Sample[*] Diachronically Like-minded User Community Detection[*]

Ali Fani
University of Windsor
afani@uwindsor.ca

Hossein Fani
University of Windsor
hfani@uwindsor.ca

ABSTRACT

This is a *sample* proposed method for your proposal. The sections and subsections are *by no means* fixed and should be indeed changed or customized according to the proposal.

KEYWORDS

More Specific Keywords, Specific Keyword, General Keyword, More General Keyword

1 PROBLEM DEFINITION

Our goal is to identify like-minded user communities whose members exhibit similar temporal dispositions towards similar topics. Here, we provide a formal statement of the problem after which we propose our approach in detail in the next section. We view the problem of like-minded user community detection as an instance of the set partitioning task on a set of users $\mathbb{U}$. A partition $\mathbb{P}$ of the set $\mathbb{U}$ of all users is a set of nonempty subsets of $\mathbb{U}$ as communities such that every user $u \in \mathbb{U}$ is in exactly one of these communities. Notationally, $\mathbb{P} = \{\mathbb{C} : \mathbb{C} \subseteq \mathbb{U}, |\mathbb{C}| \geq 1\}$ such that $\forall \mathbb{C}_i, \mathbb{C}_{j \neq i} \in \mathbb{P} : \mathbb{C}_i \cap \mathbb{C}_j = \varnothing$ and $\bigcup_{\mathbb{C} \in \mathbb{P}} \mathbb{C} = \mathbb{U}$. Since we do not consider a set with one user as a community, we relax the partition definition by assuming $|\mathbb{C}| \geq 2$ and drop the last union condition; i.e., $\mathbb{P}^* = \mathbb{P} \setminus \{\mathbb{C} : |\mathbb{C}| = 1\}$. The goal of like-minded user community detection is to infer $\mathbb{P}^*$ such that highly similar users are in the same community $\mathbb{C}$, yet users of high dissimilarity are in different communities $\mathbb{C}_i$ and $\mathbb{C}_{j \neq i}$. In our work, we consider two users to be similar if they show similar temporal inclination towards a set $\mathbb{Z}$ of possible topics.

2 PROPOSED APPROACH

Our proposed like-minded community detection method seeks to find $\mathbb{P}^*$ with respect to the temporal topic-based sense of user similarity, defined in the previous section. The approach works through three pipelined phases: temporal topic-based user modeling, user vector representation (embedding), and user community detection. In the following, we describe the details of each step.

2.1 Temporal Topic-based User Modeling

Our work relies on users' behavior towards a set of topics within time period T. To incorporate both users' topics of interest and temporality, for each user $u \in \mathbb{U}$, we model her inclination towards each topic $z \in \mathbb{Z}$ at each time interval $1 \leq t \leq \mathrm{L}$ through a matrix. The stacking of all user matrices will generate a cuboid denoted as *points of temporal interest* (PoTI). An entry in PoTI shows how much a user $u \in \mathbb{U}$ is interested in a topic $z \in \mathbb{Z}$ in time interval $1 \leq t \leq \mathrm{L}$.

Definition 2.1. **Points of Temporal Interest (PoTI).** Let $\mathbb{U}$ be a set of users, $\mathbb{M}$ be the users' posts corpus, $\mathbb{Z}$ be a set of topics, and T be a time period broken down into L intervals, *points of temporal interest* (PoTI) is a three dimensional matrix (cuboid) $\mathbb{U} \times \mathbb{Z} \times \mathrm{T} = \{y_t^u[z]\}$ where $u \in \mathbb{U}, z \in \mathbb{Z}$ and $1 \leq t \leq \mathrm{L}$ whose three dimensions correspond to users, topics and time intervals, respectively and the value $y_t^u[z]$ is the degree of u's interest in topic z at time t.

To instantiate PoTI, we need to find i) a set of topics that have been observed in time period T, i.e., $\mathbb{Z}$, and ii) each user's degree of interest at time t towards each topic $z \in \mathbb{Z}$, i.e., $y_t^u[z]$. The set of possible topics can be derived by extracting the topics available in the collection of users' posts using various existing topic detection methods in the literature including topic modeling techniques such as latent Dirichlet allocation (LDA) [1] as suggested in [4, 5]. In order to identify the set of topics, ***.

2.2 User Vector Representation (Embedding)

The key contribution of this paper is to learn user vector representations from users' topics of interest with the expectation that temporally like-minded users end up closer to each other in the vector space. We hypothesize that an appropriate embedding method would bring significant performance into our main downstream task of like-minded user community detection compared to the state of the art. To build user embeddings, we first formally formulate what we mean by a like-minded pair of users. Then, we propose an embedding method which preserves pairwise like-minded proximity of the users through maximizing the likelihood that two like-minded users stay close to each other in vector space.

2.2.1 User Like-minded Context Model. In our approach, users would be considered to be like-minded if they share similar temporal and topical interest. More formally, the more two user u_1 and u_2 share instances of $y_t^{u_1}[z] \simeq y_t^{u_2}[z]$ in the PoTI for topics $z \in \mathbb{Z}$ across different time intervals $1 \leq t \leq \mathrm{L}$, the more similar they would be.

Definition 2.2. **Region of Like-mindedness (RoL).** A three-dimensional subspace of PoTI, such as R, is defined to be a region of like-mindedness (RoL) iff (i) all the values in this subspace are *equal* with respect to a certain condition of homogeneity c; notationally, $\forall y, y' \in \mathrm{R}; c(y) = c(y')$ and (ii) it is *maximal* such that there exists no other region of like-mindedness such as R' which subsumes R.

To find all regions of like-mindedness, RoLs, in PoTI, we adopt a similar strategy to [6] where subspace submatrices are mined from three-dimensional gene-sample-time gene expression microarrays. First, we find the RoLs in user and topic dimensions at each time interval. The output is two-dimensional (2-d) RoLs indexed by the time interval $1 \leq t \leq \mathrm{L}$, i.e., RoL_t. Then, we merge RoL_t of different time intervals to build the required RoLs. The details are as follows:

[*] Link to Github or an online repo

Algorithm 1 Finding 2-d RoLs for time interval t

```
Inputs:
    users $\mathbb{U}$, topics $\mathbb{Z}$, homogeneity condition $c$, multigraph $\mathcal{G}^t$
Initialization:
    $\mathcal{R}^t = \varnothing$
    find_2d_RoLs($r = \mathbb{U} \times \varnothing, C = [z_1, z_1, z_2, z_2, ..., z_{|\mathbb{Z}|}, z_{|\mathbb{Z}|}]$)
Output: 2-d RoLs in $\mathcal{R}^t$
procedure FIND_2D_RoLs($r = A \times B, C$)
    if $(r \models c) \wedge (\nexists r' \in \mathcal{R}^t : r \subset r')$ then
        for all $r'' \in \mathcal{R}^t$ do
            if $r'' \subset r$ then
                $\mathcal{R}^t \leftarrow \mathcal{R}^t \setminus r''$
        $\mathcal{R}^t \leftarrow \mathcal{R}^t \cup r$
    for all $z_j \in \mathbb{Z}$ do
        $A \leftarrow r.A; B \leftarrow r.B \cup z_j; C \leftarrow C \setminus z_j$
        if $r.B = \varnothing$ then
            find_2d_RoLs($A \times B, C$)
        else
            for all $z_i \in r.B$ do
                for all $(z_i \rightarrow z_j) \in \mathcal{G}^t.\mathbb{E}$ do
                    $A \leftarrow r.A \cap \mathbb{U}^t_{z_i,z_j}$
                    find_2d_RoLs($A \times B, C$)
```

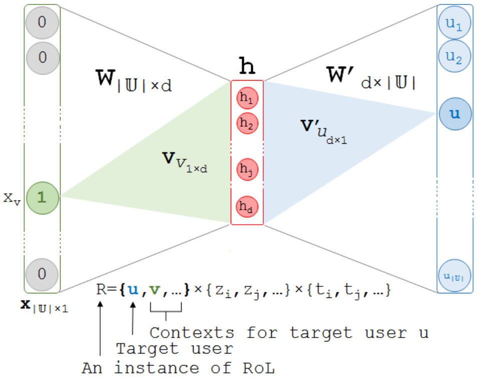
0
0
0
1
x_v
$\mathbf{x}_{|\mathbb{U}|\times 1}$
$\mathbf{W}_{|\mathbb{U}|\times d}$
$\mathbf{v}_{v_{1\times d}}$
h
h_1
h_2
h_j
h_d
$\mathbf{W'}_{d\times|\mathbb{U}|}$
$\mathbf{v'}_{u_{d\times 1}}$
u_1
u_2
u
$u_{|\mathbb{U}|}$
R={u,v,...}×{z_i,z_j,...}×{t_i,t_j,...}
Contexts for target user u
Target user
An instance of RoL

Figure 1: The neural network architecture.

Finding 2-d RoLs for the time interval t. Given the PoTI and ***

For example, Figure **??**a shows the multigraph $\mathcal{G}^{22}$ constructed from Figure **??**a for time t_{22} where the condition of homogeneity c is satisfied if the cells have a value in the range $[0.1, 1.0]$. To illustrate that there may be parallel edges, ***. To find the final 2-d RoLs for time t, we apply depth-first-search (DFS) on the multigraph $\mathcal{G}^t$ based on the pseudo code described in Algorithm 1. *** Finally, ***.

2.2.2 *User Embedding.*

Given a set of discovered regions of like-mindedness (RoLs), the context for a user u would be the set of users in each of the RoLs where u had been observed. We formulate user vector learning as a maximum likelihood (ML) optimization problem. In particular, for each user, we find her like-minded users by optimizing the conditional probability of observing users that have the same RoLs as her. To induce the user embeddings, we adopt an approach similar to [3] as follows:

Definition 2.3. **(Embedding Objective)** Given the set $\mathcal{R}$ of all regions of like-mindedness (RoLs), the embedding function $g : \mathbb{U} \rightarrow [0, 1]^d$ maps each user $u \in \mathbb{U}$ onto a d-dimensional space, such that the following objective is optimized:

$$\arg\max_{g} \sum_{R \in \mathcal{R}, u \in R} \log \Pr(u|R \setminus u) \tag{1}$$

In order to make the optimization tractable, we assume conditional independence for observing users in a RoL such as R. So,

$$\Pr(u|R \setminus u) = \prod_{v \in R \setminus u} \Pr(u|v) \tag{2}$$

To learn the user embeddings, we use a single hidden layer, fully connected neural network. The architecture of our neural network is shown in Figure 1. The hidden layer $\mathbf{h}$ is of size d, the dimensionality of the resulting user vectors, and the input and output layer is set to have as many neurons as $|\mathbb{U}|$. Thus, the input to hidden layer connections can be represented by matrix $\mathbf{W}$ of size $|\mathbb{U}| \times d$ with each row representing a vector for user $u \in \mathbb{U}$. The input layer $\mathbf{x}$ is a one-hot encoded vector and the hidden layer's neurons are all linear such that $\mathbf{h} = \mathbf{W}^\top \mathbf{x}$. Given a user v in the input layer that is taken from the context of u, i.e., u and v have been observed in the same RoL, $\mathbf{h}$ is the transpose of v's corresponding row in $\mathbf{W}$ named $\mathbf{v}_v$. In the same way, the connections from hidden layer to output layer can be described by matrix $\mathbf{W'}$ of size $d \times |\mathbb{U}|$. The prediction task could be done via a softmax function to approximate the probability of observing the target user u given user v from the same RoL, i.e.,

$$\Pr(u|v) = \frac{\exp(\mathbf{v'}_u^\top \mathbf{h})}{\sum_{w \in \mathbb{U}} \exp(\mathbf{v'}_w^\top \mathbf{h})} = \frac{\exp(\mathbf{v'}_u^\top \mathbf{v}_v)}{\sum_{w \in \mathbb{U}} \exp(\mathbf{v'}_w^\top \mathbf{v}_v)} \tag{3}$$

where v'_u is u's corresponding column of matrix $\mathbf{W'}$. With the assumption in Equation 2 and the above probability function, the objective function in Equation 1 simplifies to:

$$\arg\max_{g} \sum_{R \in \mathcal{R}, u \in R} \Big[\sum_{v \in R \setminus u} \big[(\mathbf{v'}_u^\top \mathbf{v}_v) - \log \sum_{w \in \mathbb{U}} \exp(\mathbf{v'}_w^\top \mathbf{v}_v) \big] \Big] \tag{4}$$

Our neural network is trained using stochastic gradient descent and updates $\mathbf{W}$ and $\mathbf{W'}$ gradually via backpropagation. After the training converges, each row of $\mathbf{W}$ represents the d-dimensional user embeddings.

2.3 User Community Detection

Given the user embeddings, we identify communities of users through graph-based partitioning heuristics. We represent users and their pairwise similarities through a weighted undirected graph. Precisely, let G = $(\mathbb{V}, \mathbb{E}, s)$ be a weighted user graph in time period T such that $\mathbb{V} = \mathbb{U}, \mathbb{E} = \{e_{u,v} : \forall u, v \in \mathbb{U}\}$ and the weight function $s : \mathbb{E} \rightarrow [0, 1]$ is the cosine similarity of embeddings for the incident users of an edge defined as $s(e_{u,v}) = \frac{\mathbf{v}_u \cdot \mathbf{v}_v}{||\mathbf{v}_u||_2 ||\mathbf{v}_v||_2}$. After constructing the user graph G for a given time period T, it is possible to employ a graph partitioning heuristic to extract clusters of users that form latent communities. We leverage the Louvain method (LM) [2] as it introduces linear heuristics to the problem of graph partitioning. The output is a set of induced subgraphs such as G[$\mathbb{C}$] whose vertex set $\mathbb{C} \subset \mathbb{V}$ and edge set consists of all of the edges in $\mathbb{E}$ that have both endpoints in $\mathbb{C}$. Subgraph $\mathrm{G}[\mathbb{C}]$ with $|\mathbb{C}| \geq 2$ form an instance of *temporal like-minded user community* assuming $\mathbb{C} \in \mathbb{P}^*$. The application of graph partitioning algorithms on G will produce temporal user communities $\mathbb{P}^*$ that consist of like-minded users who have contributed to the same topics with similar temporal behavior and contribution degrees.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. [n.d.]. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS 2013*.

[4] Fattane Zarrinkalam, Hossein Fani, Ebrahim Bagheri, and Mohsen Kahani. [n.d.]. Inferring Implicit Topical Interests on Twitter. In *ECIR 2016*.

[5] Fattane Zarrinkalam, Hossein Fani, Ebrahim Bagheri, and Mohsen Kahani. [n.d.]. Predicting Users' Future Interests on Twitter. In *ECIR 2017*.

[6] Lizhuang Zhao and Mohammed Javeed Zaki. [n.d.]. TriCluster: An Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data. In *SIGMOD 2005*.