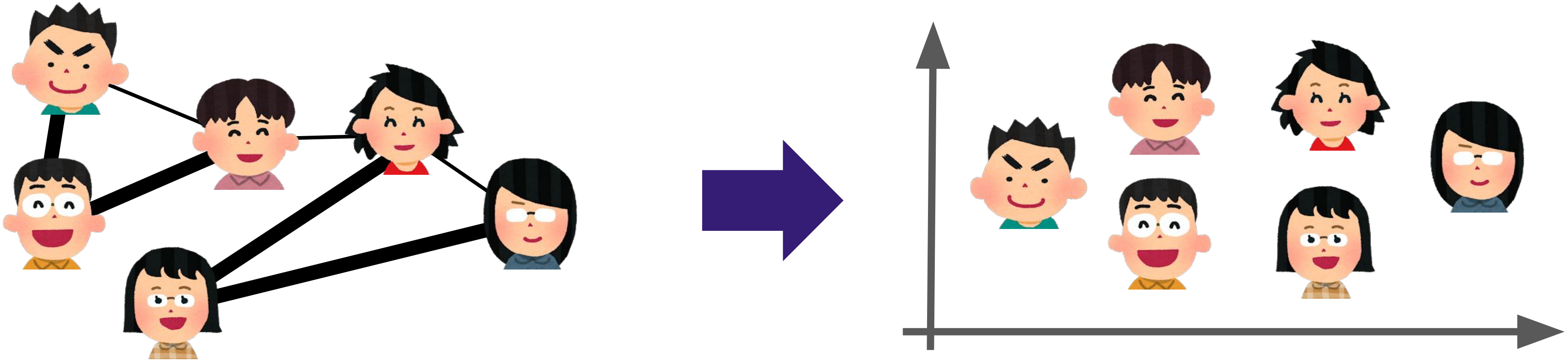
Representation Learning with Weighted Inner Product for Universal Approximation of General Similarities

Geewook Kim[1,2], Akifumi Okuno[2], Kazuki Fukui[1,2] and Hidetoshi Shimodaira[1,2]
presenter

[1]Graduate School of Informatics, Kyoto University
[2]RIKEN Center for Advanced Intelligence Project

August 15, 2019 @ IJCAI2019

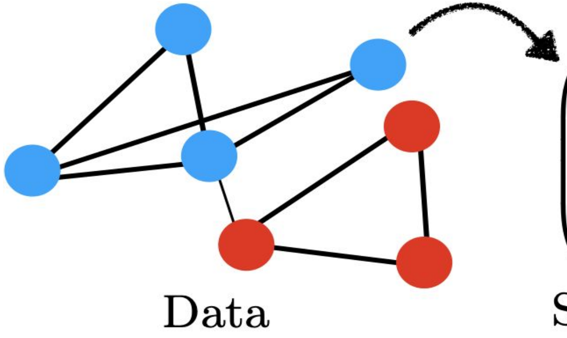
Representation Learning on graphs aims to learn useful vector representations of nodes (e.g., words, users) in a graph-structured data.



users
in a social network

user embeddings
in a vector space $\mathbb{R}^K$

Most existing methods can be generalized as follows:



$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

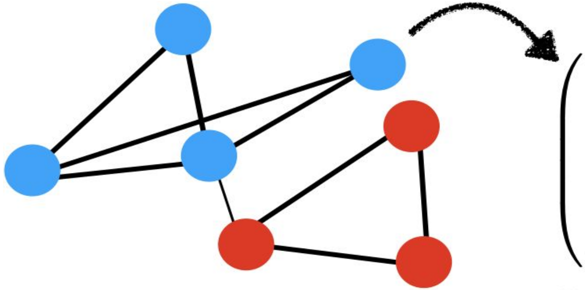
Similarity matrix $\mathbf{S}$

Similarity model $h(i,j;\boldsymbol{\theta})$

Underlying **true** structure

Estimation with model

To be more specific…



Data

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$

Similarity model $h(i,j;\boldsymbol{\theta})$

$$E(s_{ij}|i,j) = \exp(h(i,j;\boldsymbol{\theta})),$$

$$h(i,j;\boldsymbol{\theta}) = g(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$

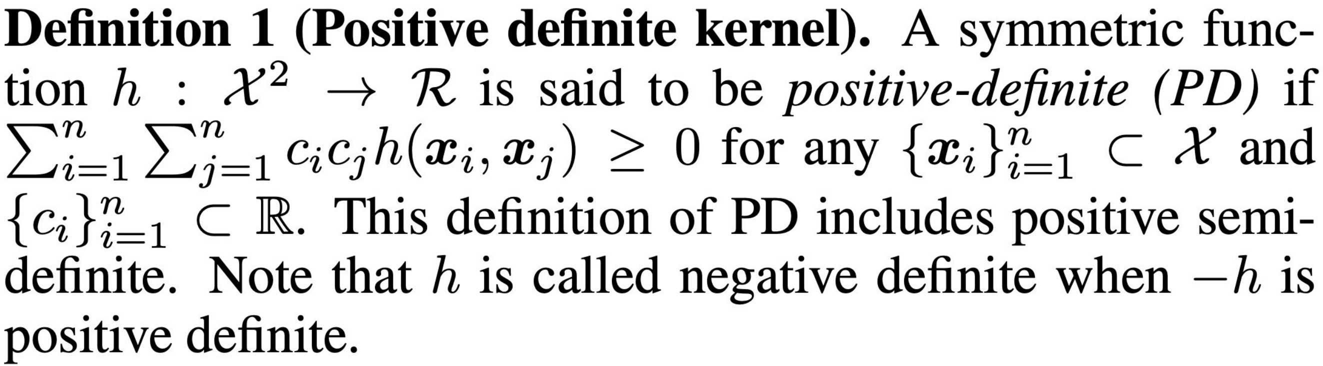
$$g(\boldsymbol{y}_i, \boldsymbol{y}_j) = \underline{\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle},$$

where $\boldsymbol{x}_i \in \mathcal{X}$ is a *data vector* (or *attribute*),
$\boldsymbol{y}_i := \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \in \mathcal{Y}$ is a *feature vector* (or *embedding*),
$\boldsymbol{f}_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathcal{Y}$ is a *embedder* (typically, NN is used),
$g : \mathcal{Y}^2 \mapsto \mathbb{R}$ is a *similarity function.*

Many methods are based on **Inner-Product Similarity (IPS)**. Why?

$$h(i, j; \boldsymbol{\theta}) = g(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$
$$g(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle,$$

➡ IPS (NN + Inner-Product) can express arbitrary **positive definite (PD) kernels** (similarities) [OHS, ICML18]



Definition 1 (Positive definite kernel). A symmetric function $h : \mathcal{X}^2 \to \mathcal{R}$ is said to be *positive-definite (PD)* if $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j h(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ for any $\{\boldsymbol{x}_i\}_{i=1}^{n} \subset \mathcal{X}$ and $\{c_i\}_{i=1}^{n} \subset \mathbb{R}$. This definition of PD includes positive semi-definite. Note that h is called negative definite when $-h$ is positive definite.

Many methods are based on **Inner-Product Similarity (IPS)**. Why?

$$h(i,j;\boldsymbol{\theta}) = g(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$

$$g(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle,$$

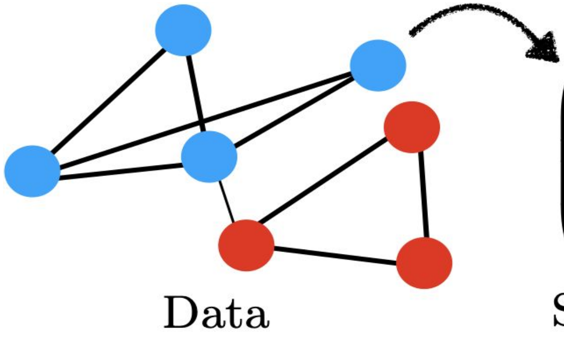
➡ IPS (NN + Inner-Product) can express arbitrary **positive definite (PD) kernels** (similarities) [OHS, ICML18]

That is, IPS can approximate any Similarity matrix $\mathbf{S}$ whose eigenvalues are all positive.

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$ Similarity model $h(i,j;\boldsymbol{\theta})$

Is PD enough? Probably not 😅



$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

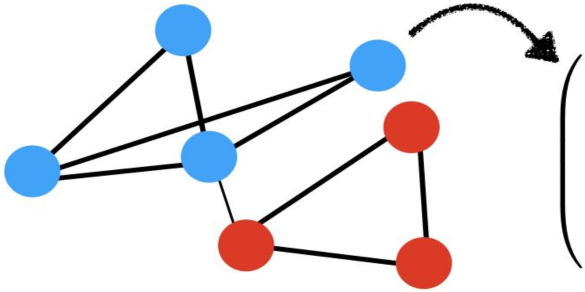
Similarity matrix $\mathbf{S}$

Similarity model $h(i,j;\boldsymbol{\theta})$

The underlying **true** structure $\mathbf{S}$ may **not PD**.

Estimation with model

Use different similarity models



Data

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$

Similarity model $h(i,j;\boldsymbol{\theta})$

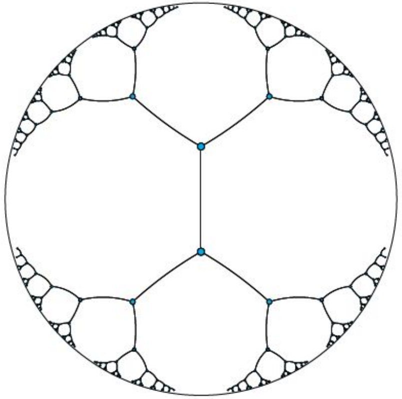
The underlying **true** structure $\mathbf{S}$ may **not PD**.

Estimation with **appropriate model**.

Use different similarity models

For example,

Poincare distance is used in Poincare embedding to embed tree-like data. [NK, NIPS17]



(b) Embedding of a tree in $\mathcal{B}^2$

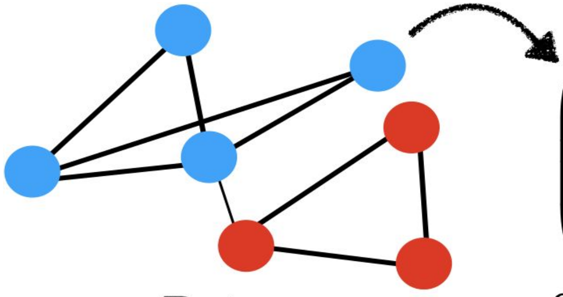
$$h(i,j;\boldsymbol{\theta}) = g(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$

$$g(\boldsymbol{y}_i, \boldsymbol{y}_j) = -\mathrm{arcosh}\Big(1 + 2\frac{\|\boldsymbol{y}_\mathrm{i} - \boldsymbol{y}_\mathrm{j}\|^2}{(1 - \|\boldsymbol{y}_\mathrm{i}\|^2)(1 - \|\boldsymbol{y}_\mathrm{j}\|^2)}\Big),$$

$$\text{where } \|\boldsymbol{y}_i\|^2 < 1, \forall i.$$

👆The image is from Figure 1-(b) in Maximillian Nickel and Douwe Kiela, “Poincaré Embeddings for Learning Hierarchical Representations.” NIPS, 2017.

Then, **similarity model selection** is a problem 🤔



Data

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$

Similarity model $h(i,j;\boldsymbol{\theta})$

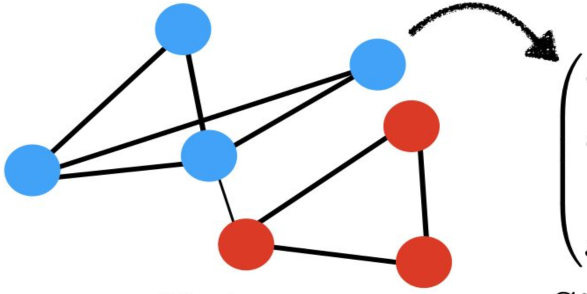
We don’t know the underlying **true** structure

We need to set **appropriate model**.

(Option 1) Inner-Product Similarity, (Option 2) Negative Poincare Distance, (Option 3) Cosine Similarity, ⋯ **so many!!!** 😫

Solution :
Highly expressive similarity models that goes beyond PD-ness [OKS, AISTATS19]

That is,



Data

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{👌}}{\overset{\text{approx.}}{\approx}} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$ — Similarity model $h(i,j;\boldsymbol{\theta})$

The model **can approximate a range of $\mathbf{S}$**
by virtue of its expressive approximation capability

Shifted Inner-Product Similarity (SIPS)

$$\begin{aligned} h_{\text{SIPS}}(i,j;\boldsymbol{\theta}) &= g((\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), u_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), (\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), u_{\boldsymbol{\theta}}(\boldsymbol{x}_j))), \\ &= \langle \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle + u_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + u_{\boldsymbol{\theta}}(\boldsymbol{x}_j). \end{aligned}$$

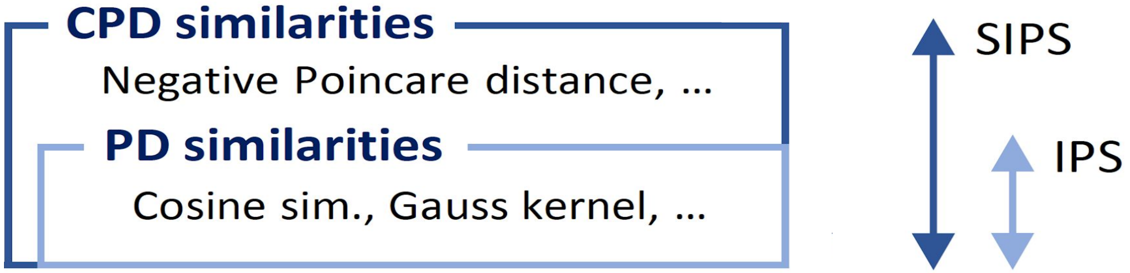
➡ SIPS can express arbitrary **conditionally positive definite (CPD) kernels** [OKS, AISTATS19]

Definition 2 (Conditionally positive definite kernel). A symmetric function $h : \mathcal{X}^2 \to \mathcal{R}$ is said to be *conditionally PD (CPD)* if $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j h(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ for any $\{\boldsymbol{x}_i\}_{i=1}^{n} \subset \mathcal{X}$ and $\{c_i\}_{i=1}^{n} \subset \mathbb{R}$ satisfying $\sum_{i=1}^{n} c_i = 0$.

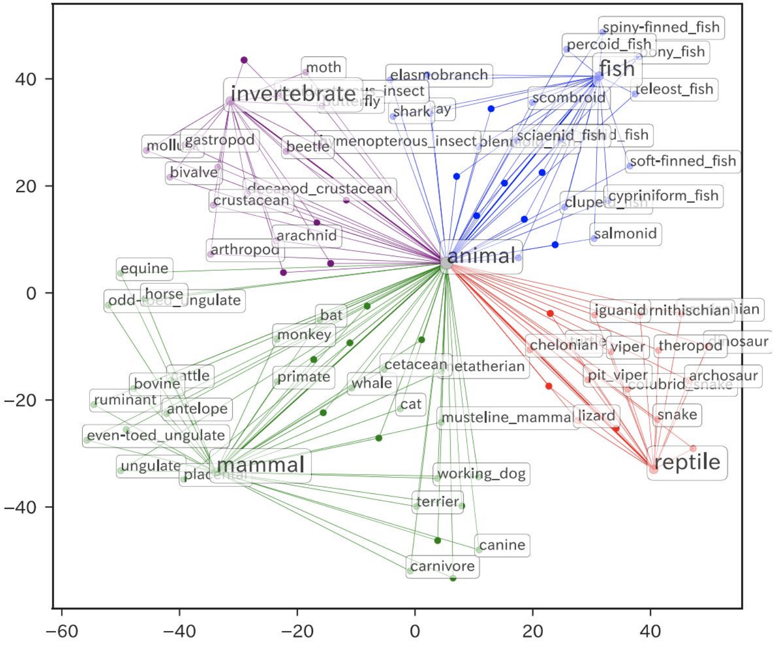
Shifted Inner-Product Similarity (SIPS)

$$
\begin{aligned}
h_{\mathrm{SIPS}}(i,j;\boldsymbol{\theta}) &= g((\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), u_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), (\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), u_{\boldsymbol{\theta}}(\boldsymbol{x}_j))), \\
&= \langle \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle + u_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + u_{\boldsymbol{\theta}}(\boldsymbol{x}_j).
\end{aligned}
$$

➡ SIPS can express arbitrary **conditionally positive definite (CPD) kernels** [OKS, AISTATS19]


CPD similarities
Negative Poincare distance, ...
PD similarities
Cosine sim., Gauss kernel, ...
SIPS
IPS

SIPS can express any **CPD similarities, and negative poincare distance is one of CPD.**



$$h_{\text{SIPS}}(i,j;\boldsymbol{\theta}) = g((\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), u_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), (\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), u_{\boldsymbol{\theta}}(\boldsymbol{x}_j))),$$

$$= \langle \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle + u_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + u_{\boldsymbol{\theta}}(\boldsymbol{x}_j).$$

$$h(i,j;\boldsymbol{\theta}) = g(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$

$$g(\boldsymbol{y}_i, \boldsymbol{y}_j) = -\text{arcosh}\left(1 + 2\frac{\|\boldsymbol{y}_{\text{i}} - \boldsymbol{y}_{\text{j}}\|^2}{(1 - \|\boldsymbol{y}_{\text{i}}\|^2)(1 - \|\boldsymbol{y}_{\text{j}}\|^2)}\right),$$

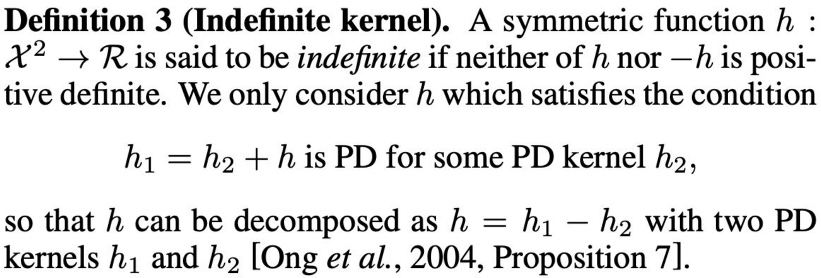
where $\|\boldsymbol{y}_i\|^2 < 1, \forall i.$

👆The image is from Figure 1 in Akifumi Okuno, Geewook Kim, and Hidetoshi Shimodaira. “Graph Embedding with Shifted Inner Product Similarity and Its Improved Approximation Capability.” AISTATS, 2019.

Inner-Product Difference Similarity (IPDS)

$$\begin{aligned} h_{\mathrm{IPDS}}(i, j; \boldsymbol{\theta}) &= g((\boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), (\boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_j))), \\ &= \langle \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle - \langle \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle \end{aligned}$$

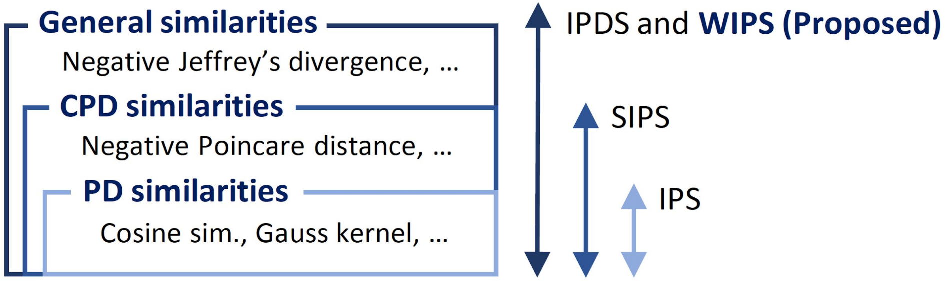
➡ IPDS can express **general similarities** (that include PD, Negative Definite and Indefinite Kernels) [OKS, AISTATS19]



Definition 3 (Indefinite kernel). A symmetric function $h : \mathcal{X}^2 \to \mathcal{R}$ is said to be *indefinite* if neither of h nor $-h$ is positive definite. We only consider h which satisfies the condition

$$h_1 = h_2 + h \text{ is PD for some PD kernel } h_2,$$

so that h can be decomposed as $h = h_1 - h_2$ with two PD kernels h_1 and h_2 [Ong *et al.*, 2004, Proposition 7].



General similarities

Negative Jeffrey’s divergence, …

CPD similarities

Negative Poincare distance, …

PD similarities

Cosine sim., Gauss kernel, …

IPDS and **WIPS (Proposed)**

SIPS

IPS

First motivation of this work :

IPDS has not been experimentally examined yet.

Let’s try IPDS on a range of applications!

First motivation of this work :
IPDS has not been experimentally examined yet.
Let’s try IPDS on a range of applications!

··· Wait🤔 what **dimensionality ratio** $\frac{q}{K-q}$ should we set?

$$h_{\mathrm{IPDS}}(i,j;\boldsymbol{\theta}) = \underset{K-q}{\underleftrightarrow{\langle \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle}} - \underset{q}{\underleftrightarrow{\langle \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle}}$$

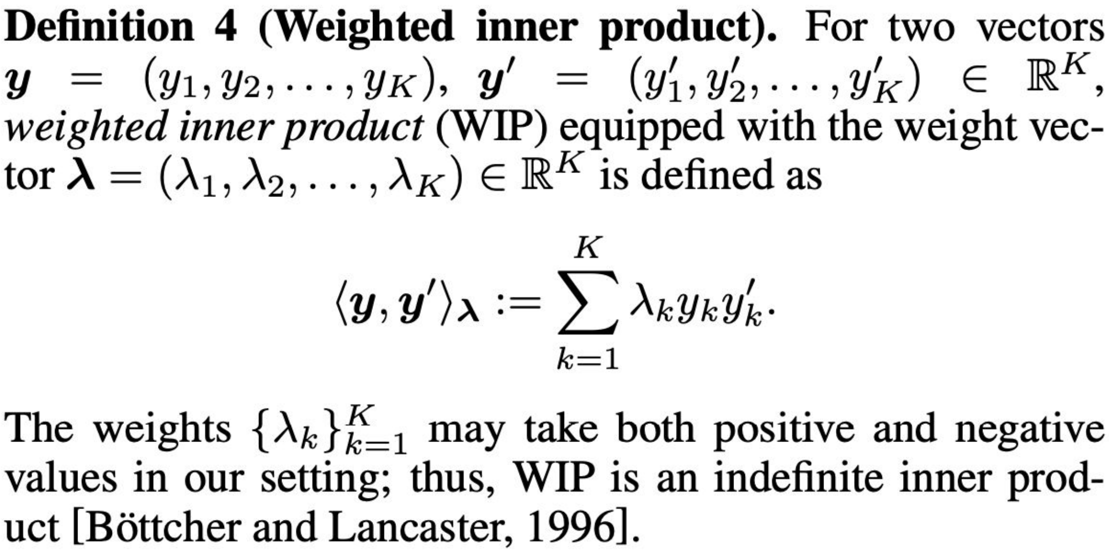
In preliminary experiments, we found the ratio in IPDS is **important** and **difficult to tune properly**.

Our proposal : A small modification to IPDS.

Weighted Inner-Product Similarity (WIPS)

$$h_{\mathrm{WIPS}}(i,j;\boldsymbol{\theta},\boldsymbol{\lambda}) = g_{\boldsymbol{\lambda}}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)),$$

$$g_{\boldsymbol{\lambda}}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle_{\boldsymbol{\lambda}}$$



Definition 4 (Weighted inner product). For two vectors $\boldsymbol{y} = (y_1, y_2, \dots, y_K)$, $\boldsymbol{y}' = (y'_1, y'_2, \dots, y'_K) \in \mathbb{R}^K$, *weighted inner product* (WIP) equipped with the weight vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_K) \in \mathbb{R}^K$ is defined as

$$\langle \boldsymbol{y}, \boldsymbol{y}' \rangle_{\boldsymbol{\lambda}} := \sum_{k=1}^{K} \lambda_k y_k y'_k.$$

The weights $\{\lambda_k\}_{k=1}^{K}$ may take both positive and negative values in our setting; thus, WIP is an indefinite inner product [Böttcher and Lancaster, 1996].

WIPS approximates arbitrary **general similarities** while it cuts out the need of tuning the dimensionality parameters in IPDS.

$$
\begin{aligned}
h_{\text{WIPS}}(i,j;\boldsymbol{\theta},\boldsymbol{\lambda}) &= \langle \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle_{\boldsymbol{\lambda}} \\
h_{\text{IPDS}}(i,j;\boldsymbol{\theta}) &= \langle \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle - \langle \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle \\
h_{\text{SIPS}}(i,j;\boldsymbol{\theta}) &= \langle \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle + u_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + u_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \\
h_{\text{IPS}}(i,j;\boldsymbol{\theta}) &= \langle \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle
\end{aligned}
$$

WIPS	$\boldsymbol{\lambda}$	$\boldsymbol{f}_{\boldsymbol{\theta}}$
IPS	$\mathbf{1}_K$	$\boldsymbol{f}_{\boldsymbol{\theta}}$
SIPS	$(\mathbf{1}_{K+1}, -1)$	$(\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}), u_{\boldsymbol{\theta}}(\boldsymbol{x}), 1, u_{\boldsymbol{\theta}}(\boldsymbol{x}) - 1)$
IPDS	$(\mathbf{1}_{K-q}, -\mathbf{1}_q)$	$(\boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}))$

Table 1: WIPS expresses the other models by specifying $\boldsymbol{\lambda}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$.

WIPS approximates arbitrary **general similarities** while it cuts out the need of tuning the dimensionality parameters in IPDS.

$$h_{\mathrm{WIPS}}(i,j;\boldsymbol{\theta},\boldsymbol{\lambda}) = \langle \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle_{\boldsymbol{\lambda}}$$

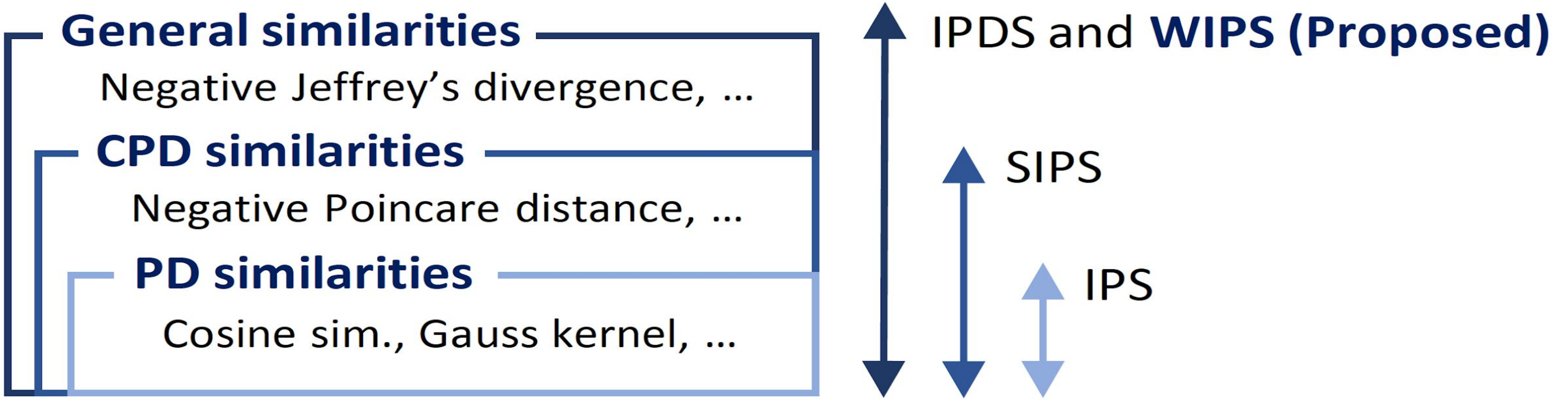
That is, simply speaking, WIPS can approximate any Similarity matrix $\mathbf{S}$ **without any condition on the eigenvalues.**

👌

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \ddots & s_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \overset{\text{approx.}}{\approx} \begin{pmatrix} h(1,1;\boldsymbol{\theta}) & h(1,2;\boldsymbol{\theta}) & \cdots & h(1,n;\boldsymbol{\theta}) \\ h(2,1;\boldsymbol{\theta}) & h(2,2;\boldsymbol{\theta}) & \ddots & h(2,n;\boldsymbol{\theta}) \\ \vdots & \ddots & \ddots & \vdots \\ h(n,1;\boldsymbol{\theta}) & h(n,2;\boldsymbol{\theta}) & \cdots & h(n,n;\boldsymbol{\theta}) \end{pmatrix}$$

Similarity matrix $\mathbf{S}$ Similarity model $h(i,j;\boldsymbol{\theta})$

So far, we’ve seen many similarity models.


General similarities
Negative Jeffrey’s divergence, …
CPD similarities
Negative Poincare distance, …
PD similarities
Cosine sim., Gauss kernel, …
IPDS and **WIPS (Proposed)**
SIPS
IPS

Using real-world datasets, we aim to assess the **approximation ability** of the similarity models as well as the **effectiveness of the learned feature vectors**.

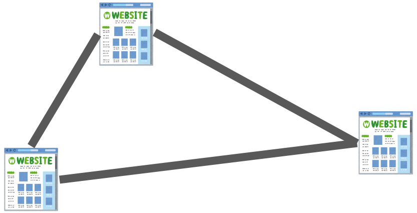
Experiments - **datasets**

3 graph-structured datasets are used.

Hypertext Network

877 nodes and 1480 links

node : webpage,
attr. : 1703 dim.
bag-of-words

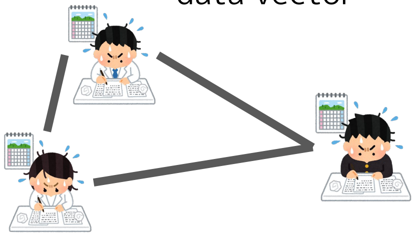


edge : hyperlink relation

Co-authorship Network

41328 nodes and 210320 links

node : author,
attr. : 43 dim.
data vector

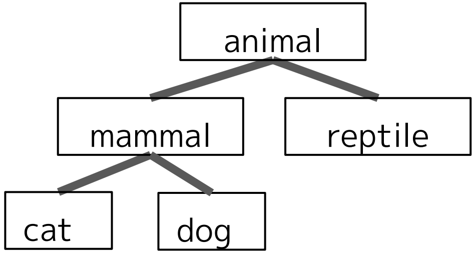


edge : co-author relation

Taxonomy Tree

37623 nodes and 312885 links

node : word,
attr. : 300 dim. pretrained
Google’s word embedding



animal

mammal

reptile

cat

dog

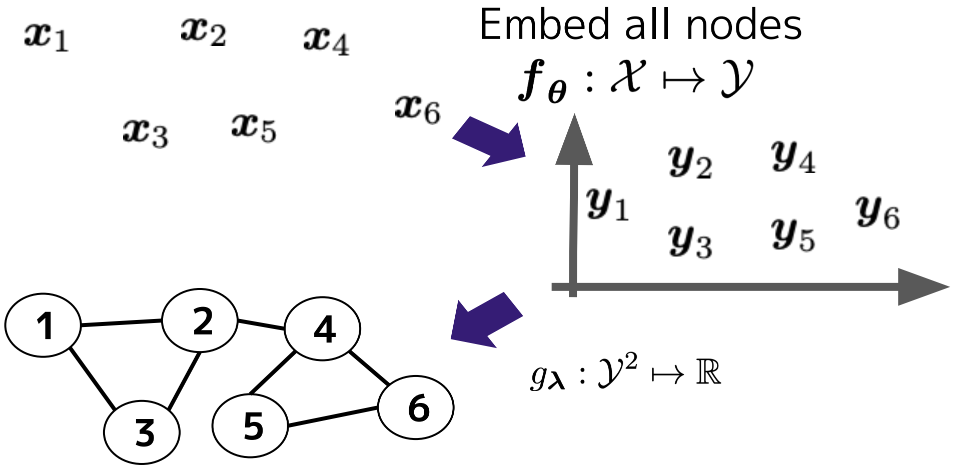
edge : hyponymy-hypernymy relation

Each webpage has
A. semantic label in {Student, Faculty, Staff, Course, Project}
B. university label in {Cornell, Texas, Washington, Wisconsin}

Graph Reconstruction

At training, **assume that all nodes and links are visible**. Use all data to train the model ($\boldsymbol{f_\theta}$ and $g_{\boldsymbol{\lambda}}$).

Then, at evaluation,



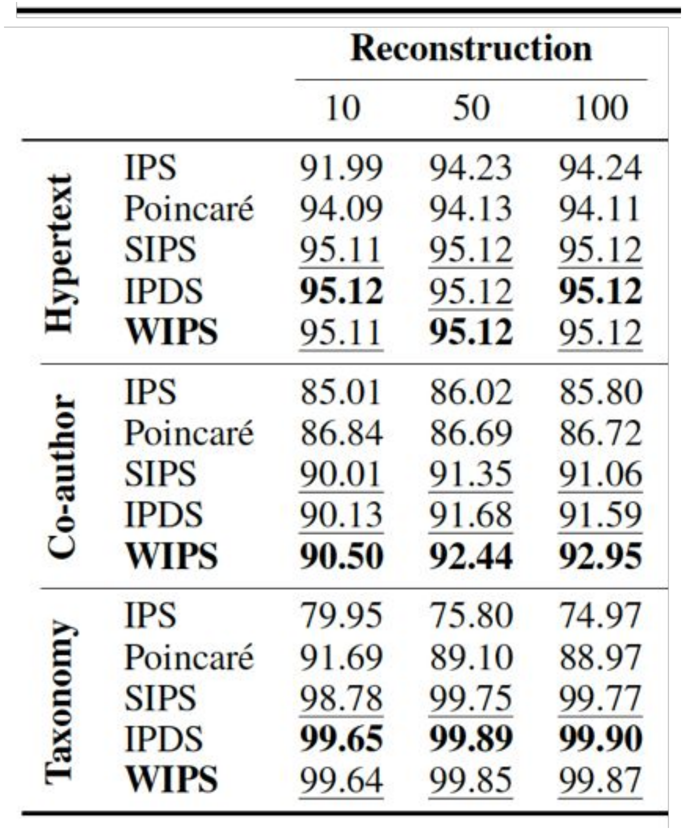
Embed all nodes

$\boldsymbol{f_\theta} : \mathcal{X} \mapsto \mathcal{Y}$

$g_{\boldsymbol{\lambda}} : \mathcal{Y}^2 \mapsto \mathbb{R}$

Predict (reconstruct) all links

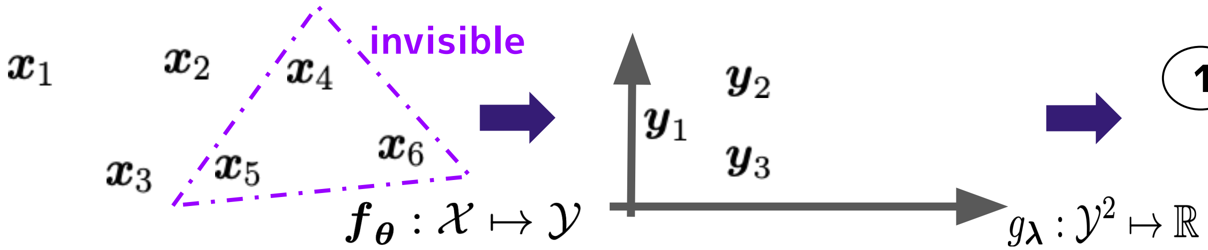
ROC-AUC for prediction errors are calculated ▶



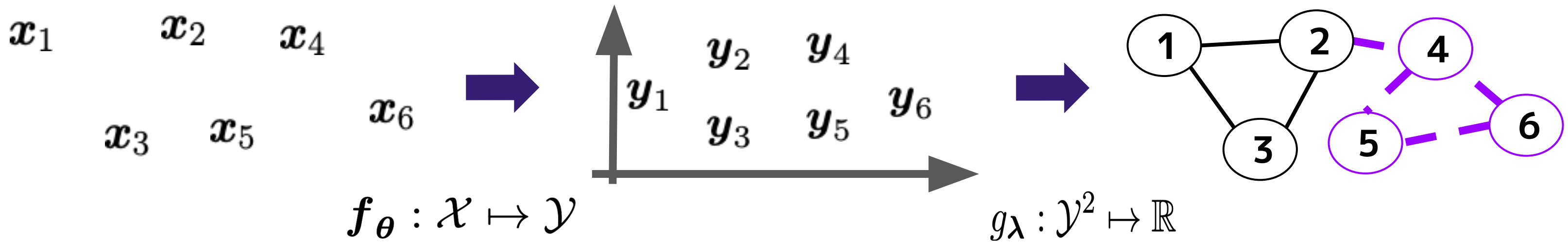
		Reconstruction		
		10	50	100
Hypertext	IPS	91.99	94.23	94.24
	Poincaré	94.09	94.13	94.11
	SIPS	95.11	95.12	95.12
	IPDS	**95.12**	95.12	**95.12**
	WIPS	95.11	**95.12**	95.12
Co-author	IPS	85.01	86.02	85.80
	Poincaré	86.84	86.69	86.72
	SIPS	90.01	91.35	91.06
	IPDS	90.13	91.68	91.59
	WIPS	**90.50**	**92.44**	**92.95**
Taxonomy	IPS	79.95	75.80	74.97
	Poincaré	91.69	89.10	88.97
	SIPS	98.78	99.75	99.77
	IPDS	**99.65**	**99.89**	**99.90**
	WIPS	99.64	99.85	99.87

Link Prediction

At training, **assume that some nodes (and its links) are invisible**.
Use only observed sub-graph to train the model ($\boldsymbol{f}_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\lambda}}$).



Then, at evaluation,



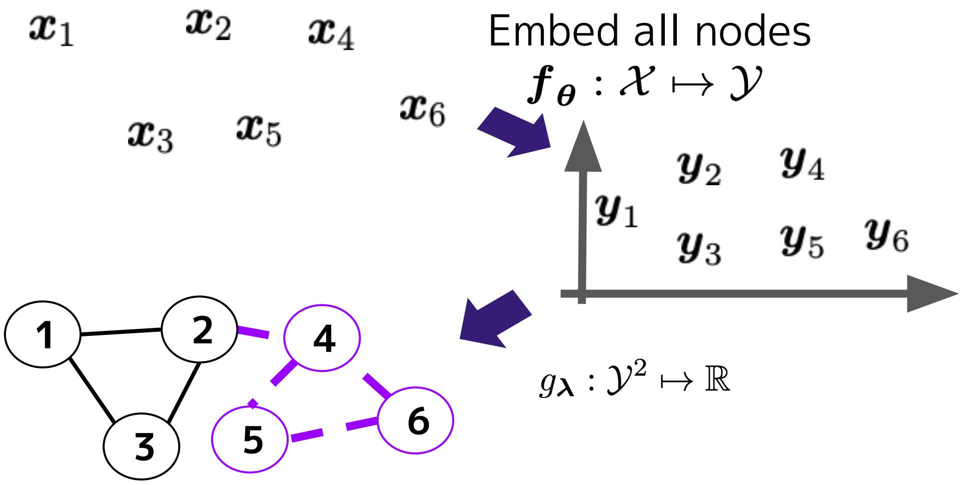
Embed all nodes

Predict (reconstruct) all **unobserved** links

Link Prediction

At training, **assume that some nodes (and its links) are invisible**.
Use only observed sub-graph to train the model ($\boldsymbol{f_\theta}$ and $g_{\boldsymbol{\lambda}}$).

Then, at evaluation,

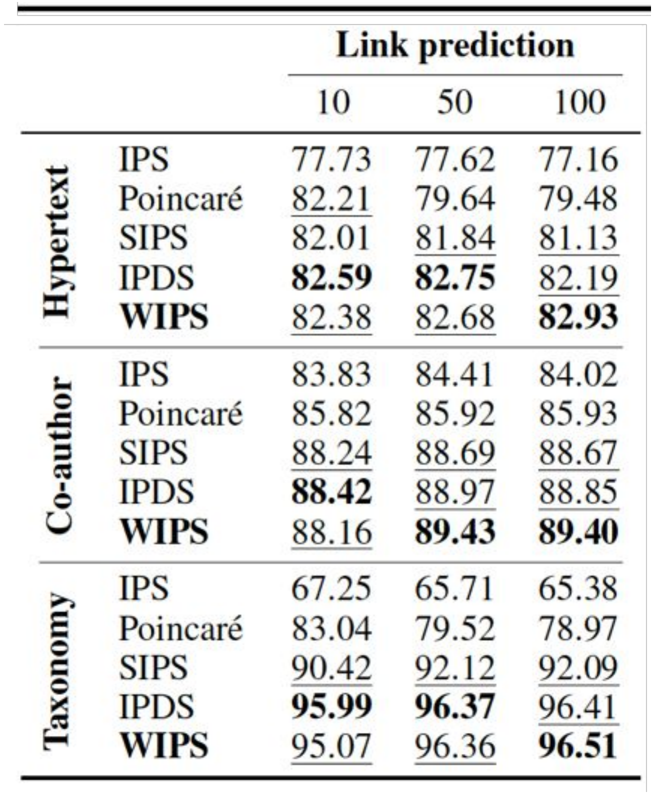


Embed all nodes
$\boldsymbol{f_\theta} : \mathcal{X} \mapsto \mathcal{Y}$

$g_{\boldsymbol{\lambda}} : \mathcal{Y}^2 \mapsto \mathbb{R}$

Predict (reconstruct) all **unobserved** links

ROC-AUC for prediction errors are calculated ▶



		Link prediction		
		10	50	100
Hypertext	IPS	77.73	77.62	77.16
	Poincaré	82.21	79.64	79.48
	SIPS	82.01	81.84	81.13
	IPDS	**82.59**	**82.75**	82.19
	WIPS	82.38	82.68	**82.93**
Co-author	IPS	83.83	84.41	84.02
	Poincaré	85.82	85.92	85.93
	SIPS	88.24	88.69	88.67
	IPDS	**88.42**	88.97	88.85
	WIPS	88.16	**89.43**	**89.40**
Taxonomy	IPS	67.25	65.71	65.38
	Poincaré	83.04	79.52	78.97
	SIPS	90.42	92.12	92.09
	IPDS	**95.99**	**96.37**	96.41
	WIPS	95.07	96.36	**96.51**

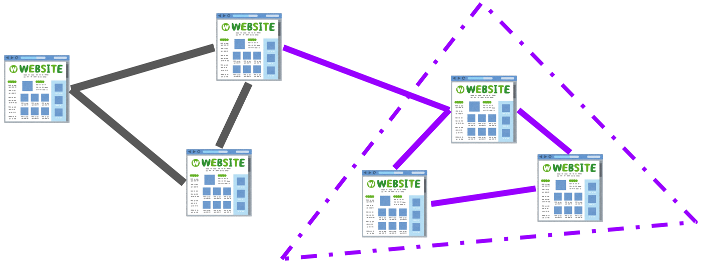
Hypertext Classification

Each webpage in Hypertext Network has

A. semantic label $\in$ {Student, Faculty, Staff, Course, Project}

B. university label $\in$ {Cornell, Texas, Washington, Wisconsin}

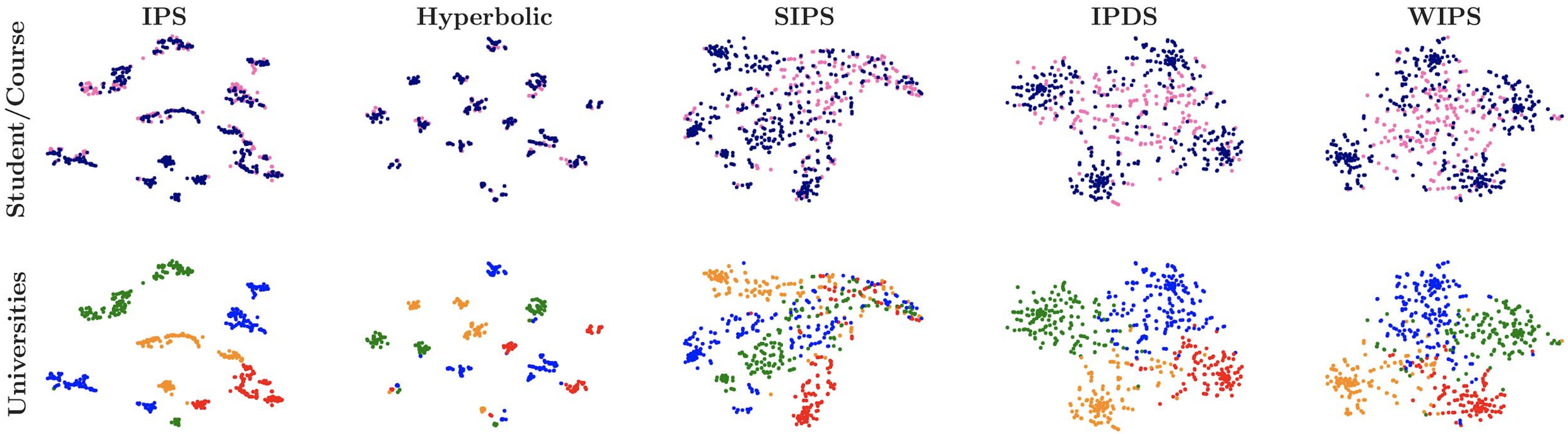
train set (observed) is used to train the embedder and classifiers



test set (invisible) is used for evaluation

	IPS	Poincaré	Hyperbolic	SIPS	IPDS	WIPS
A	56.08	46.19	47.22	69.09	71.70	**73.35**
B	91.59	30.17	93.12	93.81	93.81	**96.31**

Visualization of Hypertext Network



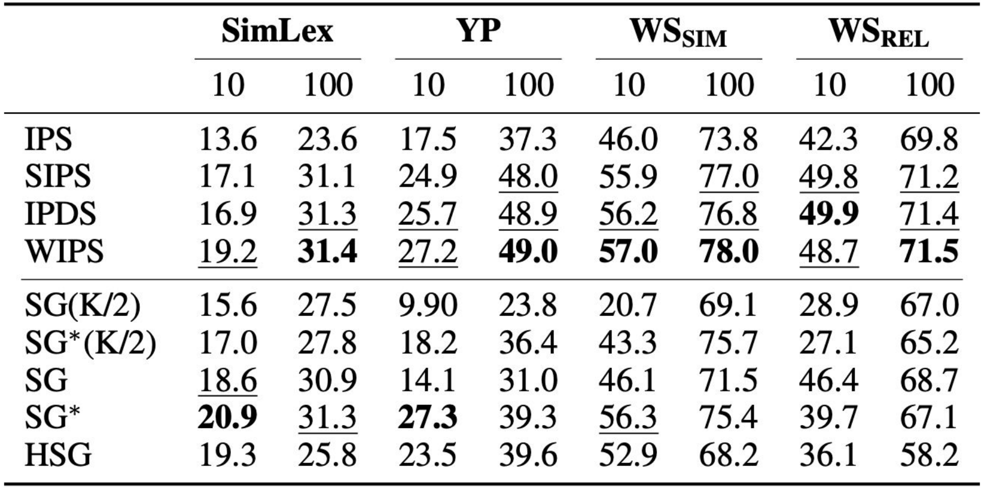
The hypertexts are colored by its semantic labels (upper) for Student (navy) and Course (pink), and also university labels (lower) for Cornell (red), Texas (orange), Washington (green) and Wisconsin (blue).

Both class labels are clearly identified with IPDS and WIPS, whereas they become obscure in the other embeddings.

Word Similarity

The similarity models are applied to Word2vec [MSCCD, NIPS13] to learn word embeddings.

The embeddings are evaluated by Spearman’s rank correlation with 4 human annotated word similarity datasets.



	SimLex		YP		WS_{SIM}		WS_{REL}	
	10	100	10	100	10	100	10	100
IPS	13.6	23.6	17.5	37.3	46.0	73.8	42.3	69.8
SIPS	17.1	31.1	24.9	48.0	55.9	77.0	49.8	71.2
IPDS	16.9	31.3	25.7	48.9	56.2	76.8	**49.9**	71.4
WIPS	19.2	**31.4**	27.2	**49.0**	**57.0**	**78.0**	48.7	**71.5**
SG(K/2)	15.6	27.5	9.90	23.8	20.7	69.1	28.9	67.0
SG*(K/2)	17.0	27.8	18.2	36.4	43.3	75.7	27.1	65.2
SG	18.6	30.9	14.1	31.0	46.1	71.5	46.4	68.7
SG*	**20.9**	31.3	**27.3**	39.3	56.3	75.4	39.7	67.1
HSG	19.3	25.8	23.5	39.6	52.9	68.2	36.1	58.2

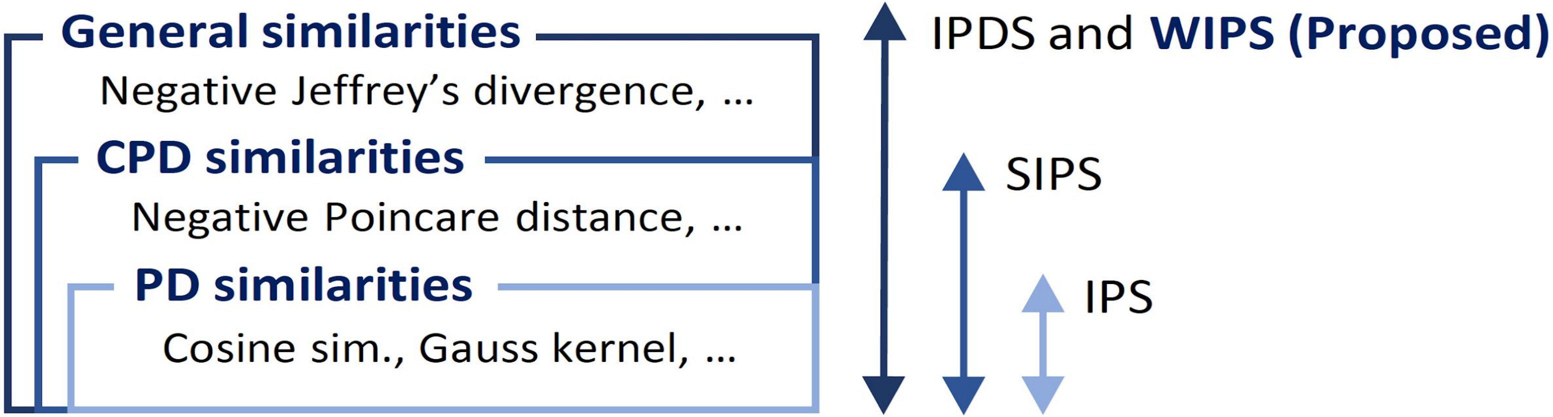
Summary

$$h_{\mathrm{WIPS}}(i,j;\boldsymbol{\theta},\boldsymbol{\lambda}) = \langle \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle_{\boldsymbol{\lambda}}$$

$$h_{\mathrm{IPDS}}(i,j;\boldsymbol{\theta}) = \langle \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{+}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle - \langle \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}^{-}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle$$

$$h_{\mathrm{SIPS}}(i,j;\boldsymbol{\theta}) = \langle \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle + u_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + u_{\boldsymbol{\theta}}(\boldsymbol{x}_j)$$

$$h_{\mathrm{IPS}}(i,j;\boldsymbol{\theta}) = \langle \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle$$


General similarities
Negative Jeffrey's divergence, ...
CPD similarities
Negative Poincare distance, ...
PD similarities
Cosine sim., Gauss kernel, ...
IPDS and WIPS (Proposed)
SIPS
IPS

Contact info : Geewook Kim (geewook@sys.i.kyoto-u.ac.jp)

References

[MSCCD, NIPS13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and Their Compositionality. In Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates Inc., 2013.

[NK, NIPS17] Maximillian Nickel and Douwe Kiela. Poincare Embeddings for Learning Hierarchical ´ Representations. In Advances in Neural Information Processing Systems 30, pages 6338–6347. Curran Associates, Inc., 2017.

[OHS, ICML18] Akifumi Okuno, Tetsuya Hada, and Hidetoshi Shimodaira. A probabilistic framework for multi-view feature learning with many-to-many associations via neural networks. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 3885–3894. PMLR, 2018.

[OKS, AISTATS19] Akifumi Okuno, Geewook Kim, and Hidetoshi Shimodaira. Graph Embedding with Shifted Inner Product Similarity and Its Improved Approximation Capability. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), volume 89 of Proceedings of Machine Learning Research, pages 644–653. PMLR, 2019.