Learning Network Dynamics from Tumblr®: A Search for Influential Users

Steven Munn[1], Kang-Yu Ni[2], and Jiejun Xu[2]

[1] University of California, Santa Barbara, Santa Barbara CA 92092, USA,
sjm@ece.ucsb.edu
[2] HRL Laboratories, Malibu CA 90265,
{kni, jxu}@hrl.com

Abstract. This work offers an original analysis of a unique data set gathered from the blogging website Tumblr by developing and applying a new data driven method for investigating network dynamics. To our knowledge, this is the first effort to analyze the spread of information on Tumblr on a such a large scale, and our method generally applies to networks where nodes have time-evolving states. We start by testing our method on simulated data, then we follow over 50,000 blogs on Tumblr over a year of activity to determine not only which blogs are influential, but more importantly, how these blogs spread their content.

Keywords: tumblr, dictionary learning, social networks, influence maximization, non-negative matrix factorization

1 Introduction

Given a large-scale social media data set from Tumblr® (a popular online microblogging platform), our goals are to, first, find the influential users and, second, identify the processes by which they spread their ideas. The first problem of finding influential users, also known as the influence maximization problem, is heavily dependent on having a good solution to our second objective. Up until now, however, most published work focuses on solving one problem or the other. To achieve our goals we will thus propose a new technique. By solving both problems together, we aim to do better at each one individually.

Our method builds on existing work in the study of network dynamics, such as state of the art research in traffic analysis, electrical grid balancing, or anomaly detection in sensor systems. This is because modeling the way blogs spread their content on Tumblr is a type of network dynamics problem itself. The blogs are interconnected in an irregular and time-varying fashion depending on how many users follow a blog or forward its content. And, the content of each blog is constantly changing depending on its activity. We can thus use ideas from general network dynamics research to reach our goals. So far, most research work has focused on how network connectivity changes over time [10]; however, the processes by which nodes change their states (this corresponds to blogs changing their content in the context of analyzing Tumblr) are often ill-understood.

To shed light on these processes, we take inspiration from signal processing on graphs, a new area of research dedicated to analyzing the states of nodes on a graph by using well-understood techniques from electrical engineering [13]. We examine the changing states of nodes by representing these states as a series of attributes on the network, and then aggregating the attributes into a matrix representation. This allows us use dictionary learning (DL) methods to obtain either a lower-dimensional representation of the attributes or a sparse representation thereof.

The main advantage of our approach is that it is agnostic of graph structure because this graph structure is often inferred from incomplete data. More importantly though, we typically do not know which aspects of the structure are important. The lower-dimensional representation obtained from dictionary learning helps us find the relevant structural features of a network from which we can infer the key nodes that drive widespread changes of state. To illustrate this idea, consider a protein interaction network. Although some proteins may react with many others, due to physical limitations, often there are restrictions on how many of these reactions can occur at the same time. These proteins may be of high degree in the network; however, only a few edges are actually relevant at any given time during the unfolding of the network process.

In this paper we motivate the use of DL with some background on signal processing on graphs. Then we explain our methodology and apply it to the study of opinion dynamics (or the spread of information), first on a set of simulation data, then with data collected from Tumblr.

2 Transforming Network Attributes

2.1 Signal Processing on Graphs

Our interest in using signal processing techniques to answer network science questions stemmed from the growing literature regarding signal processing on graphs [13]. A mainstay of this literature is the development of transforms for representing graph attributes as the linear combination of basis vectors (see [12] [6] [3] [4] for a sample of relevant papers).

The starting point for these transforms is to consider graph attributes as a vector. A graph comprises a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$. Assuming we have real-valued attributes (they could belong to any set, but we focus on the set of reals), then we can think of the attributes as the result of a function a that maps nodes to reals as in, $a : \mathcal{N} \to \mathbb{R}$.

If we label $a(n_i)$ the attribute corresponding to node i, then we define the attribute vector $\mathbf{x}$ as,

$$\mathbf{x} = \left[a(n_0), a(n_1), \ldots, a(n_{N-1})\right]^{\mathrm{T}}. \tag{1}$$

By extension, if the attributes at a given node are a time series, we can pack the attribute vectors at each time step into a matrix of attributes $\mathbf{X}$ that is $N \times t$ dimensional, where t is the number of time steps in the signal.

Perhaps the most well-known transform for graphs, the graph Fourier transform (GFT) expands attribute vectors as a linear combination of the graph Laplacian eigenvectors [5]. The GFT is especially useful for network models that represent a discretization of an underlying continuous space. The GFT assumes, however, that the graph attributes are smooth with respect to the network structure [5], which is often not the case with big data.

Another approach is to design wavelet transforms based on a model for the process underlying the graph attributes [4] [3] [6]. This works well for many applications such as traffic analysis [4], but these methods typically rely on a simple dynamical model, which makes the wavelets difficult to apply to more complicated processes. Furthermore, both the GFT and graph wavelet approaches rely heavily on graph structure, which limits their flexibility.

2.2 The Dictionary Learning Approach

Our new approach is to learn the basis vectors for our transform from the graph attributes using dictionary learning [14]. Specifically, we demonstrate this idea with non-negative matrix factorization (NMF) [9] [7] for social network activity because the matrix $\mathbf{X}$ defined in Section 2.1 has non-negative entries in that case.

NMF is an algorithm for approximating a matrix with non-negative entries as the product of two matrices, also with non-negative entries. In other words, given an input matrix $\mathbf{X}$ with entries $x_{ij} \geq 0$, NMF solves the following optimization problem,

$$\min_{\mathbf{W},\mathbf{H}} ||\mathbf{X} - \mathbf{W}\mathbf{H}||_{\mathrm{F}}^2 \tag{2}$$

subject to $w_{ij} \geq 0$, $h_{ij} \geq 0$ for all i and j, where $\mathbf{W}$ is an $N \times k$ (a number of components we set for NMF) matrix and $\mathbf{H}$ is $k \times t$.

NMF transforms $\mathbf{X}$ into a dictionary $\mathbf{W}$, where each column vector is a dictionary atom, and a coefficients matrix $\mathbf{H}$. This algorithm is useful for dimensionality reduction (setting $k < N$) [9], or for sparse encoding ($k > N$) [7]–both of which are helpful for our purposes. Dimensionality reduction will serve to identify the different sources, or starting points, for dynamic process on networks; sparse encoding will help us understand the nature of the processes unfolding on the network.

2.3 Dictionary Learning Adaptations for Improved Parallelism

For large datasets, we propose a modified approach that makes dictionary learning more efficient to run in parallel. Given a graph $\mathcal{G}$ of nodes $\mathcal{N}$ and edges $\mathcal{E}$, we can divide the graph into a set of subgraphs $\mathcal{L} = \{\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_l\}$ using community detection [1]. For each subgraph, we perform the analysis described above. That is, for subgraph $\mathcal{G}_i$, we get a signal matrix $\mathbf{X}_i$ from which we obtain a coefficients matrix $\mathbf{H}_i$ and a dictionary atoms matrix $\mathbf{W}_i$ via NMF.

For clarity, and without loss of generality, we assume that the nodes of $\mathcal{G}$ are numbered such that the nodes in subgraph $\mathcal{G}_0$ start at zero and end at $N_0 - 1$, while the nodes in $\mathcal{G}_1$ start from N_0 and end at $N_0 + N_1 - 1$, and so on. Using the full matrix $\mathbf{X}$ as input, we initialize the dictionary learning algorithm with the following dictionary matrix:

$$\mathbf{W} = \left[\begin{array}{ccccccc} | & | & & \mathbf{W_0} & \mathbf{0} & \mathbf{0} & \\ | & | & & \mathbf{0} & \mathbf{W_1} & \mathbf{0} & \\ \mathbf{r^T} & \mathbf{r^T} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{W_2} & \dots \\ | & | & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \\ \underbrace{| \quad |}_{m} & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \end{array} \right] \tag{3}$$

where m is the number of global-scale atoms we wish to learn, $\mathbf{0}$ are appropriately sized zero matrices, and $\mathbf{r}$ is a row vector with entries taken at random or following a dictionary learning initialization step (as in [2]).

Instead of optimizing the entire $\mathbf{W}$ matrix, however, we only update the first m columns. This means we are learning m new global-scale atoms, and keeping the local atoms intact.

3 Opinion Dynamics Simulations

To better explain how NMF is useful for understanding opinion dynamics, we illustrate our method with some examples using simulated data. In this section, we explain how we generated the data to best reflect the type of data available from social media.

3.1 The Decaying Cascade Model

The proposed decaying cascade model is an adaptation of the independent cascade model [8] meant to reflect the fact that ideas tend to have limited reach across a social network: Your friend's friend is usually less likely to adopt your opinions than your friend. It is also closely related to the continuous independent cascade model of Gomez et al., which better represents the temporal aspects of social media data [11]. Algorithm 1 summarizes the steps for the decaying cascade simulation.

The algorithm starts with a set of initially active persons (nodes) $\mathcal{V}_0$ and keeps track of nodes that change their states in the set of events E. After a random amount of time, a randomly selected node (the source) attempts to convince one of its neighbors (the target) to adopt its opinion. The probability of success is,

$$p\left(n_{\text{Target}}\right) = p_0 \exp\left(-d_{\mathcal{V}_0}\left(n_{\text{Target}}\right)\right) \tag{4}$$

where n_{Target} is the targeted node, p_0 a probability constant, and $d_{\mathcal{V}_0}\left(n_{\text{Target}}\right)$ is the shortest path distance on the network from the target to one of the nodes in $\mathcal{V}_0$.

Algorithm 1 Decaying Cascade

```
1: Set $t = 0$
2: Pick $\mathcal{V}_0$ from $\mathcal{N}$, and set $V = \mathcal{V}_0$
3: Initialize $E = \{(t = 0, n) : n \in \mathcal{V}_0\}$
4: while $t < T$ do
5:     $t \leftarrow t +$ Random increment
6:     Randomly pick $n_{\text{Source}} \in V$
7:     Randomly pick $n_{\text{Target}} \in$ Neighbors of $n_{\text{Source}}$
8:     Pick outcome with probability $p(n_{\text{Target}})$
9:     if Source convinces target then
10:        $V \leftarrow V \cup n_{\text{Target}}$
11:        $E \leftarrow E \cup (t, n_{\text{Target}})$
12:    else
13:        $n_{\text{Source}}$ gives up on $n_{\text{Target}}$ forever
14:    end if
15: end while
```

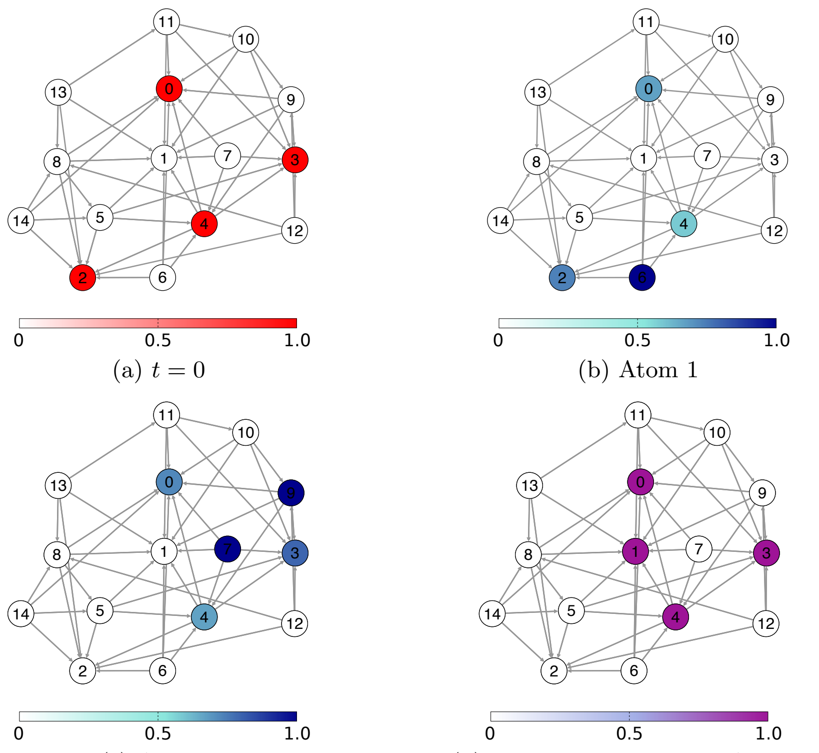
3.2 The Threshold Model

The proposed threshold model, modified from the linear threshold model [8], is exactly the same as our decaying cascade model up until the probability of convincing the target. Here, the target will only be convinced if the number of active neighboring nodes is above a predefined threshold. With a high enough threshold, the spreading of opinions will have similar properties to the decaying cascade model with respect to spatial localization across the nodes.

4 Simulation Analysis

To best reflect the idea that social media data typically involves the spreading of many ideas in parallel, we set up our simulation as follows: We select a number of "influential" nodes $\mathcal{V}_{\text{Influential}}$ (nodes that can initialize a process) and run a large number (one to two hundred) of threshold or decaying cascade model simulations where the initially actives nodes are a subset of $\mathcal{V}_{\text{Influential}}$. We then aggregate the set of events from each simulation into a global set of events for the network from which we derive the signal $\mathbf{X}$ that we use for dictionary learning.

Figure 1 (a) shows the set of nodes $\mathcal{V}_{\text{Influential}}$ in red for a series of threshold simulations where three nodes are necessary to activate a new node. In Figures (b) and (c) we can see two NMF atoms (of four emanating from the algorithm). The information spreads from the lighter blue nodes to the darker blue ones. Since the atoms are effective at showing the direction in which information is spreading, we define influence scores to find the most influential nodes as follows: We first find the top 20 atoms according to their coefficient magnitudes and then compute a weighted sum of these atoms. The influence score at each node is the average value of its neighboring nodes in the weighted sum atom. In figure (d), nodes whose neighbors have a high value in atom 1 (the top atom in our ranking for the influence scores) are colored in purple.



(a) $t = 0$ (b) Atom 1 (c) Atom 2 (d) Influence Scores from Atom 1

Fig. 1. Non-negative matrix factorization atoms for the threshold model of Section 3.2. In (a), the intially active nodes are in red. In (b) and (c), we plot the first two NMF atoms. Information is spreading from the lighter blue nodes to the darker blue ones. In (d) nodes in purple have neighbors whose values are high in atom 1.

Table 1 summarizes the results of the following experiment: On a graph with four hundred nodes, we run a series of threshold and decaying cascade model simulations, with 20 influential nodes, and generate a signal that we use for our method. We rank the nodes in terms of importance using the influence scores. Of the top 20 nodes, if the nodes match one of the initial nodes set for the simulation we count that towards the "exact node" accuracy; if it is within one hop of one of an initial node, we count that towards "within one hop" accuracy. The accuracies are averaged over 100 experiments. For comparison, Table 1 includes a baseline method in which the nodes deemed most influential are the most active ones.

To better understand how information is spreading on the network, we can set up NMF to learn an overcomplete basis of atoms. These atoms will be different depending on the process. For example, in atoms learned from cascade model simulation data, only a single node tends to be active; however, four nodes are

	Accuracy	
Algorithm	Exact node	Within one hop
Threshold Model Simulations		
NMF-based	$23.8 \pm 3.4\%$	$100.0 \pm 0.0\%$
PageRank	$15.0 \pm 0.0\%$	$15.0 \pm 0.0\%$
Baseline	$14.7 \pm 1.2\%$	$14.7 \pm 1.2\%$
Decaying Cascade Simulations		
NMF	$33.4 \pm 2.3\%$	$100.0 \pm 0.0\%$
PageRank	$15.0 \pm 0.0\%$	$15.0 \pm 0.0\%$
Baseline	$15.7 \pm 3.8\%$	$15.8 \pm 3.8\%$

Table 1. Accuracy measures for our NMF-based influence maximization algorithm. Using the two opinion dynamics models described earlier to generate data with ground truth key nodes, we compare the relative number of correctly identified sources, and the relative number of sources identified within a one-edge hop distance.

usually active when the dynamics come from a threshold model with a three-node activation requirement.

5 Real Data Results

Tumblr is a microblogging website with tens of millions of active users where people write or respond to short blog posts. In this section we use our NMF approach to find influential users on Tumblr and better understand how people's ideas spread on this social network.

For this analysis, we use the reblog network as the underlying graph structure. Users are nodes, and directed edges correspond to users who rebloged each other's content. In other words, if user i reblogs user j's posted content, we have a directed edge from i to j in the reblog network.

We demonstrate our method on a group of 59,709 users, which are picked using a community detection method. First we build the reblog network for the entire dataset. Then we find the largest connected component (approximately six million users). Finally, we use Louvain components [1] to find a community of users with a large-enough number nodes. After selecting the users, we build an attribute matrix according to their activity on Tumblr for a year. The sampling rate is twelve hours, so $\mathbf{X}$ is a 59,709 $\times$ 703 matrix.

Using the parallelism method described in Section 2.3, we partition our graph of 59,709 users into two subcomponents, and set up NMF to learn 200 atoms for each subcomponent over 100 iterations. After combining the two dictionaries following Equation 3, we then learn an additional 200 atoms over 100 iterations. All these steps take a total of 54 minutes to run on a single server node.

After ranking users according to the techniques of Section 4, we found that the most influential blogs according to our method mostly revolve around humor and fashion. Table 2 lists the top six blogs we found.

Rank	Description
1	Video collection features slapstick humor
2	Fashion and film
3	Humorous video collection with a voting system
4	Absurd-type humorous posts
5	Cartoon animation enthusiast's blog
6	Self-proclaimed novelist's posts

Table 2. The top six most influential blogs in a community of 59,709 users according to our method.

By comparison, other influence metrics including centrality measures (e.g., betweenness, degree, and PageRank®) and the popular online ranking system Alexa® also attribute high scores to nodes that our method deems influential. The ranking order, however, is usually different because the methods do not take into account how information spreads on Tumblr. This supports our assumption that even graph structures that should be highly reflective of network dynamics do not paint the full picture of how the processes are unfolding on graphs.

To examine the opinion dynamics processes in more detail, Figure 2 shows the value of the atoms at the scale of a small component in the network. Although plotting software would allow us to look at larger components, for illustration purposes in this paper, a smaller component is preferable. In Figure 2 there are two of the most important atoms. They show us how information is traveling within the component. Some paths are consistently active, while others are not.

Figure 3 offers a different perspective on the atoms by focusing on individual users. For this figure, first we rank the atoms in terms of their importance by computing the average magnitude of the coefficient corresponding to each atom over the whole signal and organize the atoms in decreasing order of this average. Then we plot the atom values in the ego network of a particular node using the top two atoms in the ranking. This shows us two common ways in which users linked to the influential node post content. Users with a strong value in one atom are likely to post within 12 hours of others in the atom who also have strong values. Users with strong values across the different atoms are generally more responsive to the central user's content. We do find, however, users who only have strong values in only a few atoms, suggesting that they only respond to particular topics from the center node.

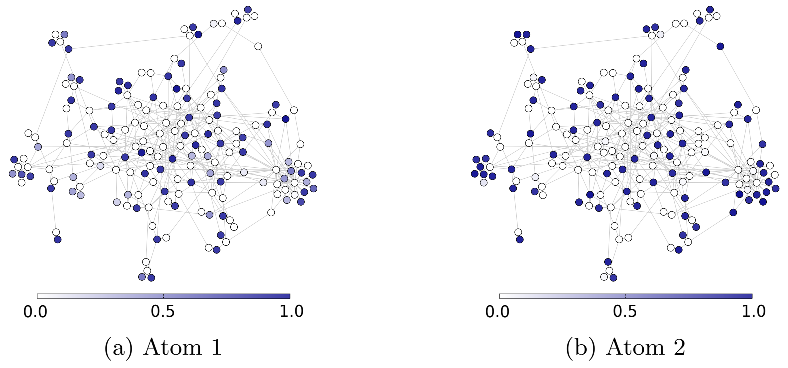


Fig. 2. A Louvain component of two of the most important atoms according to average coefficient magnitude. The atoms exemplify the characteristic ways in which the users post in this component. Some paths in the network are active in one atom but not another, which suggests that information sometimes travels along these routes.

6 Conclusion

Network dynamics underlie many real-world phenomena, and understanding them is crucial for scientific and engineering research. Focusing our attention on opinion dynamics, we set out to shed light on the processes by which people share ideas, and we studied the task of finding key sources of information or opinions. Our analysis shows that our dictionary learning-based method accomplishes these goals in a novel way and opens the way to better understanding the spread of ideas and information. Possible extensions of our work include modeling competing opinions with positive and negative signal matrix values, or difference encoding the signal over time, which could potentially help us keep track of the sources and targets of user activity.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (oct 2008), `https://doi.org/10.1088%2F1742-5468%2F2008%2F10%2Fp10008`
2. Boutsidis, C., Gallopoulos, E.: SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognition 41(4), 1350–1362 (apr 2008), `https://doi.org/10.1016%2Fj.patcog.2007.09.010`
3. Coifman, R.R., Maggioni, M.: Diffusion wavelets. Applied and Computational Harmonic Analysis 21(1), 53–94 (jul 2006), `https://doi.org/10.1016%2Fj.acha.2006.04.004`
4. Crovella, M., Kolaczyk, E.: Graph wavelets for spatial traffic analysis. In: IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428). Institute of Electrical

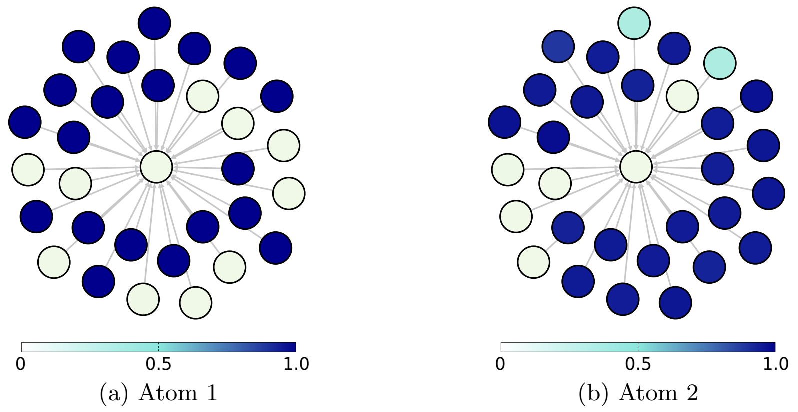


Fig. 3. Top two atoms by average coefficient magnitude values centered around an influential blog and its one-hop ego network. Both atoms typify the most prevalent ways in which users respond to the central blogs content. Some nodes with strong values in (a) have lower values in (b) this means that they respond more selectively to the center blogs posts.

and Electronics Engineers (IEEE), https://doi.org/10.1109%2Finfcom.2003.1209207

5. Grady, L.J., Polimeni, J.R.: Discrete Calculus. Springer Nature (2010)
6. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis 30(2), 129–150 (mar 2011), https://doi.org/10.1016%2Fj.acha.2010.04.005
7. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of machine learning research 5(Nov), 1457–1469 (2004)
8. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03. Association for Computing Machinery (ACM) (2003), https://doi.org/10.1145%2F956755.956769
9. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Computation 19(10), 2756–2779 (oct 2007)
10. Newman, M.E.J.: The structure and dynamics of networks. Princeton University Press, Princeton, N.J. Oxford (2006)
11. Rodriguez, M.G., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10. Association for Computing Machinery (ACM) (2010), https://doi.org/10.1145%2F1835804.1835933
12. Sandryhaila, A., Moura, J.M.F.: Discrete signal processing on graphs. IEEE Transactions on Signal Processing 61(7), 1644–1656 (apr 2013)
13. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine 30(3), 83–98 (may 2013), https://doi.org/10.1109%2Fmsp.2012.2235192
14. Tosic, I., Frossard, P.: Dictionary learning. IEEE Signal Processing Magazine 28(2), 27–38 (March 2011)