Modeling Complementarity in Behavior Data with Multi-Type Itemset Embedding

DAHENG WANG and QINGKAI ZENG, University of Notre Dame, Notre Dame, IN 46556, USA
NITESH V. CHAWLA, University of Notre Dame, Notre Dame, IN 46556, USA and Department of Computational Intelligence, Wrocław University of Science and Technology, Wrocław, Poland
MENG JIANG, University of Notre Dame, Notre Dame, IN 46556, USA

People are looking for complementary contexts, such as team members of *complementary skills* for project team building and/or reading materials of *complementary knowledge* for effective student learning, to make their behaviors more likely to be successful. Complementarity has been revealed by behavioral sciences as one of the most important factors in decision making. Existing computational models that learn low-dimensional context representations from behavior data have poor scalability and recent network embedding methods only focus on preserving the similarity between the contexts. In this work, we formulate a behavior entry as a set of context items and propose a novel representation learning method, *Multi-type Itemset Embedding*, to learn the context representations preserving the itemset structures. We propose a *measurement of complementarity* between context items in the embedding space. Experiments demonstrate both effectiveness and efficiency of the proposed method over the state-of-the-art methods on behavior prediction and context recommendation. We discover that the complementary contexts and similar contexts are significantly different in human behaviors.

CCS Concepts: • **Information systems → Data mining;**

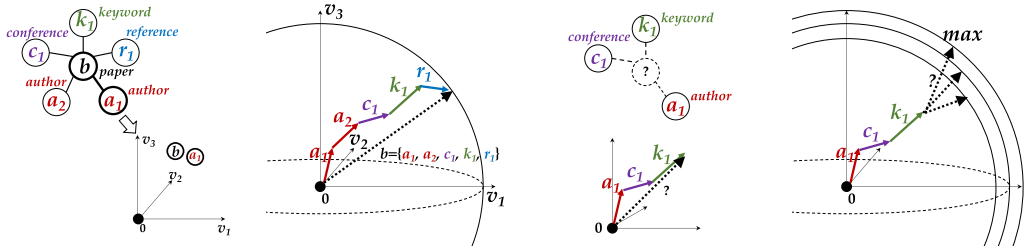
Additional Key Words and Phrases: Behavior modeling, representation learning, prediction, recommendation

ACM Reference format:
Daheng Wang, Qingkai Zeng, Nitesh V. Chawla, and Meng Jiang. 2021. Modeling Complementarity in Behavior Data with Multi-Type Itemset Embedding. *ACM Trans. Intell. Syst. Technol.* 12, 4, Article 42 (June 2021), 25 pages.
https://doi.org/10.1145/3458724

1 INTRODUCTION

Contextual behaviors are defined as the products of interaction between multiple types of contexts [25]. The multi-type contexts often include operators, goals, resources, and spatiotemporal

This research was supported in part by National Science Foundation (NSF) Grants no. IIS-1849816 and no. CCF-1901059. This research was also supported in part by the research project 2016/23/B/ST6/01735, the National Science Centre, Poland.
The affiliation for all four authors when the work was done is "University of Notre Dame, Notre Dame, IN 46556, USA". Nitesh V. Chawla is also affiliated with Department of Computational Intelligence, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.
Authors' addresses: D. Wang, Q. Zeng, and M. Jiang, University of Notre Dame, 326C Cushing Hall, Notre Dame, IN 46556; emails: {dwang8, qzeng, mjiang2}@nd.edu; N. V. Chawla, University of Notre Dame, 384 Nieuwland Science Hall, Notre Dame, IN 46556 and Department of Computational Intelligence, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland; email: nchawla@nd.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
2157-6904/2021/06-ART42 $15.00
https://doi.org/10.1145/3458724



(a) Existing embedding methods preserve pair-wise proximity (e.g., similarity)

(b) Multi-type itemset embedding preserves the structures of sets

(c) Task 1: Predict the probability of an itemset being observed

(d) Task 2: Recommend complementary items to maximize the probability.

Fig. 1. The multi-type itemset embedding method learns item representations collectively from the set structure. When it is applied to paper-publishing behavior data, the embeddings preserve the composition of a paper being successfully published instead of pairwise similarity between a paper and any item of it.

and social dimensions. Given the complexity of these components in human behaviors, it is difficult to develop a successful plan and make right decisions. For example, who should be invited to a project team to make it more successful (in terms of solving real problems and publishing at top venues)? What knowledge and skills should the team have? With the increasingly available behavior databases in fields such as social media, education, and academic research, we now have an invaluable source of information to discover behavioral patterns and support decision making. Again, let us consider the multi-dimensional paper-publishing behavior as an example: it has authors, target conference/journal, datasets, problems, methods, references, and so on. Tensor methods decompose multi-dimensional counts (e.g., numbers of co-occurrences among one author, one conference, and one reference) into low-dimensional latent vectors [23]. However, real behaviors do not guarantee single context item per dimension [19]: a paper can have multiple authors and references. Multi-contextual factor models learn latent factors of users and items through an expectation-maximization process [16], but neither of these methods can scale for massive behavior data.

Recently, embedding methods have been proposed to learn low-dimensional features of nodes on large-scale networks. Certainly, we can represent behaviors and contexts as nodes and obtain a "behavior-to-context" bipartite graph (e.g., the "paper-author/venue" graph in Figure 1(a)). The existing embedding methods preserve *pairwise similarity* between a behavior and any of its context, such as connections, common neighbors, and random walk based measures [6, 10, 28, 35, 39]. However, as having been revealed in behavioral sciences, decision makers are looking for *not similar but complementary* partners, resources, and conditions that provide extra power to make a behavior more successful. For example, partners need complementary strengths to do successful business [8, 14]; courses need complementary teaching materials to achieve effective student learning [13, 33]. Therefore, we argue that, in order to support effective decision making, the behavior embeddings should preserve the complementarity rather than similarity.

In this work, we propose a measurement of complementarity between two context items conditionally based on other contexts in the behavior: it is defined as the extent that two contexts together being in the behavior increase the *success rate* of the behavior over only one of them being included. Now we need to define the success rate of a behavior. We represent a behavior as a multi-type itemset, i.e., a set of context items of multiple types, instead of a multi-dimensional count or an explicit node. For example, as shown in Figure 1(b), the paper is represented as an itemset $b = \{a_1, a_2, c_1, k_1, r_1\}$ where a_1 and a_2 are authors, c_1 is a conference, k_1 is a keyword, and r_1 is a reference. The *success rate* of this behavior $r(\cdot)$ is defined as the probability of it being

occurred/observed. For example, $r(b) = 1$ (100%) if the paper is published; otherwise, $r(b) \in [0, 1)$. This definition can be generalized as the outcome a behavior creates, for example, if a tweet-posting behavior is represented as an itemset of user, hashtags, words, and emoticons, and its success rate can be evaluated as its popularity level, i.e., the number of times it is retweeted.

Here are the novel and challenging tasks in the field of user behavior modeling we want to solve.

Task1 (Contextual behavior prediction). Given a behavior b and its set of context items, predict the behavior b's success rate $r(b)$.

For example, given a set of authors, keywords, and a conference, predict the probability of a paper of these items being published in the conference's proceedings (see Figure 1(c)).

Task2 (Behavioral context recommendation). Given a behavior b and its set of context items, recommend complementary context items that will maximize the success rate of this behavior.

For example, given a student author, a top-tier target conference, some keywords (e.g., "data mining"), recommend other authors, proper keywords, and references that will maximize the chance of the student's paper being accepted by the conference (see Figure 1(d)).

Note that these two tasks can be found in many applications such as target advertising, personalized consulting, and military mission planning. Generally, we are looking for complementary operators/resources to maximize the chance of achieving the goal/success when being incorporated into existing operators, resources, and other contexts.

We propose a novel method that efficiently learns representations of behavior's context items of multiple types by preserving the itemset structure, or say, the complementarity betweem items. This is rather challenging. First, similarity between representations of any pair of items in the itemset is no longer capable of measuring the success rate of the itemset. Secondly, the itemset structure has heterogeneity: different context types contribute to the success at different extents. For example, it is definitely more impactful for a paper to involve one more internationally known expert than one more reference item. Thirdly, in most cases, we can only observe positive behaviors. For example, it is easy to collect tons of published papers but hard to find rejected or non-finished ones. The embedding method has to be careful to generate negative instances for embedding learning.

To address the above challenges, our method, called "*Multi-Type Itemset Embedding*," measures the success rate of an itemset in the vector space by the hyperbolic tangent of the sum of vectors of items in the itemset. First, it learns items' feature vectors collectively on both similarity (angle) between any pair of vectors and norm of each vector itself by optimizing the co-occurrence of all the items in the set. Secondly, our method assigns type weights to the context items when representing the itemset as a weighted sum of vectors, which has been demonstrated to be useful in experiments. Thirdly, we introduce two kinds of negative behavior sampling strategies, size-constrained and type distribution–constrained, to generate negative samples when they are unavailable. We propose a novel framework, called "***Complementarity in ItemSet Embedding***" (**CISE**), that first learns items' low-dimensional representations using the itemset embedding, then measures the complementarity between any pair of context items and feeds the itemset representations into a logistic/linear regression model to predict success rate or recommend items for maximizing the success rate.

Here, we summarize our contributions in this article.

- We propose to study two novel tasks in user behavior modeling: (1) contextual behavior prediction and (2) behavioral context recommendation.
- We propose *Multi-Type Itemset Embedding* method to collectively learn items' representations preserving the itemset structure. It includes a novel measurement of success rate for learning the itemset structure, considers item type's weight for heterogeneity, and conducts negative behavior sampling for representation learning if necessary.

— Based on the embeddings, we propose a measure of complementarity between contexts in a behavior. We theoretically and empirically prove it is significantly different from similarity.
— We develop the CISE framework to effectively solve these two tasks. Empirical results show new discovery of complementarity in human behaviors, which provides insights for behavioral scientists to understand behavioral mechanisms from big data.

The rest of this article is organized as follows: Section 2 reviews related work and Section 3 defines concepts and research problems. Our CISE framework is presented in Section 4. Section 5 shows experimental results and Section 6 concludes the paper.

2 RELATED WORK

In this section, we review existing methods in three relevant fields to our work, including contextual behavior modeling, network embedding, and text embedding.

2.1 Contextual Behavior Modeling

There has been a wide line of research on learning latent representations of context items [18, 41, 46, 49, 50] in behavior data toward various applications [7, 20, 32, 42, 43, 48]. Agarwal et al. proposed localized factor models combining multi-context information to improve predictive accuracy in recommender systems [1]. Jamali et al. proposed context-dependent factor models to learn latent factors of users and items for recommendation [16]. Besides factor models, tensor decompositions have been widely used for modeling multi-contextual data [31]. Jiang et al. proposed a tensor-sequence decomposition approach for discovering multi-faceted behavioral patterns [17]. Ermiş et al. studied various alternative tensor models for link prediction in heterogeneous data [9]. Lian et al. proposed regularized tensor factorization for spatiotemporal recommendation [23]. Yang et al. developed a predictive task guided tensor decomposition model for representation learning from Electronic Health Records [45]. Perros et al. designed a scalable PARAFAC2 tensor model for large and sparse datasets [29]. However, as pointed out in [19], tensor requires one value for each context dimension, which cannot support full representation for multicontextual behavior entries that may have non-value or multi-values in a dimension. Also, the computational cost of factorizing a large matrix or tensor is highly expensive. None of the existing behavior modeling methods can efficiently learn item representations to optimize success rate on behavior data.

2.2 Network Embedding

Network representation learning methods learn node representations that preserve node proximities (e.g., one-hop or two-hop connections) in network data [3, 5, 11, 24, 44, 47]. LINE [35] provided clear objectives for homogeneous network embedding that articulates what network properties are preserved. DeepWalk [28] used random walks to expand the neighborhood of a node and expected nodes with higher proximity yield similar representations. node2vec [10] presented biased random walkers to diversify the neighborhood. We have spotted a series of heterogeneous network embedding work [6, 12, 40] that capture heterogeneous structural properties in network data. Chen and Sun proposed a model that enables two features, task-guided and path-augmented, in the network embedding method to identify author names given an anonymized paper's title [4]. If we explicitly represent behavior entries as nodes and thus behavior datasets are represented as behavior-item heterogeneous bipartite networks, existing network embedding methods can be applied to learn the representations of both items and behaviors. However, these methods preserve node proximities so they can only find similar items for a behavior. Our proposed method learns the composition of context items in a behavior, i.e., the itemset structure, and preserves a behavior's success rate. It looks for complementary items that will maximize the success of a target behavior. We compare our method against the state-of-the-art network embedding algorithms.

Table 1. Symbols in This Article and their Descriptions

Symbol	Description
c, C	Context item, and the set of all items
$t, \mathcal{T}$	Context (item's) type, and the set of all types
$b, \mathcal{B}$	Behavior, and the set of all "positive" behaviors
$r(b), \hat{r}(b)$	Estimated and observed success rate of behavior b
$t(c)$	Context type of context item c
d	Number of dimensions in a low-dimensional space
$\vec{c}$	Low-dimensional vector of context item c
$\vec{b}$	Low-dimensional vector of behavior b
w_t	Learning weight of context type t

2.3 Text Embedding

With the success of deep learning techniques, representation learning becomes popular starting from practices on text data. Mikolov et al. proposed the word2vec framework to learn the distributed representations of words in natural language [26]. Pennington et al. proposed GloVe to learn word vectors from nonzero elements in a word-word co-occurrence matrix [27]. Le and Mikolov extended the embedded objects from words or phrases to paragraphs [22]. Our work focuses on representation learning from behavior data that is represented as multi-type itemsets.

3 PROBLEM DEFINITION

In this section, we define the concepts used throughout this article, and then formally define the research problems we study. Table 1 presents symbols we use in this article and their descriptions.

Definition 3.1 (Context type). A behavior includes multiple types of contexts such as individuals who make the action, behavioral goals, and resources. A **context type** is the type of a context. For example, a paper-publishing behavior has authors (as operators), a conference or journal (as behavioral goal), keywords, technical terms, datasets, and references (as resources). A tweet-posting behavior has a social network user (as operators), a specific topic, a geolocation tag, words, hashtags, urls, and emoticons (as resources). Any type of these elements can be a context type such as "author," "conference," "reference," "word." and "hashtag."

Definition 3.2 (Context item). A **context item** is a concrete item of a context type, such as an author's name, a conference's name, or a concrete publication as a reference in a paper-publishing behavior. A context item c must have a context type $t(c)$.

Definition 3.3 (Behavior and multi-type itemset). A **behavior** is a relationship among multiple types of context items. It can be represented as a set of the context items. A behavior is equivalent to a **multi-type itemset**.

For example, given a conference paper, the paper-publishing behavior is a set of author(s), conference, keyword(s), reference(s), and some other relevant items to this paper. The quantity and types of items in an itemset may vary from behavior to behavior. A paper-publishing behavior contains at least one author and keyword, exactly one conference, and multiple references. A tweet-posting behavior contains exactly one user and at least one word. To avoid ambiguity with other fields, we always refer to "context" strictly as defined in Definition 3.2 throughout this article.

Definition 3.4 (Success rate). The **success rate** of a behavior is the value (as a numeric label) denoting the real-world *success* of that behavior given its particular set of context items. For each set of context items constituting a behavior b, we use $\hat{r}(b)$ to indicate its success. where $\hat{r}(b) \geq 0$.

For a specific kind of behavior, in order to define the success rate, we should first define what is **success**. For example, for each paper in a publication dataset, we say the set of context items in this article makes a success of a paper-publishing behavior, and thus the success rate of this behavior is a positive number—we use 1 as default and we call the behavior a **"positive" behavior**. This presents a challenge, as most datasets do not include entries on unpublished works. We will overcome this obstacle in the model by adopting the strategy of negative sampling. Essentially, we assume that most of the *non-existing multi-type itemsets* indicate unsuccessful behaviors. We denote them as **"negative" behaviors** and set their success rate as 0.

Given different measurements of success, the success rate could be different. For a tweet-posting behavior, if the success is measured as the existence of its context itemsets in tweet data, we have tons of positive behaviors but no real negative ones; we have to generate non-existing itemsets as negative behaviors. If the success is measured as the behavior's popularity level, the success rate can be viewed as the number of views, likes, retweets, or shares [15]. In this case, positive behaviors are popular posts and negative behaviors are unpopular but still real posts. No non-existing itemset needs to be generated though the success rate may need to be normalized to reduce variance.

In summary, the success rate associated with each behavior may be explicit (e.g., a rating or score that the behavior received), or it may be implicit in the behavior data (e.g., the number of occurrences of the behavior). Now, we define what is behavior data.

Definition 3.5 (Behavior data). **Behavior data** is defined as $D = (\mathcal{C}, \mathcal{T}, \mathcal{B})$, where $\mathcal{C}$ is the set of unique context items, each having a particular type in $\mathcal{T}$, and $\mathcal{B}$ is the set of behaviors, each representing a relationship among one or more context items. Each behavior $b \in \mathcal{B}$ is a set of context items $b \subset \mathcal{C}$, which is associated with a nonnegative, observed success rate $\hat{r}(b)$.

If both $|\mathcal{C}|$ and $|\mathcal{B}|$ are big, we call the dataset *massive behavior data*. Based on the above concepts, we formally define the research problem as below.

Problem 1 (Massive behavior data embedding) Given a massive behavior dataset $D = (\mathcal{C}, \mathcal{T}, \mathcal{B})$, the problem of **Massive Behavior Data Embedding** aims to represent each context item $c \in \mathcal{C}$ as a low-dimensional vector $\vec{c} \in \mathbb{R}^d$, i.e., learning a function $f_D : \mathcal{C} \rightarrow \mathbb{R}^d$, where $d \ll |\mathcal{C}|$. In the space $\mathbb{R}^d$, the contributions of each context item toward a behavior's success are preserved.

As in Definition 3.3, a behavior can be generalized to a multi-type itemset of the same structure. Therefore, the problem of behavior data embedding is equivalent to multi-type itemset embedding.

Problem 2 (Multi-type itemset embedding) Given a large set of items $\mathcal{C}$, their types $\mathcal{T}$ and a large set of multi-type itemsets $\mathcal{B}$, the problem of **Multi-type Itemset Embedding** aims to represent each item $c \in \mathcal{C}$ as a low-dimensional vector $\vec{c} \in \mathbb{R}^d$, i.e., learning a function $f_D : \mathcal{C} \rightarrow \mathbb{R}^d$, where $d \ll |\mathcal{C}|$. In the space $\mathbb{R}^d$, each item's contribution toward an itemset's composition is preserved.

Much like heterogeneous information network embedding, itemset embedding aims to represent context items of various types as vectors in a low-dimensional space. Unlike network embedding, which preserves pairwise proximity between nodes, itemset embedding preserves the itemset structures. The vectors of items within an itemset may not be close to each other in $\mathbb{R}^d$, but the vectors should sum to a vector with a magnitude representative of the behavior's success.

Unlike network embedding models such as LINE [35] , DeepWalk [28], and node2vec [10] that preserve *proximities* and were evaluated on clustering tasks, our behavior data embedding model,

also a multi-type itemset embedding model, preserves the property of *success*. We will evaluate it on the two tasks of behavior modeling we have introduced in Section 1 and compete with existing works in experiments. These two tasks are challenging and important in this era of witnessing how artificial intelligence and data science significantly change our decision-making process.

4 THE CISE FRAMEWORK

Our CISE framework has two modules for learning behavior success. The first module is a multi-type itemset embedding model that learns item representations from behavior data, in which we propose a novel metric of measuring the success rate of a behavior. The second module is to feed the itemset representations into a logistic/linear regression model to predict the probability of a future behavior's success. It can also recommend complementary items to maximize the probability.

4.1 Multi-Type Itemset Embedding

A desirable embedding model for behavior data must satisfy several requirements: first, it must be able to preserve the success property of multi-type itemsets; secondly, it must scale for massive behavior data, say millions of context items and behaviors; thirdly, it must deal with behaviors with arbitrary types and quantities of context items. In this section, we present a novel multi-type itemset embedding model that satisfies all three of these requirements.

4.1.1 Model Description. We explain the embedding model to preserve success properties of behaviors. The success of a behavior refers to the success achieved given a particular set of context items. For each behavior b as a particular set of multi-type context items, we define b's low-dimensional vector representation as follows:

$$\vec{b} = \sum_{c \in b} w_{t(c)} \cdot \vec{c} \in \mathbb{R}^d, \tag{1}$$

where $\vec{c} \in \mathbb{R}^d$ is the low-dimensional vector representation (e.g., one-hot embedding) of context item c, and $w_{t(c)}$ is the type weight of c's context type. Different item types may have different levels of contributions to the behavior's success. For example, one author or keyword often contributes more to a paper's acceptance than one reference item. Considering appropriate type weights for different context items in a behavior is essential for effectively capturing its success property. Therefore, we assign type weights to the context types when we take the sum of the items' low-dimensional vectors to represent the behavior. We will discuss type-weight parameter settings in the experiments section. For a behavior b, we define b's estimated success rate as follows:

$$1r(b) = \tanh \frac{\|\vec{b}\|_2}{2} = 2 \cdot \frac{1}{1 + e^{-\|\vec{b}\|_2}} - 1, \tag{2}$$

where $\|\vec{b}\|_2 \in [0, \infty)$ is the Euclidean norm of $\vec{b}$ in the d-dimensional space. The hyperbolic tangent function $\tanh(x)$ is a rescaled version of the *logistic sigmoid* function $g(x) = \frac{1}{1+e^{-x}}$ and $\tanh(x) = 2\,g(2x) - 1$. The output range of $\tanh(x)$ is $[0, 1)$ instead of $[\frac{1}{2}, 1)$ because the norm $\|\vec{b}\|_2$ is nonnegative. Equation (2) defines a distribution $r(\cdot)$ over the entire behavior space $\mathcal{B}$, and its empirical success rate can be observed as $\hat{r}(b)$. To preserve the success properties, a straightforward way is to minimize the following objective function:

$$O = d(\hat{r}(\cdot), r(\cdot)), \tag{3}$$

where $d(\cdot, \cdot)$ is the distance between two distributions. We choose to minimize the KL-divergence of two probability distributions of the itemset's success rate. Replacing $d(\cdot, \cdot)$ with KL-divergence,

we have

$$O = -\sum_{b \in \mathcal{B}} \hat{r}(b) \log r(b). \tag{4}$$

Remark. Traditional embedding models optimize the similarity/proximity between vectors of a pair of items, i.e., $\vec{c_i} \cdot \vec{c_j}$. Our itemset embedding model optimizes the success rate based on the norm of an itemset's vector. If we expand the norm as below,

$$\|\vec{b}\|_2 = \left(\sum_{c \in b} {w_{t(c)}}^2 \cdot {\|\vec{c}\|_2}^2 + \sum_{\substack{c_i, c_j \in b \\ c_i \neq c_j}} w_{t(c_i)} w_{t(c_j)} \cdot \vec{c_i} \cdot \vec{cj} \right)^{\frac{1}{2}}, \tag{5}$$

it is worthwhile of showing that our method learns the item vectors collectively of an itemset, and it optimizes not only the vector similarity between every pair of items in the set but also the length of norm of each item's vector $\|\vec{c}\|_2$.

4.1.2 Model Optimization. As explained in Definition 3.4, our proposed model should be trained on behavior dataset $\mathcal{B}$ differently according to different measurement of success. For example, in order to optimize the objective function in Equation (4), if the success of a tweet-posting behavior is measured as the behavior's popularity level, e.g., the retweet count, $\mathcal{B}$ has both positive (popular tweets w/high retweet counts) and negative behaviors (unpopular tweets/w low retweet counts). If a behavior's success is a binary measurement of its existence given its complete set of context items, $\mathcal{B}$ has only positive behaviors, and we have to generate non-existing itemsets as negative behaviors. We introduce our optimization approaches to deal with these two scenarios as follows.

When $\mathcal{B}$ has both positive and negative behaviors. Suppose the distribution of observed success rate $\hat{r}(.)$ (e.g., the number of times a tweet being retweeted) follows the Power Law, which is often applicable in the real world. We have a reasonable number of negative behaviors in $\mathcal{B}$, and the objective function for each behavior b can be specified as follows:

$$\hat{r}(b) \log \tanh \frac{\|\vec{b}\|_2}{2} = \hat{r}(b) \log \tanh \frac{\| \sum_{c \in b} w_{t(c)} \cdot \vec{c} \|_2}{2}. \tag{6}$$

We adopt the **asynchronous stochastic gradient algorithm** (**ASGD**) [30] for optimizing Equation (6). In each step, the ASGD algorithm samples one observed behavior (can be either positive or negative) and updates the model parameters. If a behavior b is sampled, the gradient w.r.t. the embedding vector $\vec{c}$ of a context item c in b will be calculated as

$$\frac{\partial O}{\partial \vec{c}} = \frac{\hat{r}(b)}{\sinh \|\vec{b}\|_2} \cdot \frac{\partial \|\vec{b}\|_2}{\partial \vec{c}} = \frac{\hat{r}(b)}{\sinh \|\vec{b}\|_2} \cdot \frac{\partial \|\vec{b}\|_2}{\partial \vec{b}} \cdot \frac{\partial \vec{b}}{\partial \vec{c}} = \frac{w_{t(c)} \hat{r}(b)}{\|\vec{b}\|_2 \sinh \|\vec{b}\|_2} \cdot \vec{b}. \tag{7}$$

When $\mathcal{B}$ has only positive behaviors. In this case, we need to sample *non-existing multi-type itemsets* and use them as "negative behaviors." We approximate the optimization by sampling a mini-batch of behaviors that includes one positive sample and several negative samples. Inspired by the negative sampling technique proposed in [26], we propose the technique of negative behavior sampling on behavior data (see Figure 2).

Negative behavior sampling. Given the set of context items for each context type, how do we generate a non-existing multi-type itemset b' when we sample a positive itemset b from $\mathcal{B}$? We propose two different sampling strategies. The first strategy is to apply a simple size constraint on the entire itemset (say, the size of itemset b' is the same as the size of itemset b) without constraining the number of items for any type. We generate $n(t)$, which is the number of items of

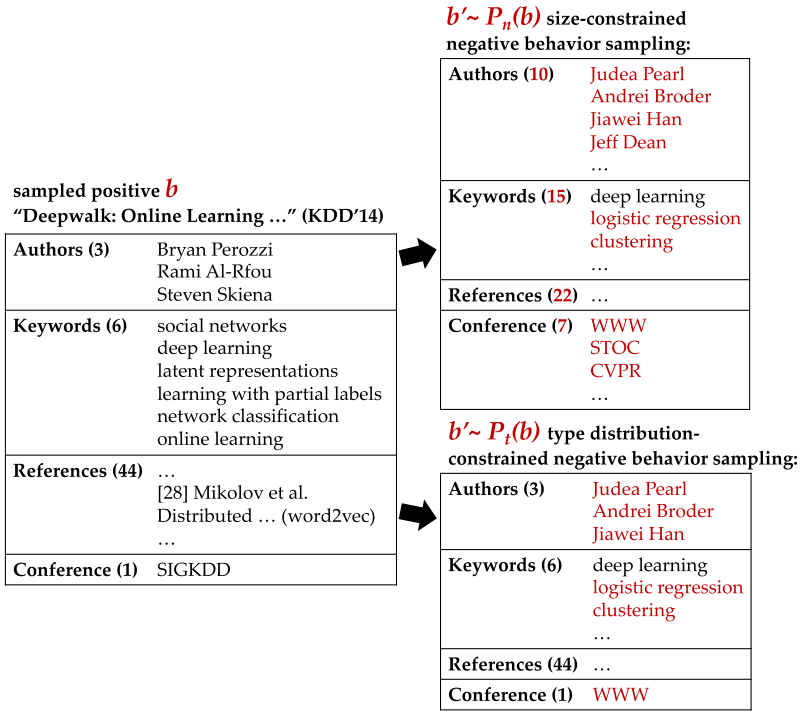


Fig. 2. Negative behavior sampling: Given a sampled "positive" behavior (e.g., an observable paper), we propose two strategies to generate negative samples: size-constrained and type distribution–constrained. Differences between two negative sampling strategies are highlighted in red.

type $t \in \mathcal{T}$, so that the sum of $n(\cdot)$ is $|b|$, and then for type t, we randomly select $n(t)$ items from C_t, which is the set of items of type t in C, and put them into b'. We denote the *size-constrained negative behavior sampling* as $b' \sim P_n(b)$ and have the following objective:

$$\hat{r}(b) \log \tanh \frac{\|\vec{b}\|_2}{2} + \sum_{k=1}^{K} \mathbb{E}_{b' \sim P_n(b)} \log \tanh \frac{\|\vec{b'}\|_2^{-1}}{2}, \tag{8}$$

whose gradient is derived as follows:

$$\frac{\partial O}{\partial \vec{c}} = w_{t(c)} \cdot \left(I_{c \in b} \cdot \frac{\hat{r}(b)}{\|\vec{b}\|_2 \sinh \|\vec{b}\|_2} \cdot \vec{b} - I_{c \in b'} \cdot \frac{1}{\|\vec{b'}\|_2^{\,3} \sinh \frac{1}{\|\vec{b'}\|_2}} \cdot \vec{b'} \right), \tag{9}$$

where I_x is an indicator function that returns 1 if x is true and 0 if x is false. Note that here $\hat{r}(b)$ is above zero. Specifically, we set $\hat{r}(b) = 1$ for paper-publishing behaviors. K is the number of negative behavior samples. We will later investigate the sensitivity of K in the experiments section.

The second strategy considers the context type distribution in a sampled positive itemset b. For each type t, we randomly select $n(t)$ context items from C_t, where $n(t)$ is the number of context items of the type t in the behavior b. We denote the *type-distribution-constrained negative behavior sampling* as $b' \sim P_t(b)$, and we are able to simply replace P_n with P_t in the objective of Equation (8) as well as in the derivative of Equation (9). We denote our method that uses size-constrained sampling as **CISE-P$_\mathbf{n}$**, denote our method that uses type-distribution-constrained sampling as **CISE-P$_\mathbf{t}$**, and compete against the state-of-the-art embedding methods in the experiments.

4.1.3 Complexity Analysis. Sampling an itemset from behavior data takes constant time, $O(1)$; and, optimization with negative sampling takes $O(d(K+1))$ time, where d is the number of dimensions and K is the number of negative samples. The number of steps is proportional to the number of behaviors/itemsets $O(|\mathcal{B}|)$. Therefore, the theoretic overall time complexity bound of the itemset embedding model is $O(dK|\mathcal{B}|)$, which is linear to the number of context items $|C|$.

4.2 Prediction and Recommendation Models

Once the representations of context items have been learned by the multi-type itemset embedding model preserving the behavior success property, we can use those low-dimensional feature vectors to solve the two behavior modeling tasks we have introduced in Section 1.

4.2.1 Contextual Behavior Prediction. Given a behavior/itemset b and its set of context items, we train a logistic/linear regression model with the itemset's representation $\vec{b}$ and label $\hat{r}(b)$ and apply the model to predict the success probability of test instances.

4.2.2 Behavioral Context Recommendation. The above two models can both be applied for recommending complementary items for a potential behavior/itemset. The goal is to maximize the predicted probability of being successful/observed. It is time-consuming (exponential time) to enumerate through all itemset candidates and compute their probability scores. So, in the experiments, we will hide only one item for each testing itemset. We leave the task of recommending the best combination of multiple items as future work.

4.3 The Measurement of Context Complementarity

After we have the latent representations of context items preserving the success property, we investigate the complementarity between two context items given other contexts in the behavior.

Definition 4.1 (Context complementarity). The conditional **context complementarity** between two context items based on a given behavior is a measure of the surplus marginal value of success rate brought to this behavior only if both of them appear in the particular behavior.

The "complementarity" could generally refer to the capability of bringing in additional benefits by incorporating the target context items. In this work, we focus on context complementarity defined as the extent that these two context items together being in the behavior increase the behavior's success rate over either one of them being included. It is carefully disigned to capture the synergistic effect created by two context items inside a behavior in terms of the success rate.

Given two context items c_i, c_j, and a behavior itemset b, assuming $c_i, c_j \in b$ and $|b| \geq 3$, the **conditional context complementarity** between c_i and c_j conditioned on b is defined as

$$cpl(c_i, c_j \mid b) = r(b) - \max\{r(b \setminus \{c_i\}), r(b \setminus \{c_j\})\}, \tag{10}$$

where $b \setminus \{c_i\}$, or $b \setminus \{c_j\}$, means context item c_i, or c_j, is excluded from the itemset b, respectively. A trivial case of $b = \{c_i, c_j\}$ can be derived as $cpl(c_i, c_j \mid b) = r(b) - \max\{r(c_i), r(c_j)\}$. Note that a composite relation $cpl(c_i|b)$ where $b = \{c_i, c_{j_1}, \ldots, c_{j_N}\}$ can be derived as $cpl(c_i|b) = \sum_{n=1}^{N} cpl(c_i, c_{j_n}|b)$. By substituting the behavior's estimated success rate given by Equation (2) inside, we can expand it into

$$cpl(c_i, c_j \mid b) = \tanh \frac{\|\vec{b}\|_2}{2} - \max\{\tanh \frac{\|\vec{b} - \vec{c}_i\|_2}{2}, \tanh \frac{\|\vec{b} - \vec{c}_j\|_2}{2}\}. \tag{11}$$

Then, the **context complementarity** between c_i, c_j can be taken as the average over the entire behavior dataset:

$$cpl(c_i, c_j) = \frac{1}{|\mathcal{B}^{\{c_i, c_j\}}|} \sum_{b \in \mathcal{B}^{\{c_i, c_j\}}} cpl(c_i, c_j \mid b), \tag{12}$$

where $\mathcal{B}^{\{c_i,c_j\}} = \{b \in \mathcal{B} \mid c_i, c_j \in b\}$ is the subset of behaviors in $\mathcal{B}$ containing both c_i and c_j. We focus on evaluating complementarity between context items of the same type in this work.

4.3.1 Properties of the Complementarity Measure.

Property 1 (Range): $cpl(c_i, c_j) \in (-1, 1)$.
Proof. For $\forall b$, we have $r(b) \in [0, 1)$, $r(b \setminus \{c_i\}) \in [0, 1)$ and $r(b \setminus \{c_j\}) \in [0, 1)$.
Thus, $-\max\{r(b \setminus \{c_i\}), r(b \setminus \{c_j\})\} \in (-1, 0]$. From Equation (10), we know $cpl(c_i, c_j) \in (-1, 1)$.

Property 2 (Zero leads to non-positive): $cpl(c_i, c_0) \leq 0$, $\vec{c}_0 = \vec{0}$.
Proof. For $\forall b$, from Equation (11), we have

$$\begin{aligned} cpl(c_i, c_0 \mid b) &= \tanh \frac{\|\vec{b}\|_2}{2} - \max\{\tanh \tfrac{\|\vec{b}-\vec{c}_i\|_2}{2}, \tanh \tfrac{\|\vec{b}\|_2}{2}\} \\ &= \min\{\tanh \tfrac{\|\vec{b}\|_2}{2} - \tanh \tfrac{\|\vec{b}-\vec{c}_i\|_2}{2}, 0\} \leq 0. \end{aligned} \tag{13}$$

So, we know $cpl(c_i, c_0) \leq 0$.

An explanation to this property is that if a context item has no "skills" (for all zero features), it may have a negative effect in terms of the complementarity with existing contexts in the behavior.

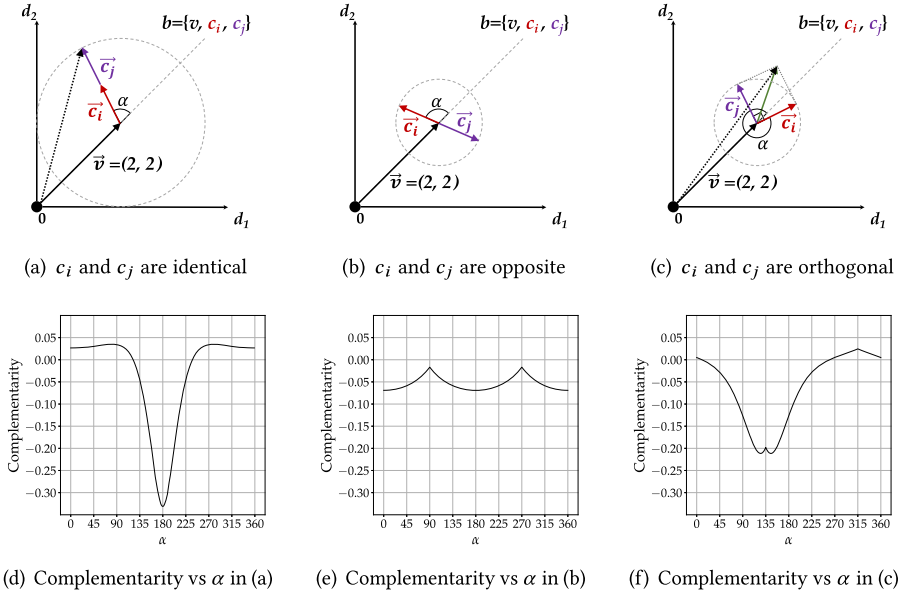
Property 3 (Symmetry): $cpl(c_i, c_j) = cpl(c_j, c_i)$.
Proof. For $\forall b$, based on Equation (10), it is evident that we have $cpl(c_i, c_j \mid b) = cpl(c_j, c_i \mid b)$ because $\max\{r(b \setminus \{c_i\}), r(b \setminus \{c_j\})\} = \max\{r(b \setminus \{c_j\}), r(b \setminus \{c_i\})\}$. Therefore, from Eq. (12), we know $cpl(c_i, c_j) = cpl(c_j, c_i)$.
Note this property may not hold between $cpl(c_i \mid b \setminus \{c_i\})$ and $cpl(c_j \mid b \setminus \{c_j\})$ when $c_i, c_j \in b$.

4.3.2 Special Cases of the Complementarity Measure. We study three cases to compare complementarity with similarity, dissimilarity, and orthogonality. We point out the uniqueness of the complementarity and its meaningfulness when being applied to behavior modeling. For all three cases, suppose the behavior has three items $b = \{v, c_i, c_j\}$ and each item is represented by two features. Suppose $\vec{v} = (2, 2)$, and $\vec{c}_i, \vec{c}_j$ are unit vectors ($\|\vec{c}_i\| = \|\vec{c}_j\| = 1$).

Special case 1: complementarity vs. similarity (if $\vec{c}_i$ and $\vec{c}_j$ are identical). In this case, the (cosine) similarity between c_i and c_j is a constant of 1. We vary $\vec{c}_i$ by changing $\alpha \in [0°, 360°)$, which denotes the counterclockwise angle between $\vec{c}_i$ and $\vec{v}$ (see Figure 3(a)). For every α value, we calculate the conditional complementarity $cpl(c_i, c_j \mid b)$ and plot the curve in Figure 3(d). We observe that the complementarity remains positive when $\alpha \in [0°, 120°)$ (due to the scale of $\vec{v}$), and becomes negative when $\vec{c}_i$ goes against $\vec{b}$. This tells us that the complementarity is not consistently fixing at 1 but is conditioned on the other item in the behavior.

Special case 2: complementarity vs. dissimilarity (if $\vec{c}_i$ and $\vec{c}_j$ are opposite). Here, we set $\vec{c}_i = -\vec{c}_j$, so the (cosine) similarity between c_i and c_j is a constant of -1. In Figure 3(e), we observe that $cpl(c_i, c_j \mid b)$ is at the peak when α is 90° or 270°, and hits the bottom when α is 0° or 180°. We can also see that the complementarity remains always below 0. This matches our intuition that we do not want to include two completely dissimilar items as they are likely to create conflicts.

Special case 3: complementarity vs. orthogonality (if $\vec{c}_i$ and $\vec{c}_j$ are orthogonal). Here, the (cosine) similarity between c_i and c_j is 0. The complementarity $cpl(c_i, c_j \mid b)$ is at the peak when $\alpha = 315°$. That is, when $\vec{c}_i + \vec{c}_j$ lies exactly like the orientation as $\vec{b}$ does. In other words, when c_i and c_j are irrelevant, $cpl(c_i, c_j \mid b)$ is maximized if the overall contribution brought by c_i and c_j aligns with b.



(a) c_i and c_j are identical (b) c_i and c_j are opposite (c) c_i and c_j are orthogonal

(d) Complementarity vs α in (a) (e) Complementarity vs α in (b) (f) Complementarity vs α in (c)

Fig. 3. Comparing complementarity with similarity, dissimilarity, and orthogonality. Top figures visualize when c_i and c_j are identical, opposite, and, orthogonal. α denotes the counterclockwise angle between $\vec{v}$ and $\vec{c_i}$, $\alpha \in [0°, 360°)$. Bottom figures plot $cpl(c_i, c_j|b)$ against α.

5 EXPERIMENTS

In this section, we first introduce the two real behavior datasets we use. For each behavior modeling task, we present validation settings, quantitative/qualitative analysis. Then, we provide case studies to highlight the key differences between context complementarity and similarity. Lastly, we analyze the sensitivity and the efficiency of the proposed method.

5.1 Datasets Description

We use two real behavior datasets to demonstrate the effectiveness of our proposed methods:

(1) We collected papers from the Microsoft Academic project to study the behavior of publishing a paper at a conference in the field of computer science. The context items of each paper includes one conference, at least one author, at least one keyword and at least one reference. We filtered out papers in which none of the authors has at least two publications in our dataset. This gave us 692,725 papers. Overall, we have 832,969 context items: 195,152 authors, 1,265 conferences, 39,756 keywords, and 596,796 references. On average, each paper has 2.31 authors, 1 conference, 6.61 keywords, and 8.79 references (18.71 context items in total).

(2) We collected tweets from the publicly available FIFA World Cup 2018 Tweets dataset to study the behavior of posting feeds on social media. After removing stop words and filtering out tweets without a valid geolocation, we had 317,549 tweets. Each tweet includes one location, at least one word, at least one hashtag, and is associated with a nonnegative retweet count. Overall, we have 114,718 context items: 25,000 words, 11,071 hashtags, and 78,647 locations. On average, each tweet has 6.95 words, 1.98 hashtags, and 1 location (9.93 items in total).

5.2 Baselines and Parameter Settings

We compare our method against the following baselines:

(1) METAPATH2VEC [6]: The state-of-the-art method of node representation learning for heterogeneous networks. It samples heterogeneous networks based on meta-path-based random walks and uses a heterogeneous Skip-gram model to perform node embeddings. We use the advanced version METAPATH2VEC++ here, which also conducts heterogeneous negative sampling to fully leverage the heterogeneity information in network.
(2) VERSE [39]: This versatile homogeneous network embedding method is able to preserve the distributions of a selected vertex-to-vertex similarity measure in a network. We use the recommended **Personalized PageRank** (**PPR**) for the similarity measure, which has been shown to produce good performance in nearly all tasks and networks in the paper.
(3) NODE2VEC [10]: This method can learn continuous feature representations for nodes in homogeneous networks by maximizing the likelihood of preserving node neighborhoods in low-dimensional feature space. The homophily and structural equivalence properties of the network are captured by conducting biased random walks on the network with interleaving breadth-first and depth-first sampling strategies.
(4) DEEPWALK [28]: It uses local information obtained from truncated uniform random walks to learn latent representations of vertices in a network. This is one of the first studies to treat walks as the equivalence of sentences and to map node neighborhood into word context in order to leverage the Skip-gram model in language modeling.
(5) LINE [35]: This homogeneous network embedding method preserves both the local and global network structures by optimizing a carefully designed objective function. Different from METAPATH2VEC [6], NODE2VEC [10], and DEEPWALK [28], it conducts edge sampling instead of random walks on the network. We use the advanced version LINE(1ST+2ST) in experiments, which concatenates node vector representations for both first- and second-order proximity.

In addition, we also investigated a few popular graph embedding methods such as spectral clustering [36] and graph factorization [2]; however, they have been shown to be significantly outperformed by LINE [35] and DEEPWALK [28] in both previous studies and in our study. Therefore, we exclude the experimental results of them due to limited space. Similarly, we report the performance of METAPATH2VEC [6] as a representative model for heterogeneous network embedding because of its proved superior performance compared with other models [12]. We also tried classical dimensionality reduction techniques like PCA [38], MDS [21], and ISOMAP [37]. Unfortunately, their high complexities do not allow them to handle our behavior datasets of such a large scale.

In the first dataset of academic papers, we have only published papers as positive behaviors. We evaluate two variants of CISE, i.e., CISE-P_n and CISE-P_t, to compare the effectiveness of different negative sampling strategies. The weights of context types are {3 (author), 1 (conference), 1 (keyword), 1 (reference)} as default. In the second dataset of tweets, we treat retweet count as the tweet-posting behavior's success rate. So, we have popular tweets of high retweet count as positive behaviors, as well as a reasonable number of tweets with low retweet count as negative behaviors. We evaluate CISE (with no negative sampling) by setting weights of context types {3 (word), 1 (hashtag), 1 (location)} as default. We also systematically examine the weights of context types using a grid search strategy. We enumerate through all combinations of type weight values in $\{1, 2, 3\}$ and report the best performance with respect to each particular type.

For fair comparison, we set the total sampling budget s to be the same for all methods. Specifically, for random walk based methods, i.e., NODE2VEC [10], DEEPWALK [28], and METAPATH2VEC [6], the total sampling budget can be computed as $s = r \cdot l \cdot |V|$ where r is repetition of walks per node, l is random walks length, and V is the set of vertices (context items and behaviors); in VERSE [39], s equals to the total number of nodes sampled from the positive distribution; in LINE [35], s equals

Table 2. CISE Outperforms Baseline Methods on Paper-Publishing Behavior Prediction

Method	Weights w_t	MAE	RMSE	Acc.	Avg. Pre.	AUC	F1	Spearman's ρ	Kendall's τ
LINE [35]		0.1318	0.2573	0.9102	0.9736	0.9735	0.9082	0.9350	0.8222
DeepWalk [28]		0.1215	0.2463	0.9175	0.9776	0.9776	0.9160	0.9461	0.8366
VERSE [39]		0.1210	0.2455	0.9182	0.9778	0.9779	0.9164	0.9468	0.8382
node2vec [10]		0.1206	0.2454	0.9180	0.9779	0.9779	0.9166	0.9469	0.8375
metapath2vec [6]		0.1181	0.2440	0.9196	0.9781	0.9781	0.9181	0.9470	0.8405
CISE-P_n (Size-constrained negative behavior sampling)	{1,1,1,1}	0.0717 (−39.3%)	0.1864 (−23.6%)	0.9537 (+3.7%)	0.9911 (+1.3%)	0.9911 (+1.3%)	0.9581 (+4.4%)	0.9819 (+3.7%)	0.9086 (+8.1%)
	{1,1,1,3}	0.1000	0.2169	0.9353	0.9861	0.9861	0.9394	0.9726	0.8723
	{1,1,3,1}	0.0684	0.1858	0.9542	0.9914	0.9914	0.9577	0.9821	0.9095
	{1,3,1,1}	0.0690	0.1861	0.9542	0.9914	0.9914	0.9577	0.9820	0.9095
	{3,1,1,1}	0.0679	0.1850	0.9546	0.9916	0.9916	0.9582	0.9825	0.9103
CISE-P_t (Type-distribution -constrained negative behavior sampling)	{1,1,1,1}	0.0584	0.1676	0.9627	0.9935	0.9935	0.9655	0.9872	0.9261
	{1,1,1,3}	0.0837	0.1994	0.9447	0.9897	0.9896	0.9491	0.9804	0.8903
	{1,1,3,1}	0.0589	0.1682	0.9625	0.9934	0.9934	0.9654	0.9870	0.9257
	{1,3,1,1}	0.0588	0.1680	0.9625	0.9935	0.9934	0.9653	0.9871	0.9257
	{3,1,1,1}	**0.0523 (−55.7%)**	**0.1619 (−33.6%)**	**0.9653 (+5.0%)**	**0.9945 (+1.7%)**	**0.9945 (+1.7%)**	**0.9681 (+5.4%)**	**0.9890 (+4.4%)**	**0.9311 (+10.8%)**

Types in w_t are {author, conf., keyword, ref.}. Improvements (%) made by CISE over the best baseline method metapath2vec are shown in parentheses. Except for MAE and RMSE, higher scores indicate better performance.

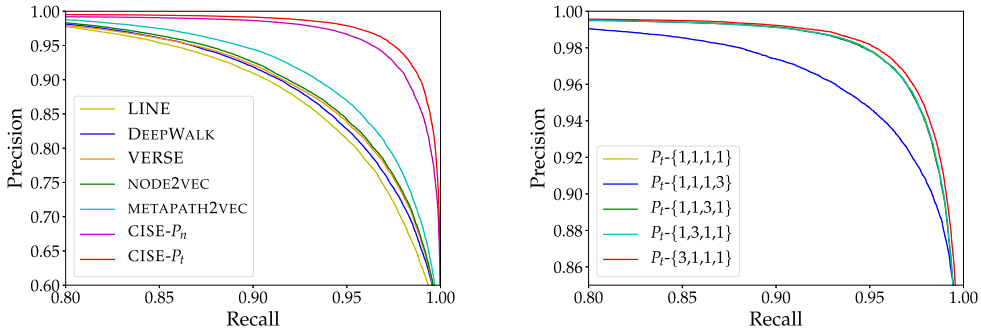
to the total numbers of edge sampling; and, for CISE, $s = \sum_{b \in R} |b|$ where R is the collection of all sampled training behaviors and s represents the total number of sampled context items. The best return and in-out hyperparameters of node2vec [10] are selected using a grid search over $p, q \in \{0.25, 0.50, 1, 2, 4\}$ as suggested by the authors. For metapath2vec [6], we use the meta-path scheme "Keyword-Author-Paper-Conference-Paper-Author-Keyword" and "Word-Hashtag-Location-Hashtag-Word" to guide random walks on two datasets, respectively. For all baselines, we generate the behavior embedding using embeddings of context item nodes according to Equation (1). The default number of dimensions d is 128; the default size of negative samples is 10. All other hyperparameters are set to typical values used in previous studies unless specified otherwise.

5.3 Contextual Behavior Prediction

We introduce validation settings of evaluating embedding methods on predicting the probability of a paper-publishing behavior's success in the publication dataset. Both quantitative and qualitative analyses are provided to demonstrate the effectiveness of CISE.

5.3.1 Validation Settings. We use 10-fold cross-validation to evaluate all methods. First, we randomly sample out 10% of behaviors (papers) for testing and use the remaining 90% for training. The whole training set is used to learn itemset embeddings. And, the same embedding initialization is shared by different methods within each fold to rule out random noise. Then, 10% of the training set (as positive instances), along with the same amount of itemsets generated by type-distribution-constrained negative behavior sampling (as negative instances), are selected out. For each method, we train a logistic regression model with the latent representations and corresponding labels of these instances. Lastly, we generate negative instances of the same size of the testing set. For each test instance b, the regression model returns a probability score $r(b)$.

Multiple evaluation metrics are used which fall into three categories. For error-based measures, we use **Mean Absolute Error** (**MAE**) and **Root Mean Squared Error** (**RMSE**) to evaluate the embedding quality. Smaller error-based measure value indicates better method performance.



(a) Comparing CISE with baseline models

(b) Comparing different type weights in CISE-P_t

Fig. 4. Precision-recall curves of the baseline methods and itemset embedding methods (with different settings of item type weights) on contextual behavior prediction.

Since each behavior (paper) is associated with a binary label, we can also use standard information retrieval metrics such as **Accuracy** (**Acc.**), **Average Precision** (**Avg. Pre.**), **Area Under the Curve** (**AUC**), and F1 score. A method that has a higher score in these metrics is better. In addition, Spearman's ρ and Kendall's τ are two ranking-based correlation coefficients computed from the ranking of prediction scores. Higher value indicates better model performance.

5.3.2 Quantitative Analysis. Table 2 presents the performance of all the baseline methods and all the variants of CISE. Figure 4 presents the precision-recall curves of these methods.

Overall performance. The best baseline method is the heterogeneous network embedding method metapath2vec [6], which gives an RMSE of 0.2440, an F1 of 0.9181, and a Kendall's τ of 0.8405. Its performance is much better than any random guessing model can achieve and tells that pairwise similarity still plays an indispensable role in predicting behaviors. For example, co-authors often work in very similar research fields. The vanilla version of our itemset embedding method is CISE-P_n with uniform type weights. Despite its simplicity, this model can score an RMSE of 0.1864 (**−23.6%** relatively), an F1 of 0.9581 (**+4.4%** relatively), and a Kendall's τ of 0.9086 (**+8.1%** relatively) when compared to the best baseline metapath2vec [6]. All improvements in parentheses are tested being statistically significant with a p-value of less than 0.05. The best variant of our itemset embedding method CISE holds (1) type-distribution-constrained negative behavior sampling strategy and (2) type weights as {3,1,1,1} (authors tend to have higher weights). It scores an RMSE of 0.1619 (**−33.6%** relatively), an F1 of 0.9681 (**+5.4%** relatively), and a Kendall's τ of 0.9311 (**+10.8%** relatively). Network embedding methods show pretty high AUC scores because the pairwise similarities between the items (e.g., authors and keywords) do have an impact on the chance of collaboration. Instead, by preserving itemset structures, our itemset embedding method CISE is able to yield **near-perfect** AUC (0.9945; **+1.7%** relatively). Figure 4(a) also shows the high effectiveness of CISE: the red curve CISE-P_t is closest to the upper right corner.

Comparing network embedding methods. First, DeepWalk [28] and node2vec [10] perform better than LINE [35] in this task. This indicates that preserving random-walk-based local network information, or node neighborhoods, is more effective than preserving connections and common neighbors on predicting a collaboration. VERSE [39] has almost the same performance as node2vec [10], indicating that the similarity metric of Personalized PageRank is also pretty helpful in this task. metapath2vec [6] learns the low-dimensional representations of nodes from richer meta-path-based features. In other words, it can model the heterogeneity of the network.

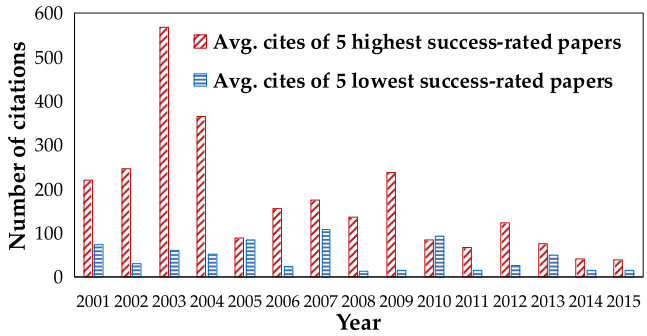


Fig. 5. Papers that have higher estimated success rates tend to be cited more (w.r.t. Google Scholar Feb.'18).

Thus, it performs the best among the baseline methods in this task. This confirms the necessity of taking multiple types of context items into consideration in the paper-publishing behaviors.

Comparing negative behavior sampling strategies. Here, we compare our CISE-P_t (type-distribution-constrained) with CISE-P_n (size-constrained) by fixing the same type weights. The CISE-P_n with uniform type weights ({1,1,1,1}) can generate an RMSE of 0.1864, an F1 of 0.9581, and a Kendall's τ of 0.9086 (**−23.6%**, **+4.4%**, and **+8.1%** relatively over METAPATH2VEC). It shows the power and effectiveness of itemset embedding on behavior prediction. We further observe that CISE-P_t, which uses the same uniform type weights, can decrease the errors to an RMSE of 0.1676 (**−10.1%** relatively over CISE-P_n), and increase the F1 score to 0.9655, Kendall's τ to 0.9261 (**+0.8%** and **+1.9%** relatively, over CISE-P_n). This demonstrates the advantage of type-distribution-constrained sampling—it considers the context types when generating negative samples, so the composition of the positive itemset is carefully modeled into the negative itemset. The best variant of CISE-P_t with type weights {3,1,1,1} generates **−12.5%** relative to RMSE, **+1.0%** relative to F1, and **+2.3%** relative to Kendall's τ compared to CISE-P_n with the same type weights.

Comparing different settings on context type weights. We adopt a grid search strategy to evaluate through the parameter space of type weight combinations in $\{1, 2, 3\}$. The best performance for each type is found and reported by setting a larger weight value (e.g., {3 (author), 1 (conference), 1 (keyword), 1 (reference)} produces the best performance of the type of author). For CISE-P_t, we further examine four settings when changing the weight of one type into 3 and keeping others fixed at 1. This puts an emphasis on one type while preserving the itemset structure. First, we note that when we set the weight of reference to 3, CISE gives us lower performance than any other type weight settings. Besides, setting conference or keyword type weight to 3 does not provide any improvements. Their curves almost overlap the curve of uniform type weights setting in Figure 4(b). Both Table 2 and Figure 4(b) demonstrate that only the type weights {3 (author),1 (conference),1 (keyword),1 (reference)} can outperform all other type weight settings. The RMSE drops from 0.1676 to 0.1619 (**−3.4%** relatively to uniform type weights); the F1 and Kendall's τ increase from 0.9655 to 0.9681 and from 0.9261 to 0.9311 (**+0.3%** and **+0.6%** relatively to uniform type weights). This matches our intuition: authors should play decisive roles in making the collaboration successful.

5.3.3 Qualitative Analysis. We answer three questions here:

Q1: Are papers of high estimated success rates not only successful but also impactful?
Yes. We collect two groups of papers in the testing set that have the highest/lowest estimated success rates given by CISE. Then, we use Google Scholar to manually collect the number of citations

of them. Figure 5 presents the average number of citations of these papers published in 15 years ranging from 2001 to 2015. By comparing these two groups of papers, we observe that papers of the highest success rate consistently have more citations than those of the lowest success rate. All these papers have been successful (with estimated success rates above 0.6), but a higher estimated success rate clearly indicates they are more likely to be impactful in the real world.

One good example is the paper *"Inferring Social Ties across Heterogeneous Networks"* [34] published in WSDM 2012. The leading author Dr. Jie Tang (Tsinghua University) is a data mining expert on a subset of the keywords such as "factor graph," "heterogeneous network," and "predictive model," and the co-author Dr. Jon Kleinberg (Cornell University) has a world-level reputation in computational social science of keywords like "social theory," "social influence," and "social ties." The collaboration between them successfully integrated their complementary expertise. As a result, they proposed an effective factor graph based predictive model of inferring social ties across heterogeneous networks. This article has the highest estimated success rate among other papers in the same conference proceedings. It has been cited more than 220 times and is ranked within the top 3 of all testing papers!

Q2: What if a negative sample, i.e., a pseudo-paper, has a good success rate? Is it possibly a good paper? Maybe. Most of the pseudo-papers have a low success rate (0.052 on avg.), but we did find some of them have a good success rate. The example below has a rate of *0.549*.

Example 5.1. Authors: Richard Sproat, Weiying Ma, Jiawei Han, Xiaoli Li; *Conference:* SIGIR; *Keywords:* text mining, Gaussian process, biological network, scalability.

In this example, Dr. Richard Sproat studies computational linguistics; Dr. Weiying Ma is an NLP and Information Retrieval scientist; Dr. Jiawei Han is famous for data mining and heterogeneous network mining; Dr. Xiao Li works on bioinformatics and bio network mining. All four keywords are quite relevant to their expertise areas. A plausible paper topic could be reduced as a scalable learning framework for biological text and network mining. However, it is still very difficult to become a real paper. CISE does not model the cost of incorporating new items into an existing itemset, which can be highly expensive as in this example. We leave this issue as a future work.

Q3: Does it mean negative samples of extremely low success rate are likely to be impossible to publish? Yes, very likely. Among the pseudo-papers having very low success rate (below 0.001), we observe a few cases: (1) the authors have little chance to build ties between each other; (2) the authors are not experts in the field of the conference, e.g., {Yan Liu, SIGGRAPH}; and, (3) the keywords are not likely to induce a plausible research topic. For instance, keywords {"heterogeneity," "degree of freedom," "biomedical"}, or keywords {"citation analysis," "chi squared statistic," "customer retention"}. It is highly unlikely for any one of these pseudo-papers to publish if no other complementary items are added into the itemset to improve its estimated success rate.

5.3.4 Visualization. Figure 6 visualizes two itemsets, i.e., papers, in a three-dimensional embedding space. The left subfigure represents the WSDM 2012 paper [34] we mentioned, and the right subfigure represents a pseudo-paper that was generated by negative behavior sampling. We visualize an itemset as the combination of the vectors of its items. The item vectors are colored by their context types. The paper's vector starts from the origin and consists of the authors' vectors, conference's vector, keywords' vectors, and references' vectors.

For the real paper, the vectors of authors Dr. Jie Tang and Dr. Jon Kleinberg, and the keyword "social networks" contribute the most to the magnitude of the real paper's vector. Interestingly, all of them have high scores on the second dimension, which indicates items of different types are well mixed in the low-dimensional embedding space. They collectively elongate the paper's vector significantly toward that dimension. On the contrary, the vectors of items in the pseudo-paper are

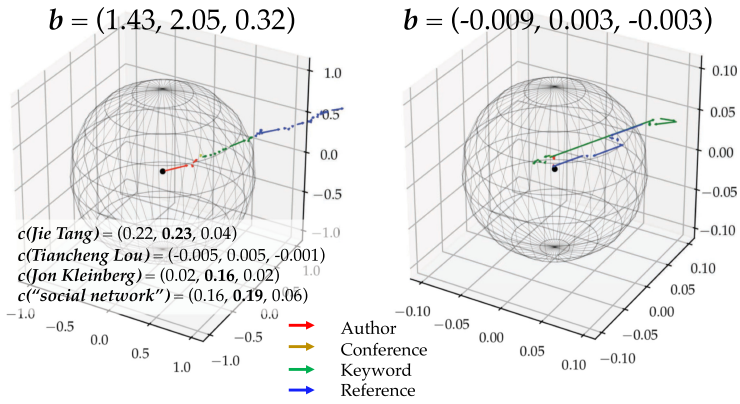


Fig. 6. Visualization of a real paper [34] (left) and a pseudo-paper (right) in three-dimensional embedding space. The item vectors are colored by their context types and learned by itemset embedding method CISE-P_t.

not always short in the space, but they are not complementary with each other. So, the itemset's vector goes back and forth in the space and finally generates a limited distance from the origin.

5.4 Behavioral Context Recommendation

We introduce validation settings and provide quantitative analysis to demonstrate the effectiveness of CISE on recommending complementary items for behaviors in the publication dataset.

5.4.1 Validation Settings. We also adopt the 10-fold cross-validation strategy. For each time of 10 folds in evaluating the behavior prediction performance, we keep the embeddings learned from the training set, and use the testing set to evaluate models' recommendation performance. So, given a particular item type, e.g., author, and a test itemset, i.e., paper, we hide one of the items of that type in the itemset. Then, we enumerate through every other item of the same type in our dataset and use the trained logistic regression model to generate the predicted success rate of incorporating this new item into the itemset. Thus, we have a complete list of predictions on that type and we rank the type items by the their correponding success rates. We assume the real hidden context item c^* of type t should be ranked at a higher position if the method makes a better recommendation.

We use $rank(c, b)$ to denote the recommendation rank of context item c on itemset b after hiding c^*, and use the **Harmonic Mean of Ranks** (**HMR**) on all test itemsets to evaluate the performance for each item type:

$$HMR(t) = \frac{|\mathcal{T}^{(+)}|}{\sum_{b \in \mathcal{T}^{(+)}} \frac{1}{rank(c \sim C^{(t)}, b)}}, \tag{14}$$

where $C^{(t)}$ is the set of items of type t and $\mathcal{T}^{(+)}$ is the testing set of all positive instances. A better method should have a smaller HMR value.

5.4.2 Quantitative Analysis. Table 3 presents the results of all methods on recommending an item for a given itemset of paper-publishing behavior. Overall, CISE consistently generates lower HMRs on all context types (with few exceptions on the conference type) than baseline methods. With respect to the negative behavior sampling strategies, CISE-P_t outperforms CISE-P_n, showing that type-distribution-constrained negative sampling is more effective in learning low-dimensional representations of itemsets for context recommendation.

Table 3. Harmonic Mean of Ranks (HMR) of CISE and Baseline Methods on Recommending Context Item for Given Itemset

Method	Weights	HMR (author)	HMR (conference)	HMR (keyword)	HMR (reference)
LINE [35]		7,023.2	25.2	80.0	6,776.2
DeepWalk [28]		6,428.1	27.3	103.2	5,043.1
VERSE [39]		5,769.2	25.8	76.3	4,492.6
node2vec [10]		5,820.7	29.2	139.7	4,057.5
metapath2vec [6]		5,489.9	24.8	162.3	4,076.1
CISE-P_n (Size-constrained negative behavior sampling)	{1,1,1,1}	5,066.7	25.6	62.6	3,730.8
	{1,1,1,3}	5,332.0	26.8	76.4	3,678.2
	{1,1,3,1}	5,109.2	26.2	58.9	3,713.0
	{1,3,1,1}	5,092.2	24.2	66.9	3,911.9
	{3,1,1,1}	4,879.7	24.4	60.2	3,829.3
CISE-P_t (Type-distribution -constrained negative behavior sampling)	{1,1,1,1}	4,303.3	25.0	55.7	3,683.8
	{1,1,1,3}	4,812.2	26.7	64.0	3,592.1
	{1,1,3,1}	4,467.4	25.5	49.3	3,743.9
	{1,3,1,1}	4,442.9	**23.2**	53.9	3,812.0
	{3,1,1,1}	**4,165.8**	24.9	**49.0**	**3,553.9**

A smaller HMR indicates a better method recommendation performance. The total number of authors, conferences, keywords, and references are 195,152, 1,265, 39,756, and 596,796, respectively.

Recommending a co-author. Among all 195,152 authors in the dataset, given a paper's conference, keywords, references, and all other authors, the real author is at the HMR of 4,165.8 on the list of authors that our CISE ranks according to the success rate of incorporating him/her into the paper. This is an extremely challenging task. The best baseline method metapath2vec [6] ranks the real author at the HMR of 5,489.9. This confirms that CISE is capable of finding more complementary co-authors to make the paper more likely to be published, based on preserving the success structures in the training itemsets.

Recommending a conference or a keyword. All the baselines and our methods can rank the real conference at the HMR around 25 among all 1,265 conferences. CISE can score an HMR of 49.0 on recommending a keyword among all 39,756 keywords while other baseline methods rank the real keyword of at least 76.3 of HMR.

Recommending a reference. This is also a challenging task comparable to recommending a co-author. Among all 596,796 references, CISE ranks the real reference at the HMR of 3,553.9. The best baseline node2vec ranks it at the HMR of 4,057.5. Given authors, keywords, and the conference, our method is more likely to recommend complementary or valuable papers to read and cite.

5.5 Social Media Behavior Data

We further validate the effectiveness of CISE in the domain of social media. In parallel to Section 5.3 and Section 5.4, we test CISE and all baseline methods on the tasks of (a) predicting the success, i.e., the popularity level, of a tweet-posting behavior and (b) recommending complementary items such as words or hashtags to any new tweet to maximize its predicting success. The validation settings are similar to previous experiments with a few key differences: (1) during the training process, CISE did not conduct any negative behavior sampling since we have observed negative behaviors of tweets with low retweet count as mentioned in Section 5.2; (2) for all baseline methods, the networks of context items and behaviors are connected by success rate (log-scale to reduce variance) weighted edges instead of unweighted edges; and (3) we train a linear regression model (w/l1 regularization) on top of context latent representations to generate prediction score $r(b)$.

Table 4. CISE Outperforms Baseline Methods on Tweet-Posting Behavior Prediction

Method	Weights w_t	MAE	RMSE	Spearman's ρ	Kendall's τ
LINE [35]		0.8798	1.0723	0.4809	0.3431
DeepWalk [28]		0.8784	1.0647	0.4894	0.3501
metapath2vec [6]		0.8669	1.0520	0.4942	0.3588
VERSE [39]		0.8508	1.0382	0.5163	0.3770
node2vec [10]		0.8419	1.0301	0.5290	0.3808
CISE (No negative behaviors sampling)	{1,1,1}	0.8117	0.9982	0.5834	0.4243
	{1,1,3}	0.8316	1.0216	0.5492	0.3962
	{1,3,1}	0.8253	1.0164	0.5516	0.4079
	{3,1,1}	**0.8069 (−4.2%)**	**0.9916 (−3.7%)**	**0.6007 (+13.6%)**	**0.4389 (+15.3%)**

Types in w_t are {word, hashtag, location}. Improvements (%) made by CISE over node2vec are shown in parentheses. Smaller MAE or RMSE value and higher scores of Spearman's ρ or Kendall's τ indicate better performance.

Table 5. Harmonic Mean of Ranks (HMR) of CISE and Baseline Methods on Recommending Words and Hashtags for Given Tweet-Posting Behavior

Method	Weights	HMR (word)	HMR (hashtag)
LINE [35]		332.6	298.2
DeepWalk [28]		286.8	469.3
metapath2vec [6]		306.5	176.5
VERSE [39]		287.3	363.8
node2vec [10]		250.3	582.1
CISE (No negative behaviors sampling)	{1,1,1}	144.0	104.2
	{1,1,3}	212.6	128.3
	{1,3,1}	193.4	**62.5**
	{3,1,1}	**132.7**	95.6

The total number of words and hashtags are 25,000 and 11,071.

The prediction results are provided in Table 4. We can see the results are inconsistent with previous experiments: CISE with uniform weigths can score an RMSE of 0.9982 (**−3.1%** relatively) and a Kendall's τ of 0.4243 (**+10.3%** relatively) when compared to the best baseline node2vec [10]; and, by setting the type weights to {3 (word),1 (hashtag),1 (location)}, CISE can further decrease the RMSE to 0.9916 (**−3.7%** relatively) and increases the Kendall's τ to 0.4389 (**+15.3%** relatively) compared with node2vec [10]. We note that CISE's type weights emphasizing the hashtags or the location of tweets produce suboptimal performance. This is due to the high coherence of hashtags within the general topic of FIFA 2018, and the location information of tweet is not greatly helpful in this task. Also, we note that metapath2vec [6] is not the best baseline method in this task and can only generate comparable results as DeepWalk [28]. This can be explained by the fact that, in different domains, it is not always obvious to design reasonable meta-paths for guiding heterogeneous random walks and capturing the semantic information among them.

The recommendation results are given in Table 5. It is evident that CISE outperforms baseline methods and scores an HMR of 132.7 for recommending a word when setting the type weights

Table 6. Top Similar and Complementary Authors with Dr. Jure Leskovec

Rank	Similarity		Complementarity	
	Author	Score	Author	Score
1	Caroline Lo	0.895	Eric Horvitz	0.192
2	Jaewon Yang	0.860	Jon Kleinberg	0.188
3	Seth Myers	0.858	Christos Faloutsos	0.177
4	Justin Cheng	0.856	Susan Dumais	0.176
5	Ashton Anderson	0.853	Héctor García-Molina	0.173
6	Mary Mcglohon	0.843	Samuel Madden	0.170
7	Gregory Kossinets	0.832	Daniel Jurafsky	0.142
8	Mohammad Mahdian	0.831	Carlos Guestrin	0.134
9	Siddharth Suri	0.828	Daniel P. Huttenlocher	0.128
10	Robert West	0.824	Natasa Milic-Frayling	0.127

Similarity is measured by cosine similarity. Complementarity is specified in Equation (12).

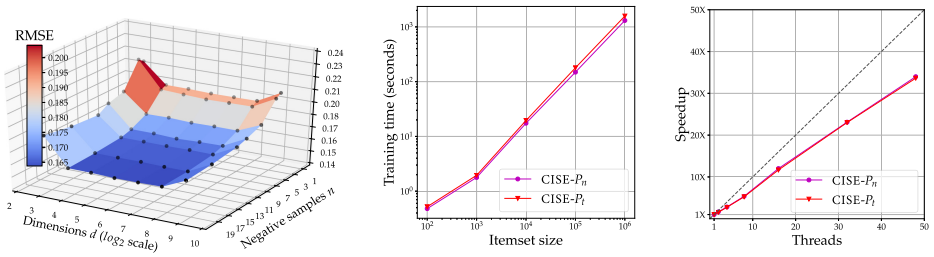
emphasizing the word type. Interestingly, when setting the type weights to emphasize the hashtag, CISE is able to produce an even smaller HMR of 62.5 for recommending a hashtag. The best baseline method NODE2VEC [10] for recommending a word can only score an HMR of 250.3, and the best baseline method METAPATH2VEC [6] for recommending a hashtag can only score an HMR of 176.5. This demonstrates CISE is capable of recommending more valuable words and hashtags to be included in a tweet to make it potentially more popular.

5.6 Context Complementarity vs. Similarity

We present case studies of comparing complementarity against similarity. We take Dr. Jure Leskovec, a famous researcher in the field of data mining (an Associate Professor from Stanford University with 45,000+ citations) as an example. In Table 6, we list Dr. Leskovec's 10 most similar authors based on the representations learned by METAPATH2VEC [6] and the 10 most complementary authors based on the representations learned by our CISE.

Top authors ranked by similarity. All 10 top similar authors have high similarity scores of greater than 0.82, while the mean similarity score for all other authors in our dataset is below 0.43. This indicates the existence of the community structure. These authors work on very similar research topics as Dr. Leskovec does. Specifically, the top five authors, as well as the 10th author, are graduated Ph.D. students advised by Dr. Leskovec; the 6th author, Dr. Mary Mcglohon, has the same advisor, Dr. Christos Faloutsos, as Dr. Leskovec has, when they were both graduate students at Carnegie Mellon University; and, the 7th, 8th, and 9th authors worked closely with Dr. Leskovec when he was a postdoctoral researcher at Cornell University working with Dr. Jon Kleinberg. On average, each one of them have 2.7 papers co-authored with Dr. Leskovec.

Top authors ranked by complementarity. Different from the most similar authors, the top 10 complementary authors presented in Table 6 have a much wider range of research interests and more diverse backgrounds: Dr. Susan Dumais studies information retrieval at Microsoft Research; Dr. Samuel Madden studies databases and distributed computing at MIT; and, Dr. Carlos Guestrin works on machine learning at the University of Washington. All of them are very influential researchers. Their collaboration with Dr. Leskovec can increase the success rate of more than 0.127 by contributing their complementary skills into the paper-publishing plan. However, being



(a) Sensitivity (b) Scalability (c) Parallelizability

Fig. 7. Sensitivity and efficiency of CISE: (a) CISE is insensitive to two parameters of embedding size d and negative sample size n; (b) CISE demonstrates linearity in running time with increasing itemset size; and (c) CISE provides good speedup with increasing threads number.

complementary does not equal to being dissimilar. The top 10 complementary authors actually have an average rank of 119.7 on the similarity ranking among 195,152 authors in our dataset. Certainly, they are still similar with Dr. Leskovec to some aspect and extent. On the other hand, the top 10 similar authors have an average rank of 44.1 on the complementarity ranking of all 55 co-authors of Dr. Leskovec. This shows that being highly similar is not equal to being complementary.

The little overlap between the top 10 similar authors and the top 10 complementary authors confirms that complementarity is significantly different from similarity. In addition, we also examine complementarity using the FIFA World Cup 2018 Tweets dataset. We generated the top 10 most similar and complementary hashtags for a set of 20 manually selected hashtags. We found they are significantly different from each other with little overlap (average Jaccard socre of 0.23). Context complementarity aims to capture the synergistic effect created by two context items inside a behavior; while, similarity measure is based on the item's community structure in the behavior-item network.

5.7 Sensitivity Analysis

There are three parameters for CISE: the number of training samples $|R|$, embedding dimensions d, and number of negative samples per positive itemset sample n. We examine CISE's sensitivity of performance on the contextual behavior prediction task in Section 5.3 over different combinations of embedding dimensions d and number of negative samples n. Specifically, we vary d in $\{2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$ and vary n in $\{1, 2, 5, 8, 10, 15, 20\}$. Results are presented in Figure 7(a).

By fixing the embedding dimensions d, it is evident that RMSE initially drops with larger negative samples n, but the marginal improvements are slower until reaching a certain elbow point ($n \geq 10$) and remain flat afterward. This indicates CISE can learn behavior representations preserving the success structure with a small number of negative samples. By fixing the negative samples n, we can see CISE is pretty stable across most dimensions except extreme ones like 2^3 and 2^9. This is because too small d value leads to loss of information and too large d includes more noise. We use the default value $d = 128$ in Section 5.3 and Section 5.4 when comparing with baseline methods as it is the typical value used in previous studies. Our experiments show we are safe to choose a much smaller value, e.g., $d = 32$, for CISE without significantly affecting performance.

5.8 Efficiency Test

We test the efficiency of CISE from two aspects: (1) its scalability over increasing sizes of input behavior datasets and (2) the speedup it can provide by parallelizing to more threads. All

experiments are conducted on a single Dell PowerEdge R920 server with Quad 16 cores, 2.3 GHz Intel Xeon CPUs, E7-4850 v4 using our publicly available performance implementation of CISE.

For scalability, we run CISE on five different scales of behavior datasets, which are built from the papers we collected from the Microsoft Academic project before filtering, with itemset number ranging from 100 to 1,000,000 (context item numbers ranging from 4,023 to 2,849,201). We can see the running time generally grows linearly with the increasing itemset size as shown in Figure 7(b). Also, CISE-P_t takes slightly more time to train than CISE-P_n because the type-distribution-constrained negative sampling strategy has finer control over the random sampling process than the size-constrained negative sampling strategy. For parallelizability, we run CISE with different number of threads on the same dataset used in Section 5 and present their speedups in Figure 7(c). The diagonal line represents the ideal speedup, which rarely occurs due to overheads. Both CISE-P_n and CISE-P_t are able to provide pretty good linear speedup when running with more threads. Specifically, they can achieve a speedup of 34 when running with 48 threads. We conducted the same suite of efficiency experiments on the FIFA World Cup 2018 Tweets dataset and observed consistent trends of both linear scalability and good parallelizability. This presents CISE a practical tool on real large-scale behavior datasets.

6 CONCLUSIONS

In this article, we considered behavior as a set of context items and targeted two novel behavior modeling tasks: (1) predicting the success rate of any set of items and (2) finding complementary items which maximize the probability of success when incorporated into an itemset. We proposed a novel scalable method, *Multi-Type Itemset Embedding*, to learn context item presentations from massive behavior data preserving the success structures. It included a novel measurement of success rate for itemset; considered type weights for heterogeneity; and conducted negative behavior sampling for representation learning. Furthermore, we proposed a measurement of context complementarity. We provided theoretical analysis showing its uniqueness when compared to similarity, dissimilarity, and orthogonality. Extensive experiments demonstrated the proposed method's superiority. Case studies showed the difference between complementarity and similarity.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Bo Long. 2011. Localized factor models for multi-context recommendation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 609–617.

[2] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. 2013. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd WWW*. ACM, 37–48.

[3] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM CIKM*. ACM, 891–900.

[4] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *Proceedings of the T10th ACM WSDM*. ACM, 295–304.

[5] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th WWW*. ACM, 29–30.

[6] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD*. ACM, 135–144.

[7] Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V. Chawla, Jinghai Rao, and Huanhuan Cao. 2012. Link prediction and recommendation across heterogeneous social networks. In *2012 IEEE 12th ICDM*. IEEE, 181–190.

[8] Jeffrey H. Dyer and Harbir Singh. 1998. The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review* 23, 4 (1998), 660–679.

[9] Beyza Ermiş, Evrim Acar, and A. Taylan Cemgil. 2015. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery* 29, 1 (2015), 203–236.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[11] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, and Jiawei Han. 2016. Large-scale embedding learning in heterogeneous event data. In *2016 IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 907–912.
[12] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, Lance Kaplan, and Jiawei Han. 2017. Embedding learning with events in heterogeneous information networks. *IEEE Trans. Knowl. Data Eng.* 29 (2017), 2428–2441.
[13] Suzanne Hidi. 2001. Interest, reading, and learning: Theoretical and practical considerations. *Educational Psychology Review* 13, 3 (2001), 191–209.
[14] Michael A. Hitt, M. Tina Dacin, Edward Levitas, Jean-Luc Arregle, and Anca Borza. 2000. Partner selection in emerging and developed market contexts: Resource-based and organizational learning perspectives. *Academy of Management Journal* 43, 3 (2000), 449–467.
[15] Yuheng Hu, Fei Wang, and Subbarao Kambhampati. 2013. Listening to the crowd: Automated analysis of events via aggregated twitter sentiment. In *IJCAI*. 2640–2646.
[16] Mohsen Jamali and Laks Lakshmanan. 2013. HeteroMF: Recommendation in heterogeneous information networks using context dependent factor models. In *Proceedings of the 22nd WWW*. ACM, 643–654.
[17] Meng Jiang, Peng Cui, Fei Wang, Xinran Xu, Wenwu Zhu, and Shiqiang Yang. 2014. FEMA: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1186–1195.
[18] Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. 2016. Little is much: Bridging cross-platform behaviors through overlapped crowds. In *30th AAAI Conference on Artificial Intelligence*.
[19] Meng Jiang, Christos Faloutsos, and Jiawei Han. 2016. Catchtartan: Representing and summarizing dynamic multicontextual behaviors. In *Proceedings of the 22nd ACM SIGKDD*. ACM, 945–954.
[20] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. 2012. Supercharging recommender systems using taxonomies for learning user purchase behavior. *VLDB* 5, 10 (2012), 956–967.
[21] Joseph B. Kruskal. 1964. Nonmetric multidimensional scaling: A numerical method. *Psychometrika* 29, 2 (1964), 115–129.
[22] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*. 1188–1196.
[23] Defu Lian, Zhenyu Zhang, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, and Xing Xie. 2016. Regularized content-aware tensor factorization meets temporal-aware location recommendation. In *2016 IEEE 16th ICDM*. IEEE, 1029–1034.
[24] Zemin Liu, Vincent W. Zheng, Zhou Zhao, Zhao Li, Hongxia Yang, Minghui Wu, and Jing Ying. 2018. Interactive paths embedding for semantic proximity search on heterogeneous graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1860–1869.
[25] David Matsumoto. 2007. Culture, context, and behavior. *Journal of Personality* 75, 6 (2007), 1285–1320.
[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.
[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.
[28] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710.
[29] Ioakeim Perros, Evangelos E. Papalexakis, Fei Wang, Richard Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. 2017. SPARTan: Scalable PARAFAC2 for large & sparse data. In *Proceedings of the 23rd ACM SIGKDD*. ACM.
[30] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*. 693–701.
[31] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM WSDM*. ACM, 81–90.
[32] Alan Said, Shlomo Berkovsky, and Ernesto W. De Luca. 2010. Putting things in context: Challenge on context-aware movie recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*. ACM, 2–6.
[33] Ellen A. Skinner and Michael J. Belmont. 1993. Motivation in the classroom: Reciprocal effects of teacher behavior and student engagement across the school year. *Journal of Educational Psychology* 85, 4 (1993), 571.
[34] Jie Tang, Tiancheng Lou, and Jon Kleinberg. 2012. Inferring social ties across heterogenous networks. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*. ACM, 743–752.
[35] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th WWW*. ACM, 1067–1077.
[36] Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery* 23, 3 (2011), 447–478.
[37] Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (2000), 2319–2323.

[38] Michael E. Tipping and Christopher M. Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, 3 (1999), 611–622.
[39] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. 2018. Verse: Versatile graph embeddings from similarity measures. In *Proceedings of the 2018 WWW*. ACM, 539–548.
[40] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1225–1234.
[41] Daheng Wang, Meng Jiang, Munira Syed, Oliver Conway, Vishal Juneja, Sriram Subramanian, and Nitesh V. Chawla. 2020. Calendar graph neural networks for modeling time structures in spatiotemporal user behaviors. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'20)*. ACM.
[42] Daheng Wang, Meng Jiang, Qingkai Zeng, Zachary Eberhart, and Nitesh V. Chawla. 2018. Multi-type itemset embedding for learning behavior success. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2397–2406.
[43] Daheng Wang, Tianwen Jiang, Nitesh V. Chawla, and Meng Jiang. 2019. TUBE: Embedding behavior outcomes for predicting success. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1682–1690.
[44] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network representation learning with rich text information. In *IJCAI*. 2111–2117.
[45] Kai Yang, Xiang Li, Haifeng Liu, Jing Mei, Guo Tong Xie, Junfeng Zhao, Bing Xie, and Fei Wang. 2017. TaGiTeD: Predictive task guided tensor decomposition for representation learning from electronic health records. In *AAAI*.
[46] Wenhao Yu, Mengxia Yu, Tong Zhao, and Meng Jiang. 2020. Identifying referential intention with heterogeneous contexts. In *Proceedings of The Web Conference (WWW'20)*. 962–972.
[47] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. 2018. Arbitrary-order proximity preserved network embedding. In *Proceedings of the 24th ACM SIGKDD*. ACM, 2778–2786.
[48] Guoshuai Zhao, Xueming Qian, and Xing Xie. 2016. User-service rating prediction by exploring social users' rating behaviors. *IEEE Transactions on Multimedia* 18, 3 (2016), 496–506.
[49] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2021. Data augmentation for graph neural networks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*.
[50] Tong Zhao, Matthew Malir, and Meng Jiang. 2018. Actionable objective optimization for suspicious behavior detection on large bipartite graphs. In *2018 IEEE International Conference on Big Data (Big Data'18)*. IEEE, 1248–1257.

Received March 2019; revised July 2020; accepted March 2021