京都大学
KYOTO UNIVERSITY

RIKEN AIP

ISI WSC 2019 – ISI World Statistics Congress 2019/08/23
Journal of Data Science, Statistics and Visualisation Session

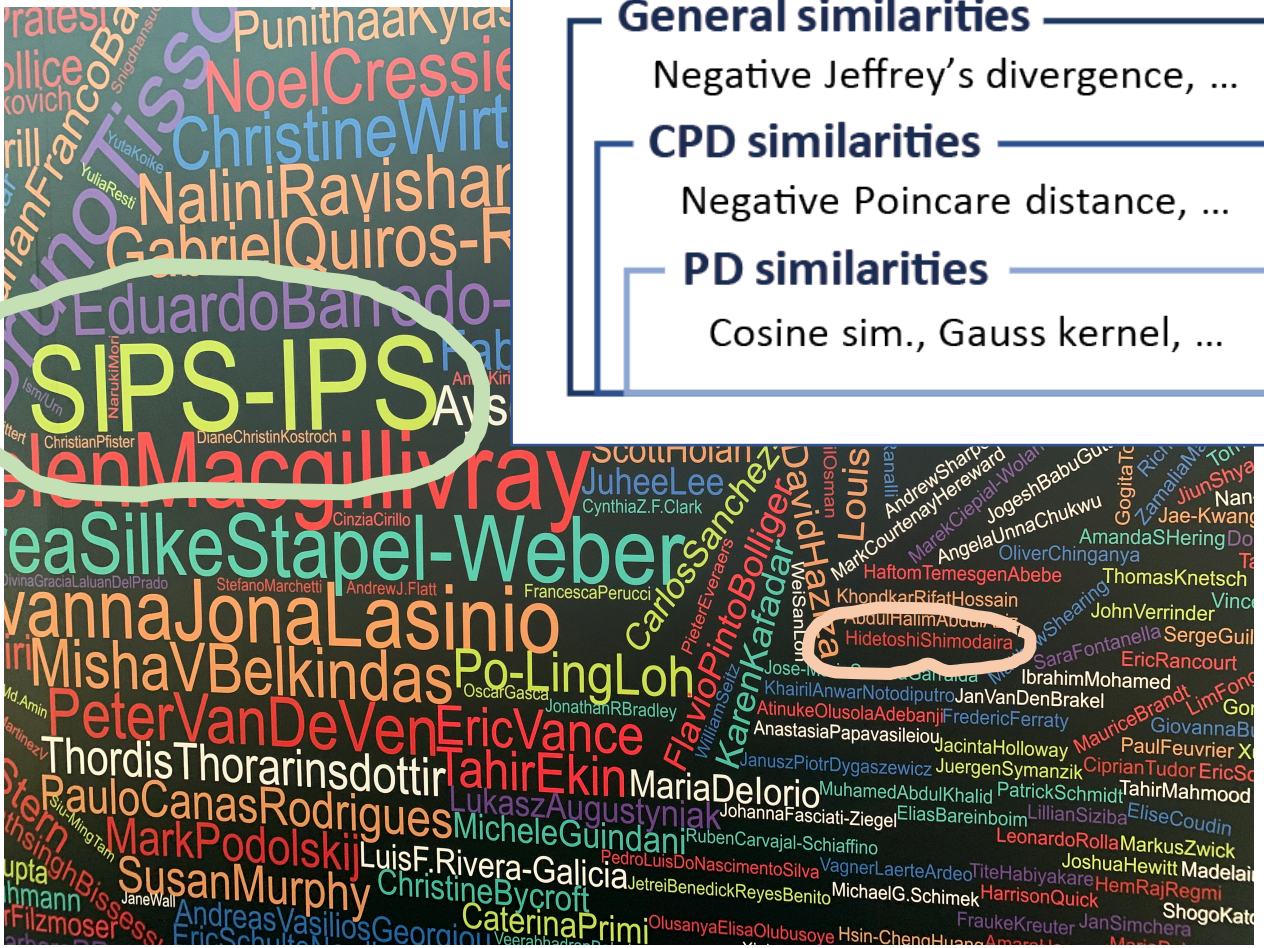
Multiview graph embedding as a generalization of canonical correlation analysis

Hidetoshi Shimodaira
Kyoto University / RIKEN AIP

In collaboration with
Kazuki Fukui, Tetsuya Hada, Geewook Kim,
Akifumi Okuno, Takamasa Oshikiri

Slides available http://stat.sys.i.kyoto-u.ac.jp/research/

“SIPS-IPS” is on the board here??

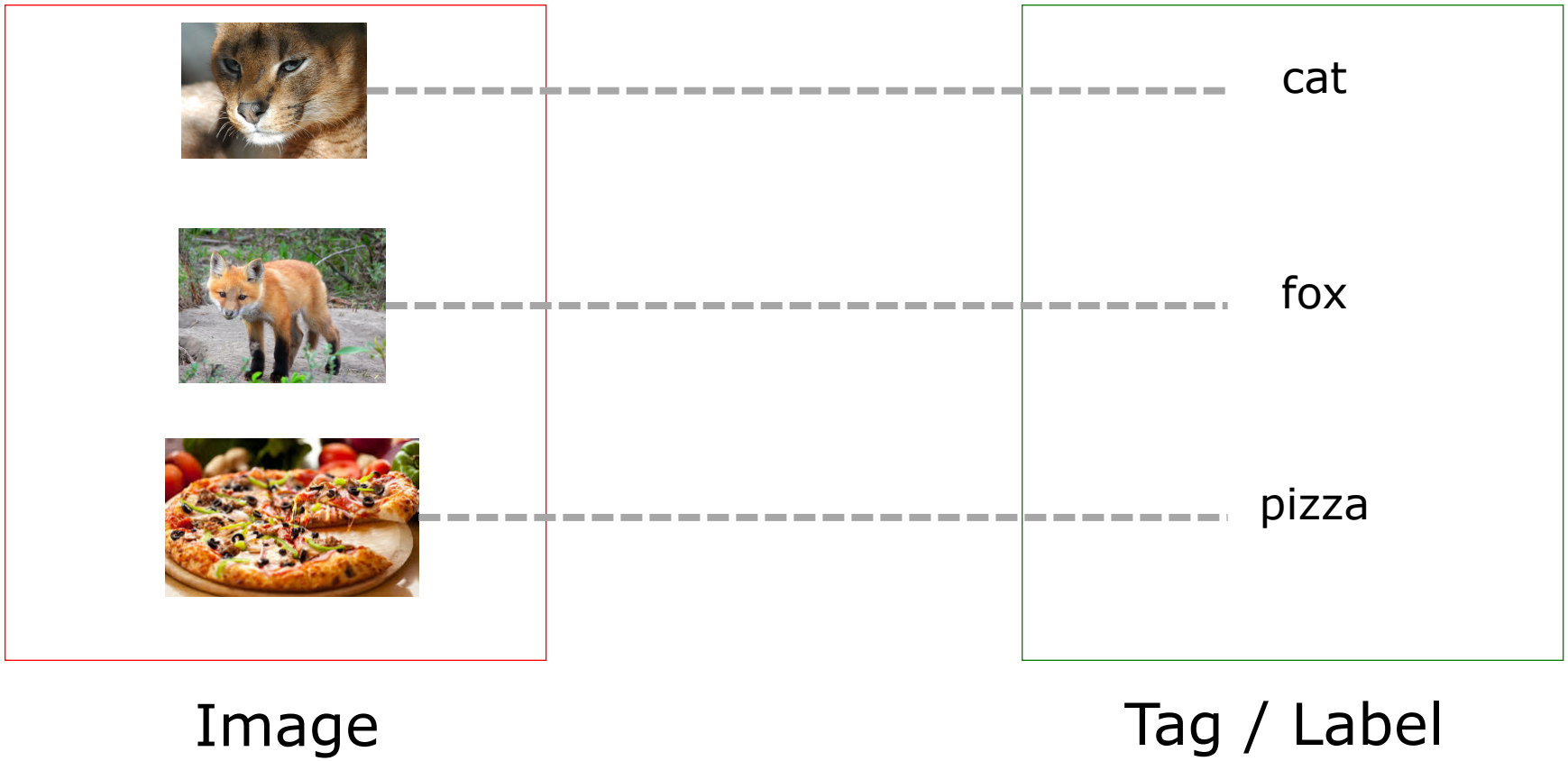


Today's Talk

- CCA is generalized to multiview graph embedding
- Further generalization to non-linear multiview graph embedding with neural networks
- A theory of representation learning: "new" similarity models beyond inner product (IPS, SIPS, WIPS)

CCA and Multiview graph embedding

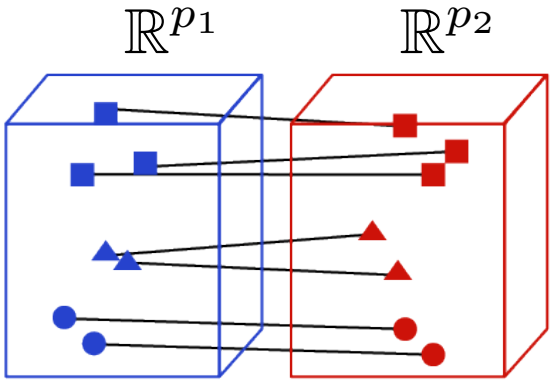
Canonical Correlation Analysis (CCA) has been used in pattern recognition



[Socher+ 2010], [Bruni+ 2014], [Lazaridou+ 2014], etc.

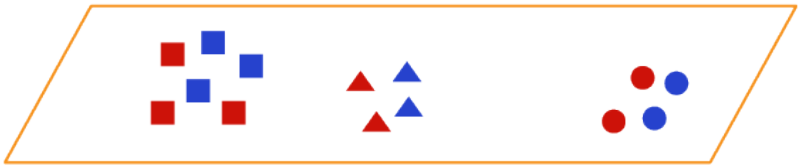
CCA maximizes the correlations or minimizes the distances

$$\boldsymbol{X}^{(1)} = \begin{bmatrix} \boldsymbol{x}_1^{(1)} \\ \vdots \\ \boldsymbol{x}_n^{(1)} \end{bmatrix}$$



$$\boldsymbol{X}^{(2)} = \begin{bmatrix} \boldsymbol{x}_1^{(2)} \\ \vdots \\ \boldsymbol{x}_n^{(2)} \end{bmatrix}$$

$$\boldsymbol{y}_i^{(1)} = \boldsymbol{x}_i^{(1)} \boldsymbol{A}^{(1)} \qquad \boldsymbol{y}_i^{(2)} = \boldsymbol{x}_i^{(2)} \boldsymbol{A}^{(2)}$$



$\mathbb{R}^K$

$$\underbrace{\sum_{i=1}^{n} \|\boldsymbol{y}_i^{(1)} - \boldsymbol{y}_i^{(2)}\|^2}_{\text{minimize}} = \underbrace{\sum_{i=1}^{n} \|\boldsymbol{y}_i^{(1)}\|^2 + \sum_{i=1}^{n} \|\boldsymbol{y}_i^{(2)}\|^2}_{\text{constraints}} - \underbrace{2 \sum_{i=1}^{n} \langle \boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)} \rangle}_{\text{maximize}}$$

Hotelling (1936). Biometrika: "**RELATIONS BETWEEN TWO SETS OF VARIATES**"

Equivalent formulations of CCA

$$\max_{\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)}} \operatorname{tr} \boldsymbol{A}^{(1)\top} \boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(2)} \boldsymbol{A}^{(2)}$$

subject to $\boldsymbol{A}^{(1)\top} \boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(1)} \boldsymbol{A}^{(1)} = \boldsymbol{A}^{(2)\top} \boldsymbol{X}^{(2)\top} \boldsymbol{X}^{(2)} \boldsymbol{A}^{(2)} = \boldsymbol{I}_K / 2$

$$\min_{\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)}} \|\boldsymbol{x}_i^{(1)} \boldsymbol{A}^{(1)} - \boldsymbol{x}_i^{(2)} \boldsymbol{A}^{(2)}\|^2$$

subject to $\boldsymbol{A}^{(1)\top} \boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(1)} \boldsymbol{A}^{(1)} + \boldsymbol{A}^{(2)\top} \boldsymbol{X}^{(2)\top} \boldsymbol{X}^{(2)} \boldsymbol{A}^{(2)} = \boldsymbol{I}_K$

$$\max_{\boldsymbol{A} \in \mathbb{R}^{N \times K}} \boldsymbol{A}^\top \boldsymbol{H} \boldsymbol{A}$$

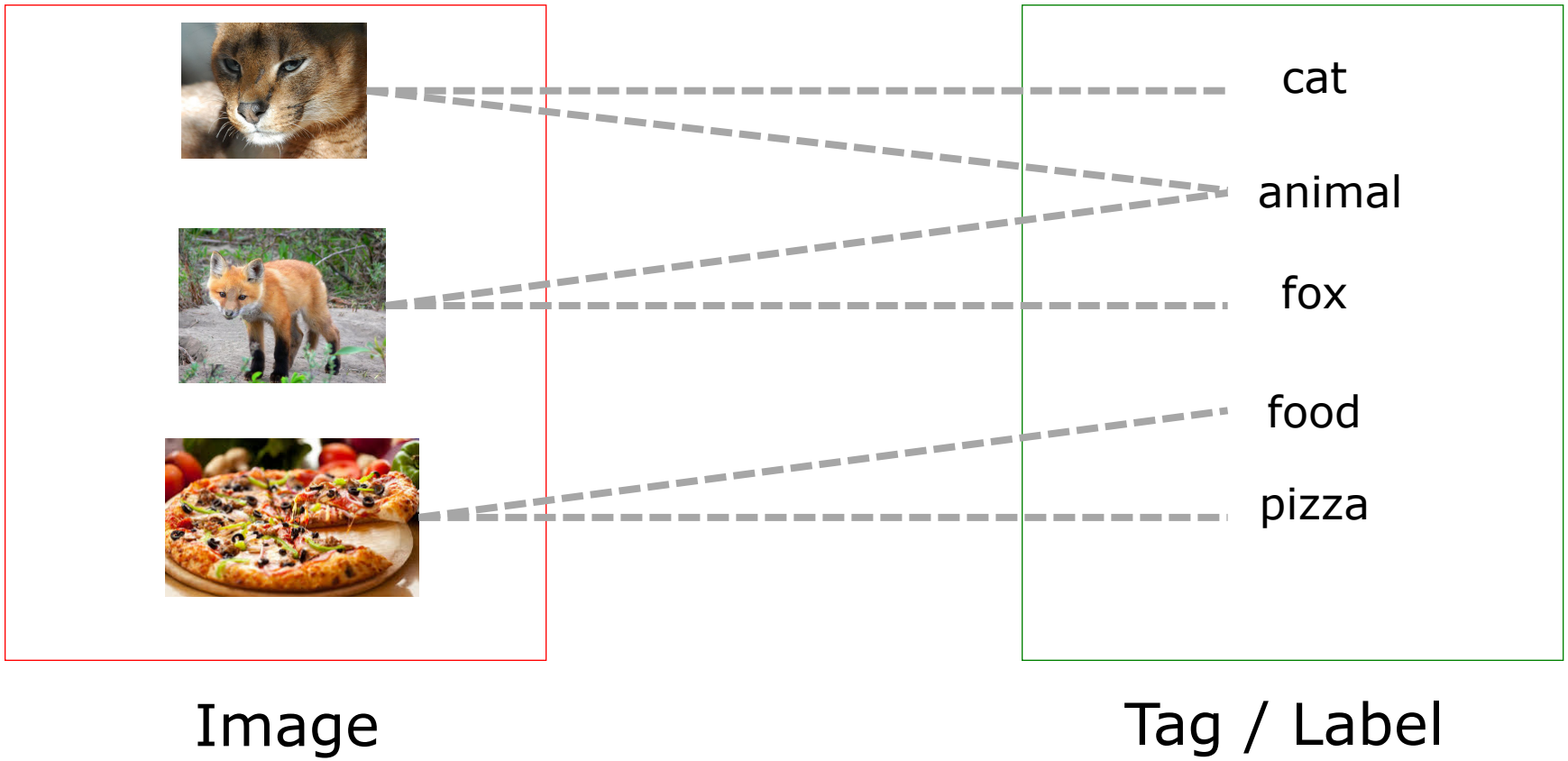
subject to $\boldsymbol{A}^\top \boldsymbol{G} \boldsymbol{A} = \boldsymbol{I}_K$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \boldsymbol{A}^{(2)} \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} & \boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(2)} \\ \boldsymbol{X}^{(2)\top} \boldsymbol{X}^{(1)} & \end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(1)} & \\ & \boldsymbol{X}^{(2)\top} \boldsymbol{X}^{(2)} \end{bmatrix}$$

This formulation is generalized to graph embedding

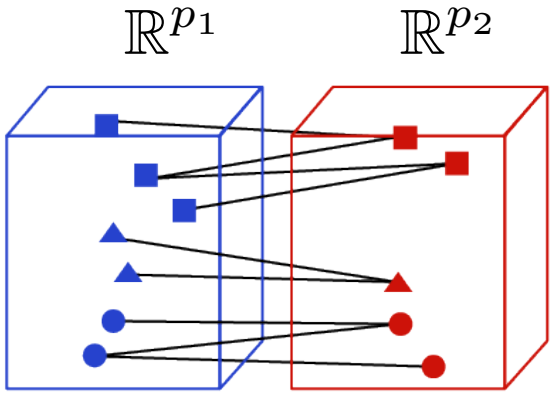
Many-to-many relations



ml-CCA [Ranjan+ 2015], cluster CCA [Rasiwasia+ 2014], etc.

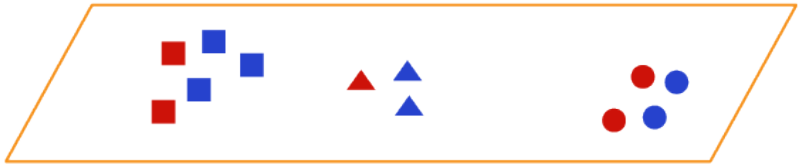
CCA is generalized to graph-embedding (two or more views)

$$\boldsymbol{X}^{(1)} = \begin{bmatrix} \boldsymbol{x}_1^{(1)} \\ \vdots \\ \boldsymbol{x}_{n_1}^{(1)} \end{bmatrix}$$



$$\boldsymbol{X}^{(2)} = \begin{bmatrix} \boldsymbol{x}_1^{(2)} \\ \vdots \\ \boldsymbol{x}_{n_2}^{(2)} \end{bmatrix}$$

$$\boldsymbol{y}_i^{(1)} = \boldsymbol{x}_i^{(1)} \boldsymbol{A}^{(1)} \quad w_{ij}^{(12)} \quad \boldsymbol{y}_j^{(2)} = \boldsymbol{x}_j^{(2)} \boldsymbol{A}^{(2)}$$



$\mathbb{R}^K$

$$\min_{\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)}} \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} w_{ij}^{(12)} \|\boldsymbol{y}_i^{(1)} - \boldsymbol{y}_j^{(2)}\|^2 \quad \text{or} \quad \max_{\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)}} \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} w_{ij}^{(12)} \langle \boldsymbol{y}_i^{(1)}, \boldsymbol{y}_j^{(2)} \rangle$$

Cross-domain matching correlation analysis (CDMCA) generalizes several multivariate analysis methods such as CCA, multiset-CCA, and PCA. Shimodaira (2016). Neural Networks.
"Cross-validation of matching correlation analysis by resampling matching weights"

CDMCA (2-view case)

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \boldsymbol{A}^{(2)} \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}^{(11)} & \boldsymbol{W}^{(12)} \\ \boldsymbol{W}^{(21)} & \boldsymbol{W}^{(22)} \end{bmatrix} \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^{(1)} & \\ & \boldsymbol{M}^{(2)} \end{bmatrix} \qquad m_i = \sum_{j=1}^{N} w_{ij}$$

CCA is given by $\boldsymbol{W} = \begin{bmatrix} & \boldsymbol{I}_n \\ \boldsymbol{I}_n & \end{bmatrix}$ $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{I}_n & \\ & \boldsymbol{I}_n \end{bmatrix}$

$$\max_{\boldsymbol{A} \in \mathbb{R}^{N \times K}} \boldsymbol{A}^\top \boldsymbol{H} \boldsymbol{A} \qquad \boldsymbol{H} = \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}^{(1)\top} \boldsymbol{W}^{(11)} \boldsymbol{X}^{(1)} & \boldsymbol{X}^{(1)\top} \boldsymbol{W}^{(12)} \boldsymbol{X}^{(2)} \\ \boldsymbol{X}^{(2)\top} \boldsymbol{W}^{(21)} \boldsymbol{X}^{(1)} & \boldsymbol{X}^{(2)\top} \boldsymbol{W}^{(22)} \boldsymbol{X}^{(2)} \end{bmatrix}$$

subject to $\boldsymbol{A}^\top \boldsymbol{G} \boldsymbol{A} = \boldsymbol{I}_K$ $\qquad \boldsymbol{G} = \boldsymbol{X}^\top \boldsymbol{M} \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}^{(1)\top} \boldsymbol{M}^{(1)} \boldsymbol{X}^{(1)} & \\ & \boldsymbol{X}^{(2)\top} \boldsymbol{M}^{(2)} \boldsymbol{X}^{(2)} \end{bmatrix}$

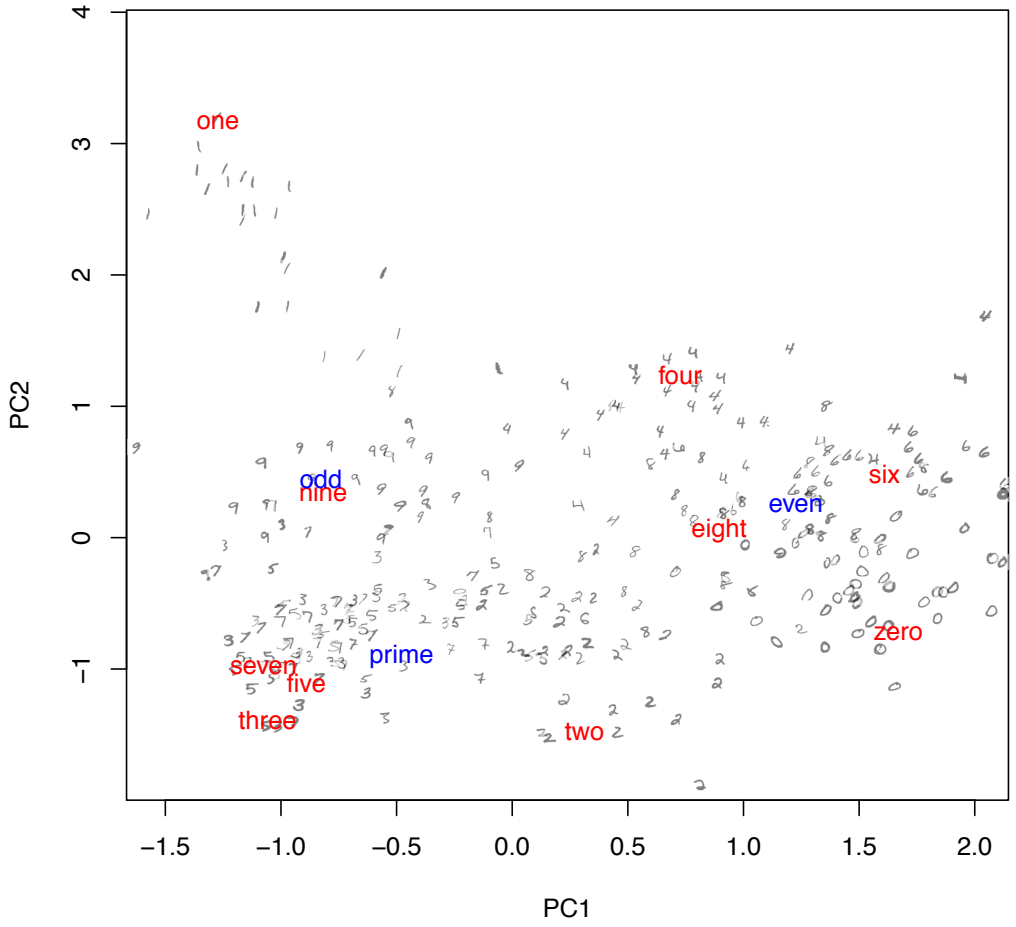
CDMCA is solved as a generalized eigenvalue problem

$$\boldsymbol{G}^{-1/2\top} \boldsymbol{H} \boldsymbol{G}^{-1/2} \boldsymbol{u_i} = \lambda_i \boldsymbol{u}_i \qquad \lambda_1 \geq \lambda_2 \geq \cdots$$

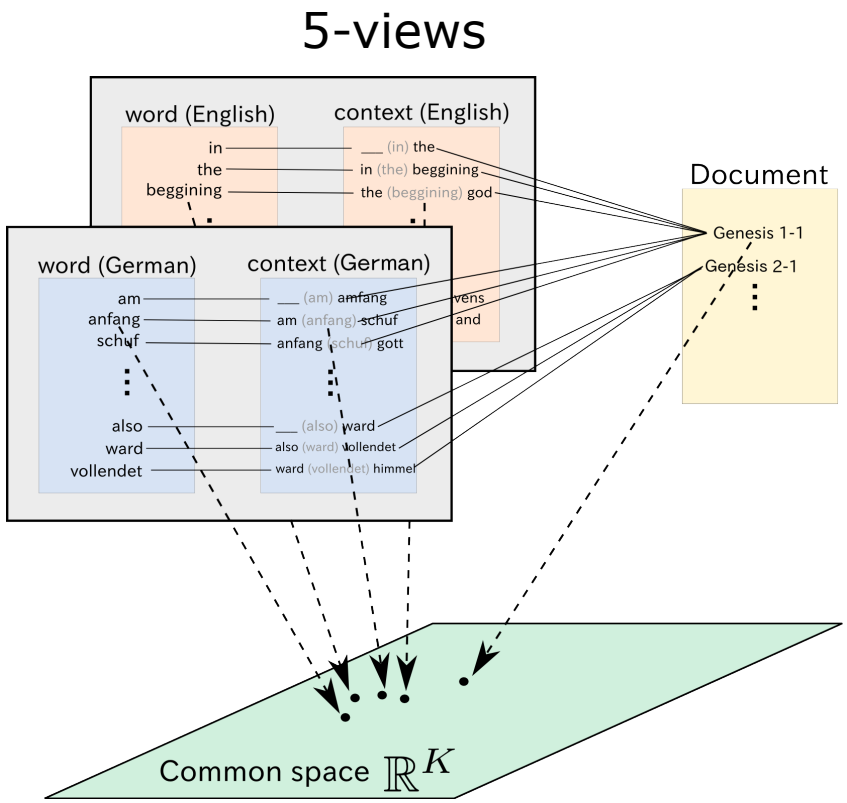
$$\hat{\boldsymbol{A}} = \boldsymbol{G}^{-1/2} (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)$$

MNIST handwritten digits

3-views: {0,1,…,9} – images – {prime,even,odd}



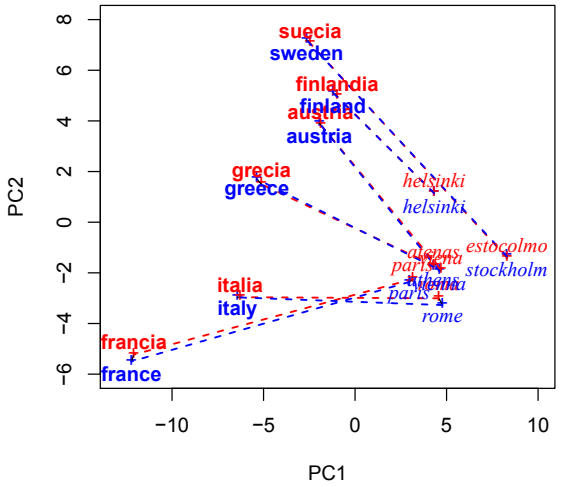
Bilingual word representations



$$\mathbf{X} = \begin{pmatrix} \mathbf{V}^{(1)} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{C}^{(1)} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{V}^{(2)} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{C}^{(2)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{D} \end{pmatrix},$$

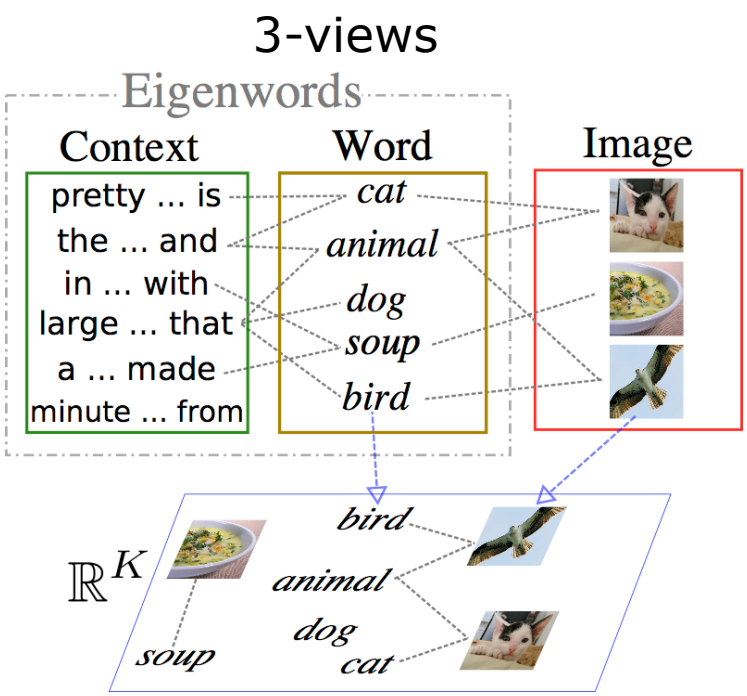
$$\mathbf{W} = \begin{pmatrix} \mathbf{O} & \mathbf{I}_{T^{(1)}} & \mathbf{O} & \mathbf{O} & \tilde{\mathbf{J}}^{(1)} \\ \mathbf{I}_{T^{(1)}} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \tilde{\mathbf{J}}^{(1)} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{I}_{T^{(2)}} & \tilde{\mathbf{J}}^{(2)} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}_{T^{(2)}} & \mathbf{O} & \tilde{\mathbf{J}}^{(2)} \\ \tilde{\mathbf{J}}^{(1)\top} & \tilde{\mathbf{J}}^{(1)\top} & \tilde{\mathbf{J}}^{(2)\top} & \tilde{\mathbf{J}}^{(2)\top} & \mathbf{O} \end{pmatrix},$$

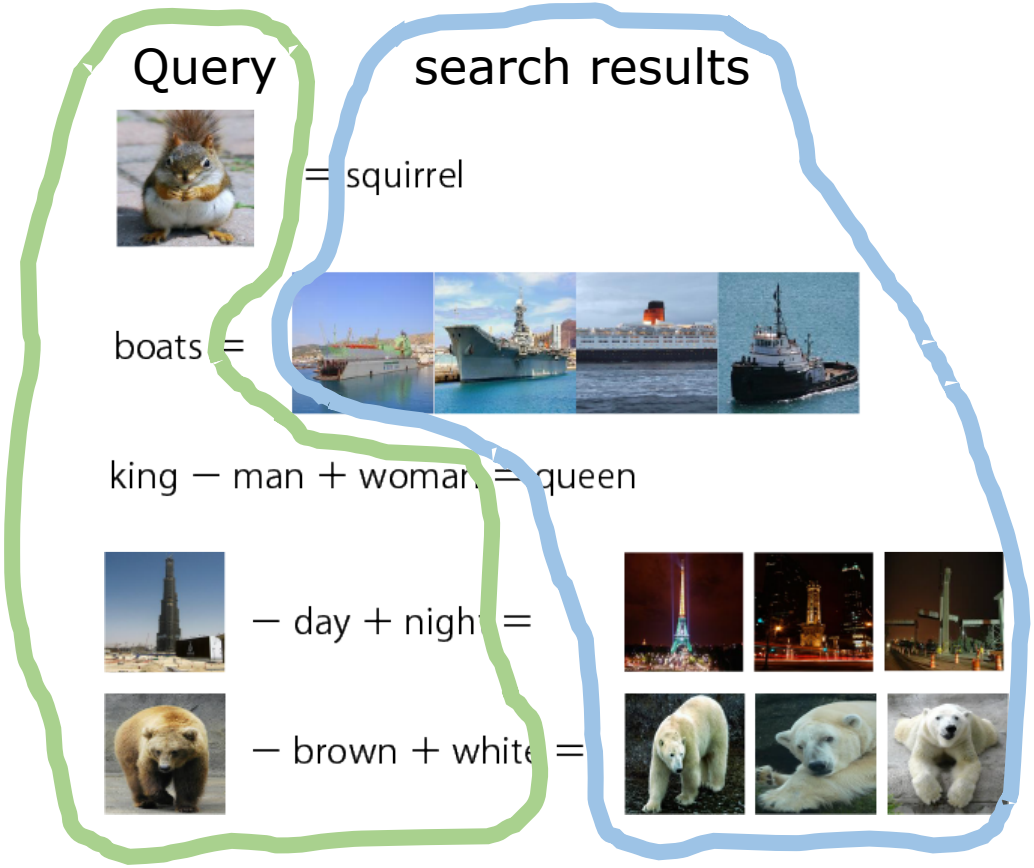
$$\mathbf{A}^\top = (\mathbf{A}_V^{(1)\top}, \mathbf{A}_C^{(1)\top}, \mathbf{A}_V^{(2)\top}, \mathbf{A}_C^{(2)\top}, \mathbf{A}_D^\top).$$



T. Oshikiri, K. Fukui, H. Shimodaira (2016). Cross-Lingual Word Representations via Spectral Graph Embeddings, The 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016.

Flickr image-tag search and “linguistic regularities”

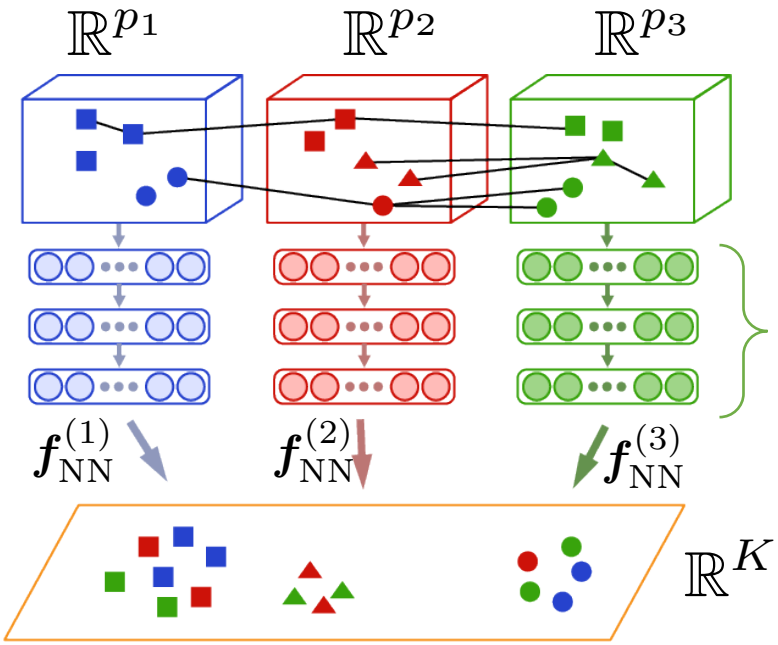




Fukui, Oshikiri, Shimodaira (2017). Spectral Graph-Based Method of Multimodal Word Embedding. TextGraphs-11, the Workshop on Graph-based Methods for Natural Language Processing, ACL 2017.

Graph embedding with Neural Networks

PMvGE: Probabilistic Multi-view Graph Embedding



$$i = 1, \ldots, n$$
$$\boldsymbol{x}_i \in \mathbb{R}^{p_{d_i}}$$
$$d_i \in \{1, 2, 3\}$$

Non-linear transformations via Neural Networks

$$\boldsymbol{y}_i = \boldsymbol{f}_{\mathrm{NN}}^{(d_i)}(\boldsymbol{x}_i)$$
$$w_{ij} \sim \mathrm{Po}(\alpha^{(d_i, d_j)} e^{\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle})$$

Likelihood is optimized by SGD without the constraints on y's

Okuno, Hada, Shimodaira (2018). A probabilistic framework for multi-view feature learning with many-to-many associations via neural networks. The International Conference on Machine Learning, ICML 2018.

Optimization of PMvGE

Log-likelihood: $$\ell_n(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}_n} w_{ij}\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle - \sum_{(i,j)\in\mathcal{I}_n} \exp(\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle)$$

$$\mathcal{P}_n = \{(i,j) : w_{ij} > 0,\ 1 \le i < j \le n\}$$

$$\mathcal{I}_n = \{(i,j) : 1 \le i < j \le n\}$$

In each iteration, Stochastic Gradient Decent with mini-batch uses:

$$\ell_{\text{mini}}(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}_{\text{mini}}} w_{ij}\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle - \tau \sum_{(i,j)\in\mathcal{I}_{\text{mini}}} \exp(\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle)$$

$\mathcal{P}_{\text{mini}}, \mathcal{I}_{\text{mini}}$ are mini-batch sampling of $\mathcal{P}_n, \mathcal{I}_n$

MLE estimates the true similarity up to a constant factor

$$\lim_{n\to\infty} \exp(\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle) = \text{const} \times \mathbb{E}(w_{ij}|\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Okuno and Shimodaira (arXiv 2019) Hyperlink Regression via Bregman Divergence

PMvGE generalizes CCA and CDMCA to non-linear transformations

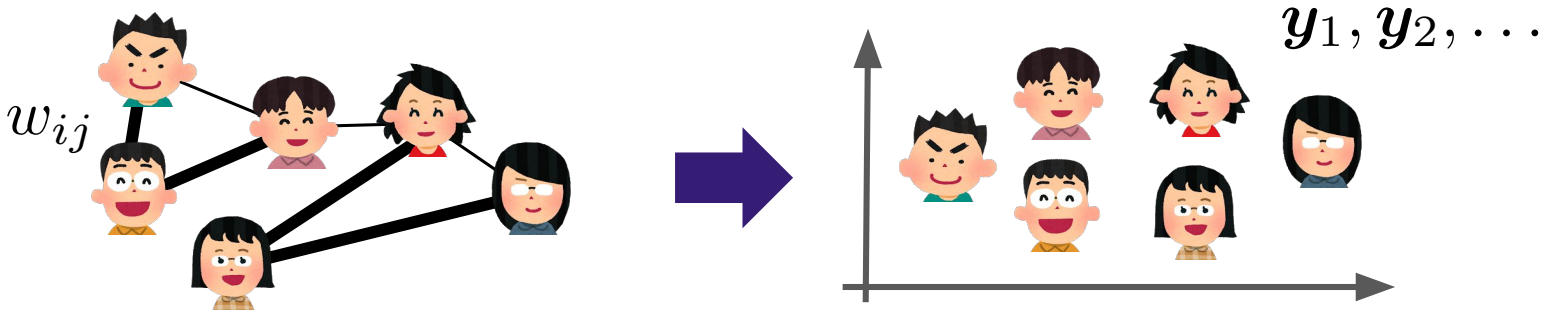
(Nv): Number of views, (MM): Many-to-many, (NL): Non-linear, (Ind): Inductive, (Lik): Likelihood-based.
PMvGE has all the properties. Nv = D represents that the method can deal with arbitrary number of views.

	(Nv)	(MM)	(NL)	(Ind)	(Lik)
CCA (Hotelling 1936)	2			✓	
Deep CCA (Andrew et al. 2013)	2		✓	✓	
MCCA (Kettenring 1971)	D			✓	
SGE (Belkin et al. 2001)	0	✓			
LINE (Tang et al. 2015)	0	✓			✓
LPP (He et al. 2004)	1	✓		✓	
CvGE (Huang et al. 2013)	2	✓		✓	
CDMCA (Shimodaira 2016)	D	✓		✓	
DeepWalk (Perozzi et al. 2014)	0	✓			✓
SBM (Holland et al. 1983)	1	✓		✓	✓
GCN (Kipf et al. 2017)	1	✓	✓		✓
GraphSAGE (Hamilton et al. 2017)	1	✓	✓	✓	✓
IDW (Dai et al. 2018)	1	✓	✓	✓	✓
PMvGE (Proposed)	D	✓	✓	✓	✓

Representation learning

Similarities are expressed via vector representations

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 $\mathbb{R}^K$

$$\mathbb{E}(w_{ij}|\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(s_{ij})$$

Similarity: s_{ij}

Attributes: $\boldsymbol{x}_i$

$$s_{ij} = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$$

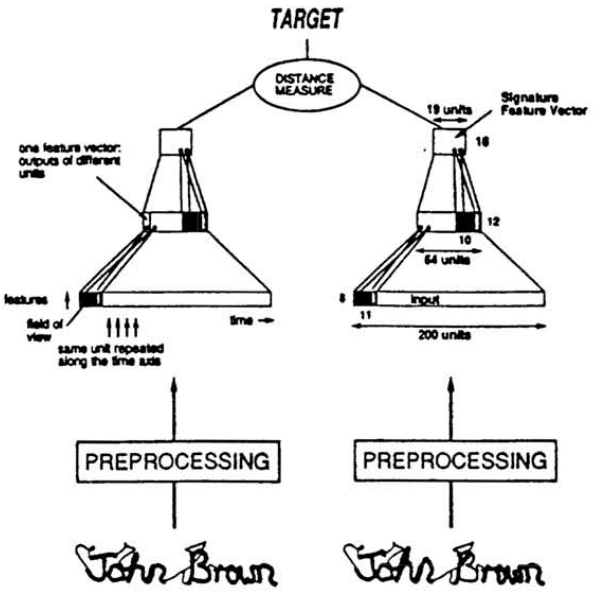
Embedding: $\boldsymbol{y}_i$
(vector representation)

NN-based similarity learning: Siamese (twin) neural networks

$$\langle \boldsymbol{y}, \boldsymbol{y}' \rangle = y_1 y_1' + \cdots + y_K y_K'$$

$\boldsymbol{y} = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x})$

(K-dim vector)



$\boldsymbol{y}' = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}')$

(K-dim vector)

$\boldsymbol{x}$ $\boldsymbol{x}'$

Bromley, Guyon, Le Cun, Sackinger, Shah (1994). "Signature verification using a "Siamese" time delay neural network." NIPS.

Why many methods are based on inner product of vector representations?

The true underlying similarity

$$s(\boldsymbol{x}, \boldsymbol{x}')$$

Inner Product Similarity (IPS)
"Neural networks + Inner product"

$$h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$$

$$s_{ij} = s(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad h_{ij} = h_{\mathrm{IPS}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix} \overset{\text{approx.}}{\approx} \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & & \ddots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nn} \end{bmatrix}$$

Inner product is a key to good vector representations

The distributive law

$$\langle \boldsymbol{a} + \boldsymbol{b}, \boldsymbol{c} \rangle = \langle \boldsymbol{a}, \boldsymbol{c} \rangle + \langle \boldsymbol{b}, \boldsymbol{c} \rangle$$

"linguistic regularities"

$$\langle \boldsymbol{y}_{\text{king}} - \boldsymbol{y}_{\text{man}} + \boldsymbol{y}_{\text{woman}}, \boldsymbol{y}' \rangle = \langle \boldsymbol{y}_{\text{king}}, \boldsymbol{y}' \rangle - \langle \boldsymbol{y}_{\text{man}}, \boldsymbol{y}' \rangle + \langle \boldsymbol{y}_{\text{woman}}, \boldsymbol{y}' \rangle$$

will be maximized by $\boldsymbol{y}' = \boldsymbol{y}_{\text{queen}}$

Inner Product Similarity (IPS) can approximate any Positive Definite (PD) similarity

Theorem 1 (Okuno, Kim, Shimodaira 2018)

$s_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}')$ is a positive definite (PD) similarity.

For arbitrary small $\epsilon > 0$

there exists $h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$ such that

$$|s_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}') - h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}')| < \epsilon \quad \text{for any} \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$$

$$s_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}') \Leftrightarrow h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}')$$

Proof: assuming s_{PD} is continuous on a compact set *X*, the theorem follows by combining Mercer's theorem of kernels and Universal approximation theorem of NN (Funahashi, 1989; Cybenko, 1989; Yarotsky, 2017; Telgarsky, 2017)

Positive definite kernel

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j s_{\mathrm{PD}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

for arbitrary $c_1, c_2, \ldots, c_n \in \mathbb{R}, \quad \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathcal{X}$

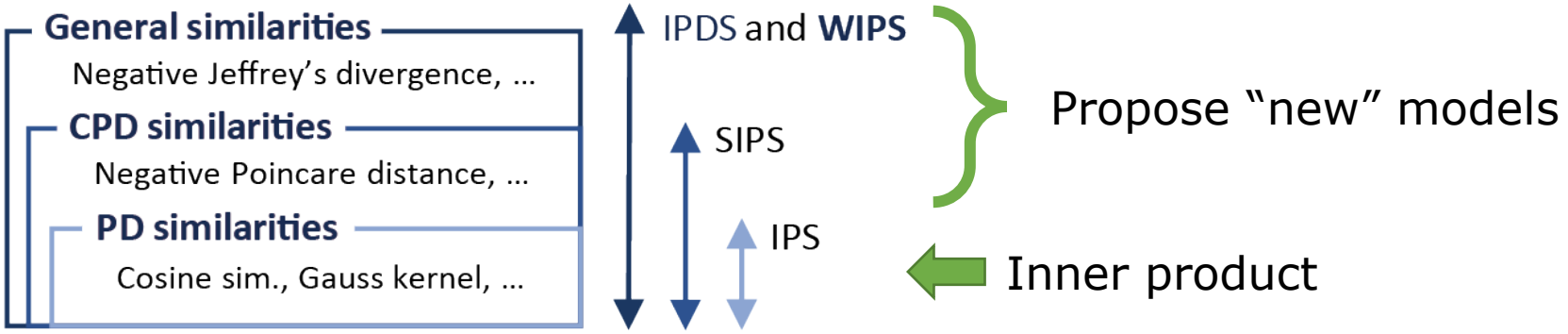
(example) $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}),\ \boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{x}'); \quad s_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}') = g(\boldsymbol{y}, \boldsymbol{y}')$

$g(\boldsymbol{y}, \boldsymbol{y}') = \langle \boldsymbol{y}, \boldsymbol{y}' \rangle$ It is obvious that IPS is PD.

$$g(\boldsymbol{y}, \boldsymbol{y}') = \left\langle \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}, \frac{\boldsymbol{y}'}{\|\boldsymbol{y}'\|} \right\rangle$$

(counter example) $g(\boldsymbol{y}, \boldsymbol{y}') = -\|\boldsymbol{y} - \boldsymbol{y}'\|^2$

A hierarchy of similarity models



Note: we are not attempting

$$s_{\text{general}}(\boldsymbol{x}, \boldsymbol{x}') \Leftrightarrow f_{\text{NN}}(\boldsymbol{x}, \boldsymbol{x}')$$

Adding bias terms to IPS

Okuno, Kim, Shimodaira (2019). Graph Embedding with Shifted Inner Product Similarity and Its Improved Approximation Capability. AISTATS2019.

Inner product similarity (IPS)

$$h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle f_{\mathrm{NN}}(\boldsymbol{x}), f_{\mathrm{NN}}(\boldsymbol{x}') \rangle$$

Shifted Inner product similarity (SIPS)

$$h_{\mathrm{SIPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle + u_{\mathrm{NN}}(\boldsymbol{x}) + u_{\mathrm{NN}}(\boldsymbol{x}')$$

Bias terms have been used in some applications
(e.g. recommender systems)

Shifted Inner Product Similarity (SIPS) can approximate any Conditionally Positive Definite (CPD) similarity

Theorem 2 (Okuno, Kim, Shimodaira 2019)

$s_{\mathrm{CPD}}(\boldsymbol{x}, \boldsymbol{x}')$ is a conditionally positive definite (CPD) similarity.

For arbitrary small $\epsilon > 0$

there exists $h_{\mathrm{SIPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle + u_{\mathrm{NN}}(\boldsymbol{x}) + u_{\mathrm{NN}}(\boldsymbol{x}')$

such that $|s_{\mathrm{CPD}}(\boldsymbol{x}, \boldsymbol{x}') - h_{\mathrm{SIPS}}(\boldsymbol{x}, \boldsymbol{x}')| < \epsilon \quad \text{for any} \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$

$$s_{\mathrm{CPD}}(\boldsymbol{x}, \boldsymbol{x}') \Leftrightarrow h_{\mathrm{SIPS}}(\boldsymbol{x}, \boldsymbol{x}')$$

Proof: combining Mercer's theorem + the Universal approximation theorem + Berg et al. (1984)

Conditionally Positive Definite (CPD) kernel

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j s_{\mathrm{CPD}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

for arbitrary $c_1, c_2, \ldots, c_n \in \mathbb{R}, \quad \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathcal{X}$

with the condition $\sum_{i=1}^{n} c_i = 0$

(example) $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}),\ \boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{x}'); \quad s_{\mathrm{CPD}}(\boldsymbol{x}, \boldsymbol{x}') = g(\boldsymbol{y}, \boldsymbol{y}')$

$$g(\boldsymbol{y}, \boldsymbol{y}) = -\mathrm{distance}(\boldsymbol{y}, \boldsymbol{y}')$$

$$d_{\mathrm{Euclid}}(\boldsymbol{y}, \boldsymbol{y}') = \|\boldsymbol{y} - \boldsymbol{y}'\|_2^{\alpha}, \quad 0 < \alpha \leq 2$$

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

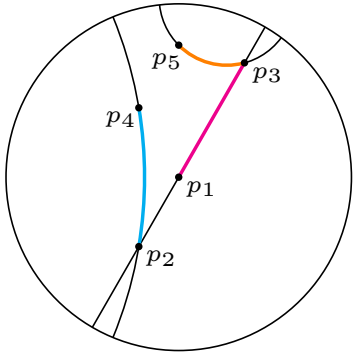
$$d_{\text{Wasserstein-}p}(\boldsymbol{y}, \boldsymbol{y}') = \inf_{\pi \in \Pi(\boldsymbol{y}, \boldsymbol{y}')}\left(\int\int_{\mathcal{Z}\times\mathcal{Z}} d_Z(z, z')^p d\pi(z, z')\right)^{1/p}$$

p=1 if $-d_Z$ is CPD
p=2 if Z = R

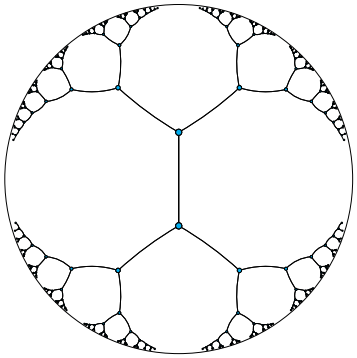
The reason why bias terms are necessary for CPD kernel

$$-\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{y}'\|^2 = \langle \boldsymbol{y}, \boldsymbol{y}' \rangle - \frac{1}{2}\|\boldsymbol{y}\|^2 - \frac{1}{2}\|\boldsymbol{y}'\|^2$$

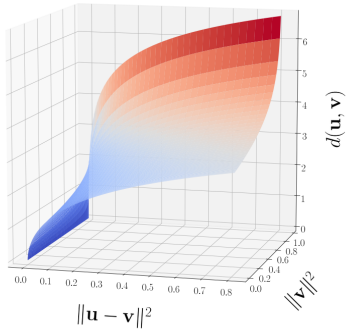
A tree structure is easily expressed in Poincaré Nickel & Kiela (NIPS 2017)



(a) Geodesics of the Poincaré disk

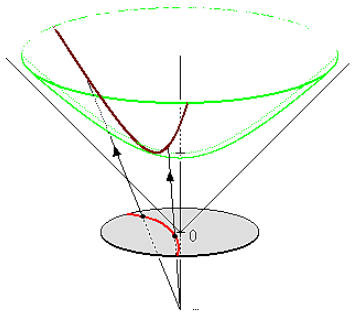


(b) Embedding of a tree in $\mathcal{B}^2$



(c) Growth of Poincaré distance

Figure 1: (a) Due to the negative curvature of $\mathcal{B}$, the distance of points increases exponentially (relative to their Euclidean distance) the closer they are to the boundary. (c) Growth of the Poincaré distance $d(\boldsymbol{u}, \boldsymbol{v})$ relative to the Euclidean distance and the norm of $\boldsymbol{v}$ (for fixed $\|\boldsymbol{u}\| = 0.9$). (b) Embedding of a regular tree in $\mathcal{B}^2$ such that all connected nodes are spaced equally far apart (i.e., all black line segments have identical hyperbolic length).



Hyperbolid model and Poincare disk model (from Wikipedia)

Circle Limit III (from Wikipedia)

Inner-Product Difference Similarity (IPDS) can approximate any general similarities

$$s_{\mathrm{general}}(\boldsymbol{x}, \boldsymbol{x}') = s^{+}_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}') - s^{-}_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{x}')$$

(Indefinite kernel: Ong et al., 2004)

Theorem 3 (Okuno, Kim, Shimodaira 2019)

$$h_{\mathrm{IPDS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}') \rangle - \langle \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$$

$$s_{\mathrm{general}}(\boldsymbol{x}, \boldsymbol{x}') \Leftrightarrow h_{\mathrm{IPDS}}(\boldsymbol{x}, \boldsymbol{x}')$$

Vector representation: $\boldsymbol{y} = (\boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x})) \in \mathbb{R}^{K}$

$$\boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}) \in \mathbb{R}^{K-q}, \quad \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x}) \in \mathbb{R}^{q}$$

Choosing q from (0,K) is important but it can be difficult in practice

Weighted Inner Product Similarity (WIPS) avoids model selection

G. Kim, A. Okuno, K. Fukui & H. Shimodaira (2019). Representation Learning with Weighted Inner Product for Universal Approximation of General Similarities. Proceedings of 28th International Joint Conference on Artificial Intelligence (IJCAI-19).

Weighted inner product allows negative weight values

$$\langle \boldsymbol{y}, \boldsymbol{y}' \rangle_{\boldsymbol{\lambda}} = \lambda_1 y_1 y'_1 + \lambda_2 y_2 y'_2 + \cdots + \lambda_K y_K y'_K$$

$\lambda_1, \lambda_2, \ldots, \lambda_K \in \mathbb{R}$ are parameters to be learned

Theorem 4 (Kim, Okuno, Fukui, Shimodaira 2019)

$$h_{\mathrm{WIPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle_{\boldsymbol{\lambda}}$$

$$s_{\mathrm{general}}(\boldsymbol{x}, \boldsymbol{x}') \Leftrightarrow h_{\mathrm{WIPS}}(\boldsymbol{x}, \boldsymbol{x}')$$

WIPS performs very well

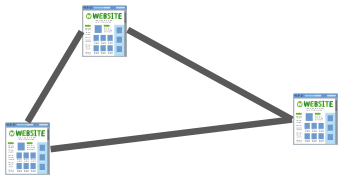
Experiments - **datasets**

3 graph-structured datasets are used.

Hypertext Network

877 nodes and 1480 links

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



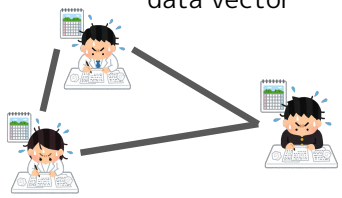
edge : hyperlink relation

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

Co-authorship Network

41328 nodes and 210320 links

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

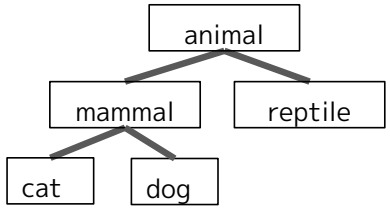


edge : co-author relation

Taxonomy Tree

37623 nodes and 312885 link

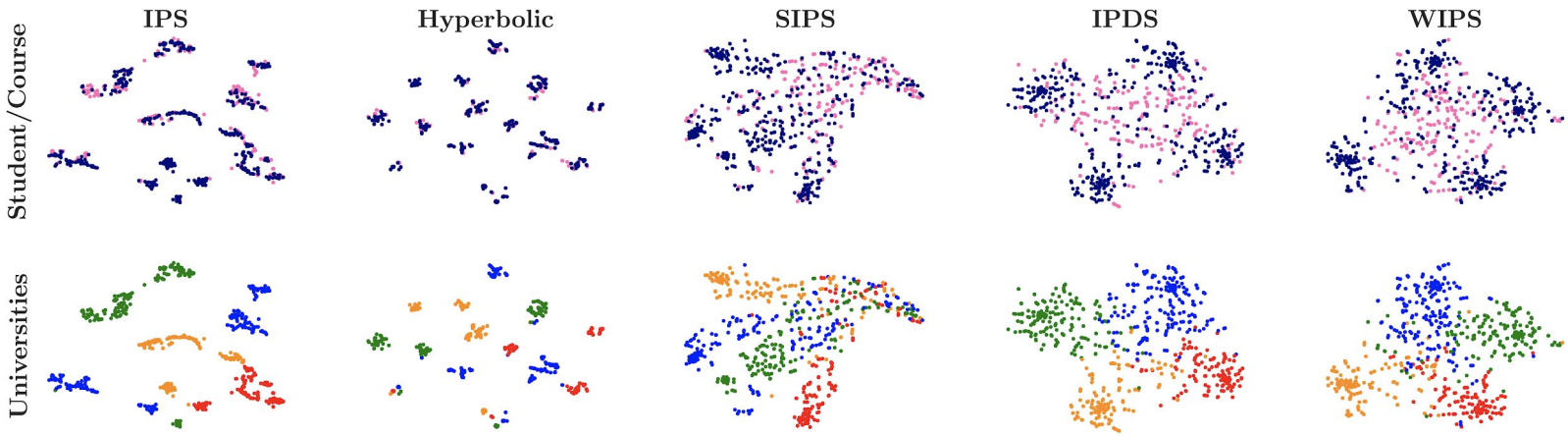
node : word,
attr. : 300 dim. pretrained
Google's word embedding



edge : hyponymy-hypernymy relation

From presentation slides of Kim, Okuno, Fukui, Shimodaira (IJCAI 2019)

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.

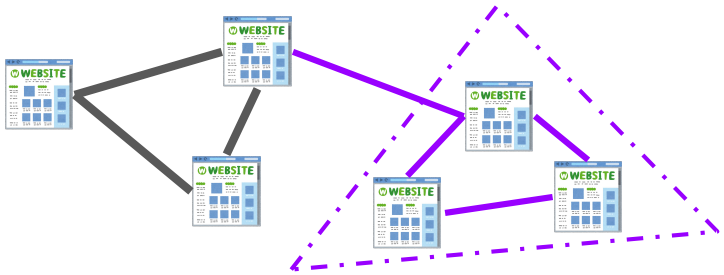
Hypertext Classification

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



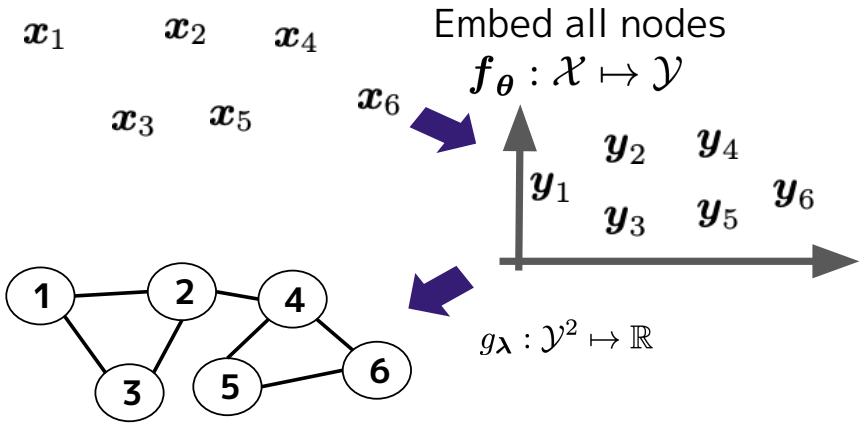
test set (invisible) is used for evaluation

	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**

Graph Reconstruction

At training, **assume that all nodes and links are visible**. Use all data to train the model (f_θ and g_λ).

Then, at evaluation,



Predict (reconstruct) all links

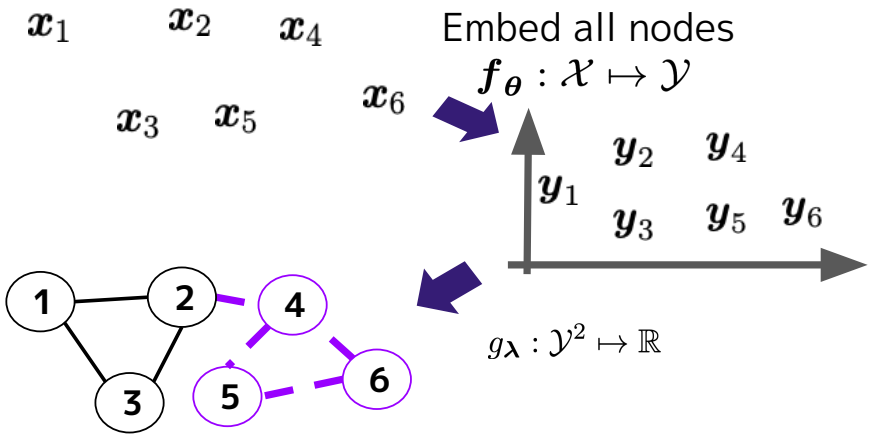
ROC-AUC for prediction errors are calculated ▶

		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 (f_θ and g_λ).

Then, at evaluation,

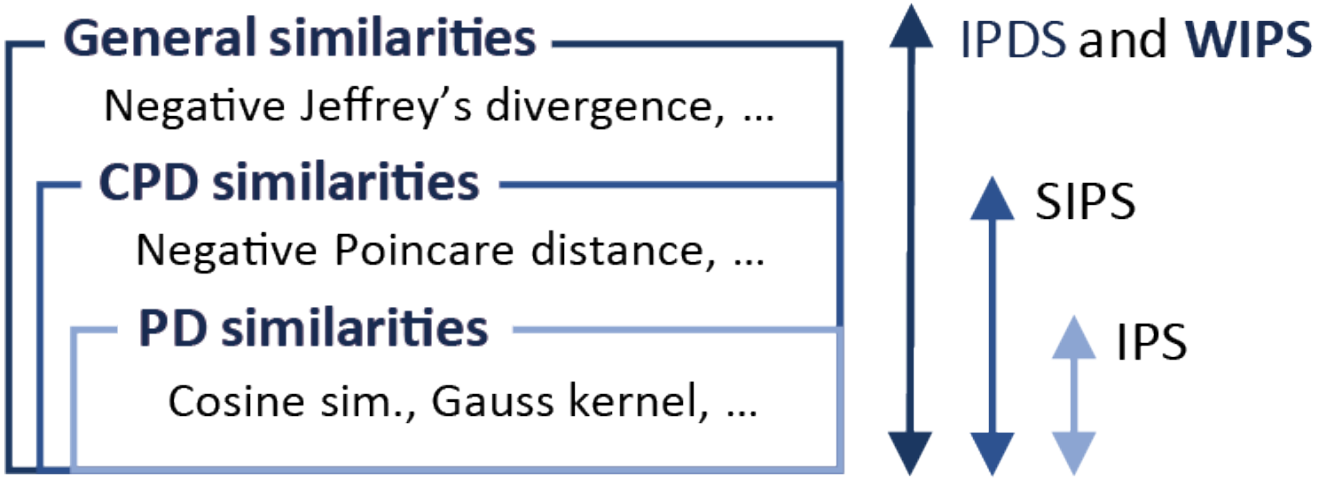


Predict (reconstruct) all
unobserved links

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**

Summary of similarity models



$$h_{\mathrm{WIPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle_{\boldsymbol{\lambda}}$$

$$h_{\mathrm{IPDS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}^{+}_{\mathrm{NN}}(\boldsymbol{x}') \rangle - \langle \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}^{-}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$$

$$h_{\mathrm{SIPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle + u_{\mathrm{NN}}(\boldsymbol{x}) + u_{\mathrm{NN}}(\boldsymbol{x}')$$

$$h_{\mathrm{IPS}}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}), \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{x}') \rangle$$

References

- A. Okuno and H. Shimodaira. Hyperlink Regression via Bregman Divergence. arXiv:1908.02573, 2019
- G. Kim, A. Okuno, K. Fukui & H. Shimodaira, Representation Learning with Weighted Inner Product for Universal Approximation of General Similarities, Proceedings of 28th International Joint Conference on Artificial Intelligence (IJCAI-19), Main track, 5031-5038, 2019
- A. Okuno & H. Shimodaira, Robust Graph Embedding with Noisy Link Weights, Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 89, 664-673, 2019
- A. Okuno, G. Kim & H. Shimodaira, Graph Embedding with Shifted Inner Product Similarity and Its Improved Approximation Capability, Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 89, 644-653, 2019
- A. Okuno, T. Hada & H. Shimodaira, A probabilistic framework for multi-view feature learning with many-to-many associations via neural networks, Proceedings of the 35th International Conference on Machine Learning (ICML), PMLR 80, 3888-3897, 2018
- K. Fukui, T. Oshikiri & H. Shimodaira, Spectral Graph-Based Method of Multimodal Word Embedding. TextGraphs-11, the Workshop on Graph-based Methods for Natural Language Processing, ACL 2017
- M. Nickel and D. Kiela, Poincare´Embeddings for Learning Hierarchical Representations. In Advances in Neural Information Processing Systems (NIPS) 30, 6338–6347, 2017
- K. Fukui, A. Okuno & H. Shimodaira, Image and tag retrieval by leveraging image-group links with multi-domain graph embedding, 2016 IEEE International Conference on Image Processing (ICIP), 221–225, 2016
- T. Oshikiri, K. Fukui & H. Shimodaira, Cross-Lingual Word Representations via Spectral Graph Embeddings, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), 493–498, 2016
- H. Shimodaira, Cross-validation of matching correlation analysis by resampling matching weights, Neural Networks, 75, 126-140, 2016
- Harold Hotelling, Relations Between Two Sets of Variates, Biometrika, 28, 321-377, 1936