Graph Embedding with Shifted Inner Product Similarity and Its Improved Approximation Capability

Akifumi Okuno[†,‡] okuno@sys.i.kyoto-u.ac.jp
Geewook Kim[†,‡] geewook@sys.i.kyoto-u.ac.jp
Hidetoshi Shimodaira[†,‡] shimo@i.kyoto-u.ac.jp
[†]Graduate School of Informatics, Kyoto University, [‡]RIKEN Center for Artificial Intelligence Project (AIP)

Abstract

We propose *shifted inner-product similarity (SIPS)*, which is a novel yet very simple extension of the ordinary inner-product similarity (IPS) for neural-network based graph embedding (GE). In contrast to IPS, that is limited to approximating positive-definite (PD) similarities, SIPS goes beyond the limitation by introducing bias terms in IPS; we theoretically prove that SIPS is capable of approximating not only PD but also conditionally PD (CPD) similarities with many examples such as cosine similarity, negative Poincaré distance and negative Wasserstein distance. Since SIPS with sufficiently large neural networks learns a variety of similarities, SIPS alleviates the need for configuring the similarity function of GE. Approximation error rate is also evaluated, and experiments on two real-world datasets demonstrate that graph embedding using SIPS indeed outperforms existing methods.

1 INTRODUCTION

Graph embedding (GE) of relational data, such as texts, images, and videos, etc., now plays an indispensable role in machine learning. To name but a few, words and contexts in a corpus constitute relational data, and their vector representations obtained by skip-gram model (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are often used in natural language processing. More classically, a similarity graph is constructed from data vectors, and nodes are embedded to a lower dimensional space where connected nodes are closer to each other (Cai et al., 2018).

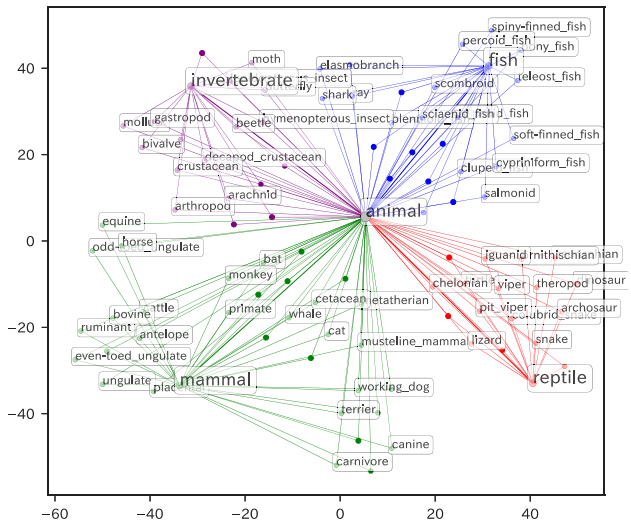

Figure 1: Visualization of word feature vectors for WordNet dataset computed by GE with our proposed SIPS model. See Supplement A for details.

Embedding is often designed so that the inner product between two vector representations in Euclidean space expresses their similarity. In addition to its interpretability, the inner product similarity has the following two desirable properties: (1) The vector representations are suitable for downstream tasks as feature vectors because machine learning methods are often based on inner products (e.g., kernel methods). (2) Simple vector arithmetic in the embedded space may represent similarity arithmetic such as the "linguistic regularities" of word vectors (Mikolov et al., 2013b). The latter property comes from the distributive law of inner product $\langle \boldsymbol{a}+\boldsymbol{b}, \boldsymbol{c}\rangle = \langle \boldsymbol{a}, \boldsymbol{c}\rangle + \langle \boldsymbol{b}, \boldsymbol{c}\rangle$, which decomposes the similarity of $\boldsymbol{a}+\boldsymbol{b}$ and $\boldsymbol{c}$ into the sum of the two similarities. For seeking the word vector

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

$\boldsymbol{y}' = \boldsymbol{y}_{\text{queen}}$, we maximize $\langle \boldsymbol{y}_{\text{king}} - \boldsymbol{y}_{\text{man}} + \boldsymbol{y}_{\text{woman}}, \boldsymbol{y}' \rangle = \langle \boldsymbol{y}_{\text{king}}, \boldsymbol{y}' \rangle - \langle \boldsymbol{y}_{\text{man}}, \boldsymbol{y}' \rangle + \langle \boldsymbol{y}_{\text{woman}}, \boldsymbol{y}' \rangle$ in Eq. (3) of Levy and Goldberg (2014). Thus solving analogy questions with vector arithmetic is mathematically equivalent to seeking a word which is similar to king and woman but is different from man.

Although classical GE has been quite successful, it considers simply the graph structure, where data vectors (pre-obtained attributes such as color-histograms of images), if any, are used only through the similarity graph. To fully utilize data vectors, neural networks (NNs) are incorporated into GE so that data vectors are converted to new vector representations (Kipf and Welling, 2016; Zhanga et al., 2017; Hamilton et al., 2017; Dai et al., 2018), which reduces to the classical GE by taking 1-hot vectors as data vectors. While these methods consider 1-view setting, multi-view setting is considered in Probabilistic Multi-view Graph Embedding (Okuno et al., 2018, PMvGE), which generalizes existing multivariate analysis methods (e.g., PCA and CCA) and NN-extensions (Andrew et al., 2013, DCCA) as well as graph embedding methods such as Locality Preserving Projections (He and Niyogi, 2004; Yan et al., 2007, LPP), Cross-view Graph Embedding (Huang et al., 2012, CvGE), and Cross-Domain Matching Correlation Analysis (Shimodaira, 2016, CDMCA). In these methods, the inner product of two vector representations obtained via NNs represents the strength of association between the corresponding two data vectors. The vector representations and the inner products are referred to as *feature vectors* and *Inner Product Similarities (IPS)*, respectively, in this paper.

IPS is considered to be highly expressive for representing the association between data vectors due to the Universal Approximation Theorem (Funahashi, 1989; Cybenko, 1989; Yarotsky, 2017; Telgarsky, 2017, UAT) for NN, which proves that NNs having many hidden units approximate arbitrary continuous functions within any given accuracy. However, since IPS considers the inner product of two vector-valued NNs, the UAT is *not* directly applicable to the whole network with the constraints at the final layer. Thus the approximation capability of IPS is yet to be clarified.

For that reason, Okuno et al. (2018) incorporates UAT into Mercer's theorem (Minh et al., 2006) and proves that IPS approximates any similarity based on Positive Definite (PD) kernels arbitrary well. For example, IPS can learn cosine similarity, because it is a PD kernel. This result shows not only the validity but also the fundamental limitation of IPS, meaning that the PD-ness of the kernels is required for IPS.

To overcome the limitation, similarities based on specific kernels other than the inner product have received considerable attention in recent years. One example is Poincaré embedding (Nickel and Kiela, 2017) which is an NN-based GE using Poincaré distance for embedding vectors in hyperbolic space instead of Euclidean space. Hyperbolic space is especially compatible with computing feature vectors of tree-structured relational data (Sarkar, 2011). While these methods efficiently compute reasonable low-dimensional feature vectors by virtue of specific kernels, their theoretical differences from IPS is not well understood.

In order to provide theoretical insights on these methods, in this paper, we will point out that some specific kernels are not PD by referring to existing studies. To deal with such non-PD kernels, we consider Conditionally PD (CPD) kernels (Berg et al., 1984; Schölkopf, 2001) which include PD kernels as special cases. We then propose a novel model named *Shifted IPS (SIPS)* that approximates similarities based on CPD kernels within any given accuracy. Interestingly, negative Poincaré distance is already proved to be CPD (Faraut and Harzallah, 1974) and it is not PD. So, similarities based on this kernel can be approximated by SIPS but not by IPS. Although we can think of a further generalization beyond CPD, this is only touched in Supplement E by defining *inner product difference similarity (IPDS)* model.

Our contribution is summarized as follows:

(1) We show that IPS cannot approximate a non-PD kernel; we propose SIPS to go beyond the limitation, and prove that SIPS can approximate any CPD similarities arbitrary well.

(2) We evaluate the error rate for SIPS to approximate CPD similarities, by incorporating neural networks such as multi-layer perceptron and deep neural networks.

(3) We conduct numerical experiments on two real-world datasets, to show that graph embedding using SIPS outperforms recent graph embedding methods.

This paper is an extension of Okuno and Shimodaira (2018) presented at ICML2018 workshop.

2 BACKGROUND

We work on an undirected graph consisting of n nodes $\{v_i\}_{i=1}^n$ and link weights $\{w_{ij}\}_{i,j=1}^n \subset \mathbb{R}_{\geq 0}$ satisfying $w_{ij} = w_{ji}$ and $w_{ii} = 0$, where w_{ij} represents the strength of association between v_i and v_j. The data vector representing the attributes (or side-information) at v_i is denoted as $\boldsymbol{x}_i \in \mathbb{R}^p$. If we have no attributes,

we use 1-hot vectors in $\mathbb{R}^n$ instead. We assume that the observed dataset consists of $\{w_{ij}\}_{i,j=1}^n$ and $\{\boldsymbol{x}_i\}_{i=1}^n$.

Let us consider a simple random graph model for the generative model of random variables $\{w_{ij}\}_{i,j=1}^n$ given data vectors $\{\boldsymbol{x}_i\}_{i=1}^n$. The conditional distribution of w_{ij} is specified by a *similarity function* $h(\boldsymbol{x}_i, \boldsymbol{x}_j)$ of the two data vectors. Typically, Bernoulli distribution $P(w_{ij} = 1|\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$ with sigmoid function $\sigma(x) := (1 + \exp(-x))^{-1}$ for 0-1 variable $w_{ij} \in \{0, 1\}$, and Poisson distribution $w_{ij} \sim \mathrm{Po}(\exp(h(\boldsymbol{x}_i, \boldsymbol{x}_j)))$ for non-negative integer variable $w_{ij} \in \{0, 1, \ldots\}$ are used to model the conditional probability. These models are in fact specifying the conditional expectation $E(w_{ij}|\boldsymbol{x}_i, \boldsymbol{x}_j)$ by $\sigma(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$ and $\exp(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$, respectively, and they correspond to logistic regression and Poisson regression in the context of generalized linear models.

These two generative models are closely related. Let $w_{ij} \sim \mathrm{Po}(\lambda_{ij})$ with $\lambda_{ij} = \exp(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$. Then Supplement B shows that

$$P(w_{ij} = 1 \mid \boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma(h(\boldsymbol{x}_i, \boldsymbol{x}_j)) + O(\lambda_{ij}^3) \tag{1}$$

and $P(w_{ij} \geq 2) = O(\lambda_{ij}^2)$, indicating that, for sufficiently small λ_{ij}, the Poisson model is well approximated by the Bernoulli model. Since these two models are not very different in this sense, we consider only the Poisson model in this paper.

We write the similarity function as

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) := g(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{f}(\boldsymbol{x}_j)), \tag{2}$$

where $\boldsymbol{f} : \mathbb{R}^p \to \mathbb{R}^K$ is a continuous function and $g : \mathbb{R}^{K \times K} \to \mathbb{R}$ is a symmetric continuous function. For two data vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, their feature vectors are defined as $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)$ and $\boldsymbol{y}_j = \boldsymbol{f}(\boldsymbol{x}_j)$, thus the similarity function is also written as $g(\boldsymbol{y}_i, \boldsymbol{y}_j)$. In particular, we consider a vector-valued neural network (NN) $\boldsymbol{y} = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x})$ for computing the feature vector, then $g(\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j))$ is especially called siamese network (Bromley et al., 1994) in neural network literature. The original form of siamese network uses the cosine similarity for g, but we can specify other types of similarity function. By specifying the inner product $g(\boldsymbol{y}, \boldsymbol{y}') = \langle \boldsymbol{y}, \boldsymbol{y}' \rangle$, the similarity function (2) becomes

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j) \rangle. \tag{3}$$

We call (3) as Inner Product Similarity (IPS) model. IPS commonly appears in a broad range of methods, such as DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015), node2vec (Grover and Leskovec, 2016), Variational Graph AutoEncoder (Kipf and Welling, 2016), and GraphSAGE (Hamilton et al., 2017). Multi-view extensions (Okuno et al., 2018) with views $d = 1, \ldots, D$, are easily obtained by preparing a neural network $\boldsymbol{f}_{\mathrm{NN}}^{(d)}$ for each view.

3 PD SIMILARITIES

In order to prove the approximation capability of IPS given in eq. (3), Okuno et al. (2018) incorporates the UAT for NN (Funahashi, 1989; Cybenko, 1989; Yarotsky, 2017; Telgarsky, 2017) into Mercer's theorem (Minh et al., 2006). In this section, we review their assertion that shows uniform convergence of IPS to any PD similarity. To show the result in Theorem 3.2, we first define a kernel and its positive-definiteness.

Definition 3.1 For some set $\mathcal{Y}$, a symmetric continuous function $g : \mathcal{Y}^2 \to \mathbb{R}$ is called a *kernel* on $\mathcal{Y}^2$.

Definition 3.2 A kernel g on $\mathcal{Y}^2$ is said to be *Positive Definite (PD)* if satisfying $\sum_{i=1}^n \sum_{j=1}^n c_i c_j g(\boldsymbol{y}_i, \boldsymbol{y}_j) \geq 0$ for arbitrary $c_1, c_2, \ldots, c_n \in \mathbb{R}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \mathcal{Y}$.

For instance, cosine similarity $g(\boldsymbol{y}, \boldsymbol{y}') := \langle \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2}, \frac{\boldsymbol{y}'}{\|\boldsymbol{y}'\|_2} \rangle$ is a PD kernel on $(\mathbb{R}^p \setminus \{\boldsymbol{0}\})^2$. Its PD-ness immediately follows from $\sum_{i=1}^n \sum_{j=c}^n c_i c_j g(\boldsymbol{y}_i, \boldsymbol{y}_j) = \| \sum_{i=1}^n c_i \frac{\boldsymbol{y}_i}{\|\boldsymbol{y}_i\|_2} \|_2^2 \geq 0$ for arbitrary $\{c_i\}_{i=1}^n \subset \mathbb{R}$ and $\{\boldsymbol{y}_i\}_{i=1}^n \subset \mathcal{Y}$. Also polynomial kernel, Gaussian kernel, and Laplacian kernel are PD (Berg et al., 1984).

Definition 3.3 A function $h(\boldsymbol{x}, \boldsymbol{x}') := g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}'))$ with a continuous function $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ and a kernel $g : \mathcal{Y}^2 \to \mathbb{R}$ is called a *similarity* on $\mathcal{X}^2$.

For a PD kernel g, the similarity h is also a PD kernel on $\mathcal{X}^2$, since $\sum_{i=1}^n \sum_{j=1}^n c_i c_j h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{i=1}^n \sum_{j=1}^n c_i c_j g(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{f}(\boldsymbol{x}_j)) \geq 0$.

Briefly speaking, a similarity h is used for measuring how similar two data vectors are, while a kernel g is used to compare feature vectors.

The following theorem (Minh et al., 2006) shows existence of a series expansion of any PD kernel, which has been utilized in kernel methods in machine learning (Hofmann et al., 2008).

Theorem 3.1 (Mercer's theorem) For some compact set $\mathcal{Y} \subset \mathbb{R}^{K^*}$, $K^* \in \mathbb{N}$, we consider a positive definite kernel $g_* : \mathcal{Y}^2 \to \mathbb{R}$. Then, there exist non-negative eigenvalues $\{\lambda_k\}_{k=1}^\infty$, $\lambda_1 \geq \lambda_2 \geq \cdots$, and continuous eigenfunctions $\{\phi_k\}_{k=1}^\infty$ such that

$$g_*(\boldsymbol{y}_*, \boldsymbol{y}'_*) = \sum_{k=1}^\infty \lambda_k \phi_k(\boldsymbol{y}_*) \phi_k(\boldsymbol{y}'_*), \tag{4}$$

for all $\boldsymbol{y}_*, \boldsymbol{y}'_* \in \mathcal{Y}$, where the series convergences absolutely for each $(\boldsymbol{y}_*, \boldsymbol{y}'_*)$ and uniformly for $\mathcal{Y}$.

Note that the condition (2) in Minh et al. (2006), i.e., $\int_{\mathcal{Y}} \int_{\mathcal{Y}} g_*(\boldsymbol{y}_*, \boldsymbol{y}'_*)\, d\boldsymbol{y}_*\, d\boldsymbol{y}'_* < \infty$, holds since g_* is continuous and $\mathcal{Y}$ is compact. The theorem can be extended

to closed set $\mathcal{Y}$, but we assume compactness for simplifying our argument.

It is obvious that IPS is always PD, because $\sum_{i=1}^n \sum_{j=1}^n c_i c_j \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j)\rangle = \| \sum_{i=1}^n c_i \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i)\|_2^2 \geq 0$. We would like to show the converse: IPS approximates any PD similarities. This is given by the Approximation Theorem (AT) for IPS below, which is Theorem 5.1 ($D = 1$) of Okuno et al. (2018). The idea is to incorporate the UAT for NN into Mercer's theorem (Theorem 3.1).

Theorem 3.2 (AT for IPS) For $\mathcal{X} = [-M, M]^p$, $M > 0$, and some compact set $\mathcal{Y} \subset \mathbb{R}^{K^*}$, $K^* \in \mathbb{N}$, we consider a continuous function $\boldsymbol{f}_* : \mathcal{X} \to \mathcal{Y}$ and a PD kernel $g_*^{(\mathrm{PD})} : \mathcal{Y}^2 \to \mathbb{R}$. Let $\sigma(\cdot)$ be ReLU or an activation function which is non-constant, continuous, bounded, and monotonically-increasing. Then, for arbitrary $\varepsilon > 0$, by specifying sufficiently large $K \in \mathbb{N}, m_f = m_f(K) \in \mathbb{N}$, there exist $\boldsymbol{A} \in \mathbb{R}^{K \times m_f}, \boldsymbol{B} \in \mathbb{R}^{m_f \times p}, \boldsymbol{c} \in \mathbb{R}^{m_f}$ such that

$$\left| g_*^{(\mathrm{PD})}\left(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')\right) - \left\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\right\rangle \right| < \varepsilon$$

for all $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2$, where $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\sigma}(\boldsymbol{B}\boldsymbol{x} + \boldsymbol{c})$ is a 1-hidden layer neural network with m_f hidden units and K outputs, and $\boldsymbol{\sigma}(\boldsymbol{x})$ is element-wise $\sigma(\cdot)$ function.

See Supplement A of Okuno et al. (2018) for the proof. It is based on the series expansion $g_*^{(\mathrm{PD})}(\boldsymbol{y}_*, \boldsymbol{y}'_*) = \sum_{k=1}^\infty \lambda_k \phi_k(\boldsymbol{y}_*)\phi_k(\boldsymbol{y}'_*)$ of Mercer's theorem (Theorem 3.1) for arbitrary PD kernel $g_*^{(\mathrm{PD})}$. This expansion indicates with a vector-valued function $\tilde{\boldsymbol{\phi}}_K(\boldsymbol{x}) := (\lambda_1^{1/2}\phi_1(\boldsymbol{f}_*(\boldsymbol{x})), \ldots, \lambda_K^{1/2}\phi_K(\boldsymbol{f}_*(\boldsymbol{x})))$ that

$$\langle \tilde{\boldsymbol{\phi}}_K(\boldsymbol{x}), \tilde{\boldsymbol{\phi}}_K(\boldsymbol{x}')\rangle \to g_*^{(\mathrm{PD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')),\ K \to \infty,$$

for all $\boldsymbol{x}, \boldsymbol{x}'$. Considering a vector-valued NN $\boldsymbol{f}_{\mathrm{NN}} : \mathbb{R}^p \to \mathbb{R}^K$ that approximates $\tilde{\boldsymbol{\phi}}_K$, the IPS $\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\rangle \approx \langle \tilde{\boldsymbol{\phi}}_K(\boldsymbol{x}), \tilde{\boldsymbol{\phi}}_K(\boldsymbol{x}')\rangle$ converges to $g_*^{(\mathrm{PD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}'))$ as $K \to \infty$, thus proving the assertion. In addition to the uniform convergence shown in Theorem 3.2, the approximation error rate will be evaluated in Section 5.

Unlike Mercer's theorem which indicates only the existence of the feature map $\tilde{\boldsymbol{\phi}}_K$, Theorem 3.2 shows that a neural network $\boldsymbol{f}_{\mathrm{NN}} : \mathbb{R}^p \to \mathbb{R}^K$ can be implemented so that the IPS $\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\rangle$ eventually approximates the PD similarity $g_*^{(\mathrm{PD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}'))$ arbitrary well.

Note that Theorem 3.2 is AT for IPS which shows only the existence of NNs with required accuracy. Although we do not go further in this paper, consistency of the maximum likelihood estimation implemented as SGD is discussed in Section 5.2 and Supplement B of Okuno et al. (2018) for showing that IPS actually learns any PD similarities by increasing n.

4 CPD SIMILARITIES

Theorem 3.2 shows that IPS approximates any PD similarities arbitrary well. However, similarities in general are not always PD. To deal with non-PD similarities, we consider a class of similarities based on Conditionally PD (CPD) kernels (Berg et al., 1984; Schölkopf, 2001) which includes PD kernels as special cases. We then extend IPS to approximate CPD similarities.

Someone may wonder why only similarities based on inner product are considered in this paper. In fact, it is obvious that a real-valued NN $f_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{x}')$ with sufficiently many hidden units approximates any similarity $h(\boldsymbol{x}, \boldsymbol{x}')$ arbitrary well. This is an immediate consequence of the UAT directly applied to $f_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{x}')$. Therefore, considering the form $\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\rangle$ or its extension just makes the problem harder. Our motivation in this paper is that we would like to utilize the feature vector $\boldsymbol{y} = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x})$ with nice properties such as "linguistic regularities" which may follow from the constraint of the inner product.

The remaining of this section is organized as follows. In Section 4.1, we point out the fundamental limitation of IPS to approximate a non-PD similarity. In Section 4.2, we define CPD kernels with some examples. In Section 4.3, we propose a novel Shifted IPS (SIPS), by extending the IPS. In Section 4.4, we give interpretations of SIPS and its simpler variant C-SIPS. In Section 4.5, we prove that SIPS approximates CPD similarities arbitrary well.

4.1 Fundamental Limitation of IPS

Let us consider the negative squared distance (NSD) $g(\boldsymbol{y}, \boldsymbol{y}') = -\|\boldsymbol{y} - \boldsymbol{y}'\|_2^2$ and the identity map $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}$. Then the similarity function

$$h(\boldsymbol{x}, \boldsymbol{x}') = g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = -\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$$

defined on $\mathbb{R}^p \times \mathbb{R}^p$ is not PD but CPD, which is defined later in Section 4.2. Regarding the NSD similarity, Proposition 4.1 shows a strictly positive lower bound of approximation error for IPS.

Proposition 4.1 For all $M > 0, p, K \in \mathbb{N}$, and a set of all $\mathbb{R}^K$-valued continuous functions $\mathfrak{S}(K)$, we have

$$\inf_{\boldsymbol{f} \in \mathfrak{S}(K)} \frac{1}{(2M)^{2p}} \int_{[-M,M]^p} \int_{[-M,M]^p} \left| -\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 - \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')\rangle \right| \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \geq \frac{2pM^2}{3}.$$

The proof is given in Supplement C.1.

Since $\mathfrak{S}(K)$ includes neural networks, Proposition 4.1 indicates that IPS does not approximate NSD similarity arbitrary well, even if NN has a huge amount of hidden units with sufficiently large output dimension.

4.2 CPD Kernels and Similarities

Here, we introduce similarities based on Conditionally PD (CPD) kernels (Berg et al., 1984; Schölkopf, 2001) in order to consider non-PD similarities which IPS does not approximate arbitrary well. We first define CPD kernels.

Definition 4.1 A kernel g on $\mathcal{Y}^2$ is called *Conditionally PD (CPD)* if $\sum_{i=1}^n \sum_{j=1}^n c_i c_j g(\boldsymbol{y}_i, \boldsymbol{y}_j) \geq 0$ holds for arbitrary $c_1, c_2, \ldots, c_n \in \mathbb{R}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \mathcal{Y}$ with the constraint $\sum_{i=1}^n c_i = 0$.

The difference between the definitions of CPD and PD kernels is whether it imposes the constraint $\sum_{i=1}^n c_i = 0$ or not. According to these definitions, CPD kernels include PD kernels as special cases. For a CPD kernel g, the similarity h is also a CPD kernel on $\mathcal{X}^2$.

A simple example of CPD kernel is $g(\boldsymbol{y}, \boldsymbol{y}') = -\|\boldsymbol{y} - \boldsymbol{y}'\|_2^\alpha$ for $0 < \alpha \leq 2$ defined on $\mathbb{R}^K \times \mathbb{R}^K$. Other examples are $-(\sin(y - y'))^2$ and $-\mathbf{1}_{(0,\infty)}(y + y')$ on $\mathbb{R} \times \mathbb{R}$. CPD-ness is a well-established concept with interesting properties (Berg et al., 1984): For any function $u(\cdot)$, $g(\boldsymbol{y}, \boldsymbol{y}') = u(\boldsymbol{y}) + u(\boldsymbol{y}')$ is CPD. Constants are CPD. The sum of two CPD kernels is also CPD. For CPD kernels g with $g(\boldsymbol{y}, \boldsymbol{y}') \leq 0$, CPD-ness holds for $-(-g)^\alpha$ ($\alpha \in (0, 1]$) and $-\log(1 - g)$.

Example 4.1 (Poincaré distance) For open unit ball $B^K := \{\boldsymbol{y} \in \mathbb{R}^K \mid \|\boldsymbol{y}\|_2 < 1\}$, we define a distance between $\boldsymbol{y}, \boldsymbol{y}' \in B^K$ as

$$d_{\text{Poincaré}}(\boldsymbol{y}, \boldsymbol{y}') := \cosh^{-1}\left(1 + 2\frac{\|\boldsymbol{y} - \boldsymbol{y}'\|_2^2}{(1 - \|\boldsymbol{y}\|_2^2)(1 - \|\boldsymbol{y}'\|_2^2)}\right), \quad (5)$$

where $\cosh^{-1}(z) = \log(z + \sqrt{z^2 - 1})$. Considering the generative model of Section 2 with 1-hot data vectors, Poincaré embedding (Nickel and Kiela, 2017) learns parameters $\boldsymbol{y}_i$, $i = 1, \ldots, n$, by fitting $\sigma(-d_{\text{Poincaré}}(\boldsymbol{y}_i, \boldsymbol{y}_j))$ to the observed $w_{ij} \in \{0, 1\}$. Lorentz embedding (Nickel and Kiela, 2018) reformulate Poincaré embedding with a specific variable transformation, that enables more efficient computation.

Interestingly, negative Poincaré distance is proved to be CPD in Faraut and Harzallah (1974, Corollary 7.4).

Proposition 4.2 $-d_{\text{Poincaré}}$ is CPD on $B^K \times B^K$.

$-d_{\text{Poincaré}}$ is strictly CPD in the sense that $-d_{\text{Poincaré}}$ is not PD. A counter-example of PD-ness is, for example, $n = 2, K = 2, c_1 = c_2 = 1, \boldsymbol{y}_1 = (1/2, 1/2), \boldsymbol{y}_2 = (0, 0) \in B^2$.

Another interesting example of CPD kernels is negative Wasserstein distance.

Example 4.2 (Wasserstein distance) Let $\boldsymbol{Z}$ be a metric space endowed with a metric d_Z, which we call as "ground distance". For $q \geq 1$, let $\mathcal{Y}$ be the space of all measures μ on $\boldsymbol{Z}$ satisfying $\int_{\boldsymbol{Z}} d_Z(\boldsymbol{z}, \boldsymbol{z}_0)^q \mathrm{d}\mu(\boldsymbol{z}) < \infty$ for some $\boldsymbol{z}_0 \in \boldsymbol{Z}$. The q-Wasserstein distance between $\boldsymbol{y}, \boldsymbol{y}' \in \mathcal{Y}$ is defined as

$$d_W^{(q)}(\boldsymbol{y}, \boldsymbol{y}') := \left(\inf_{\pi \in \Pi(\boldsymbol{y}, \boldsymbol{y}')} \iint_{\boldsymbol{Z} \times \boldsymbol{Z}} d_Z(\boldsymbol{z}, \boldsymbol{z}')^q \mathrm{d}\pi(\boldsymbol{z}, \boldsymbol{z}')\right)^{1/q}.$$

Here, $\Pi(\boldsymbol{y}, \boldsymbol{y}')$ is the set of joint probability measures on $\boldsymbol{Z} \times \boldsymbol{Z}$ having marginals $\boldsymbol{y}, \boldsymbol{y}'$. Wasserstein distance is used for a broad range of methods, such as Generative Adversarial Networks (Arjovsky et al., 2017) and AutoEncoder (Tolstikhin et al., 2018).

Some cases of negative Wasserstein distance are proved to be CPD.

Proposition 4.3 $-d_W^{(1)}$ is CPD on $\mathcal{Y}^2$ if $-d_Z$ is CPD on $\boldsymbol{Z}^2$. $-d_W^{(2)}$ is CPD on $\mathcal{Y}^2$ if $\boldsymbol{Z}$ is a subset of $\mathbb{R}$.

$-d_W^{(1)}$ is known as the negative earth mover's distance, and its CPD-ness is discussed in Gardner et al. (2017). The CPD-ness of a special case of $-d_W^{(2)}$ is shown in Kolouri et al. (2016) Corollary 1. However, we note that negative Wasserstein distance, in general, is not necessarily CPD. As Proposition 4.3 states, $\boldsymbol{Z}$ is required to be a subset of $\mathbb{R}$ when considering $q > 1$.

4.3 Proposed Models

For approximating CPD similarities, we propose a novel similarity model

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{f}_{\text{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\text{NN}}(\boldsymbol{x}_j) \rangle + u_{\text{NN}}(\boldsymbol{x}_i) + u_{\text{NN}}(\boldsymbol{x}_j), \quad (6)$$

where $\boldsymbol{f}_{\text{NN}} : \mathbb{R}^p \to \mathbb{R}^K$ and $u_{\text{NN}} : \mathbb{R}^p \to \mathbb{R}$ are vector-valued and real-valued NNs, respectively. We call (6) as Shifted IPS (SIPS) model, because the IPS $\langle \boldsymbol{f}_{\text{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\text{NN}}(\boldsymbol{x}_j) \rangle$ given in (3) is shifted by the offset $u_{\text{NN}}(\boldsymbol{x}_i) + u_{\text{NN}}(\boldsymbol{x}_j)$. For illustrating how SIPS expresses CPD similarities, let us consider the NSD discussed in Section 4.1:

$$-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 = \langle \sqrt{2}\boldsymbol{x}_i, \sqrt{2}\boldsymbol{x}_j \rangle - \|\boldsymbol{x}_i\|_2^2 - \|\boldsymbol{x}_j\|_2^2$$

is expressed by SIPS with $\boldsymbol{f}_{\text{NN}}(\boldsymbol{x}) = \sqrt{2}\boldsymbol{x}$ and $u_{\text{NN}}(\boldsymbol{x}) = -\|\boldsymbol{x}\|_2^2$. Later, we show in Theorem 4.1 that SIPS approximates any CPD similarities arbitrary well.

We also consider a simplified version of SIPS. By assuming $u_{\mathrm{NN}}(\boldsymbol{x}) = -\gamma/2$ for all $\boldsymbol{x}$, SIPS reduces to

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j) \rangle - \gamma, \tag{7}$$

where $\gamma \in \mathbb{R}$ is a parameter to be estimated. We call (7) as Constantly-Shifted IPS (C-SIPS) model.

If we have no attributes, we use 1-hot vectors for $\boldsymbol{x}_i$ in $\mathbb{R}^n$ instead, and $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i) = \boldsymbol{y}_i \in \mathbb{R}^K$, $u_{\mathrm{NN}}(\boldsymbol{x}_i) = u_i \in \mathbb{R}$ are model parameters. Then SIPS reduces to the matrix decomposition model with biases

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle + u_i + u_j. \tag{8}$$

This model is widely used for recommender systems (Koren et al., 2009) and word embedding such as GloVe (Pennington et al., 2014), and SIPS is considered as its generalization.

4.4 Interpretation of SIPS and C-SIPS

Here we illustrate the interpretation of the proposed models by returning back to the setting in Section 2. We consider a simple generative model of independent Poisson distribution with mean parameter $E(w_{ij}) = \exp(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$. Then SIPS gives a generative model

$$w_{ij} \overset{\text{indep.}}{\sim} \mathrm{Po}\Big(\beta(\boldsymbol{x}_i)\beta(\boldsymbol{x}_j) \exp(\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j) \rangle)\Big), \tag{9}$$

where $\beta(\boldsymbol{x}) := \exp(u_{\mathrm{NN}}(\boldsymbol{x})) > 0$. Since $\beta(\boldsymbol{x})$ can be regarded as the "importance weight" of data vector $\boldsymbol{x}$, SIPS naturally incorporates the weight function $\beta(\boldsymbol{x})$ to probabilistic models used in a broad range of existing methods. Similarly, C-SIPS gives a generative model

$$w_{ij} \overset{\text{indep.}}{\sim} \mathrm{Po}\Big(\alpha \exp(\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j) \rangle)\Big), \tag{10}$$

where $\alpha := \exp(-\gamma) > 0$ regulates the sparseness of $\{w_{ij}\}$. The generative model (10) is already proposed as 1-view PMvGE (Okuno et al., 2018).

It was shown in Supplement C of Okuno et al. (2018) that PMvGE (based on C-SIPS) approximates CDMCA when w_{ij} is replaced by δ_{ij} in the constraint (8) therein, and this result can be extended so that PMvGE with SIPS approximates the original CDMCA using w_{ij} in the constraint.

4.5 Approximation Theorems

It is obvious that SIPS is always CPD, because $\sum_{i=1}^n \sum_{j=1}^n c_i c_j \big(\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_j) \rangle + u_{\mathrm{NN}}(\boldsymbol{x}_i) + u_{\mathrm{NN}}(\boldsymbol{x}_j)\big) = \| \sum_{i=1}^n c_i \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i) \|_2^2 + 2\big(\sum_{i=1} c_i\big)\big(\sum_{j=1} c_j u_{\mathrm{NN}}(\boldsymbol{x}_j)\big) \geq 0$ for any c_i's with $\sum_{i=1}^n c_i = 0$. We would like to show the converse: SIPS approximates any CPD similarities, and thus it overcomes the fundamental limitation of IPS. This is given in Theorem 4.1 below, by extending Theorem 3.2 of IPS to SIPS. Theorem 4.2 also proves that C-SIPS given in eq. (7) approximates CPD similarities in a weaker sense.

Theorem 4.1 (AT for SIPS) For $\mathcal{X} = [-M, M]^p$, $M > 0$, and some compact set $\mathcal{Y} \subset \mathbb{R}^{K^*}$, $K^* \in \mathbb{N}$, we consider a continuous function $\boldsymbol{f}_* : \mathcal{X} \to \mathcal{Y}$ and a CPD kernel $g_*^{(\mathrm{CPD})} : \mathcal{Y}^2 \to \mathbb{R}$. Let $\sigma(\cdot)$ be ReLU or an activation function which is non-constant, continuous, bounded, and monotonically-increasing. Then, for arbitrary $\varepsilon > 0$, by specifying sufficiently large $K \in \mathbb{N}, m_f = m_f(K) \in \mathbb{N}, m_u \in \mathbb{N}$, there exist $\boldsymbol{A} \in \mathbb{R}^{K \times m_f}, \boldsymbol{B} \in \mathbb{R}^{m_f \times p}, \boldsymbol{c} \in \mathbb{R}^{m_f}, \boldsymbol{e} \in \mathbb{R}^{m_u}, \boldsymbol{F} \in \mathbb{R}^{m_u \times p}, \boldsymbol{o} \in \mathbb{R}^{m_u}$ such that

$$\Big| g_*^{(\mathrm{CPD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) - \big(\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle + u_{\mathrm{NN}}(\boldsymbol{x}) + u_{\mathrm{NN}}(\boldsymbol{x}')\big) \Big| < \varepsilon$$

for all $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2$, where $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\sigma}(\boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}) \in \mathbb{R}^K$ and $u_{\mathrm{NN}}(\boldsymbol{x}) = \langle \boldsymbol{e}, \boldsymbol{\sigma}(\boldsymbol{F}\boldsymbol{x} + \boldsymbol{o}) \rangle \in \mathbb{R}$ are one-hidden layer neural networks with m_f and m_u hidden units, respectively, and $\boldsymbol{\sigma}(\boldsymbol{x})$ is element-wise $\sigma(\cdot)$ function.

The proof is in Supplement C.2. It stands on Lemma 2.1 in Berg et al. (1984), which shows the equivalence of CPD-ness of $g_*^{(\mathrm{CPD})}(\boldsymbol{y}, \boldsymbol{y}')$ and PD-ness of

$$\begin{aligned} g_0(\boldsymbol{y}, \boldsymbol{y}') :=& g_*^{(\mathrm{CPD})}(\boldsymbol{y}, \boldsymbol{y}') + g_*^{(\mathrm{CPD})}(\boldsymbol{y}_0, \boldsymbol{y}_0) \\ & - g_*^{(\mathrm{CPD})}(\boldsymbol{y}, \boldsymbol{y}_0) - g_*^{(\mathrm{CPD})}(\boldsymbol{y}', \boldsymbol{y}_0) \end{aligned} \tag{11}$$

for any fixed $\boldsymbol{y}_0 \in \mathcal{Y}$. Using g_0 and $h_*(\boldsymbol{x}) := g_*^{(\mathrm{CPD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{y}_0) - \frac{1}{2} g_*^{(\mathrm{CPD})}(\boldsymbol{y}_0, \boldsymbol{y}_0)$, we write

$$\begin{aligned} & g_*^{(\mathrm{CPD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) \\ & \quad = g_0(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) + h_*(\boldsymbol{x}) + h_*(\boldsymbol{x}'). \end{aligned} \tag{12}$$

AT for IPS shows that $\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$ approximates $g_0(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}'))$ arbitrary well, and UAT for NN shows that $u_{\mathrm{NN}}(\boldsymbol{x})$ approximates $h_*(\boldsymbol{x})$ arbitrary well, thus proving the theorem.

Theorem 4.2 (AT for C-SIPS) Symbols and assumptions are the same as those of Theorem 4.1. For arbitrary $\varepsilon > 0$, by specifying sufficiently large $K \in \mathbb{N}$, $m_f = m_f(K) \in \mathbb{N}$, $r > 0$, there exist $\boldsymbol{A} \in \mathbb{R}^{K \times m_f}$, $\boldsymbol{B} \in \mathbb{R}^{m_f \times p}$, $\boldsymbol{c} \in \mathbb{R}^{m_f}$, $\gamma = O(r^2)$ such that

$$\Big| g_*^{(\mathrm{CPD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) - \big(\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle - \gamma\big) \Big| < \varepsilon + O(r^{-2})$$

for all $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2$, where $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\sigma}(\boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}) \in \mathbb{R}^K$ is a one-hidden layer neural network with m_f hidden units.

The proof is in Supplement C.3.

There is an additional error term of $O(r^{-2})$ in Theorem 4.2. A large r will reduce the error, but then large $\gamma = O(r^2)$ value may lead to unstable computation for finding an optimal NN. Conversely, a small r increases the upper bound of the approximation error . Thus, if available, we prefer SIPS in terms of both computational stability and small approximation error.

5 APPROXIMATION ERROR RATE

Thus far, we showed universal approximation capabilities of IPS and SIPS in Theorems 3.2 and 4.1. In this section, we evaluate error rates for these approximation theorems, by assuming some additional conditions. They are used for employing the theorems for eigenvalue decay rate of PD kernels (Cobos and Kühn, 1990, Theorem 4) and approximation error rate for NNs (Yarotsky, 2018).

Conditions on the similarity function: We consider the following conditions on the function $\boldsymbol{f}_*$ and the kernel g_* for the underlying true similarity $g_*(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}'))$.

(C-1) Eigenfunctions $\{\phi_k(\boldsymbol{y})\}_{k=1}^{\infty}$ of $g_*(\boldsymbol{y}, \boldsymbol{y}')$ defined in Theorem 3.1 are continuously differentiable, i.e., C^1, and uniformly bounded in the sense of $\sup_{k\in\mathbb{N},\boldsymbol{y}\in\mathcal{Y}} |\phi_k(\boldsymbol{y})| < \infty$ and $\sup_{k\in\mathbb{N},\boldsymbol{y}\in\mathcal{Y}} \lambda_k \|\partial\phi_k(\boldsymbol{y})/\partial\boldsymbol{y}\|_2^2 < \infty$.

(C-2) $g_*(\boldsymbol{y}, \boldsymbol{y}')$ is C^1.

(C-3) $\boldsymbol{f}_*$ is C^1.

NN architecture: As we considered in Theorems 3.2 and 4.1, we employ a set of K-dimensional vector-valued NNs for $\mathcal{X} = [-M, M]^p$. The activation function is confined to ReLU $\sigma(z) := \max\{0, z\}$. Let $L \in \mathbb{N}$ be the number of hidden layers, i.e., depth, of the NN, and let $W \in \mathbb{N}$ be the total number of weights in the NN. For example, $L = 1$ and W is the number of elements in $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{c}$ in Theorems 3.2. Instead of the fixed network architecture, here we consider a class of architectures specified by W with a specific growing rate of the depth L. For $0 \le \alpha \le 1$, define a set of all possible NNs with the constraint as

$$\mathfrak{S}_\alpha(W, K) := \{\boldsymbol{f}_{\mathrm{NN}} : \mathcal{X} \to \mathbb{R}^K \mid \boldsymbol{f}_{\mathrm{NN}} \text{ has } W \text{ weights with depth } L = O((W/K)^\alpha)\}, \quad (13)$$

where $W/K \to \infty$. This is a simple extension of the case $K = 1$ considered in Yarotsky (2018), where $\alpha = 0$ and $\alpha = 1$ correspond to constant-depth shallow NNs and constant-width deep NNs, respectively.

Theorem 5.1 (Approx. error rate for IPS) Symbols and assumptions are the same as those of Theorem 3.2 except for the additional conditions (C-1) and (C-2) for $g_*^{(\mathrm{PD})}$ and (C-3) for $\boldsymbol{f}_*$. Instead of the 1-hidden layer NN, we consider the set of NNs $\boldsymbol{f}_{\mathrm{NN}} \in \mathfrak{S}_\alpha(W_f, K)$ for $W_f \in \mathbb{N}$. Then the approximation error rate of IPS is given by

$$\inf_{\boldsymbol{f}_{\mathrm{NN}}\in\mathfrak{S}_\alpha(W_f,K)} \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \left| g_*^{(\mathrm{PD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) - \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\rangle \right| = O\Big(K^{-\frac{1}{K^*}} + K^{\frac{1}{2}+\frac{1+\alpha}{p}} W_f^{-\frac{1+\alpha}{p}}\Big). \quad (14)$$

Proof is in Supplement D.3. In the above result, $O(K^{-1/K^*})$ is attributed to truncating (4) at K terms in Mercer's theorem and $O(K^{\frac{1}{2}+\frac{1+\alpha}{p}} W_f^{-\frac{1+\alpha}{p}})$ is attributed to the approximation error of $\boldsymbol{f}_{\mathrm{NN}}$. The error rate for SIPS is similarly evaluated, but it includes the error rate for newly incorporated NN u_{NN}.

Theorem 5.2 (Approx. error rate for SIPS) Symbols and assumptions are the same as those of Theorem 4.1 except for the additional conditions (C-1) for g_0 of (11), (C-2) for $g_*^{(\mathrm{CPD})}$, and (C-3) for $\boldsymbol{f}_*$. Instead of the 1-hidden layer NN, we consider the set of NNs $\boldsymbol{f}_{\mathrm{NN}} \in \mathfrak{S}_\alpha(W_f, K)$ for $W_f \in \mathbb{N}$ and $u_{\mathrm{NN}} \in \mathfrak{S}_\alpha(W_u, 1)$ for $W_u \in \mathbb{N}$. Then the approximation error rate of SIPS is given by

$$\inf_{\substack{\boldsymbol{f}_{\mathrm{NN}}\in\mathfrak{S}_\alpha(W_f,K)\\ u_{\mathrm{NN}}\in\mathfrak{S}_\alpha(W_u,1)}} \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \Big| g_*^{(\mathrm{CPD})}(\boldsymbol{f}_*(\boldsymbol{x}), \boldsymbol{f}_*(\boldsymbol{x}')) - (\langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')\rangle + u_{\mathrm{NN}}(\boldsymbol{x}) + u_{\mathrm{NN}}(\boldsymbol{x}')) \Big| = O\Big(K^{-\frac{1}{K^*}} + K^{\frac{1}{2}+\frac{1+\alpha}{p}} W_f^{-\frac{1+\alpha}{p}} + W_u^{-\frac{1+\alpha}{p}}\Big). \quad (15)$$

Proof is in Supplement D.4.

In Theorems 5.1 and 5.2, the commonly appearing term $O(K^{-1/K^*})$ may be a bottleneck when K^* is very large. We may specify $W_f = O(K^{1+\frac{p}{1+\alpha}(\frac{1}{K^*}+\frac{1}{2})}) \approx O(K^{1+\frac{p}{2(1+\alpha)}})$ and $W_u = O(K^{\frac{p}{(1+\alpha)K^*}})$ so that the overall approximation error rate is $O(K^{-1/K^*})$.

6 EXPERIMENTS

In this section, we evaluate similarity models (NSD, Poincaré, IPS, SIPS) on two real-world datasets: Co-authorship network dataset (Prado et al., 2013) in Section 6.1 and WordNet dataset (Miller, 1995) in Section 6.2. Details of experiments are shown in Supplement A.

Table 1: Experiments on Co-authorship network and WordNet evaluated by ROC-AUC score (higher is better). Sample average and the standard deviation of 5 runs are shown.

	Co-authorship network				WordNet			
	$K=2$	$K=5$	$K=10$	$K=20$	$K=2$	$K=5$	$K=10$	$K=20$
NSD	**0.8220** $\pm$ 0.010	0.8655 $\pm$ 0.014	0.8771 $\pm$ 0.012	0.8651 $\pm$ 0.033	0.7924 $\pm$ 0.0072	0.8997 $\pm$ 0.0009	0.9569 $\pm$ 0.0005	0.9836 $\pm$ 0.0001
Poincaré	0.7071 $\pm$ 0.021	0.8738 $\pm$ 0.001	0.8822 $\pm$ 0.001	0.8835 $\pm$ 0.001	0.8401 $\pm$ 0.0073	**0.9792** $\pm$ 0.0006	**0.9866** $\pm$ 0.0003	0.9851 $\pm$ 0.0002
IPS	0.7802 $\pm$ 0.005	0.8830 $\pm$ 0.001	0.8955 $\pm$ 0.001	0.8956 $\pm$ 0.001	0.7245 $\pm$ 0.0056	0.7604 $\pm$ 0.0055	0.7688 $\pm$ 0.0023	0.7918 $\pm$ 0.0018
SIPS	0.7811 $\pm$ 0.001	**0.8853** $\pm$ 0.001	**0.8964** $\pm$ 0.002	**0.8974** $\pm$ 0.001	**0.9632** $\pm$ 0.0008	0.9766 $\pm$ 0.0006	0.9825 $\pm$ 0.0005	**0.9865** $\pm$ 0.0004

6.1 Experiment on Co-authorship Network

Co-authorship network dataset (Prado et al., 2013) consists of $n = 42,252$ nodes and $210,320$ undirected edges. Each node v_i represents an author, and data vector $\boldsymbol{x}_i \in \mathbb{R}^{33}$ ($p = 33$) represents the numbers of publications in 29 conferences/journals and 4 microscopic topological properties describing the direct neighborhood of the node. Adjacency matrix $\boldsymbol{W} = (w_{ij}) \in \{0,1\}^{n\times n}$ represents the co-authorship relations: $w_{ij} = w_{ji} = 1$ if v_i and v_j have any co-authorship relation, and $w_{ij} = w_{ji} = 0$ otherwise.

Preprocessing: We split authors into training set (90%) and test set (10%). Co-authorship relations for the test set are treated as unseen. We use 10% of the training set as validation set.

Author feature vectors: Using the data vectors for authors $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^p$, feature vectors $\{\boldsymbol{y}_i\}_{i=1}^n \subset \mathbb{R}^K$ are computed via a neural network $\boldsymbol{y}_i = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i)$. We employ 1-hidden layer perceptron with $10,000$ hidden units and ReLU activation function. For implementing SIPS, one of the K output units of $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i)$ is used for the bias term $u_i = u_{\mathrm{NN}}(\boldsymbol{x}_i)$, so actually the feature vector is computed as $(\boldsymbol{y}_i, u_i) = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}_i) \in \mathbb{R}^K$ with $\boldsymbol{y}_i \in \mathbb{R}^{K-1}$. Model parameters are trained by maximizing the objective

$$\sum_{1\le i\ne j\le n} w_{ij} \log \frac{\exp(h(\boldsymbol{x}_i, \boldsymbol{x}_j))}{\sum_{k\in\mathcal{S}_r(\mathcal{N}_{ij})} \exp(h(\boldsymbol{x}_i, \boldsymbol{x}_k))}, \qquad (16)$$

where $h : \mathcal{X}^2 \to \mathbb{R}$ is a similarity function and $\mathcal{S}_r(\mathcal{N}_{ij})$ is a subset that consists of $r = 10$ entries randomly sampled from $\mathcal{N}_{ij} := \{k | 1 \le k \le n, w_{ik} = 0\} \cup \{j\}$.

Similarity models: (i) NSD uses $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2$. (ii) Poincaré embedding (Nickel and Kiela, 2017) uses $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = -d_{\mathrm{Poincare}}(\boldsymbol{y}_i, \boldsymbol{y}_j)$ defined in (5). (iii) IPS uses $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$. (iv) SIPS uses $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle + u_i + u_j$.

Results: Models are evaluated by ROC-AUC (Bradley, 1997) on the task of predicting unseen co-authorship relations. ROC-AUC scores are shown on the left-hand side of Table 1. Although NSD demonstrates a good performance for $K = 2$, SIPS outperforms the other methods for $K = 5, 10, 20$.

6.2 Experiment on WordNet

WordNet dataset (Miller, 1995) is a lexical resource that contains a variety of nouns and their relations. For instance, a noun "mammal" represents a superordinate concept of a noun "dog", thus these two words have hypernymy relation. We preprocess WordNet dataset in the same way as Nickel and Kiela (2017). We used a subset of the graph with $n = 4027$ nouns and $53,905$ hierarchical relations by extracting all the nouns subordinate to "animal". Each noun is represented by v_i, and relations are represented by adjacency matrix $\boldsymbol{W} = (w_{ij}) \in \{0,1\}^{n\times n}$, where $w_{ij} = w_{ji}$ represents any hypernymy relation, including transitive closure, between v_i and v_j.

Word feature vectors: Since nodes have no attributes, data vectors are formally treated as 1-hot vectors in $\mathbb{R}^n$. Instead of learning neural networks, the distributed representations $\{\boldsymbol{y}_i\}_{i=1}^n \subset \mathbb{R}^K$ of words are learned by maximizing the objective (16) with $r = 20$ for NSD, Poincaré and IPS, and $\{(\boldsymbol{y}_i, u_i)\}_{i=1}^n \subset \mathbb{R}^K$ are learned for SIPS. Similarity models are the same as those of Section 6.1.

Results: Models are evaluated by ROC-AUC of reconstruction error on the task of reconstructing hierarchical relations in the same way as Nickel and Kiela (2017). ROC-AUC score is listed on the right-hand side of Table 1. SIPS outperforms the other methods for $K = 2, 20$, and it is competitive to Poincaré embedding for $K = 5, 10$.

7 CONCLUSION

We proposed a novel shifted inner-product similarity (SIPS) for graph embedding (GE), that is theoretically proved to approximate arbitrary conditionally positive-definite (CPD) similarities including negative Poincaré distance. Since SIPS automatically approximates a wide variety of similarities, SIPS alleviates the need for configuring the similarity function of GE.

Acknowledgement

This work was partially supported by JSPS KAKENHI grant 16H02789 to HS and 17J03623 to AO.

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep Canonical Correlation Analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1247–1255.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

Berg, C., Christensen, J., and Ressel, P. (1984). *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Graduate Texts in Mathematics. Springer New York.

Bradley, A. P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(7):1145–1159.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Cai, H., Zheng, V. W., and Chang, K. (2018). A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*.

Cobos, F. and Kühn, T. (1990). Eigenvalues of integral operators with positive definite kernels satisfying integrated hölder conditions over metric compacta. *Journal of Approximation Theory*, 63(1):39–55.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

Dai, Q., Li, Q., Tang, J., and Wang, D. (2018). Adversarial Network Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Faraut, J. and Harzallah, K. (1974). Distances hilbertiennes invariantes sur un espace homogène. In *Annales de l'institut Fourier*, volume 24, pages 171–217. Association des Annales de l'Institut Fourier.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192.

Gardner, A., Duncan, C. A., Kanno, J., and Selmic, R. R. (2017). On the Definiteness of Earth Mover's Distance and Its Relation to Set Intersection. *IEEE Transactions on Cybernetics*.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data mining (SIGKDD)*, pages 855–864. ACM.

Hamilton, W. L., Ying, Z., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems (NIPS)*, pages 1025–1035.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv preprint arXiv:1502.01852*.

He, X. and Niyogi, P. (2004). Locality Preserving Projections. In *Advances in Neural Information Processing Systems (NIPS)*, pages 153–160.

Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220.

Huang, Z., Shan, S., Zhang, H., Lao, S., and Chen, X. (2012). Cross-view Graph Embedding. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 770–781.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N. and Welling, M. (2016). Variational Graph Auto-Encoders. *NIPS Workshop*.

Kolouri, S., Zou, Y., and Rohde, G. K. (2016). Sliced Wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5258–5267.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.

Levy, O. and Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Minh, H. Q., Niyogi, P., and Yao, Y. (2006). Mercer's Theorem, Feature Maps, and Smoothing. In *International Conference on Computational Learning Theory (COLT)*, pages 154–168. Springer.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6341–6350.

Nickel, M. and Kiela, D. (2018). Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Okuno, A., Hada, T., and Shimodaira, H. (2018). A probabilistic framework for multi-view feature learning with many-to-many associations via neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3885–3894.

Okuno, A. and Shimodaira, H. (2018). On representation power of neural network-based graph embedding and beyond. In *ICML Workshop (TADGM)*.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: Online Learning of Social Representations. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data mining (SIGKDD)*, pages 701–710. ACM.

Prado, A., Plantevit, M., Robardet, C., and Boulicaut, J.-F. (2013). Mining graph topological patterns: Finding covariations among vertex descriptors. *IEEE Transactions on Knowledge and Data Engineering*, 25(9):2090–2104.

Sarkar, R. (2011). Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer.

Schölkopf, B. (2001). The kernel trick for distances. In *Advances in Neural Information Processing Systems (NIPS)*, pages 301–307.

Shimodaira, H. (2016). Cross-validation of matching correlation analysis by resampling matching weights. *Neural Networks*, 75:126–140.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). LINE: Large-scale Information Network Embedding. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 1067–1077.

Telgarsky, M. (2017). Neural networks and rational functions. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2018). Wasserstein Auto-Encoders. In *Proceedings of the International Conference on Representation Learning (ICLR)*.

Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S. (2007). Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(1):40–51.

Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114.

Yarotsky, D. (2018). Optimal approximation of continuous functions by very deep ReLU networks. *Proceedings of Machine Learning Research*, 75:639–649. The 31st Annual Conference on Learning Theory (COLT 2018).

Zhanga, D., Yinb, J., Zhuc, X., and Zhanga, C. (2017). User profile preserving social network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.