HID: Hierarchical Multiscale Representation Learning for Information Diffusion

Honglu Zhou , **Shuyuan Xu** , **Zuohui Fu** ,
Gerard de Melo , **Yongfeng Zhang** and **Mubbasir Kapadia**
Department of Computer Science, Rutgers University, New Brunswick, NJ 08901
{honglu.zhou, shuyuan.xu, zuohui.fu}@rutgers.edu,
gdm@demelo.org, yongfeng.zhang@rutgers.edu, mk1353@cs.rutgers.edu

Abstract

Multiscale modeling has yielded immense success on various machine learning tasks. However, it has not been properly explored for the prominent task of information diffusion, which aims to understand how information propagates along users in online social networks. For a specific user, whether and when to adopt a piece of information propagated from another user is affected by complex interactions, and thus, is very challenging to model. Current state-of-the-art techniques invoke deep neural models with vector representations of users. In this paper, we present a **H**ierarchical **I**nformation **D**iffusion (`HID`) framework by integrating user representation learning and multiscale modeling. The proposed framework can be layered on top of *all* information diffusion techniques that leverage user representations, so as to boost the predictive power and learning efficiency of the original technique. Extensive experiments on three real-world datasets showcase the superiority of our method.

1 Introduction

The accurate prediction of *information diffusion* is beneficial to a wide range of applications. For instance, it may help in modeling user behavior such as clicks and commenting, so that user interest is better captured in recommender systems [Leskovec *et al.*, 2007; Xian *et al.*, 2019]. It can also support persuasion campaigns targeting public opinion [Nadeau *et al.*, 2008], and bring similar advantages to other tasks such as influencer identification and viral marketing [Guille *et al.*, 2013]. Information diffusion is an incredibly well-studied topic. While traditional stochastic probabilistic-based models dominated for many years [Kempe *et al.*, 2003; Leskovec *et al.*, 2007], recent advances include deep diffusion models [Wang and Li, 2019; Yang *et al.*, 2019], which benefit from their robust generalization abilities. Nevertheless, how to accurately represent the users to better capture the process of information diffusion remains well-known as a difficult problem [Wang *et al.*, 2019]. Specifically, whether and when a user is likely to adopt a specific piece of information may depend on a multitude of complex interactions, of which the ultimate cause remains unknown [Guille *et al.*, 2013]. To tackle this, existing deep models tend to adopt *user representation learning*. In this paper, we propose a novel *framework* (Fig. 1) for these models that boosts their performance and learning efficiency.

In particular, our work is motivated by the importance of the user's role in the information diffusion process [Wang *et al.*, 2019]. In order to capture the user diffusion behavior, previous works either devise complex custom models [Feng *et al.*, 2018], or rely on external knowledge such as the friendship network and user profile [Yang *et al.*, 2019; Lagnier *et al.*, 2013], which is often less organized, anonymous, or inaccessible due to privacy policies [Mano and Ishikawa, 2010]. Therefore, we seek to exploit the traces of the user diffusion behavior themselves (i.e., diffusion paths), from which we extract multiple aspects of user behavior. The idea draws inspiration from recent achievements that multiscale modeling has made on various machine learning tasks [Alber *et al.*, 2019]. Data from the real world can naturally be encoded at multiple scales, which can serve as rich and reliable resources for feature representations in deep learning. There is an urgent need for the deep models to exploit these implicit multiple scales. However, the exploration of multiscale modeling in information diffusion has largely been neglected. Against this background, we explore multiscale modeling in a user representation learning framework. The ***challenges*** are mainly three-fold: (1) how to discover and induce multiple scales, and at the same time preserve the individual behavior patterns in the original data while the scales are implicit; (2) how to transfer knowledge among multiple scales in an efficient and effective way; and (3) how to design generalizable strategies that apply to most state-of-the-art diffusion models.

To address the above-mentioned problems, we propose the **H**ierarchical **I**nformation **D**iffusion (`HID`) framework. Our method has several appealing properties, most notably:

- `HID` aids the learning of user representations, with multiple scales of summarization over the diffusion paths. This is accomplished through *upscaling*, by means of grouping the users and then coarsening the diffusion paths at each scale. A k-order diffusion proximity matrix is proposed to support this. Fewer users and activities at the coarsened scale also brings about efficiency.
- *Downscaling* achieves an exchange of knowledge about users across scales, inspired by studies on artificial neu-

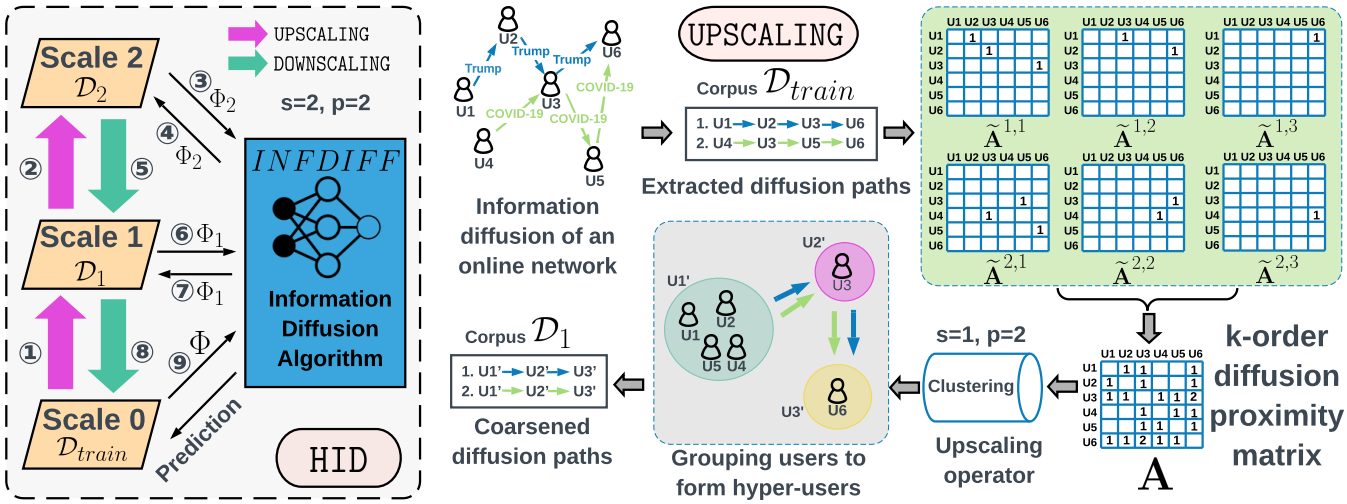

Figure 1: Methodology overview. *Left*: Framework overview of HID with a toy example. Numbers next to the arrow indicate the order of the algorithm workflow. *Right*: The UPSCALING procedure for generating multiple scales of information diffusion data corpus.

ral networks revealing that the importance of a neuron for the model performance correlates with the extent to which its weight distribution after training differs from the initialization [Meyes *et al.*, 2019].

- Combining upscaling and downscaling yields a hierarchical learning framework that has the potential to group inactive users with others, so that more cycles of optimization are gained for the inactive users compared to sampling-based optimization without multiscaling.

Contribution. We propose HID[1] for information propagation in online networks. To the best of our knowledge, this is the first method to integrate user representation learning and multiscale modeling. HID captures the multiscale information adoption proximity between users, and can be layered on top of *all* existing information diffusion techniques with a user vector representation learning component. Through extensive experiments, we showcase the benefits of multiscale learning on 3 real datasets. Our method yields substantially higher predictive power and learning efficiency, and achieves empirical gains of up to **14.76**% MAP score.

2 Related Work

Information Diffusion with User Representations. [Bourigault *et al.*, 2014; Bourigault *et al.*, 2016; Gao *et al.*, 2017] utilize representation learning to predict unseen diffusion. [Feng *et al.*, 2018] further consider network structure and user interest similarity. [Singh *et al.*, 2017] learn user representations with a novel path sampling procedure to gather contexts for each user. [Wang *et al.*, 2019] learn network regularized role-based user representations. [Zhang *et al.*, 2018] learn user embeddings while regularizing the information diffusion process using community structures. [Yang *et al.*, 2018] employ an attention mechanism and convolutional network with a user embedding layer. [Wang *et al.*, 2018] employ a recurrent network with a user embedding layer and explore both diffusion paths and graph by structure attention.

[1]https://github.com/hongluzhou/HID

Multiscale Representation Learning. Multiscale representation learning for information diffusion is *unexplored*. [Yang *et al.*, 2019] jointly handle both micro- and macroscopic prediction, and [Wang and Li, 2019] adopt a hierarchical attention mechanism. Both models can benefit from HID. Additionally, [Zhang *et al.*, 2017] capture hierarchical community-level information diffusion, but as a probabilistic model. Multiscale representation learning for graphs, in contrast, has recently received substantial attention [Yang *et al.*, 2017; Shen and Chung, 2017; Sang *et al.*, 2019; Morris *et al.*, 2019; Chen *et al.*, 2018; Perozzi *et al.*, 2017]. Though closely related, we show that multiscale graph representation techniques fail to tackle information diffusion in Sec. 5.1.

3 Method

3.1 Notations and Definitions

An online social network (OSN) can be described as a platform that permits its users to share information [Guille *et al.*, 2013]. Formally, the set of users that an OSN has is denoted as V. A corpus $\mathcal{D}$ is a set of diffusion paths, typically extracted from an OSN and generated by its $|\mathrm{V}|$ users during the information-sharing process. Our predictive task is information diffusion.

Definition 1. *(Diffusion Path): A sequence of non-repetitive users who have shared a certain information unit, denoting the ordered user path capturing the order in which the users of the OSN adopted the piece of information.*

Definition 2. *(Information Diffusion): Given $|\mathrm{V}|$ users and a corpus $\mathcal{D}_{\text{train}}$, the task aims to complete the diffusion paths in the corpus $\mathcal{D}_{\text{test}}$ with users in V. The first user (i.e., source user) of each diffusion path in $\mathcal{D}_{\text{test}}$ is known.*

Popular deep learning-based approaches commonly model information diffusion with user representation learning. We define a user representation as a mapping $\Phi : v \in \mathrm{V} \mapsto \mathbb{R}^d$, which maps every user to a d-dimensional vector in the Euclidean space ($d \ll |\mathrm{V}|$). In practice, Φ is a $|\mathrm{V}| \times d$ matrix estimated from a training corpus.

Algorithm 1 $\texttt{HID}(s, p, d, \mathcal{D}_{\text{train}}, \mathcal{F})$

Input:
the number of coarse scales s
the coarsening rate between adjacent two scales p
the user embedding dimensionality d
an arbitrary information diffusion algorithm that leverages user representations $\mathcal{F}$
a corpus $\mathcal{D}_{\text{train}}$ with user set V
Output: latent representation of users $\Phi : v \in \text{V} \mapsto \mathbb{R}^d$

$\mathcal{D}_0 \leftarrow \mathcal{D}_{\text{train}}$
$\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_s \leftarrow \texttt{UPSCALING}(s, p, \mathcal{D}_0)$
Initialize Φ_s
$\Phi_s \leftarrow \textit{INFDIFF}(\Phi_s, \mathcal{D}_s, d, \mathcal{F})$
for $i = s - 1$ to 0 **do**
    $\Phi_i \leftarrow \texttt{DOWNSCALING}(\Phi_{i+1}, \mathcal{D}_i, \mathcal{D}_{i+1})$
    $\Phi_i \leftarrow \textit{INFDIFF}(\Phi_i, \mathcal{D}_i, d, \mathcal{F})$
end for
$\Phi \leftarrow \Phi_0$
return Φ

3.2 Problem Formulation

We desire to learn Φ, a latent representation of users, to better predict information diffusion. Current approaches for estimating Φ suffer from two main disadvantages: (1) the multiscale property is not considered, and (2) their stochastic optimization can easily fall into local minima due to troublesome initial configurations. In light of these deficiencies, we introduce *hierarchical user representation learning* for modelling information diffusion in an OSN:

Given the number of coarse scales s, simplify the corpus $\mathcal{D}_{\text{train}}$ to a series of successively coarser corpora, $\mathcal{D}_1$, $\mathcal{D}_2$, $\ldots$, $\mathcal{D}_s$, with respective user sets V_1, V_2, $\ldots$, V_s ($|\text{V}| > |\text{V}_1| > \cdots > |\text{V}_s|$). Learn the corresponding family of user representations Φ_1, Φ_2, $\ldots$, Φ_s and then obtain the finest-granularity user representation Φ to predict the diffusion paths in $\mathcal{D}_{\text{test}}$.

3.3 Proposed Approach

Algorithm 1 outlines the overall HID approach. Our underlying method is general and can be applied to *any* technique for information diffusion that leverages user representations. We denote the information diffusion algorithm that can benefit from HID as *INFDIFF*. The *INFDIFF* procedure will learn user representations and predict information diffusion.

Upscaling. We present UPSCALING in Algorithm 2. The upscaling process achieves a successive abstraction over the users. Abstraction is done such that the coarser scale will have fewer users, but the key characteristics of the diffusion paths are mostly retained. Specifically, the change in diffusion paths is minimized and the user ordering is preserved.

Consider a corpus $\mathcal{D}_{\text{train}}$ of an OSN represented as a diffusion graph $G = (\text{V}, \text{E})$, where each node is a user in V and there is a directed edge pointing from node i to node j for every ordered successive user pair (user i, user j) in the corpus $\mathcal{D}_{\text{train}}$. Accordingly, each diffusion path can be viewed as a path in this diffusion graph G.

Recall that [Qiu *et al.*, 2018; Perozzi *et al.*, 2017] have shown that popular graph embedding approaches are implicitly factoring a matrix containing entries of $\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^k$, where k is the window size over the random walk, and the entry $\mathbf{A}^k_{ij}$ is the number of paths between nodes i and j of length k. We define matrix $\widetilde{\mathbf{A}}^{m,k}$, a $|\text{V}| \times |\text{V}|$ matrix, for each diffusion path, representing the adjacency matrix of diffusion path m with step size k. Entry $\widetilde{\mathbf{A}}^{m,k}_{ij}$ is the number of times that the ordered pair of user i and user j with step k appeared in the diffusion path m, and thus can only be 1 or 0.

In this paper, the multiscale property serves the following functions: (1) capturing directed as well as both local and long-distance information adoption proximity between two different users, and (2) the consideration of distinct connections of information adoption patterns in terms of different transitional orders and diffusion paths. To capture the multiscale property of a corpus $\mathcal{D}_{\text{train}}$, we define the following k-order diffusion proximity matrix:

$$\mathbf{A} = \sum_{m=1}^{|\mathcal{D}_{\text{train}}|} \sum_{k=1}^{l} \widetilde{\mathbf{A}}^{m,k} + \left(\sum_{m=1}^{|\mathcal{D}_{\text{train}}|} \sum_{k=1}^{l} \widetilde{\mathbf{A}}^{m,k} \right)^{\text{T}}, \quad (1)$$

where $|\mathcal{D}_{\text{train}}|$ is the total number of diffusion paths in $\mathcal{D}_{\text{train}}$ and l is the maximum possible step size in $\mathcal{D}_{\text{train}}$ (i.e., the length of the longest diffusion path minus one). The step of calculating $\mathbf{A}$ is called $ObtainProximity$. $\mathbf{A}$ accounts for the bi-directional co-occurrence patterns of users along diffusion paths, and may be decomposed or transformed, with subcomponents having a genuine practical interpretation, e.g.,

$$\mathbf{A_n} = \sum_{m=1}^{|\mathcal{D}_{\text{train}}|} \sum_{k=1}^{\tau} \widetilde{\mathbf{A}}^{m,k}, \quad (2)$$

where τ is a pre-defined neighborhood threshold ($\tau < l$) and thus $\mathbf{A_n}$ considers neighboring patterns in terms of similar information adoption time of users. Further,

$$\mathbf{A_o} = \sum_{m=1}^{|\mathcal{D}_{\text{train}}|} \left[\mathbf{i}_{e_m} \cdot \sum_{k=1}^{l} \widetilde{\mathbf{A}}^{m,k}_{e_m} \right], \quad (3)$$

where e_m is the source user's index in Φ of diffusion path m, and $\mathbf{i}_{e_m}$ is a $|\text{V}|$-dimensional vector, which serves as an indicator function (having a 1 in the e_m-th entry and 0s elsewhere). In this manner, $\mathbf{A_o}$ considers patterns of whether a user would diffuse information originating from another user.

UPSCALING groups users to form hyper-users via clustering. UPSCALING operates in a bottom-up manner across all scales, and at each scale, users who have similar adoption patterns would form a hyper-user. Based upon $\mathbf{A}$ at scale i, we apply $UpscalingOperator$ to form hyper-users at scale $i+1$. Possible upscaling operators include Hierarchical Agglomerative Clustering (HAC), Spectral Clustering, K-means, etc. The number of users at scale $i + 1$ is defined by p, i.e., the coarsening rate between adjacent two scales, and is calculated as the number of users at scale i divided by p.

The $RewriteCorpus$ procedure then updates the corpus $\mathcal{D}_i$ and obtains $\mathcal{D}_{i+1}$ by replacing user ID at scale i with the corresponding hyper-user ID at scale $i + 1$, while ensuring the resulting diffusion path is still valid. Since we want to make sure the change in diffusion paths is minimized and

Algorithm 2 UPSCALING$(s, p, \mathcal{D}_{\text{train}})$

Input:
the number of coarse scales s
the coarsening rate between adjacent two scales p
a corpus $\mathcal{D}_{\text{train}}$ with user set V
Output: a series of successively coarser corpora, $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_s$

$\mathcal{D}_0 \leftarrow \mathcal{D}_{\text{train}}$
for $i = 0$ to $s - 1$ **do**
  $\mathbf{A}_i \leftarrow ObtainProximity(\mathcal{D}_i)$
  $\mathrm{V}_{i+1} \leftarrow UpscalingOperator(\mathbf{A}_i, \mathrm{V}_i, p)$
  $\mathcal{D}_{i+1} \leftarrow RewriteCorpus(\mathcal{D}_i, \mathrm{V}_i, \mathrm{V}_{i+1})$
end for
return $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_s$

the user ordering is preserved, we only consider the first occurrence of every hyper-user. In addition, if the number of users on a coarsened diffusion path is less than 3, we discard that diffusion path, because many diffusion models require at least 3 users on a diffusion path [Bourigault *et al.*, 2014; Gao *et al.*, 2017]. Corpora $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_s$ is thus obtained.

Downscaling. The DOWNSCALING process maps a coarser user representation Φ_{i+1} at scale $i+1$ to its subsequent finer-grained user representation Φ_i at scale i, which is used as the initialization point for learning the finer-grained user representation. Specifically, for each user at scale i, DOWNSCALING finds the user's corresponding hyper-user at scale $i+1$, then assigns the *learned* hyper-user's representation (returned by *INFDIFF*) as that user's *initial* representation. DOWNSCALING transfers the knowledge of user representations across scales, and thus provides good initializations for learning, effectively avoiding troublesome configurations that are more prone to local minima in non-convex optimization [Chen *et al.*, 2018].

4 Experimental Setup

We apply HID to current deep learning-based information diffusion models that involve user representation learning. For each, we compare accuracy and efficiency changes, with and without applying HID. In addition, HARP and Walklets are multiscale graph representation techniques to compare the multiscale representation learning aspect.

4.1 Baseline Methods

CDK [Bourigault *et al.*, 2014] induces user embeddings such that users contaminated earlier are closer to the source user than users contaminated later or not contaminated.

CSDK [Bourigault *et al.*, 2014] generates user embeddings such that users contaminated first are closer to the source user after applying the information embedding as an offset, than users contaminated later, or not contaminated at all.

Forest [Yang *et al.*, 2019] is a deep diffusion model based on reinforcement learning (RL). RL incorporates the diffusion size information into the recurrent neural network model.

HARP [Chen *et al.*, 2018] is a method for learning node embeddings of a graph by compressing the input graph prior to embedding it. HARP is a meta-strategy, proven to improve the state-of-the-art algorithms for embedding graphs.

Walklets [Perozzi *et al.*, 2017] is an approach for learning multiscale representations of nodes in a graph, by subsampling short random walks (i.e., "skipping" over steps).

Dataset	Memetracker	Twitter	Digg
Number of Users	994	1,222	500
Number of Links	32,652	166,889	486,354
Number of Diffusion Paths	4,319	9,761	3,553
Avg. Diffusion Path Length	8.56	18.10	137.89

Table 1: Statistics of datasets used in our experiments.

4.2 Datasets

Three real datasets are used. Table 1 gives the statistics.

Memetracker [Leskovec *et al.*, 2009]. This corpus contains blog posts and Web news article from August 1, 2008 to April 30, 2009, and is often used for research on information diffusion [Bao *et al.*, 2016]. Each website or blog is considered as a user. We consider a subset with roughly one thousand of most active users. Such filtering is common for fast verification of model performance [Wang *et al.*, 2019].

Twitter [Yang and Leskovec, 2011]. This dataset retrieves Twitter tweets from June 1, 2009 to December 31, 2009. For each tweet, author, time, and content (i.e., information) are available. We consider users with at least 600 tweets and then discard diffusion paths with fewer than 10 users.

Digg [Hogg and Lerman, 2012]. This dataset contains stories promoted to Digg's front page over a period of a month in 2009. For each story, all Digg users who have voted for the story up to the time of data collection are captured. We consider the 500 most active users.

4.3 Evaluation Metrics

For each dataset, the set of diffusion paths is randomly split into two parts: 80% for training and validation ($\mathcal{D}_{\text{train}}$), and the remainder for testing ($\mathcal{D}_{\text{test}}$). We evaluate the performance through the widely adopted metric Mean Average Precision (MAP)[2] [Bourigault *et al.*, 2014; Bourigault *et al.*, 2016; Wang and Li, 2019], computed as

$$\text{MAP} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{t \in \mathcal{D}_{\text{test}}} \frac{\sum_{k=1}^{|t|} P@k \times \text{isInfected}(k)}{|t|}, \quad (4)$$

where t is a diffusion path in $\mathcal{D}_{\text{test}}$, and $P@k$ is the precision at rank k, i.e., the percentage of infected users among the top k users. $\text{isInfected}(k)$ is 1 when the k-th user truly participates in this diffusion path, and 0 otherwise.

4.4 Parameter Settings

The hyper-parameters are chosen based on validation performance. For CDK, the maximum training epoch was $8,000$ and per epoch the number of samples was $5,000$. The initial learning rate was 0.01 with a decay of 1×10^{-6}. CSDK shared the same parameters, except $10,000$ for the number of samples per epoch and 1×10^{-12} for decay. For Forest, HARP, and Walklets, we used the parameters suggested by the authors. Forest used a maximum training epoch of 24. We built the diffusion graph (Sec. 3) for HARP and Walklets. Since the friendship network of Digg is available, we also tried the friendship network, reporting whichever gave better results (diffusion graph for HARP and friendship network for

[2] Additional metrics such as Area Under the Receiver Operating Characteristic Curve were computed, but we opted not to report due to the similar trend that all metrics indicate and limited space.

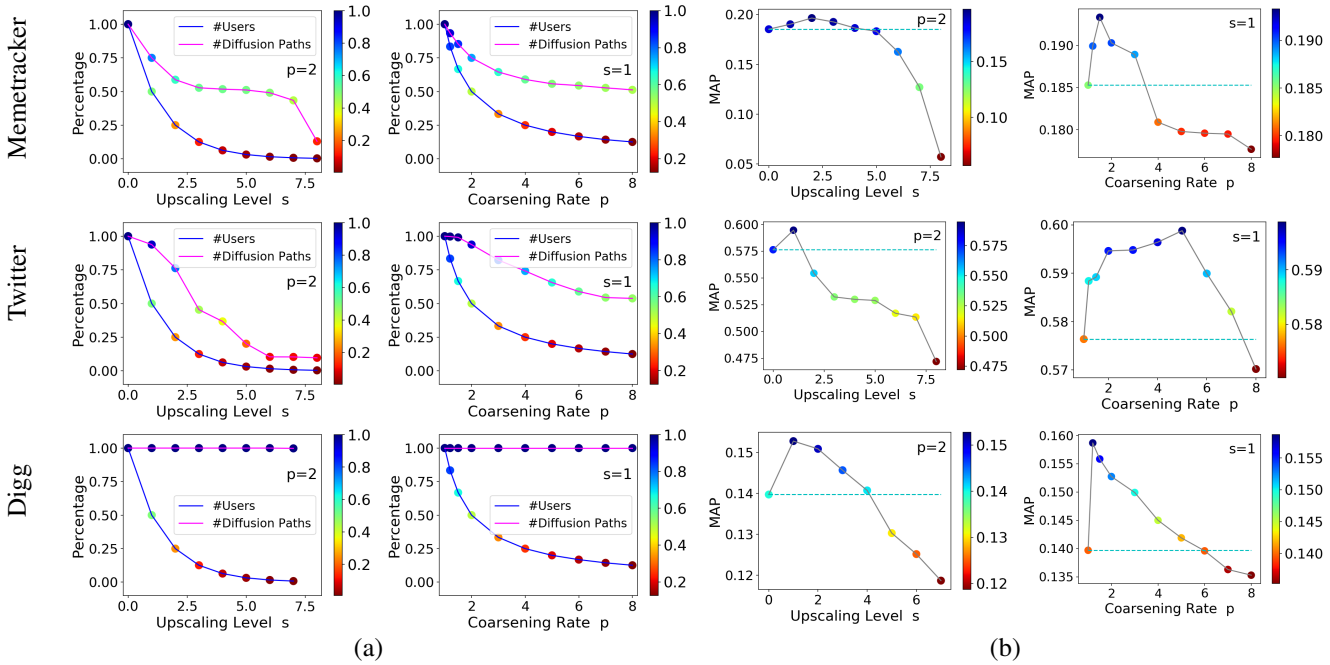

Figure 2: Parameter sensitivity analysis showing the upscaling effect on the original corpus (a) and on HID performance (b). The dotted horizontal line in cyan shows the MAP score achieved by the baseline CDK (i.e., before applying HID). The first 3 scales tend to affect the corpus the most, but are most likely to achieve improvements. Less than 10% of users remain after 4 scales when $p = 2$, suggesting the critical point for coarsening. The extent of coarsening varies across datasets, which implies the level of difficulty for HID to bring improvements on the dataset (less affecting the number of diffusion paths, easier to bring improvements, e.g., Digg and counterexample Twitter).

Walklets). We used the HARP algorithm type that gave the best performance. Specifically, HARP(node2vec) for Memetracker, HARP(LINE) for Twitter and HARP(DeepWalk) for Digg. Using HARP and Walklets, during the prediction, we rank the source user's neighbors according to the Euclidean distance in the user representation space (same technique introduced in CDK). To ensure a fair comparison, we reduced the number of training epochs for the HID enhanced version, according to the parameter s. Specifically, if the maximum training epoch of the baseline is e, the HID enhanced version uses $e/(s+1)$. The same set of the above hyper-parameters was used for all datasets. The user representation dimensionality was 64. HID requires two hyper-parameters, s and p. For results in Table 2, different values of s in $\{1, 2, 3\}$ and different values of p in $\{1.2, 1.5, 2, 3, 4\}$ were tried with a grid search using the validation data before choosing the best-performing settings, which in general vary across datasets, baseline techniques, and the upscaling operators. How these two hyper-parameters affect HID is discussed in Sec. 5.2.

5 Evaluation

5.1 Results

We present the results in Table 2. HID consistently produces better results than all compared methods. On Memetracker, the relative gains of HID over CDK, CSDK, and Forest are 6.1%, 4.34%, and 4.72%, respectively. On Twitter, the baselines tend to perform very well with high MAP scores (e.g., CDK has a MAP score of 0.5763). Still, HID is able to bring performance gains. On Twitter, the improvements introduced by HID(CDK), HID(CSDK), and HID(Forest) are 3.61%, 2.47%, and 2.16%. Given the long diffusion paths of Digg (see Table 1), upscaling succeeds at preserving the characteristics of the original corpus (Fig. 2), which provides more opportunities for HID to be effective. The gains are particularly striking on Digg: HID(CDK), HID(CSDK), and HID(Forest) outperform the baselines by 14.75%, 3.28%, and 12.6%, respectively. The somewhat more modest gain of HID(CSDK) on Digg might stem from the additional learning of information representations, which plays a dominant role in CSDK's diffusion modeling process. The improvements introduced by HID are *statistically significant*: every experiment is repeated 5 times to ensure the reliability of our results, and we consistently see higher accuracy with our framework.

In Table 2, results on different upscaling operators are also demonstrated. In most cases, HAC yields the best results, and occasionally, is significantly better than K-means and Spectral Clustering (e.g., on Digg). This might be because HID operates in a hierarchical way, which accords with HAC. Spectral Clustering tends to give the lowest performance gains. Spectral clustering treats the data points as vertices of a graph, and connects vertices that are close enough. Noisy online data without well-separated connected components might be the reason why Spectral Clustering is less suitable.

We also compare HID with HARP and Walklets in Table 2. HARP and Walklets are multiscale graph representation learning techniques. Between the two, HARP gives the better results. However, suffering from the closed-world assumption [Guille *et al.*, 2013], neither of them are able to effectively tackle the prediction task of information diffusion. On the 3 datasets, HARP and Walklets give worse or only comparable results to CDK, CSDK, and Forest, which are algorithms designed for information diffusion and avoid

Algo	Memetracker		Twitter		Digg	
	MAP	Gain	MAP	Gain	MAP	Gain
CDK	*0.1852*	N/A	*0.5763*	N/A	*0.1397*	N/A
HID_H	$0.1965^\star$	**6.1**	$0.5971^\star$	**3.61**	$0.1603^\star$	**14.75**
HID_K	$0.1899^\star$	**2.54**	$0.5885^\star$	**2.12**	$0.1455^\star$	**4.15**
HID_S	0.1855	**0.16**	0.5773	**0.17**	0.1451^*	**3.87**
CSDK	*0.2074*	N/A	*0.5712*	N/A	*0.1492*	N/A
HID_H	$0.2164^\star$	**4.34**	$0.5844^\star$	**2.31**	0.1541^*	**3.28**
HID_K	0.2098^*	**1.16**	0.5804^*	**1.61**	0.1516^*	**1.61**
HID_S	0.2082	**0.39**	$0.5853^\star$	**2.47**	0.1501	**0.6**
Forest	*0.3244*	N/A	*0.5915*	N/A	*0.1500*	N/A
HID_H	$0.3397^\star$	**4.72**	$0.6019^\star$	**1.76**	$0.1689^\star$	**12.6**
HID_K	$0.3376^\star$	**4.07**	$0.6043^\star$	**2.16**	$0.1583^\star$	**5.53**
HID_S	$0.3296^\star$	**1.6**	$0.6029^\star$	**1.93**	$0.1564^\star$	**4.27**
HARP	*0.1704*	**99.35***	*0.5745*	**5.19***	*0.1376*	**22.75***
Walklets	*0.1581*	**114.86***	*0.5744*	**5.21***	*0.1208*	**39.82***

Table 2: Performance on test data and gains of HID in percentage (bolded). Results of the compared methods are in italics. Subscript of HID shows the upscaling operator, where 'H', 'K', and 'S' stands for 'HAC', 'K-means', and 'Spectral' Clustering. Gains of HARP and Walklets are calculated using the best HID result. Every experiment is repeated 5 times and the mean metric value is reported. $(*,\star)$ indicates statistically superior performance to the compared method at a significance level of $(0.05, 0.001)$ using a standard paired t-test.

the need of a pre-defined graph. Approaches requiring a pre-defined graph are said to be limited in terms of their ability to explain future diffusion and are less optimal [Bourigault *et al.*, 2014; Gao *et al.*, 2017]. Based upon the above, HID, a framework for enhancing information diffusion techniques, gives superior performance compared to HARP and Walklets.

5.2 Parameter Sensitivity

HID involves 2 parameters: the number of coarse scales s and the coarsening rate p. Using CDK as the baseline with HAC, we examine how different choices of s and p affect the extent of coarsening on the original corpus and HID performance.

We show the effect on the original corpus in Fig. 2 (a). The first column reports the effect on the relative total number of users and diffusion paths, i.e., the ratio of the number of users/diffusion paths of the coarsened scale to that of the original corpus, with varied values of s when $p = 2$; and the second column shows the same but with varied values of p when $s = 1$. The first 3 scales tend to affect the corpus the most. Fewer than 10% of users remain after 4 scales when $p = 2$, and approximately only 20% remain when p is greater than 5 with $s = 1$, both indicating the critical point that might suffer from performance loss. The level of coarsening varies across datasets, which implies the level of difficulty for HID to bring improvements. Among the 3 datasets, s and p tend to affect Twitter the most, which explains why improvements on Twitter are relatively hard to achieve (Sec. 5.1).

We measure HID performance on the testing set as a function of s and of p in Fig. 2 (b). With regard to s, the performance of HID first improves as s increases, then decreases as s keeps increasing. The performance is expected to increase when coarsening is appropriate. In HID, the coarsening operates over the users. The coarse scale learns initial user representations for the finer scales, so a suitable extent of coarsening with an adequate coarsening strategy will ensure the characteristics of users' behavior are preserved in the coarsened scales. When s becomes too large, the number of users and diffusion paths remaining in the coarsest scale will be too small to serve as a summary of the original corpus. This is why the performance would decrease afterwards, which is not surprising. Similarly, we observe that there is a range of acceptable values for p, after which p becomes too large and destroys the original patterns that are crucial for information diffusion prediction. Both parameters have a fairly high impact on the performance. The range of best s (or p) can be implied by Fig. 2 (a). E.g., on Memetracker, the critical point for s to affect the original corpus is around 4, hence, as shown in Fig. 2 (b), the range of best s is from 1 to 4. Practitioners can use Fig. 2 (a) to facilitate the selection of s and p.

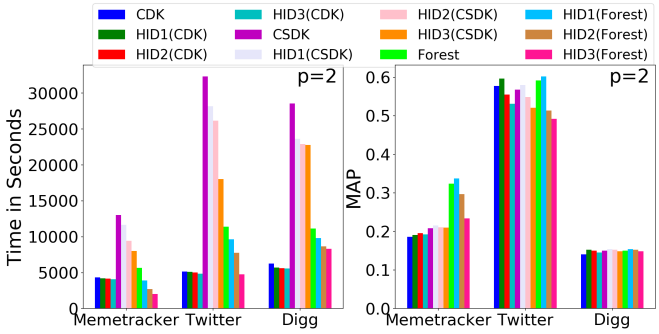

Figure 3: Efficiency comparison between methods and their MAP scores. The number after "HID" is the value for s. HID is always more efficient as s increases. A trade-off between efficiency and effectiveness is sometimes observable when s is greater than 1.

5.3 Scalability

In Fig. 3, we compare the run time of HID and other techniques. All models run on a single machine with 256 GB memory, 48 CPU cores at 2.30GHz, and an NVIDIA Quadro K6000 graphics card. Though sharing the same total number of training epochs, HID is always more efficient compared to the baseline, and becomes more efficient as s increases. This is because fewer users and activities are involved at the coarsened scales. The run time decreases almost linearly with respect to the remaining corpus size. A trade-off between efficiency and performance is sometimes observed when $s > 1$ (e.g., Twitter). Still, HID can be a valuable tool for large-scale online networks due to its substantial scalability.

6 Conclusion

This paper presents a novel hierarchical framework for the task of information diffusion. The proposed framework HID can be layered on top of *all* information diffusion techniques that leverage user representations. HID facilitates more efficient learning and accurate prediction. Extensive experiments show the superior performance. In the future, we hope to extend HID to model additional aspects (e.g., information representations, combination of different upscaling operators, mix of knowledge by downscaling) so as to obtain even further improvements.

Acknowledgements

This research was supported in part by NSF IIS-1703883 and NSF S&AS-1723869.

References

[Alber *et al.*, 2019] Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *npj Digital Medicine*, 2019.

[Bao *et al.*, 2016] Qing Bao, William K Cheung, and Jiming Liu. Inferring motif-based diffusion models for social networks. In *IJCAI*, pages 3677–3683, 2016.

[Bourigault *et al.*, 2014] Simon Bourigault, Cedric Lagnier, et al. Learning social network embeddings for predicting information diffusion. In *WSDM*. ACM, 2014.

[Bourigault *et al.*, 2016] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. Representation learning for information diffusion through social networks: an embedded cascade model. In *WSDM*, pages 573–582. ACM, 2016.

[Chen *et al.*, 2018] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks. In *Proceedings of AAAI*, 2018.

[Feng *et al.*, 2018] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. Inf2vec: Latent representation model for social influence embedding. In *ICDE*, pages 941–952. IEEE, 2018.

[Gao *et al.*, 2017] Sheng Gao, Huacan Pang, Patrick Gallinari, et al. A novel embedding method for information diffusion prediction in social network big data. *IEEE Transactions on Industrial Informatics*, 13(4):2097–2105, 2017.

[Guille *et al.*, 2013] Adrien Guille, Hakim Hacid, Cecile Favre, et al. Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2), 2013.

[Hogg and Lerman, 2012] Tad Hogg and Kristina Lerman. Social dynamics of digg. *EPJ Data Science*, 1(1):5, 2012.

[Kempe *et al.*, 2003] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.

[Lagnier *et al.*, 2013] Cédric Lagnier, Ludovic Denoyer, et al. Predicting information diffusion in social networks using content and user's profiles. In *ECIR*, 2013.

[Leskovec *et al.*, 2007] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[Leskovec *et al.*, 2009] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *SIGKDD*, pages 497–506. ACM, 2009.

[Mano and Ishikawa, 2010] Masanori Mano and Yoshiharu Ishikawa. Anonymizing user location and profile information for privacy-aware mobile services. In *SIGSPATIAL Workshop on LBSN*. ACM, 2010.

[Meyes *et al.*, 2019] Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, et al. Ablation studies in artificial neural networks. *CoRR*, abs/1901.08644, 2019.

[Morris *et al.*, 2019] Christopher Morris, Martin Ritzert, et al. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.

[Nadeau *et al.*, 2008] Richard Nadeau, Neil Nevitte, Elisabeth Gidengil, and André Blais. Election campaigns as information campaigns: who learns what and does it matter? *Political Communication*, 25(3):229–248, 2008.

[Perozzi *et al.*, 2017] Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don't walk, skip!: online learning of multi-scale network embeddings. In *ASONAM*, pages 258–265. ACM, 2017.

[Qiu *et al.*, 2018] Jiezhong Qiu, Yuxiao Dong, et al. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*, 2018.

[Sang *et al.*, 2019] Lei Sang, Min Xu, et al. AAANE: Attention-based adversarial autoencoder for multi-scale network embedding. In *PAKDD*. Springer, 2019.

[Shen and Chung, 2017] Xiao Shen and Fu-Lai Chung. Deep network embedding with aggregated proximity preserving. In *ASONAM*, pages 40–43. ACM, 2017.

[Singh *et al.*, 2017] Harvineet Singh, Amitabha Bagchi, and Parag Singla. Learning user representations in online social networks using temporal dynamics of information diffusion. *arXiv preprint arXiv:1710.07622*, 2017.

[Wang and Li, 2019] Zhitao Wang and Wenjie Li. Hierarchical diffusion attention network. In *IJCAI*, pages 3828–3834. AAAI Press, 2019.

[Wang *et al.*, 2018] Zhitao Wang, Chengyao Chen, and Wenjie Li. A sequential neural information diffusion model with structure attention. In *CIKM*. ACM, 2018.

[Wang *et al.*, 2019] Zhitao Wang, Chengyao Chen, et al. Information diffusion prediction with network regularized role-based user representation learning. *TKDD*, 2019.

[Xian *et al.*, 2019] Yikun Xian, Zuohui Fu, S. Muthukrishnan, et al. Reinforcement knowledge graph reasoning for explainable recommendation. *SIGIR*, 2019.

[Yang and Leskovec, 2011] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186. ACM, 2011.

[Yang *et al.*, 2017] Cheng Yang, Maosong Sun, Zhiyuan Liu, and Cunchao Tu. Fast network embedding enhancement via high order proximity approximation. In *IJCAI*, 2017.

[Yang *et al.*, 2018] Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, et al. Neural diffusion model for microscopic cascade prediction. *CoRR*, abs/1812.08933, 2018.

[Yang *et al.*, 2019] Cheng Yang, Jian Tang, Maosong Sun, Ganqu Cui, et al. Multi-scale information diffusion prediction with reinforced recurrent networks. In *AAAI*, 2019.

[Zhang *et al.*, 2017] Yuan Zhang, Tianshu Lyu, and Yan Zhang. Hierarchical community-level information diffusion modeling in social networks. In *SIGIR*. ACM, 2017.

[Zhang *et al.*, 2018] Yuan Zhang, Tianshu Lyu, et al. Cosine: Community-preserving social network embedding from information diffusion cascades. In *AAAI*, 2018.