ & + \lambda \sum_{u=1}^{N} \|\mathbf{I}_u\|_F + \gamma \sum_{u=1}^{N}\sum_{k=1}^{K} \|\mathbf{I}_{uk}\|_2 \\ \text{s.t.} \quad & \mathbf{I}_{uk} \geq 0, \qquad 1 \leq u \leq N, \quad 1 \leq k \leq K, \end{aligned} \tag{5}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm. The penalty term $\|\mathbf{I}_u\|_F$ encourages the entire matrix $\mathbf{I}_u$ to be zero altogether, which means that the node u is non-influential for all different contagions. If the estimated $\|\mathbf{I}_u\|_F > 0$ (i.e., the matrix $\mathbf{I}_u$ is non-zero), a fine-grained selection is performed by the penalty $\sum_{u=1}^{N}\sum_{k=1}^{K} \|\mathbf{I}_{uk}\|_2$, which is essentially a group-Lasso penalty (Yuan and Lin 2006) and can encourage the sparsity of vectors $\{\mathbf{I}_{uk}\}$. The sparsity-level (i.e., the number of selected nodes for each contagion) is controlled by the regularization parameters λ and γ. In particular, for a specific contagion k, one can identify the most influential nodes $\mathcal{U}_k$ with respect to this contagion as:

$$\mathcal{U}_k = \{u : \|\widehat{\mathbf{I}}_{uk}\|_2 > 0\}, \tag{6}$$

where $\widehat{\mathbf{I}}_{uk}$ is the optimal solution of (5).

With the estimated $\widehat{\mathbf{I}}_{uk}$, one can predict the total volume of the contagion k at $T+1$ by $\widehat{V}_k(T+1) = \sum_{u=1}^{N}\sum_{l=0}^{L-1} M_{uk}(T-l)I_{uk}(T+1)$. Therefore, our MSLIM can simultaneously conduct contagion-sensitive volume prediction and influential node detection in a unified framework.

We further note that as compared to the traditional multi-task sparse learning with the l_1/l_2 mixed-norm penalty (Obozinski, Wainwright, and Jordan 2009) (l_1 corresponds to the summation and l_2 corresponds to the vector l_2-norm), the formulation of MSLIM is much more complicated, which makes the optimization (5) quite difficult. Firstly, the traditional multi-task sparse learning only has one l_1/l_2 mixed-norm penalty on the regression coefficients. In contrast, MSLIM in (5) has both l_1/l_F penalty $\sum_{u=1}^{N} \|\mathbf{I}_u\|_F$ and l_1/l_2 penalty $\sum_{u=1}^{N}\sum_{k=1}^{K} \|\mathbf{I}_{uk}\|_2$; and these two penalties are intertwined together in the sense that $\mathbf{I}_{uk}$ is a sub-matrix of $\mathbf{I}_u$. In addition, since the influence is always "positive" in the basic LIM model (1), MSLIM requires the non-negativity of $\{\mathbf{I}_{uk}\}$, which is generally not required by multi-task sparse learning. The complicated structure of the penalty function in MSLIM makes the corresponding optimization very challenge. Thus traditional methods for multi-task sparse learning (e.g., (Chen et al. 2009; Liu, Ji, and Ye 2009)) cannot be directly applied. The work in (Chen et al. 2012; Chen and Liu 2012) proposed a smoothing proximal gradient method which can be used for our problem and achieves the convergence rate of $O(1/\delta)$ where δ is the desired accuracy. In the next subsection, we show how to adapt the FISTA algorithm (Beck and Teboulle 2009) to solve MSLIM in (5) with a faster convergence rate of $O(1/\sqrt{\delta})$.

Optimization

To solve MSLIM, we adopt the accelerated gradient method, in particular, the popular fast iterative shrinkage-thresholding algorithm (FISTA) in (Beck and Teboulle 2009). Since it only utilizes the gradient information of the squared loss, it is efficient and can be scaled to very large problems. To guarantee the optimal first-order convergence rate of an accelerated gradient method, it requires that one key step called *proximal mapping* has an exact analytical solution. For our problem, the proximal mapping takes the form:

$$\text{argmin}_{\mathbf{I}\geq 0} \sum_{u=1}^{N} \frac{1}{2}\|\mathbf{I}_u - \mathbf{W}_u\|_F^2 + \frac{\lambda}{L}\sum_{u=1}^{N}\|\mathbf{I}_u\|_F + \frac{\gamma}{L}\sum_{u=1}^{N}\sum_{k=1}^{K}\|\mathbf{I}_{uk}\|_2, \tag{7}$$

where $\mathbf{W}_u$ is a given matrix with the same size as $\mathbf{I}_u$ and L is the Lipschitz constant for the gradient of the squared loss. The analytical solution of (7) is presented in Appendices. With the close-form solution of the proximal mapping step, we could directly apply FISTA (Beck and Teboulle 2009) to solve MSLIM in (5), which achieves the optimal first-order convergence rate of $O(1/\sqrt{\delta})$, where δ is the desired optimization accuracy.

Experiments

In this section, we evaluate the performance of MSLIM on a Twitter dataset. We first describe the data collection and topic (contagion) extraction procedures, and then compare MSLIM with several competitors on the prediction task. In addition, we also present some qualitative analysis results on the detected influential users(nodes) for different topics.

Data collection

In our experiment, we used the tweets from Twitter, which are text-based messages of up to 140 characters. Prior to crawling a corpus of tweets, we need to collect a set of Twitterers who composed those tweets. To that end, we resorted to the public company profiles on TechCrunch [1], and extracted a list of 1000 Twitter usernames. TechCrunch is one of the leading technology media sites, dedicated to profiling startups, new products, and breaking finance and tech news. Using the Twitter search API [2], we crawled all the tweets of these twitter users between January 2009 and November 2011, which include the full text, the author, and the timestamp for each tweet. In addition, we also collected the profile for each individual user, including followers count, the location, a short biography, etc. We conducted a standard Twitter-data cleaning procedure by removing from tweets URLs, shortened URLs, stop words, special symbols, words containing non-English characters, numbers, punctuation, as well as words in the form of "@username". After the preprocessing, we converted each tweet into a bag-of-words representation. In our dataset, on average, each user has 2,611.825 tweets and the maximum number of tweets for a specific user is 10,986. Following (Yang and Leskovec 2010), we further selected $N = 786$ users out of 1000 users with at least 1,000 tweets during these three years and used them to construct the so-called active node set for modeling the total volume of contagions to construct our implicit network.

[1] http://techcrunch.com/

[2] http://apiwiki.twitter.com/Twitter-API-Document

Topic modeling

When applying MSLIM, the first thing is to define semantically meaningful contagions (corresponding to topics of tweets). A straightforward way of defining topics is to directly use words as topics (e.g., LinkedIn). However, a single word may not be rich enough to represent a broad topic (e.g., social network sites). Another possible way is to use the hashtag (e.g., #SouthAfrica). However, the frequency of "#hashtag" is low in our tweet corpus, rendering the use of them in defining topics inappropriate. In our experiment, we constructed the topics using the Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) , which is a renowned generative probabilistic model for topic discovery. We took the LDA implementation (Phan and Nguyen 2007) , which uses the Gibbs sampling for parameter estimation and inference (Griffiths and Steyvers 2004); and we set the number of topics $K = 50$. For each topic, we only keep the top 100 words since probability on these 100 words has already achieved a significant portion. Table 2 lists the top 10 words from four example topics learned by LDA.

For each tweet $\mathbf{w}$, LDA will make inference on $\Pr(\text{topic of } \mathbf{w} = k|\mathbf{w})$. Thus, we could simply set the topic for each tweet $\mathbf{w}$ using the maximum a posterior rule, i.e, the topic for $\mathbf{w}$ is $\max_{1\leq k\leq K} \Pr(\text{topic of } \mathbf{w} = k|\mathbf{w})$. Given the topic for each tweet, we can directly construct the matrix $\mathbf{M}_k$, where $M_{u,k}(t)$ is the number of times that the user u mentioned the topic k in $[t-1, t]$. The volume $V_k(t)$ is defined by the total number of tweets that the topic k appears in $[t-1, t]$ over the entire 1,000 users.

Quantitative analysis

In this section, we evaluate the prediction performance of MSLIM. Volume $V_k(t)$ of a contagion k can be viewed as a time series in t. We thus evaluate MSLIM on a time series prediction task using the prediction mean-squared error (MSE) as the evaluation metric:

$$\text{MSE} := \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{T}\left(\sum_{t=1}^{T}(\widehat{V}_k(t) - V_k(t))^2\right)\right), \tag{8}$$

where $\widehat{V}_k(t)$ is the predicted total volume for the contagion k at the time t using the data from previous time.

To apply the model, some parameters (e.g.,time-tag L, regularization parameters λ, γ) need to be determined. In our time-series data analysis, it is hard to apply cross validation since if the validation set is in the middle of the entire time sequence, the training set will not be consecutive in time. Therefore, a common way to tune parameters is to split the data into two portions: a training set and a validation set. In particular, we split the first 60% tweets in time as the training set and the last 40% as the validation set. We choose the best set of parameters that lead to the minimum prediction MSE on the validation set. In our experiment, we set one day as the time unit. Parameter L denotes how long

Apple	apple	iphone	ios	ipad	ipod	verizon	siri	sprint	itunes	jailbreak
Samsung	phone	android	samsung	tablet	galaxy	phones	nexus	smartphone	tablets	dual
Startup	startup	funding	entrepreneurs	startups	raises	clients	investors	techcrunch	partners	founders
Finance	nytimes	wsj	nyc	gutschein	stock	bloomberg	tax	finance	bonus	euro

Table 2: Top words for four interesting topics learned by LDA.

it takes the influence of a user to decay to zero. We set L to 5 since we observe when L exceeds 5 (days), the performance becomes stable (i.e, the performance for $L = 5$ and $L = 10$ are roughly the same). In our experiment, we first use the validation set to tune the parameters γ and λ, which determine the number of selected influential users for each topic. Then we combine the training and validation sets to re-train the model with the best selected parameters and estimate the influence function for each user and topic. The final MSE is reported on the entire time span.

Here, we compare our MSLIM algorithm to several competitors on detecting influential nodes in social networks:

- **In-degree selection:** select users based on their total number of followers (i.e.,in-degree).
- **PageRank:** select users with only link structure of the network (Brin and Page 1998).
- **TwitterRank:** one of the state-of-the-art methods for selecting topic-sensitive influential users (Weng et al. 2010).
- **Single-task:** a single-task version of MSLIM which separately runs MSLIM on each single topic.

All the first three methods require the network structure given as the prior knowledge. In our dataset, we use the standard "following relationship" to construct the network structure. The first two methods, which are the most classical methods for hub node detection, select a common set of users across all different topics; while the TwitterRank is a topic-sensitive selection method. With the selected users from each method, we adopted the LIM for the prediction task. Here, single-task version of MSLIM is listed as the last competitor to investigate the benefit of introducing topic-sensitive influence function in LIM.

The comparison of the prediction MSE is presented in Table 3. Note that we select the same number of influential users as MSLIM for a fair comparison. As shown from Table 3, the In-degree selection performs the worst, followed by PageRank, and then single-task version of MSLIM. TwitterRank performs better since it considers the topic information. Our MSLIM outperforms all the competitors in terms of MSE. Again, we note that TwitterRank uses the information about network, which is unavailable in some practical situations. In contrast, our MSLIM does not require the network structure and thus has wider applications.

We also point out that the runtime of MSLIM is slightly longer than that of LIM due to the complicated penalty. In our experiments, we implement both MSLIM and LIM on a standard PC with 4GB RAM in MATLAB. For the best chosen configuration of the parameters, the LIM takes 110.7 seconds while MSLIM takes 170.6 seconds.

	Indegree + LIM	Pagerank + LIM	Twitterrank +LIM	SingleTask	MultiTask
MSE	29.98	22.95	16.21	20.24	13.03

Table 3: Comparison of the prediction MSE.

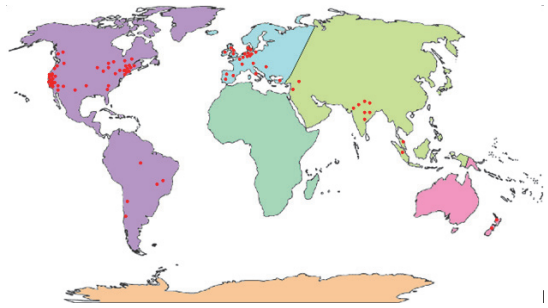

Figure 4: Location of selected influential users for aggregation of 50 topics

	NA	Europe
Apple	22/35	6/35
Samsung	6/31	18/31

Table 4: Location distribution of influential users for topics "Apple" and "Samsung".

	US West	US East
Startup	10/29	4/29
Finance	5/26	9/26

Table 5: Location distribution of influential users for topics "Startup" and "Finance"

Qualitative analysis

In this section, we present some interesting patterns of the most influential users selected from MSLIM. For each topic k, we detect a set of the most influential users $\mathcal{U}_k$ in (6). The union set of the selected users (i.e., $\mathcal{U} = \{u : \|\widehat{\mathbf{I}}_u\|_F > 0\}$) contains 86 users while some users are influential on a variety of topics. We plot the locations of these 86 users as shown in Figure 4. We omit four of them since they do not provide their locations in their profile. We can observe that most of the influential users (42 out of 82) are located in North America (NA), followed by the Europe (22 out of 82). There is no influential user selected in Africa, Australia or Antarctica. Such a result is quite expectable because the total 1000 users are closely related to internet media and high technology and North America and Europe are centers for high-technology.

By analyzing the influential users selected for each topic, we show some interesting findings for the four topics in Table 2. We omit those selected users without location information. The topic "Apple" has more influential users in

North America while "Samsung" are more popular in Europe. For the topic "Startup", nearly 30% of influential users are located in the west coast of USA. This is because the Silicon Valley in the bay area has many startups in the high-technology sector. In contrast, most influential users for the topic "Finance" are located in the east coast of the US, which is the world center of finance. We also analyze the influential users' short biography and find that some influential users for "Apple" are marketing managers, graphic designers and news sources while all of influential users for "Samsung" are related to high-technology. Moveover, influential users for "Startup" include more IT related users but less news sources and financial media related users.

Appendices

We present the analytical solution for the proximal mapping in (7). As we can see, Eq. (7) can be decomposed into N independent problems where each problem only involves $\mathbf{I}_u$:

$$\text{argmin}_{\mathbf{I}_u \geq 0} \frac{1}{2}\|\mathbf{I}_u - \mathbf{W}_u\|_F^2 + \frac{\lambda}{L}\|\mathbf{I}_u\|_F + \frac{\gamma}{L}\sum_{k=1}^{K}\|\mathbf{I}_{uk}\|_2 \quad (9)$$

For simplicity, we use $\mathbf{X}$, $\mathbf{W}$, λ, γ to denote $\mathbf{I}_u$,$\mathbf{W}_u$, $\frac{\lambda}{L}$ and $\frac{\gamma}{L}$ respectively so that Eq. (9) can be written as:

$$\text{argmin}_{\mathbf{X}\geq 0} \frac{1}{2}\|\mathbf{X} - \mathbf{W}\|_F^2 + \lambda\|\mathbf{X}\|_F + \gamma\sum_{k=1}^{K}\|\mathbf{X}_k\|_2, \quad (10)$$

where $\mathbf{X} \in \mathbb{R}^{L\times K}$ and $\mathbf{X}_k$ is the k-th column of $\mathbf{X}$. Now we only need to find the analytical solution of (10), i.e., the solution not obtained by another optimization procedure.

Utilizing the dual-norm, we could write $\|\mathbf{X}\|_F = \max_{\|\mathbf{Y}\|_F\leq\lambda}\langle\mathbf{X},\mathbf{Y}\rangle$ and $\|\mathbf{X}_k\|_2 = \max_{\|\mathbf{Z}_k\|_2\leq\gamma}\langle\mathbf{Z}_k,\mathbf{X}_k\rangle$; and hence (10) can be reformulated as:

$$\max_{\|\mathbf{Y}\|_F\leq\lambda}\max_{\|\mathbf{Z}_k\|_2\leq\gamma}\left(\min_{\mathbf{X}\geq 0}\frac{1}{2}\|\mathbf{X}-\mathbf{W}\|_F^2 + \langle\mathbf{X},\mathbf{Y}\rangle + \sum_{k=1}^{K}\langle\mathbf{Z}_k,\mathbf{X}_k\rangle\right) \quad (11)$$

Assuming that $\mathbf{Y}$ and $\{\mathbf{Z}_k\}_{k=1}^K$ are given, using the KKT condition, we can obtain the optimal $\mathbf{X}$ by solving the inner minimization problem in (11):

$$x_{lk} = \max(w_{lk} - y_{lk} - z_{lk}, 0). \quad (12)$$

We plug (12) back into (11). Now our goal is to solve $\mathbf{Y}$ and $\{\mathbf{Z}_k\}_{k=1}^K$ in the following problem:

$$\max_{\|\mathbf{Y}\|_F\leq\lambda}\max_{\|\mathbf{Z}_k\|_2\leq\gamma}\sum_{l,k}\Bigg[\left(-\frac{1}{2}(y_{lk}+z_{lk})^2 + (y_{lk}+z_{lk})w_{lk}\right)\mathbb{I}(w_{lk} > y_{lk}+z_{lk}) + \frac{1}{2}w_{lk}^2\mathbb{I}(w_{lk}\leq y_{lk}+z_{lk})\Bigg], \quad (13)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Let $s_{lk} = y_{lk} + z_{lk}$ and define

$$\phi(s_{lk}) = (-\frac{1}{2}s_{lk}^2 + s_{lk}w_{lk})\mathbb{I}(w_{ik} > s_{lk}) + \frac{1}{2}w_{lk}^2\mathbb{I}(w_{lk}\leq s_{lk}) \quad (14)$$

To maximize (13) over $\mathbf{Y}, \mathbf{Z}_k$, we discuss it case by case:

1. When $w_{lk} \leq 0$, the solution $s_{lk} = 0$ has already achieved the maximum value of $\phi(s_{lk})$. Therefore, we could simply set $y_{lk} = z_{lk} = 0$ and obtain $x_{lk} = 0$ according to (12).
2. When $w_{lk} > 0$, we define $\mathcal{P} = \{\{l,k\} : w_{lk} > 0\}$ to be the set of indices such that $w_{lk} > 0$. Let $\text{vec}(\mathbf{W}(\mathcal{P}))$ be the vectorization of w_{lk} with $(l,k) \in \mathcal{P}$.
   (a) If $\|\text{vec}(\mathbf{W}(\mathcal{P}))\|_2 \leq \lambda$, we could set $\mathbf{Y}(\mathcal{P}) = \mathbf{W}(\mathcal{P})$ and all other elements in $\mathbf{Y}$ to zero so that $\|\mathbf{Y}\|_F \leq \lambda$; and set $\mathbf{Z} = 0$. Then, Eq. (11), i.e., $\sum_{(l,k)\in\mathcal{P}}\phi(s_{lk})$, achieves its maximum. According to (12), we have $\mathbf{X}(\mathcal{P}) = 0$.
   (b) If $\|\text{vec}(\mathbf{W}(\mathcal{P}))\|_2 > \lambda$, for each column of $\mathbf{W}_k$, let $\text{vec}(\mathbf{W}_k(\mathcal{P}))$ be the vectorization of elements w_{lk} with $(l,k) \in \mathcal{P}$ for a fixed k.
      i. If $\|\text{vec}(\mathbf{W}_k(\mathcal{P}))\|_2 \leq \gamma$, we could set $\mathbf{Z}_k(\mathcal{P}) = \mathbf{W}_k(\mathcal{P})$ and other elements in $\mathbf{Z}_k$ to zero so that $\|\text{vec}(\mathbf{Z}_k(\mathcal{P}))\|_2 \leq \gamma$; and set $\mathbf{Y}_k(\mathcal{P}) = 0$. Then, $\sum_{l:(l,k)\in\mathcal{P}}\phi(s_{lk})$ achieves the maximum for a fixed k. According to (12), we have $\mathbf{X}_k(\mathcal{P}) = 0$.
      ii. If $\|\text{vec}(\mathbf{W}_k(\mathcal{P}))\|_2 > \gamma$, we can imply that both $\mathbf{Y}_k(\mathcal{P})$ and $\mathbf{Z}_k(\mathcal{P})$ as the same direction as $\mathbf{W}_k(\mathcal{P})$. According to the constraint $\|\mathbf{Z}_k\|_2 \leq \gamma$:

$$\mathbf{Z}_k(\mathcal{P}) = \frac{\gamma}{\|\mathbf{W}_k(\mathcal{P})\|_2}\mathbf{W}_k(\mathcal{P}). \quad (15)$$

Now define $\overline{w}_{lk} = w_{lk} - \frac{\gamma}{\|\mathbf{w}_k^{\mathcal{P}}\|_2}w_{lk}$ and $\overline{\mathbf{W}}$ to be the matrix of $\overline{w}_{lk}$. Let $\Theta = \{k : \|\mathbf{w}_k^{\mathcal{P}}\|_2 > \gamma\}$, $\Omega = \{(l,k) : (l,k) \in \mathcal{P} \wedge k \in \Theta\}$. By plugging the solution of $\mathbf{Z}_k(\mathcal{P})$ into (13), Eq. (13) can be written as the following optimization problem:

$$\max_{\|\mathbf{Y}\|_F\leq\lambda,\mathbf{Y}(\Omega^C)=0}\sum_{(l,k)\in\Omega}\Bigg[-\frac{1}{2}(y_{lk}-\overline{w}_{lk})^2\mathbb{I}(\overline{w}_{lk} > y_{lk}) + \frac{1}{2}w_{lk}^2\mathbb{I}(\overline{w}_{lk}\leq y_{lk})\Bigg] \quad (16)$$

Since $\|\text{vec}(\mathbf{W}_k(\Omega))\|_2 > \gamma$, we have $\overline{w}_{lk} > 0$ for all $(l,k) \in \Omega$. Now

   A. $\|\text{vec}\left(\overline{\mathbf{W}}(\Omega)\right)\|_2 \leq \lambda$, we set $y_{lk} = \bar{w}_{lk}$ for $(l,k) \in \Omega$ and $y_{lk} = 0$ for $k \in \Theta$ but $(l,k) \notin \Omega$. Therefore, we have $x_{lk} = 0$ for $(l,k) \in \Omega$. Combining with the previous discussions, we can further infer that the entire $\mathbf{X} = 0$.
   B. $\|\text{vec}\left(\overline{\mathbf{W}}(\Omega)\right)\|_2 > \lambda$, we have

$$\mathbf{Y}(\Omega) = \frac{\lambda}{\|\text{vec}(\overline{\mathbf{W}}(\Omega))\|_2}\overline{\mathbf{W}}(\Omega) \quad (17)$$

Therefore, according to (12), we have $\mathbf{X}(\Omega^C) = 0$ and for each $(l,k) \in \Omega$:

$$\begin{aligned} x_{lk} &= w_{lk} - z_{lk} - y_{lk} \quad (18)\\ &= w_{lk} - \frac{\gamma}{\|\text{vec}\left(\mathbf{w}_k(\mathcal{P})\right)\|_2}w_{lk} - \frac{\lambda}{\|\text{vec}\left(\overline{\mathbf{W}}(\Omega)\right)\|_2}\overline{w}_{lk}\\ &= w_{lk}\left(1 - \frac{\gamma}{\|\text{vec}\left(\mathbf{w}_k(\mathcal{P})\right)\|_2}\right)\left(1 - \frac{\lambda}{\|\text{vec}\left(\overline{\mathbf{W}}(\Omega)\right)\|_2}\right)\end{aligned}$$

By combining the case 1, 2(a), 2(b)i, 2(b)iiA and 2(b)iiB, we obtain the analytical solution of $\mathbf{X}$.

References

Bakshy, E.; Hofman, J. M.; Mason, W. A.; and Watts, D. J. 2011. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the 4th ACM international conference on Web search and data mining*.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Science* 2:183–202.

Biessmann, F.; Papaioannou, J.-M.; Braun, M.; and Harth, A. 2012. Canonical trends: Detecting trend setters in web data. In *Proceedings of the 29th International Conference on Machine Learning*.

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

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Network ISDN Systems* 30:107–117.

Cha, M.; Haddadi, H.; Benevenuto, F.; and Gummadi, K. P. 2010. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social*.

Chen, X., and Liu, H. 2012. An efficient optimization algorithm for structured sparse cca, with applications to eqtl mapping. *Statistics in Biosciences* 4(1):3–26.

Chen, X.; Pan, W.; Kwok, J.; and Carbonell, J. 2009. Accelerated gradient method for multi-task sparse learning problem. In *Proceedings of the 9th IEEE International Conference on Data Mining*.

Chen, X.; Lin, Q.; Kim, S.; Carbonell, J.; and Xing, E. P. 2012. Smoothing proximal gradient method for general structured sparse learning. *Annals of Applied Statistics* 6(2):719–752.

Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Griffiths, T. L., and Steyvers, M. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences*.

Kempe, D.; Kleinberg, J.; and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Kempe, D.; Kleinberg, J.; and Éva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd international conference on Automata, Languages and Programming*.

Liu, J.; Ji, S.; and Ye, J. 2009. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*.

Ma, H.; Yang, H.; Lyu, M. R.; and King, I. 2008. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*.

Nesterov, Y. 2003. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Pub.

Obozinski, G. R.; Wainwright, M. J.; and Jordan, M. I. 2009. High-dimensional union support recovery in multivariate regression. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*.

Phan, X.-H., and Nguyen, C.-T. 2007. GibbsLDA++: A C/C++ implementation of Latent Dirichlet Allocation (LDA).

Richardson, M., and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Weng, J.; Lim, E.-P.; Jiang, J.; and He, Q. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the 3rd ACM international conference on Web search and data mining*.

Yang, J., and Leskovec, J. 2010. Modeling information diffusion in implicit network. In *Proceedings of the10th IEEE International Conference on Data Mining*.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68:49–67.