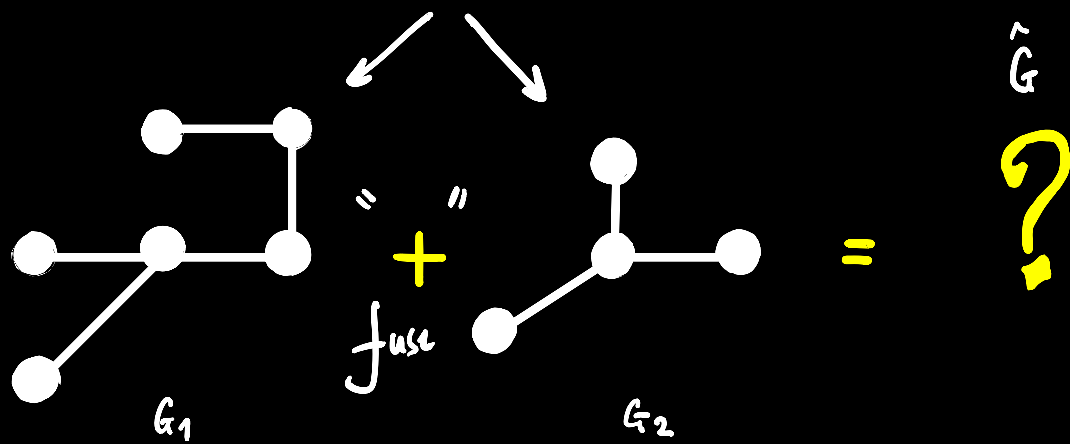
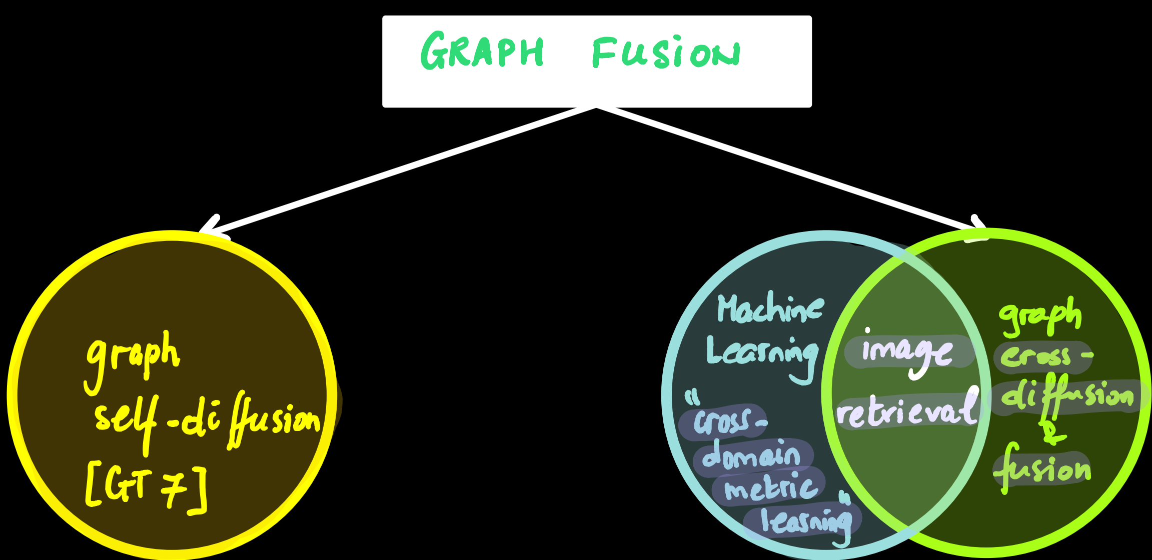
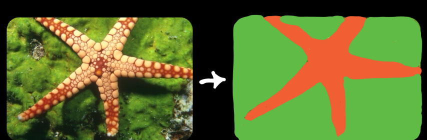
GraphT_8

Given two graphs G_1 and G_2





[Wang et al., CVPR 2012]



Unsupervised Metric Fusion by Cross Diffusion

Bo Wang[1], Jiayan Jiang[2], Wei Wang[4], Zhi-Hua Zhou[4], and Zhuowen Tu[2,3]
[1]Department of Computer Science, University of Toronto
[2]Lab of Neuro Imaging, University of California, Los Angeles
[3]Microsoft Research Asia
[4]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China

Abstract

Metric learning is a fundamental problem in computer vision. Different features and algorithms may tackle a problem from different angles, and thus often provide complementary information. In this paper, we propose a fusion algorithm which outputs enhanced metrics by combining multiple given metrics (similarity measures). Unlike traditional co-training style algorithms where multi-view features or multiple data subsets are used for classification or regression, we focus on fusing multiple given metrics through diffusion process in an unsupervised way. Our algorithm has its particular advantage when the input similarity matrices are the outputs from diverse algorithms. We provide both theoretical and empirical explanations to our method. Significant improvements over the state-of-the-art results have been observed on various benchmark datasets. For example, we have achieved 100% accuracy (no longer the bull's eye measure) on the MPEG-7 shape dataset. Our method has a wide range of applications in machine learning and computer vision.

1. Introduction

Data samples are often given as high dimensional points, whereas they live in much lower intrinsic spaces (manifolds); utilizing the intrinsic data manifold structure therefore is an important topic in learning and vision [29, 21]; computing faithful manifold metrics (distance/similarity measures) leads to good performances in a wide range of applications in classification, segmentation, regression, image search/retrieval, and visualization [24, 16]. For high dimensional data, a direct approach of distance metric learning (e.g. Mahalanobis distances) [15, 36, 23, 7] is often used in the context of supervised learning.

One idea to derive a good distance measure is to explicitly construct a new embedding space with distance propagation which is more faithful to the manifold structure and hence induces a better distance notion. The same idea can be extended to semi-supervised cases, where a limited portion of data labels are given. For example, label propagation [40] and its variants [32] use a diffusion process to propagate labels to the unlabeled data samples along the manifold. One promising approach, diffusion map [5], defines a new metric, diffusion distances, between data samples; an input similarity matrix is then improved through a diffusion process.

The diffusion/propagation process improves the individual inputs; in practice, we are often given multiple measures by different algorithms/metrics, which are also complementary to each other. We therefore are facing an additional fusion [17] task on top of the metric learning problem.

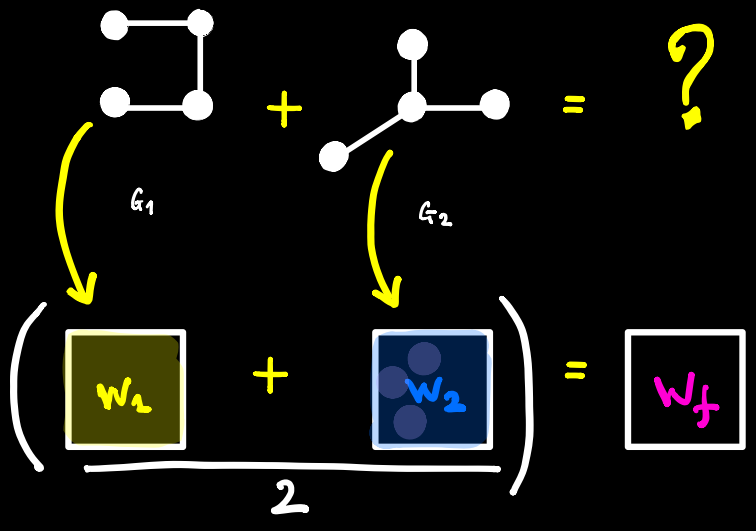
From a different angle, co-training style algorithms allow classifiers trained on different views of the features [4, 6, 31, 25] or different subsets of the training data [9, 39] to pull out more samples from unlabeled data to help each other. PAC bounds were given for co-training on multi-view features [6] and tri-training on multi-classifiers [34]. Recently, co-regularization has been adopted for multi-view learning [31, 27]. However, there are few algorithms which address the problem in an unsupervised manner. In [38] an unsupervised Bayesian kernel was proposed to fuse the information induced from multiple views. This simple fusion technique suffers from being sensitive to the parameters in the algorithm and noise in the data. In this paper, we develop a dynamic process to fuse multiple metrics in an unsupervised way.

In terms of application, shape/image retrieval is an important topic in computer vision; due to being intrinsically high-dimensional and ambiguous, shape/image retrieval remains a challenging problem. Recent advances in this category attempts to apply transductive or semi-supervised learning [37] to enhance the retrieval results. Depending on the affinity relationship on a simple graph, semi-supervised learning techniques could be utilized to boost the image retrieval performance. However, it may not be sufficient to represent the full affinity relations by only one fixed graph.

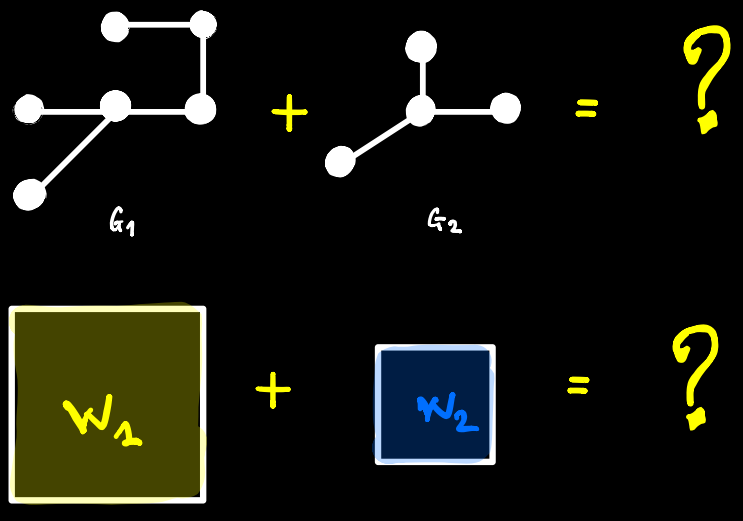
In this paper, we propose a metric learning algorithm, *cross-diffusion process*, for generating enhanced similarity

How to merge / fuse / integrate graphs ?

case 1 $|G_1| = |G_2|$ same number of nodes



case 2 $|G_1| \neq |G_2|$



☹ Limited literature on graph fusion methods.

☺ Design new methods to fuse multi-dimensional graphs.

Previously

* $\mathcal{G}raph\mathcal{T}_7$

original image we want to segment

desired output: segmentation map / labeled image

Table 2: *Affinity Learning via Self-diffusion for Image Segmentation and Clustering (Wang et al., CVPR 2012).*

Mathematical notation	Definition
N	number of nodes in a graph
x_i	i^{th} data point (or sample) comprising m features
$d(i,j)$	distance between samples $x_i \in \mathbb{R}^m$ and $x_j \in \mathbb{R}^m$
W	graph weighted adjacency matrix (or data similarity matrix) $\in \mathbb{R}^{N\times N}$
$W(i,j) = exp(-d^2(i,j)/\sigma^2)$	the edge weight connecting nodes i and j
σ	kernel size
$D(i,i) = \sum_{k=1}^{n} W(i,k)$	strength of node i and D is a diagonal matrix
$P = D^{-1}W$	transition kernel = row-normalized similarity matrix W
$W_t = W_{t-1}P + I$	smoothed similarity matrix at iteration t
$W^* = W_t^* D^{-1}$	W_t^* self-normalization

diffuse W via the kernel P

$$W_t = W_{t-1}P + I$$
$$W_{t+1} = W_t P + I$$
$$= (W_{t-1} P + I) P + I$$
$$= W_{t-1} P^2 + P + I$$

Theorem $\mathcal{G}raph\mathcal{T}_5$

The number of walks of length k joining any two nodes v_i and v_j of a binary graph is given by the (i,j) entry of the matrix W^k:

$$N_k(v_i, v_j) = W^k(i,j)$$

* Cross-diffusion for
1. graph fusion
2. metric (data similarity) fusion
   → cross-domain metric learning

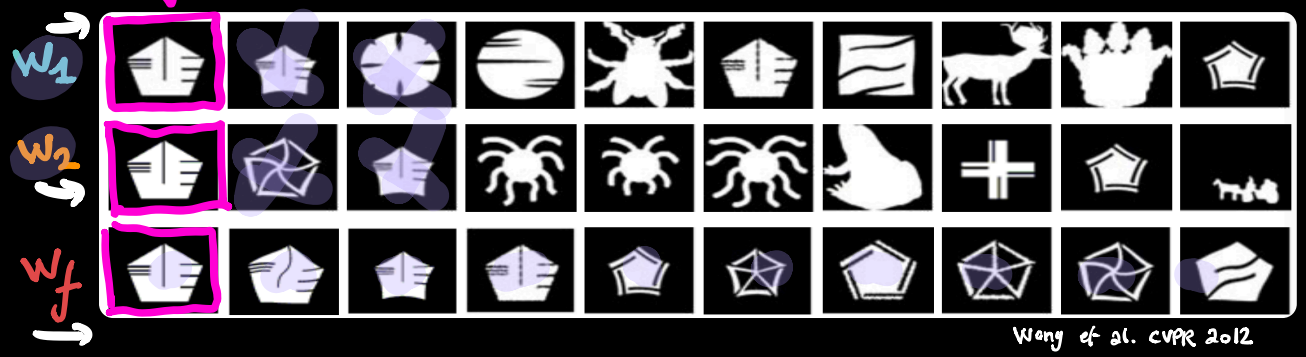
[Wang et al., CVPR 2012]

⇓

More powerful generalization of graph diffusion concept

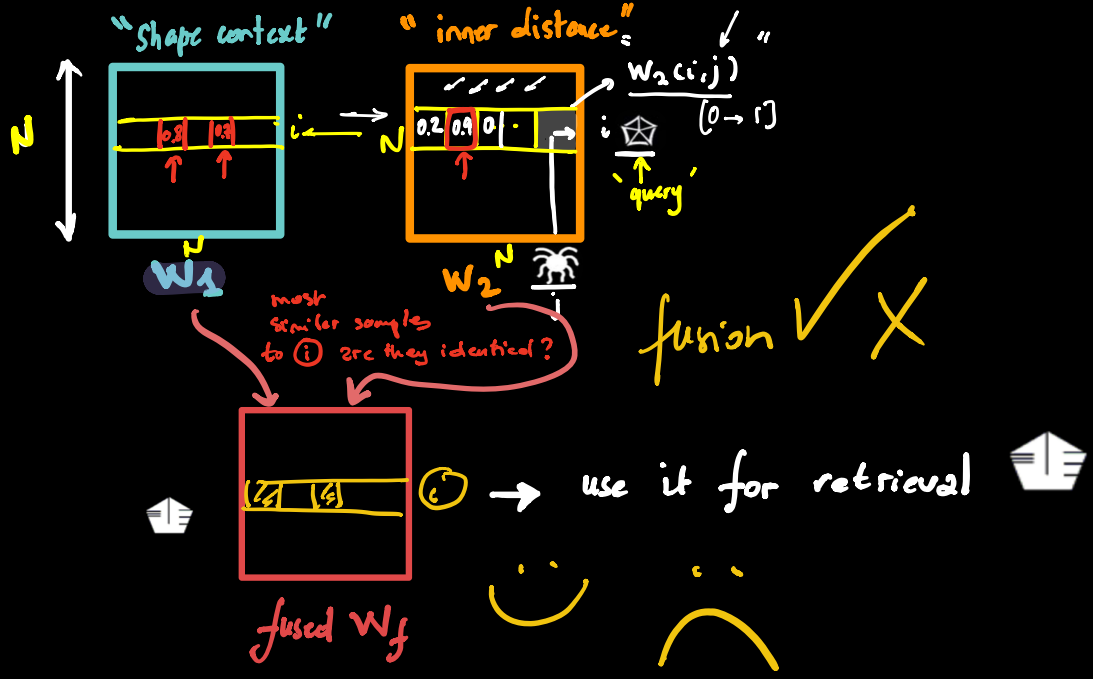
* What are the potential applications of unsupervised metric fusion ?

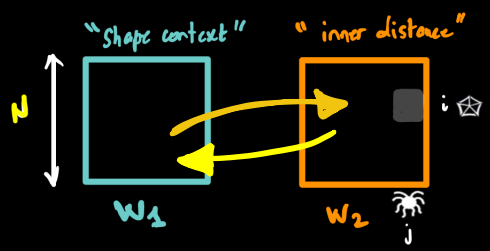
query image
↓ retrieve all similar "images"

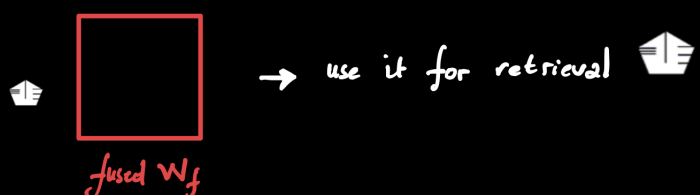


Wang et al. CVPR 2012

- Given a dataset with different N shapes, build two similarity matrices W_1 and W_2 capturing different types of pair wise relationships between samples :







problem I want to solve

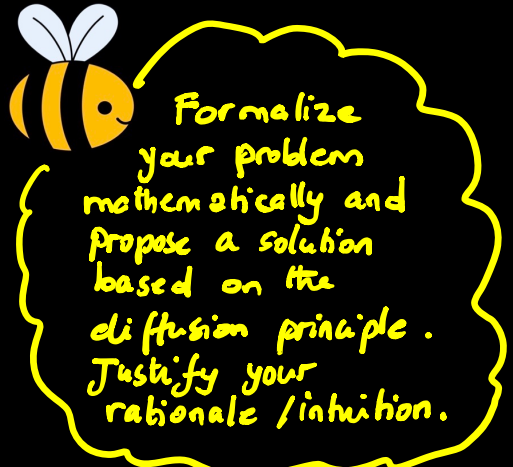


Table 2: *Affinity Learning via Self-diffusion for Image Segmentation and Clustering (Wang et al., CVPR 2012).*

Mathematical notation	Definition
N	number of nodes in a graph
x_i	i^{th} data point (or sample) comprising m features
$d(i,j)$	distance between samples $x_i \in \mathbb{R}^m$ and $x_j \in \mathbb{R}^m$
W	graph weighted adjacency matrix (or data similarity matrix) $\in \mathbb{R}^{N\times N}$
$W(i,j) = exp(-d^2(i,j)/\sigma^2)$	the edge weight connecting nodes i and j
σ	kernel size
$D(i,i) = \sum_{k=1}^{n} W(i,k)$	strength of node i and D is a diagonal matrix
$P = D^{-1}W$	transition kernel = row-normalized similarity matrix W
$W_t = W_{t-1}P + I$	smoothed similarity matrix at iteration t
$W^* = W_t^* D^{-1}$	W_t^* self-normalization

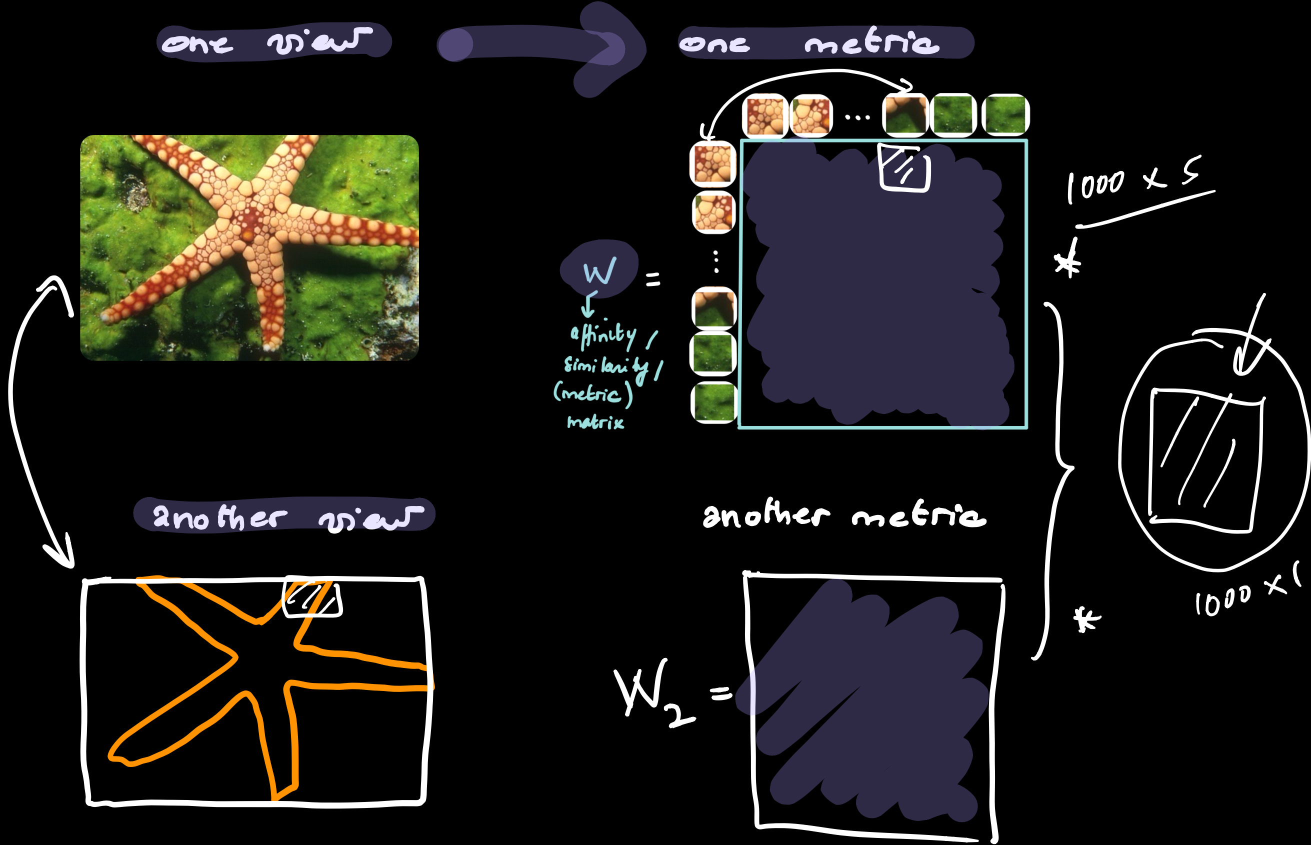
How to solve it using graph diffusion?

→ Solution proposed by Bo Wang et al. CVPR 2012 → Nature 2014.

[Wang et al., CVPR 2012]

NEW IDEA = CROSS-DIFFUSE TO FUSE

- One sample can be represented by different views → present complementary information about the sample.



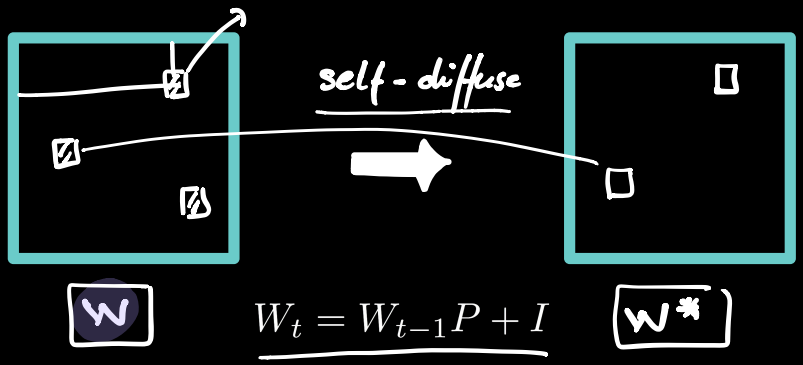
- Propose a **fusion algorithm** which outputs enhanced **metrics** by combining multiple given metrics (**similarity measures**).

TRADITIONAL	NOVEL
* train a model using **multi-view data** samples to { **classify** / **regress** / **cluster** }	* fuse multiple given **metrics** (**data views**) in a fully **unsupervised** way.

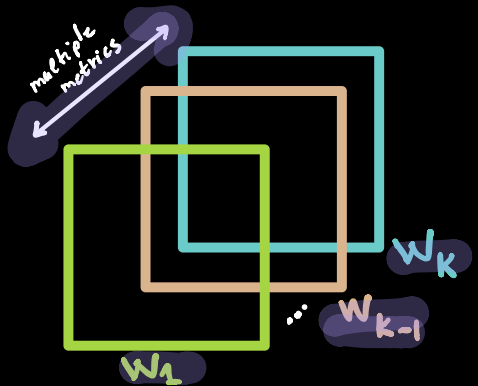
$\mathcal{M} \infty \mathcal{L}_2$

Types of ML methods:
1. cross-validation demo < LOO / 5 fold / 10 fold
2. supervised learning <predictive learning> { classification / regression }
3. unsupervised learning (clustering)
4. semi-supervised learning & more

☺ [GT 7] The **diffusion/propagation** process improves the **individual inputs**.



* In practice, we are often given **multiple measures** by different algorithms / metrics, which are **complementary to each other**.



⬇

we are facing an additional **graph fusion** task on top of the **graph learning** problem.

- Given a finite weighted graph $G = (V, E, W)$

  - V: set of vertices based on the dataset $X = \{x_i,\ i = 1, \dots, N\}$

  - E: a set of edges
  - W: non-negative symmetric weight function

    $W : E \to [0,1]$

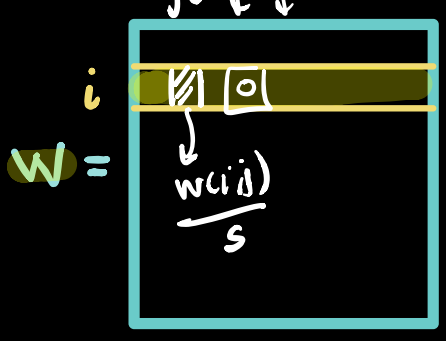
    $W(i,j)$ → similarity measure between samples i and j.

- A natural kernel acting on functions on V can be defined by normalization of the weight matrix as follows:

$$P(i,j) = \frac{W(i,j)}{s = \sum_{k \in V} W(i,k)}$$

$W(i,j)$: element (i,j) in graph adjacency matrix W

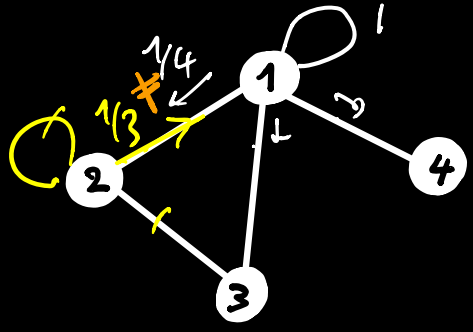
s: sum of weights of adjacent neighbors k of node i including self-adjacency



$$\sum_{j \in V} P(i,j) = 1$$

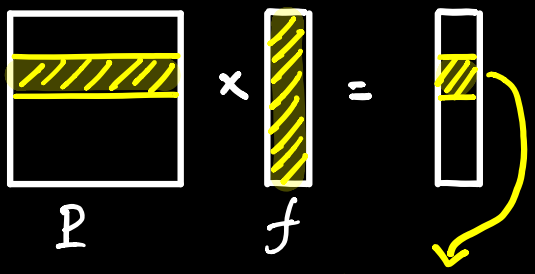
Is P symmetric?

$P(i,j) \neq P(j,i)$

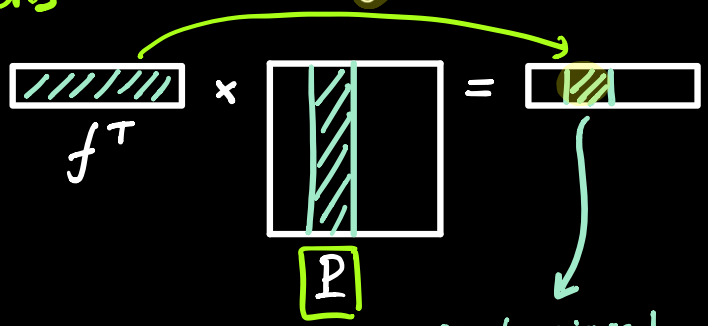


• For any label vector or probability distribution f :

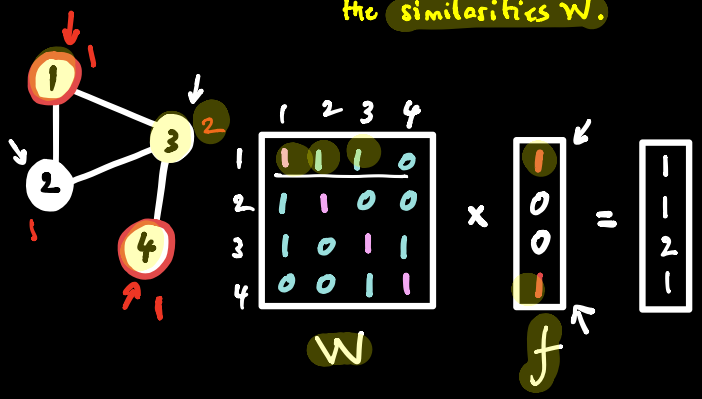
Pf ⟶ "diffusion on graphs" ⟵ $f^T P$

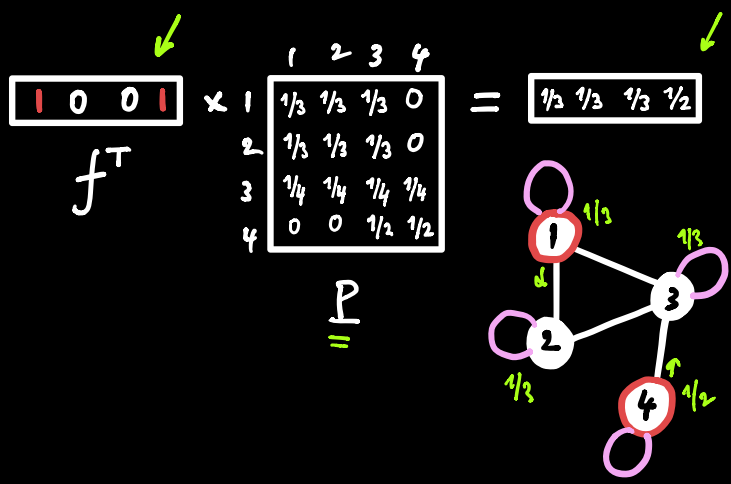


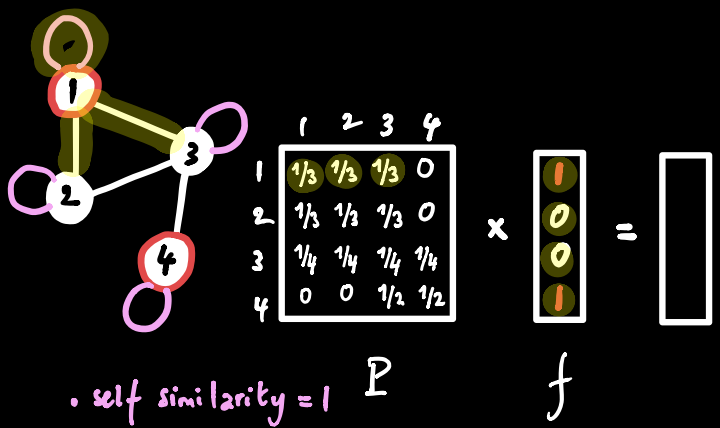
local averaging operation with locality measured by the similarities W.



can be viewed as a Markov walk of the vector f on G.







• self similarity = 1

→ Both Pf and $f^T P$ can loosely be considered as a diffusion process.

How this'll help us solve our graph cross-diffusion and fusion problem?

* Let $X \triangleq (x_1, \dots, x_N)$ denote a dataset with weight matrix W.

• **Step 1** : built a subgraph $\mathcal{G}$ of G where local similarities (high values) are preserved.

Hypothesis : local similarities are more reliable than far-away ones → can be propegated to non-local points through a diffusion process on G.

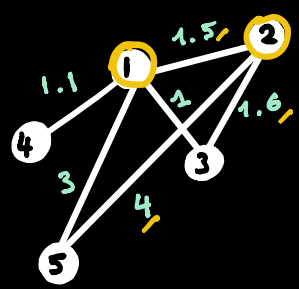
$\mathcal{G} = (V, E, \mathcal{W})$

↳ localized similarity matrix (local neighborhood)

$$\mathcal{W}(i,j) = \begin{cases} W(i,j) \text{ if } x_j \in KNN(x_i) \\ 0 \text{ otherwise} \end{cases}$$

3 = k nearest neighbors

K nearest neighbors of node x_i → neighboring samples most similar to x_i



	1	2	3	4	5
1	1	1.5	1	1.1	3
2	1.5	1	1.6	0	4
3	1	1.6	1	0	0
4	1.1	0	0	1	0
5	3	4	0	0	1

W

K = 1 →

	1	2	3	4	5
1		0	1	0	0
2	1.5		0	0	0
3					
4					
5					

$\mathcal{W}$

• **Step 2** : build the corresponding local kernel S :

$$S(i,j) = \frac{\mathcal{W}(i,j)}{\sum_{x_k \in KNN(x_i)} \mathcal{W}(i,k)}$$

normalized local similarity matrix → $S(i,j)$; $\in \mathbb{R}^{N \times N}$

localized adjacency matrix → $\mathcal{W}(i,j)$

row-wise normalization based on the most similar K samples x_k to sample x_i

$$P(i,j) = \frac{W(i,j)}{\sum_{k \in V} W(i,k)}$$

element (i,j) in graph adjacency matrix W

sum of weights of adjacent neighbors k of node i

$$S(i,j) = \frac{W(i,j)}{\sum_{x_k \in KNN(x_i)} W(i,k)}$$

normalized local similarity matrix

$S(i,j) \in \mathbb{R}^{N \times N}$

localized adjacency matrix

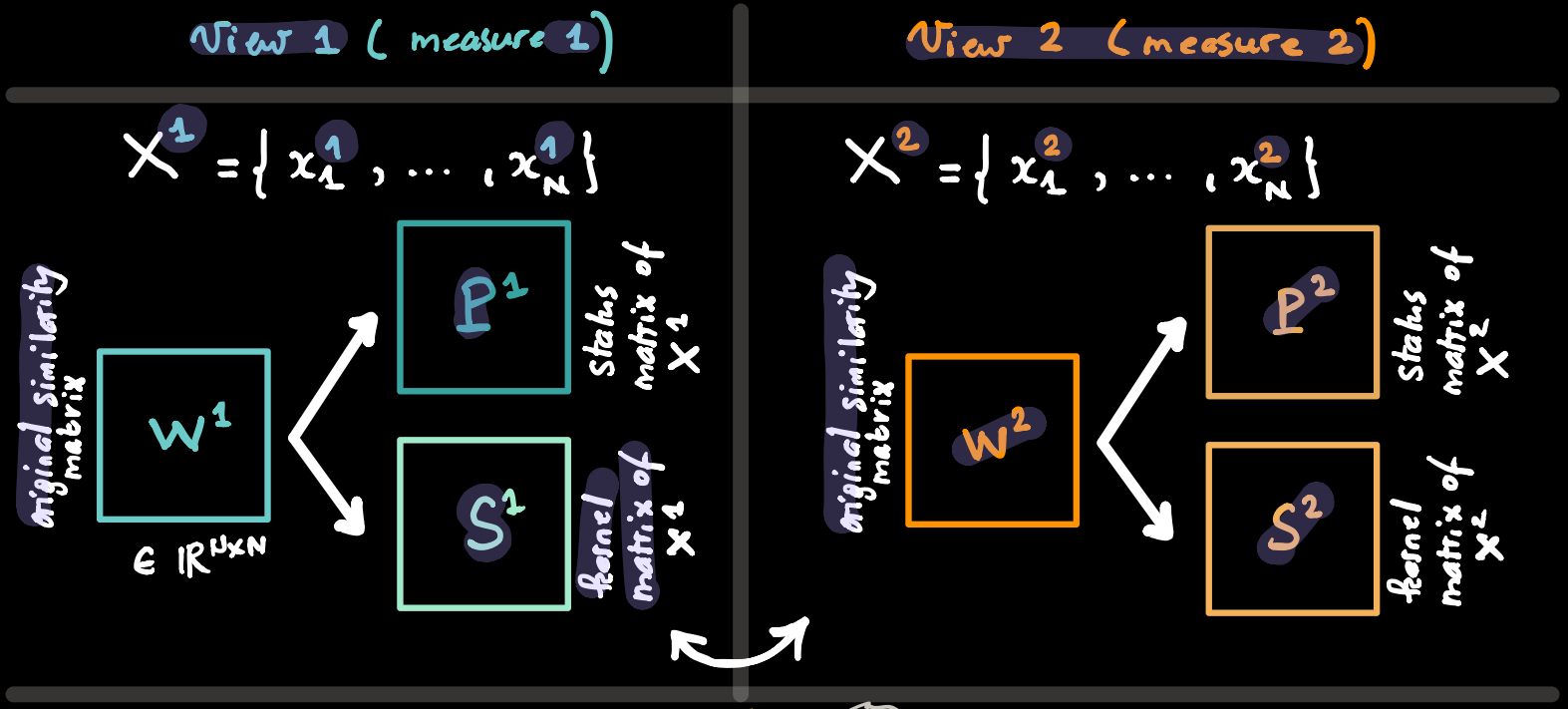
row-wise normalization based on the most similar k samples x_k to sample x_i

* P carries the full information about the similarity of each data point to all others.
* P = status matrix

* S only encodes the similarity to nearby points (local neighborhood).
* S = kernel matrix

Idea: diffuse P along the local structure of S (using kernel S).

- **Step 3**: graph / metric fusion via cross-diffusion with $m = 2$ similarity measures



Pf $f^T P$

iteratively cross-diffuse

$$P^1_{t+1} = S^1 \times P^2_t \times (S^1)^T + \eta I$$

$$P^2_{t+1} = S^2 \times P^1_t \times (S^2)^T + \eta I$$

local topology of G_1

global topology of G_2

diffusing P^2 across S^1

global

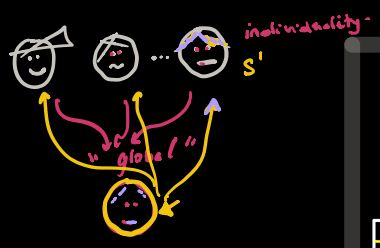
"local"

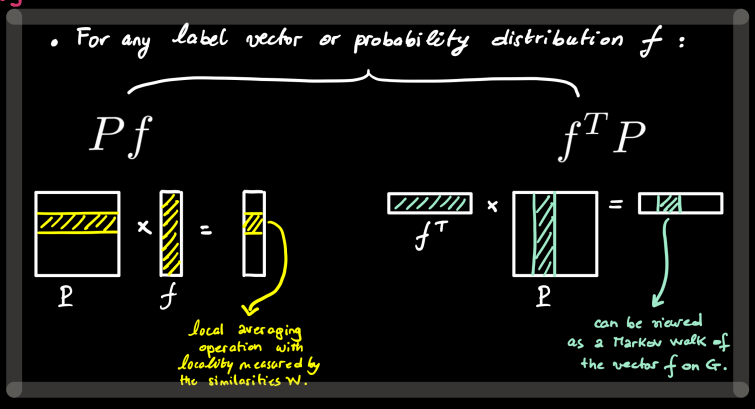
$P^1_0 = P^1$

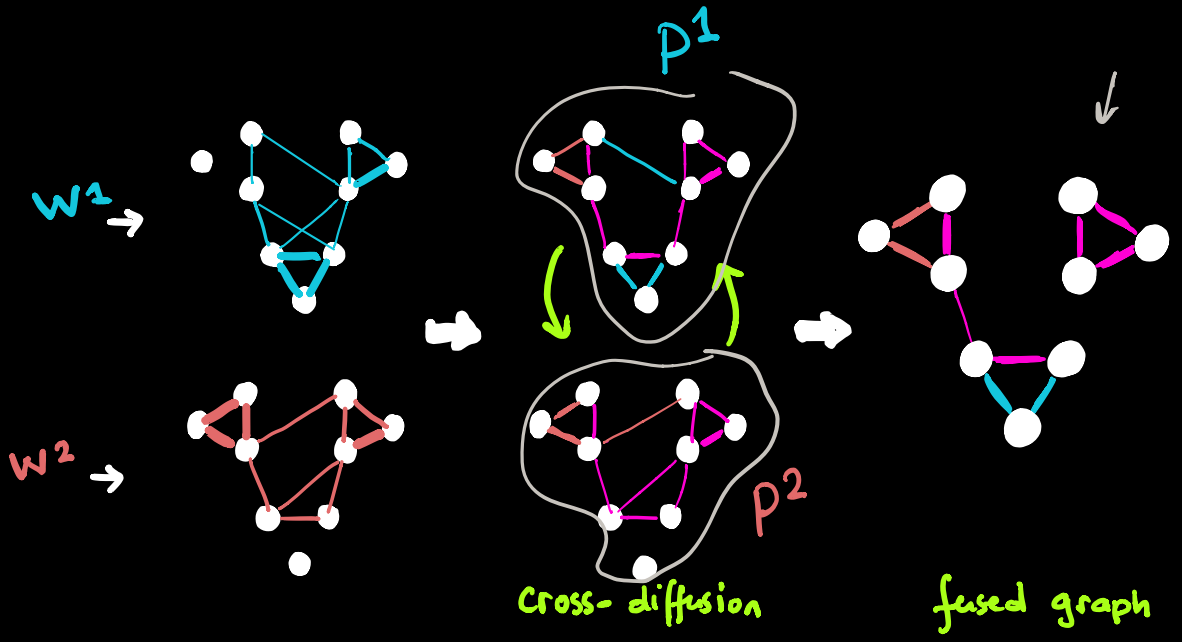
$P^2_0 = P^2$

regularization avoiding the loss of the self-similarity through diffusion.

(but it is optional)







$$P^1_{t+1} = S^1 \times P^2_t \times (S^1)^T + \eta I$$

$$P^2_{t+1} = S^2 \times P^1_t \times (S^2)^T + \eta I$$

exchanging the status matrices each time and generate two parallel inter-changing diffusion processes.

P^1_t : status matrix after t iterations

* After t^* steps, The overall status matrix (i.e., fused matrix) is computed as :

$$P^c = \frac{1}{2}\left(P^1_{t^*} + P^2_{t^*}\right)$$

P^c : fused matrix

• **Step 4** : generalization of graph **cross-diffusion** and **fusion** for **m > 2** graphs

① **cross-diffuse**

$$P^1_{t+1} = S^1 \times \frac{P^2_t + \cdots + P^m_t}{(m-1)} \times (S^1)^T + \eta I$$

$$P^2_{t+1} = S^2 \times P^1_t \times (S^2)^T + \eta I$$

$$\vdots$$

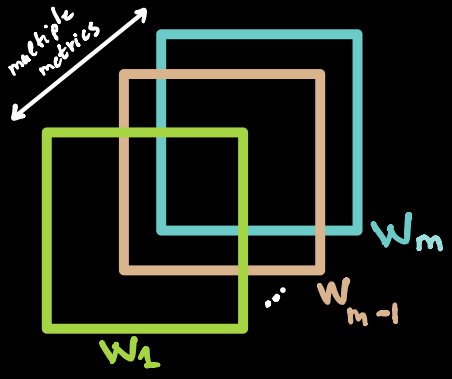
$$P^m_{t+1} = S^m \times P^1_t \times (S^m)^T + \eta I$$

regularization

② **fuse**

$$P^c = \frac{1}{2}\left(P^1_{t^*} + P^2_{t^*}\right)$$

Generalize the proposed **fusion approach** to m views



① **cross-diffuse**

$$P^i_{t+1} = S^i \times \left(\frac{1}{m-1}\sum_{j \neq i} P^j_t\right) \times (S^i)^T + \eta I$$

$i = 1, \ldots, m$

average of all other global / status matrices

② **fuse**

$$P^{(c)} = \frac{1}{m}\sum_{i=1}^{m} P^i_{t^*}$$

- Metric fusion for data clustering:

$$\begin{cases} \min_{Q \in \mathbb{R}^{n \times c}} Trace(Q^T L^\dagger Q) \\ s.t.\ Q^T Q = I \end{cases}$$

$$L^\dagger = I - D^{-1/2} W D^{-1/2}$$

Normalized symmetric graph Laplacian

$W^1 \rightarrow$ ⋮ $W^m \rightarrow$ $W_f = P^c$

Given a single dataset with m views $\{X^1, .., X^m\}$, how to update the clustering model above in such a way it leverages the fused similarity matrix?

What to change in the equations above?

---

* Variant of the proposed solution in (Wang et al. CVPR 2012) where P is normalized differently to avoid scaling self-similarity:

Status matrix (global)

$$P(i,j) = \begin{cases} \frac{W(i,j)}{2\sum_{k \neq i} W(i,k)}, j \neq i \\ 1/2, j = i \end{cases}$$

Kernel matrix (local)

$$S(i,j) = \begin{cases} \frac{W(i,j)}{\sum_{k \in N_i} W(i,k)}, j \in N_i \\ 0, \ otherwise \end{cases}$$

(Wang et al. 2014, Nature Methods)

Table 3: *Unsupervised Metric Fusion by Cross Diffusion; Bo Wang et al. CVPR 2012 (for metric-fusion)*

.

Symbol	Formulation
Unsupervised view-centralized network fusion	iteratively diffuse the global information from all other views along the local struc. of the sim. matrix an input similarity matrix is then improved through a diffusion process. better if the metrics are complementary to each other. **How to fuse the information induced from multiple views?** **Cross diffusion process with m similarity measures (or metrics)**
Multiple-metric fusion algorithm High-dim data metric learning **Proposed cross-diffusion**	outputs enhanced metrics by combining multiple given metrics (similarity measures) e.g., Mahalanobis distances often used in the context of supervised learning. inspired from the work of *co-training* (Blum et al. 1998) different from *co-transduction* (Bai et al. ECCV 2010)
$G = (V, E, W)$	$X = \{x_i, i = 1, \ldots, N\}$ dataset, $x_i \in \mathbb{R}^d$ (i^{th} sample with d features)
P	natural kernel acting on functions on V, can be defined by normalizing $W \in \mathbb{R}^{N \times N}$: $P(i,j) = \frac{W(i,j)}{\sum_{k \in V} W(i,k)}$ $P = D^{-1}W$ carries the full information about the similarity of each data point to all others so that $\sum_{j \in V} P(i,j) = 1$, P is asymmetric after normalization Pf, f label vector, is a local averaging operation, with locality measured by W $f^T P$, f^T is transpose of f, can be viewed as Markov walk of the vector f **both $f^T P$ and Pf can be loosely considered as a diffusion process.**
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$	a subgraph of G with local structure based on neighborhood **assumption:** local similarities (high values) are more reliable than far-away ones; → local similarities can be propagated to non-local points through a diffusion
$\mathcal{W}$	measures local affinity using KNN, $\mathcal{W}(i,j) = W(i,j)$ if $x_j \in KNN(x_i)$, 0 otherwise
S	derived from $\mathcal{W}$, $S(i,j) = \frac{\mathcal{W}(i,j)}{\sum_{x_k \in KNN(x_i)} \mathcal{W}(i,k)}$ encodes the similarity to nearby data points
Case $m = 2$, 2 metrics	$P^1_{t+1} = S^1 \times P^2_t \times (S^1)^T + \eta I$ $P^2_{t+1} = S^2 \times P^1_t \times (S^2)^T + \eta I$
Generalization	$P^i_{t+1} = S^i \times (\frac{1}{m-1} \sum_{j \neq i} P^j_t) \times (S^i)^T + \eta I$ for $i = 1, \ldots, m$
Final status matrix	$P^{(c)} = \frac{1}{m} \sum_{i=1}^m P^{(i)}_t$
$\begin{cases} min_{Q \in \mathbb{R}^{n \times C}} Trace(Q^T L^\dagger Q) \\ s.t.\ Q^T Q = I \end{cases}$	clustering
$L^\dagger = I - D^{-1/2} W D^{-1/2}$	normalized Laplacian matrix