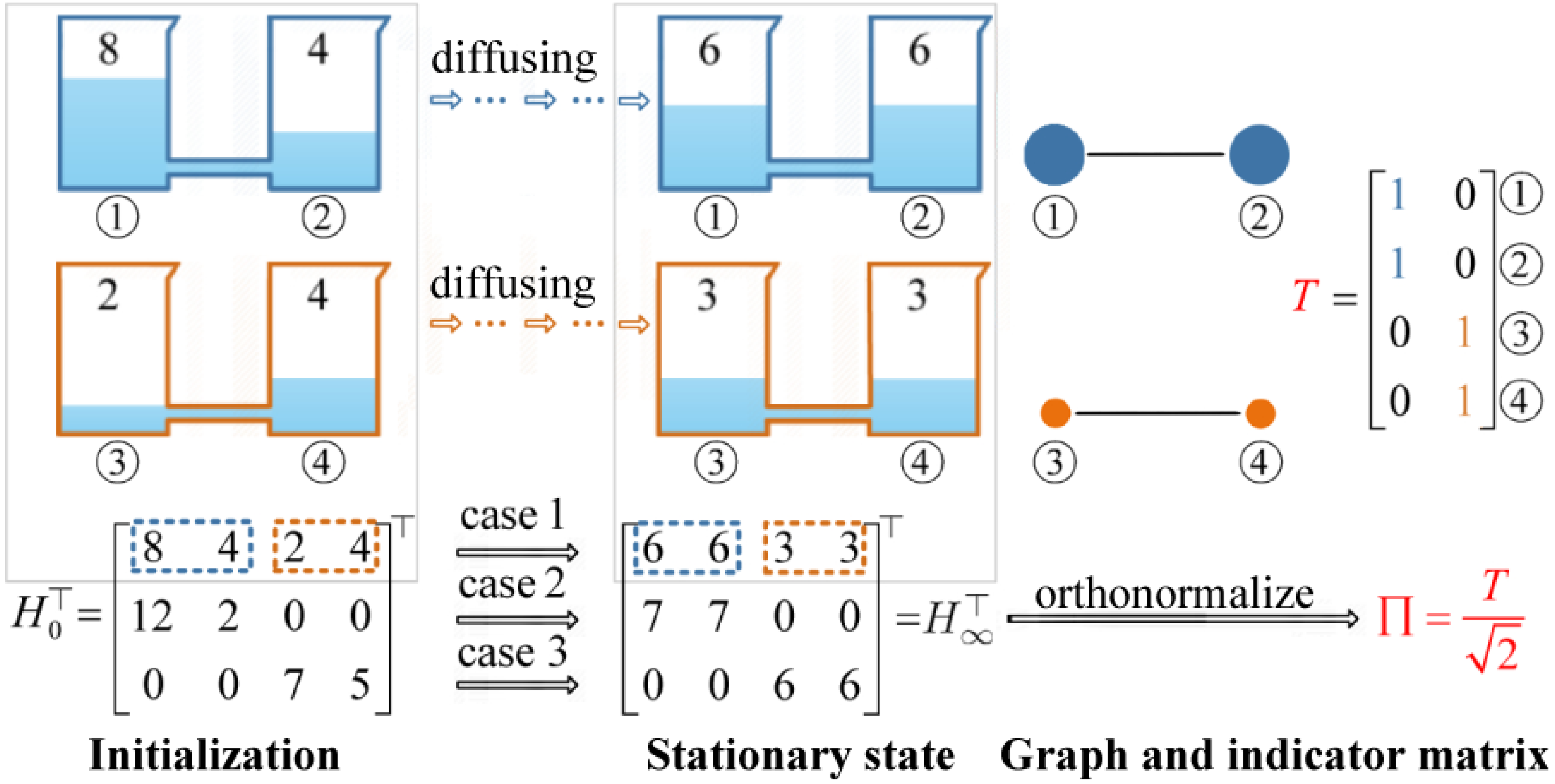
Stationary Diffusion State Neural Estimation for Multiview Clustering

Chenghua Liu, Zhuolin Liao, Yixuan Ma and Kun Zhan*

{liuchh20,liaozhl20,mayx2021,kzhan}@lzu.edu.cn

AAAI-22

Problem



Contribution

1. We introduce the Stationary Diffusion State Neural Estimation (SDSNE).
2. We show the multiview clustering utility of this estimator is derived from the shared parameter between views.
3. We design a co-supervised loss to guide SDSNE in achieving the stationary state.

Diffusion

We define the transition probability matrix $P = [p_{ij}], \forall ij, p_{ij} = p_{ji} \geqslant 0, \sum_i p_{ij} = 1$, and the graph diffusion process is given by

$$\boldsymbol{h} \leftarrow P\boldsymbol{h} \tag{1}$$

The following statements are equivalent for the Markov chain determined by P

1. The Markov chain is stationary at the state of $\boldsymbol{\pi}$.
2. $\boldsymbol{\pi} = P\boldsymbol{\pi}$.
3. $\sum_{i,j=1}^{n} p_{ij}(\pi_i - \pi_j)^2 = 0$.

Estimator

The k eigenvectors corresponding to eigenvalue 1 of P constructs the matrix $\Pi = [\boldsymbol{\pi}_i]$, and $H = [\boldsymbol{h}_i]$ is constrained by $H^\top H = I$. Then, the inequality $0 = \sum_{i=1}^{k} \boldsymbol{\pi}_i^\top (I - P)\boldsymbol{\pi}_i = \min_{H^\top H = I} \mathrm{Tr}(H^\top (I-P)H) \leqslant \sum_{i=1}^{k} \boldsymbol{h}_i^\top (I - P)\boldsymbol{h}_i$ holds. Then, we have

$$\mathrm{Tr}(\Pi^\top (I-P)\Pi) \leqslant \mathrm{Tr}(H^\top (I-P)H)$$

we use

$$\mathrm{Tr}(H^\top (I-P)H) \tag{2}$$

as its loss function and define Eq. (1) to be its neural layer.

Hyper transition matrix

Given different transition matrices, $P^{(1)}$ and $P^{(2)}$, in two views, we construct a hyper-transition matrix with them,

$$\boldsymbol{P} = P^{(1)} \otimes P^{(2)} \tag{3}$$

Method

The diffusion with $\boldsymbol{P}$ is given by

$$\boldsymbol{g} \leftarrow \boldsymbol{P}\boldsymbol{g} = \mathrm{vec}\big(P^{(2)} S (P^{(1)})^\top\big) \tag{4}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization by stacking columns one by one and $\mathrm{vec}(S) = \boldsymbol{g}$. We model the diffusion as a layer of the neural network and we share model parameter W in different views, for $\forall\, v \in \{1, 2, \ldots, n_v\}$

$$H^{(v)} \leftarrow P^{(v)} W (P^{(v)})^\top . \tag{5}$$

Fuse the learned features $H^{(v)}$ to obtain a unified global feature by,

$$H = \alpha \sum_{v=1}^{n_v} H^{(v)} + (1-\alpha) I \tag{6}$$

According to Eq. (2), the loss function guides SDSNE in obtaining a stationary state by minimizing,

$$\mathcal{L} = \sum_{v=1}^{n_v} \big(\mathrm{Tr}(H^\top (I - \hat{P}^{(v)})H) + \mu \|H^{(v)}\|_{\mathrm{F}}\big) \tag{7}$$

Algorithm

Input: $\mathcal{X} = \{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\}$;
Output: H;
Initialize α, μ, W, and $epoch_{\max}$;
for $v \in \{1, 2, \ldots, n_v\}$ **do**
    Construct $A^{(v)}$ by the Gaussian kernel with $X^{(v)}$;
    Calculate the degree matrix $D^{(v)}$ of $A^{(v)}$;
    Normalize $A^{(v)}$ by $P^{(v)} = (D^{(v)})^{-\frac{1}{2}} A^{(v)} (D^{(v)})^{-\frac{1}{2}}$;
repeat
    for $v \in \{1, 2, \ldots, n_v\}$ **do**
        $H^{(v)} \leftarrow P^{(v)} W (P^{(v)})^\top$;
        Calculate the degree matrix $\hat{D}^{(v)}$ of $H^{(v)}$;
        $\hat{P}^{(v)} = (\hat{D}^{(v)})^{-\frac{1}{2}} H^{(v)} (\hat{D}^{(v)})^{-\frac{1}{2}}$;
    Update H by Eq. (6);
    Update $\mathcal{L}$ by Eq. (7);
    Update W by the gradient descent algorithm.
until *converged*

Results

Clustering on multiview datasets. We show the clustering accuracy between SDSNE and two representative method: CGD and O2MAC on six benchmark datasets.

We perform the k-means clustering and spectral clustering to obtain the clustering results (denoted by $\mathrm{SDSNE_{km}}$ and $\mathrm{SDSNE_{sc}}$, respectively).

In most cases, SDSNE outperforms other state-of-the-art methods.

More algorithms and measure metrics can refer to our paper.

Dataset	BBC Sport	MSRC-v1	100 Leaves	Three Sources	Scene-15	Reuters
CGD	0.974±0.004	0.910±0.006	0.859±0.005	0.781±0.006	0.428±0.004	0.492±0.004
O2MAC	0.964±0.008	0.709±0.030	0.557±0.009	0.755±0.026	0.309±0.013	0.459±0.039
$\mathrm{SDSNE_{km}}$	0.969±0.000	0.943±0.000	0.962±0.000	0.828±0.000	0.443±0.000	0.516±0.000
$\mathrm{SDSNE_{sc}}$	0.985±0.000	0.933±0.000	0.957±0.000	0.935±0.000	0.436±0.000	0.522±0.000