Multi-View Diffusion Process for Spectral Clustering and Image Retrieval

presented by

Qilin Li
Curtin University

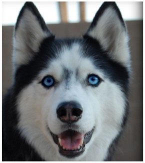
qilin.li@curtin.edu.au

Overview

Introduction

- What is affinity (similarity) learning?

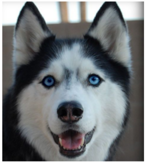
A



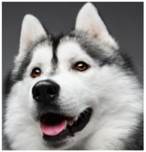
B

Low affinity score

A



B



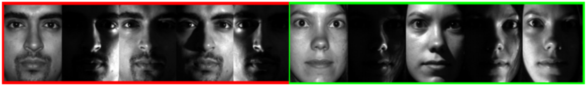
High affinity score

Introduction (cont.)

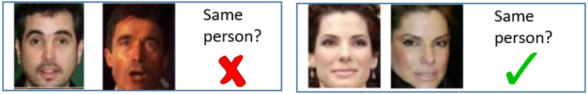
- Why affinity learning?

retrieval

clustering



verification



......

Related works

- The Gaussian kernel affinity

$$A_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right) \tag{1}$$

  - The bandwidth σ controls how fast the affinity vary based on the Euclidean distance
  - A global σ cannot fit non-uniform data well

Related works (cont.)

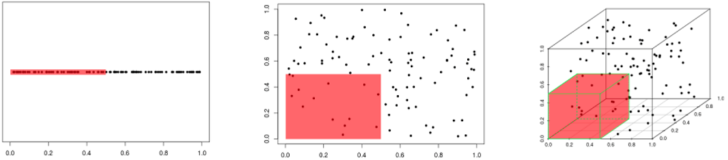
- Locally adapted Gaussian kernel affinity

$$A_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right) \tag{2}$$

  - σ_i adapted to the local structure, *e.g.*, the mean distance to kNN of x_i

Related works (cont.)

- All Gaussian kernels suffer from the "curse of dimensionality"
  - As the dimension increases, the available data become sparse

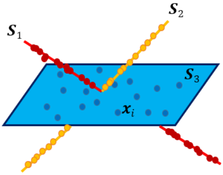


1D: 42% data captured   2D: 14% data captured   3D: 7% data captured

  - The Euclidean distance tends to be large all the time in this sparse space
  - The Gaussian kernel affinity is not appropriate for high-dimensional data, such as images

Related works (cont.)

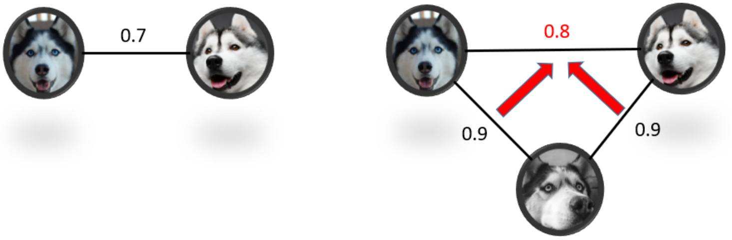
- Sparse representation affinity [1]
  - Manifold assumption: high-dimensional data lie in low-dimensional manifolds (subspaces)
  - Sparse constraint encourages the usage of data points from the same subspace for reconstruction



$$\min_{C} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda\|\mathbf{C}\|_1 \quad \text{s.t.} \quad \text{diag}(\mathbf{C}) = 0, \tag{3}$$

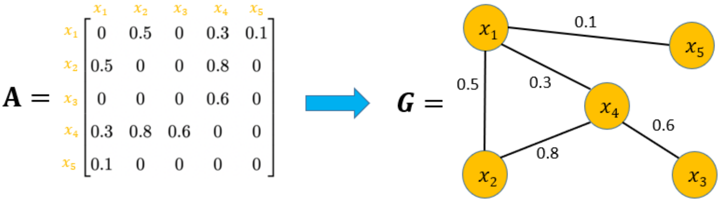
- The affinity matrix $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^\top$

- Key idea: use **neighbor information** to augment pairwise affinity
  - Intuitively, if x_i is similar to x_k and x_j is also similar to x_k, then the affinity value between x_i and x_j should be increased



Diffusion process (cont.)

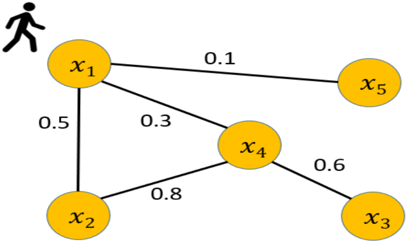
- Graph representation $G = (V, E)$ of affinity matrix
  - Vertices are data points, and edge weights are affinity values



affinity matrix   affinity graph

Diffusion process (cont.)

- Diffusion process as a Markov Random Walk on the graph
  - Affinity values A_{ij} can be interpreted as the transition probability of walking from $Node_i$ to $Node_j$
  - *e.g.*, starting from x_1, it has a 0.5 chance walking to x_2 and 0.3 chance to x_4, **in one step**

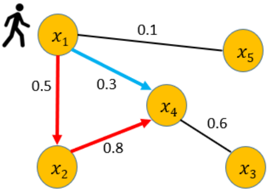


Diffusion process (cont.)

- Random Walk with many steps
  - **A** is the transition probability of Random Walk in **one** step
  - $\mathbf{A}^2$ is the transition probability of Random Walk in **two** steps
  - . . .

$$\mathbf{A}=\begin{bmatrix}0.00 & 0.50 & 0.00 & 0.30 & 0.10\\ 0.50 & 0.00 & 0.00 & 0.80 & 0.00\\ 0.00 & 0.00 & 0.00 & 0.60 & 0.00\\ 0.30 & 0.80 & 0.60 & 0.00 & 0.00\\ 0.10 & 0.00 & 0.00 & 0.00 & 0.00\end{bmatrix}$$

$$\mathbf{A}^2=\begin{bmatrix}0.35 & 0.24 & 0.18 & 0.40 & 0.00\\ 0.24 & 0.89 & 0.48 & 0.15 & 0.05\\ 0.18 & 0.48 & 0.36 & 0.00 & 0.00\\ 0.40 & 0.15 & 0.00 & 1.09 & 0.03\\ 0.00 & 0.05 & 0.00 & 0.03 & 0.01\end{bmatrix}$$

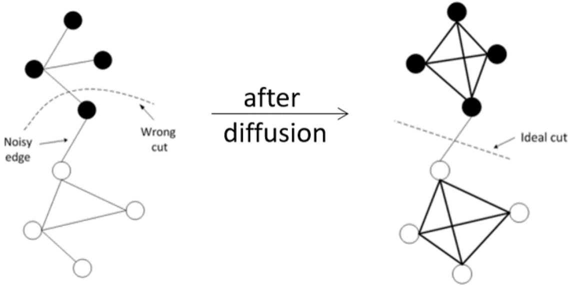


The probability of walking from x_1 to x_4:

- In one step: $A_{14}=0.3$
- In two steps: $A^2{}_{14}=A_{12}\cdot A_{24}=0.4$
- ......

Diffusion process (cont.)

- **The power of affinity matrix $\mathbf{A}^t$ updates pairwise affinity using the contextual information, which is their affinity to neighbor nodes.**
  - Pull together similar nodes
  - Push away dissimilar nodes
  - *e.g.*, in a graph cut problem:



Diffusion process (cont.)

- Larger t means more neighbors are considered
- What t to use?
- The Random Walk is a stationary stochastic process

$$\mathbf{A}^t \to \Pi \quad \text{as} \quad t \to \infty \tag{4}$$

where Π is a stochastic matrix with all its rows equal to π (a left eigenvector of $\mathbf{A}$)

- Find a proper $t \in (0, \infty)$ is hard

Diffusion process (cont.)

- Instead, we could use all t together!

$$\mathbf{A}^0 + \mathbf{A}^1 + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^t = \sum_{i=0}^{t} \mathbf{A}^t. \tag{5}$$

- That is, the pairwise affinity is updated as the summation of all transition probabilities of walking from one node to another, **in any number of steps**

Diffusion process (cont.)

- If the eigenvalues of **A** are bounded in (-1, 1) (which can be easily achieved), then it can be shown that

$$\sum_{i=0}^{t} \mathbf{A}^{t} = (\mathbf{I} - \mathbf{A})^{-1}, \tag{6}$$

where **I** is the identity matrix.

Diffusion process (cont.)

- This can be generalized to the high-order tensor

$$\sum_{i=0}^{t} \mathbb{A}^{t} = (\mathbb{I} - \mathbb{A})^{-1}, \tag{7}$$

  where $\mathbb{A}$ is the Kronecker product $\mathbb{A} = \mathbf{A} \bigotimes \mathbf{A}$.

- Diffusion on higher-order tensor makes use of more contextual information

Diffusion process (cont.)

- $(\mathbb{I} - \mathbb{A})^{-1}$ is a diffusion kernel of size $nn \times nn$. We can obtain a $n \times n$ affinity matrix $\mathbf{A}^*$ by

$$\mathbf{A}^* = vec^{-1}((\mathbb{I} - \mathbb{A})^{-1} vec(\mathbb{I})) \tag{8}$$

  where $vec : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{mn}$ is an isomorphic operator that stacks the columns of a matrix into a column vector. Its inverse is denoted as vec^{-1}.

- The computation cost is prohibitive due to the tensor inverse

Iterative algorithm

- It can be shown that the following iterative algorithm converges to the same closed-form solution (8) [2]. Initialize $\mathbf{A}^{(1)} = \mathbf{A}$ and then

$$\mathbf{A}^{(t+1)} = \mathbf{A}\mathbf{A}^{(t)}\mathbf{A}^{\top} + \mathbf{I}. \tag{9}$$

- $\lim_{t\to\infty} \mathbf{A}^{(t+1)} = vec^{-1}((\mathbb{I} - \mathbb{A})^{-1} vec(\mathbb{I}))$

Optimization framework

- Surprisingly, the diffusion process can also be formulated as an optimization problem [3]

$$\min_{\hat{A}} \frac{1}{2} \sum_{i,k,j,m=1}^{n} \underbrace{A_{ik} A_{jm} \left(\frac{\hat{A}_{ij}}{\sqrt{d_i d_j}} - \frac{\hat{A}_{km}}{\sqrt{d_k d_m}} \right)^2}_{\text{smoothness}} + \mu \sum_{i,j=1}^{n} \underbrace{\left(\hat{A}_{ij} - I_{ij} \right)^2}_{\text{fitness}}$$

  where $d_i = \sum_{j=1}^{n} A_{ij}$ is the degree of vertex x_i.
- The optimal solution of diffusion process is a tradeoff between **smoothness** and **fitness**

Diffusion based sparse subspace clustering (DSSC) [4]

- Unsupervised affinity learning
- Sparse coding + diffusion process

Motion segmentation

Methods	LRR	LRSC	LSR	BDSSC	SSC	SSSC	DSSC
2 motions	3.76	3.69	2.20	2.29	1.95	1.94	**1.68**
3 motions	9.92	7.69	7.13	4.95	4.94	4.92	**4.64**
All	5.15	4.59	3.31	2.89	2.63	2.61	**2.35**

Face clustering

Methods	LRR	LRSC	LSR	BDSSC	SSC	SSSC	DSSC
2 subjects	6.74	3.15	6.72	3.90	1.87	1.27	**0.61**
3 subjects	9.30	4.71	9.25	17.70	3.35	2.71	**1.25**
5 subjects	13.94	13.06	13.87	27.50	4.32	3.41	**2.80**
8 subjects	25.61	21.25	25.98	33.20	5.99	4.15	**4.04**
10 subjects	29.53	29.58	28.33	39.53	7.29	5.16	**4.84**

- Code: https://github.com/qilinli/Diffusion_based_Sparse_Subspace_Clustering-DSSC

Self-reinforced diffusion process (SRD) [5]

- Semi-supervised affinity learning
- Diffusion process with label information guidance
- The diffusion process contains two terms. The first term is about message passing among neighbors. What about the second term?

$$\mathbf{A}^{(t+1)} = \mathbf{S}\mathbf{A}^{(t)}\mathbf{S}^\top + \mathbf{I} \tag{10}$$

- Google's PageRank system in a nutshell:

$$\mathbf{A}^{(t+1)} = \alpha\mathbf{S}\mathbf{A}^{(t)} + (1-\alpha)\mathbf{Y} \tag{11}$$

- $\mathbf{Y}$ is a personalized jumping matrix between webpages based on user preference
- The identity matrix $\mathbf{I}$ acts as a **prior** affinity matrix

SRD

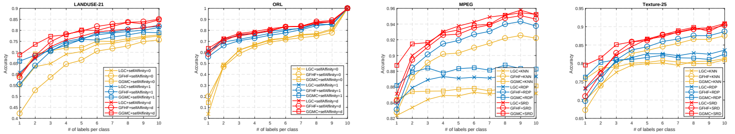
- The prior can be updated if additional information is given, *e.g.*, labels of data points

$$\mathbf{A}^{(t+1)} = \mathbf{S}\mathbf{A}^{(t)}\mathbf{S}^{\top} + \mathbf{Y} \tag{12}$$

where

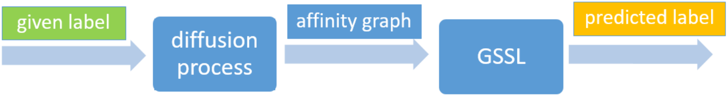
$$Y_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ have the same label} \\ 0 & \text{otherwise} \end{cases}$$

- Self-affinity helps the diffusion process to absorb contextual information more effectively
- Code: https://github.com/qilinli/Self_Supervised_Diffusion



Alternating diffusion process (ADP) [6]

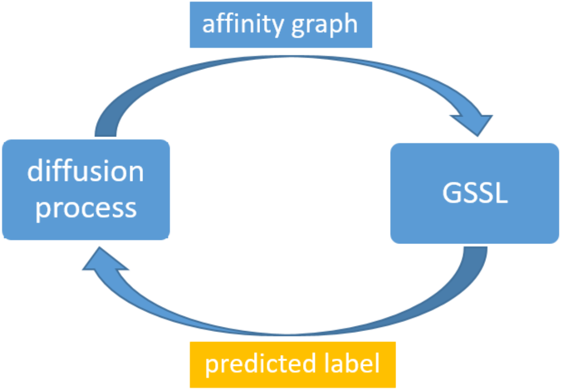
- Self-supervised affinity learning
- Diffusion process with guidance of self-predicted labels
- Rethink the SRD algorithm



- If the given label can aid the diffusion process, the predicted label may also help

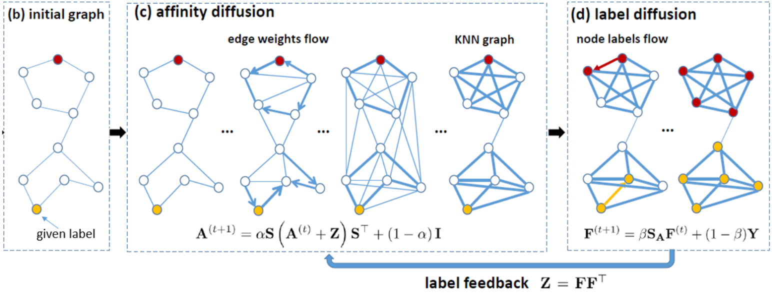
Alternating diffusion process (ADP) [6] (cont.)

- The predicted label as feedback to “self-supervise” the diffusion process



- The affinity graph and predicted label are updated alternately in an iterative manner

- The workflow of ADP



- Code: https://github.com/qilinli/Alternating_Diffusion_Process

ADP experiments

- Ablation study on synthetic data

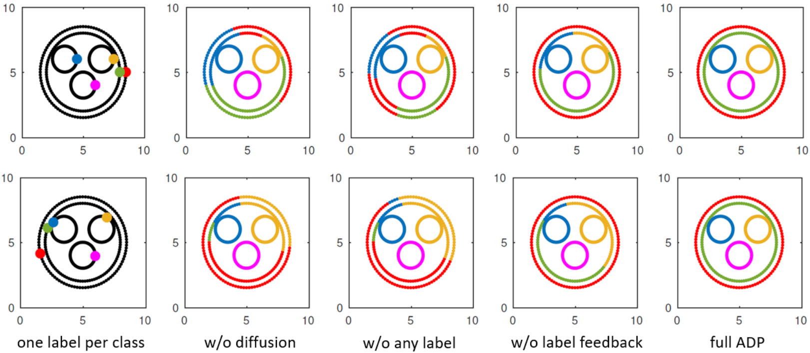


Figure 1: Semi-supervised classification results on "Five Circles" dataset with one label per class.

Today

- Paper [7]:
  Q. Li, S. An, L. Li, W. Liu, and Y. Shao, "Multi-view diffusion process for spectral clustering and image retrieval," IEEE Transactions on Image Processing, 2023.
- Code:
  https://github.com/qilinli/Multi-View-Diffusion-MVD

Multi-view learning

- Data can often be characterised by multiple representations
- E.g., An image can be encoded by SIFT, CNN, ResNet, Transformer...
- E.g., a webpage can be represented by its text, images, hyperlinks, keywords...
- Multi-view learning, cross-modal learning
- The goal is to fuse consensus or complementary information among views

Multi-view learning (cont.)

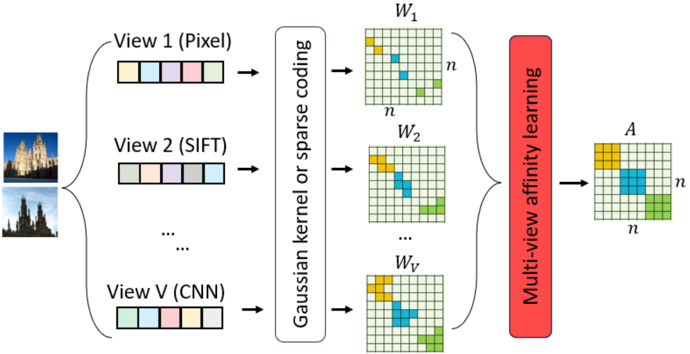
- Multi-view learning is versatile, and can take place at any stage
- E.g., in classification (representation, metric, prediction):

Multi-view representation learning	Multi-view metric learning	Multi-view prediction
• canonical correlation analysis • Discriminative analysis • Deep learning based • ...	• t-SVD • Essential tensor • Weight learning • ...	• Co-training • ...

- Today we will focus on multi-view matric learning

Multi-view learning (cont.)

- In the context of affinity learning, the problem is: "given a set of affinity matrices $\{W_1, W_2, ..., W_n\} \in \mathrm{R}^{n \times n}$, the goal is to learn a unified affinity matrix $A \in \mathrm{R}^{n \times n}$ that leverages all views"



- A naive idea is to compute an average of all V views: $\frac{1}{V}\sum_{v=1}^{V} \mathbf{W}^{(v)}$
- What is the problem?

Problem formulation

- A better idea is to compute a weighted sum: $\mathbf{Z} = \sum_{v=1}^{V} \beta_v \mathbf{W}^{(v)}$
- There are still problems:
- 1) How to define weights β?
- 2) $\mathbf{Z}$ is a linear combination of $\mathbf{W}$ and thus limited by inputs?

Problem formulation (cont.)

- We propose to use the consensus $\mathbf{Z}$ as a plausible prior and enforce the target affinity matrix $\mathbf{A}$ to be close to $\mathbf{Z}$, resulting in the following optimization problem:

$$\min_{\mathbf{A},\boldsymbol{\beta}} \quad \|\mathbf{A} - \mathbf{Z}(\boldsymbol{\beta})\|_F^2 + \frac{1}{2}\lambda\|\boldsymbol{\beta}\|_2^2, \tag{13}$$
$$s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq 0$$

where $\lambda > 0$ is a regularizer favouring balanced weights among views.The constraints on $\boldsymbol{\beta}$ ensure all the weights are in the range of zero to one and sum up to one.

Problem formulation (cont.)

- The next question is how to quantify the "goodness" of input affinity matrices $\mathbf{W}$ so that appropriate weights can be assigned?
- This is where we involve the **smoothness** term in the diffusion process, i.e., for each affinity matrix $\mathbf{W}^{(v)}$, we compute a scalar measurement q_v as:

$$\frac{1}{2}\sum_{i,j,k,l=1}^{n} \mathbf{W}_{ij}^{(v)}\mathbf{W}_{kl}^{(v)} \left(\frac{\mathbf{A}_{ki}}{\sqrt{\mathbf{D}_{ii}^{(v)}\mathbf{D}_{kk}^{(v)}}} - \frac{\mathbf{A}_{lj}}{\sqrt{\mathbf{D}_{jj}^{(v)}\mathbf{D}_{ll}^{(v)}}} \right)^2 \quad (14)$$

- The intuition is that $sim(a, c)$ should be enlarged if both $sim(a, b)$ and $sim(b, c)$ are large, where $sim(\cdot, \cdot)$ is the similarity between two data points. A widely used information retrieval technique, named automatic query expansion uses similar idea.

Problem formulation (cont.)

- Put everything together, we obtain the final optimisation problem:

$$\min_{\mathbf{A},\boldsymbol{\beta}} \quad \boldsymbol{\beta}^\top \mathbf{q}\left(\mathbf{A}\right) + \mu \|\mathbf{A} - \mathbf{Z}\left(\boldsymbol{\beta}\right)\|_F^2 + \frac{1}{2}\lambda \|\boldsymbol{\beta}\|_2^2 \qquad (15)$$
$$s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq 0.$$

  where $\mu > 0$ is a balance hyperparameter.
- 1) smoothness, 2) fitness, 3) regularizer
- Solve by an alternating optimization routine of two subproblems

Optimization solver

$$\min_{\mathbf{A},\boldsymbol{\beta}} \quad \boldsymbol{\beta}^\top \mathbf{q}(\mathbf{A}) + \mu \|\mathbf{A} - \mathbf{Z}(\boldsymbol{\beta})\|_F^2 + \frac{1}{2}\lambda \|\boldsymbol{\beta}\|_2^2 \qquad (16)$$
$$s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq 0.$$

- Subproblem 1): fix β, update $\mathbf{A}$
- Subproblem 2): fix $\mathbf{A}$, update β

Optimization solver (cont.)

- Subproblem 1): fix β, update $\mathbf{A}$
- Closed-form solution:

$$\mathbf{A}^* = vec^{-1}\left(\left(1-\sum_v \alpha_v\right)\left(\mathbb{I}-\sum_v \alpha_v \mathbb{S}^{(v)}\right)^{-1} vec(\mathbf{Z})\right), \quad (17)$$

where $\alpha_v = \frac{1}{1+\mu}\beta_v$ and $\mathbb{I}$ is the identity matrix of the appropriate size, $\mathbb{S} = \mathbf{S} \otimes \mathbf{S}$ is the Kronecker product of the normalized affinity matrix $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, and $vec : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ is the operation that stacks the column of a matrix into a vector. The inverse of *vec* exists and is denoted as vec^{-1}.

- Iterative solver:

$$\mathbf{A}_{(t+1)} = \sum_{v=1}^{V} \alpha_v \mathbf{S}^{(v)} \mathbf{A}_{(t)} \mathbf{S}^{(v)\top} + \left(1-\sum_{v=1}^{V}\alpha_v\right)\mathbf{Z}. \quad (18)$$

Optimization solver (cont.)

$$\min_{\mathbf{A},\boldsymbol{\beta}} \quad \boldsymbol{\beta}^\top \mathbf{q}\left(\mathbf{A}\right) + \mu \|\mathbf{A} - \mathbf{Z}\left(\boldsymbol{\beta}\right)\|_{\mathsf{F}}^2 + \frac{1}{2}\lambda \|\boldsymbol{\beta}\|_2^2 \qquad (19)$$
$$s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq 0.$$

- Subproblem 2): fix **A**, update β
- The problem can be re-written as:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{Q}\boldsymbol{\beta} + \mathbf{f}^\top \boldsymbol{\beta} \qquad (20)$$
$$s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq 0,$$

which can be directly solved using the standard quadratic programming routine.

Experiments and Results

- Experiment 1): Sanity check on Cifar10
  - 5 views: pixel, LeNet (75.2%), VGG (91.5%), ResNet (94.6%), SENet (94.7%)
  - 3 weighting strategies: Naive, RED [3], MVD

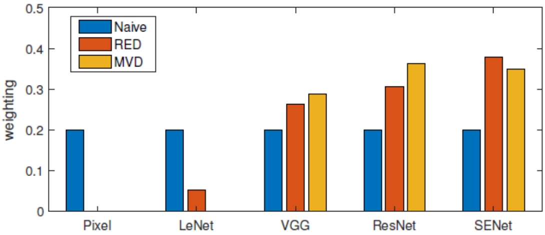


Fig. 3. Weights learned on multi-view representations of the Cifar10 dataset.

Experiments and Results (cont.)

Table 1: Clustering performance (%) on Cifar10 with single view representation and multi view representation.

Metric	Single view					Multi view		
	Pixel	LeNet	VGG	ResNet	SENet	Naive	RED	MVD
ACC	15.00	31.85	81.85	92.71	93.61	93.55	94.64	**95.32**
NMI	3.50	31.63	77.96	84.65	86.08	85.97	87.41	**89.35**

- Multi-view is not guaranteed to be better
- Weights learned by MVD make more sense and achieve best results

Experiments and Results (cont.)

- Experiment 2): Image retrieval on Oxford5K and Paris6K

TABLE II
RETRIEVAL PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON OXFORD 5K AND PARIS 6K. R-MAC FEATURES ARE EXTRACTED BASED ON [50] USING VGG AND RESNET101 AS THE BACKBONE.

Method	Feature	Oxford 5k	Paris 6k
*k*NN search	R-MAC(VGG)	79.5	84.5
*k*NN search + AQE [42]		85.4	88.4
Iscen's diffusion [19]		85.7	94.1
Yang's diffusion [31]		89.7	94.7
*k*NN search	R-MAC(ResNet)	83.9	93.8
*k*NN search + AQE [42]		89.6	95.3
Iscen's diffusion [19]		87.1	96.5
Yang's diffusion [31]		92.6	97.1
Naive	Both	88.2	96.1
RED [37]		92.8	97.3
Proposed MVD		**94.4**	**97.7**

Experiments and Results (cont.)

Experiments and Results (cont.)

- Experiment 3): Clustering on 13 benchmark datasets
- Comparison to 6 state-of-the-art multi-view clustering approaches

Table 2: Statistics of multi-view datasets for clustering.

Dataset	Type	# Instances	# Classes	# Views	View 1	View 2	View 3	View 4	View 5	View 6	View 7
ORL	Face	400	40	3	Intensity (4096)	LBP (3304)	Gabor (6750)				
Yale	Face	165	15	3	Intensity (4096)	LBP (3304)	Gabor (6750)				
Reuters	Text	1200	6	5	English (2000)	French (2000)	German (2000)	Italian (2000)	Spanish (2000)		
BBC-Sport	Text	544	5	2	Seg1 (3183)	Seg2 (3203)					
CiteSeer	Text	3312	6	2	Citations (3312)	Content (3703)					
Reuters-21578	Text	1500	6	5	English (2000)	French (2000)	German (2000)	Italian (2000)	Spanish (2000)		
Flower17	Flower	1360	17	7	Color	Texture	Shape	HOG	HSV	SIFT bdy	SIFT int
UCI-digits	Digits	2000	10	6	PIX (240)	FOU (76)	FAC (216)	ZER (47)	KAR (64)	MOR (6)	
NUS-WIDE	Object	2000	31	5	CH (65)	CM (226)	CORR (145)	EDH (74)	WT (129)		
MSRC-v1	Object	210	7	5	CM (24)	HOG (576)	GIST (512)	LBP (256)	CENT (254)		
ALOI	Object	10800	100	4	CS (77)	HAR (13)	HSB (64)	RGB (125)			
Caltech20	Object	2386	20	6	Gabor (48)	WM (40)	CENT (254)	HOG (1984)	GIST (512)	LBP (928)	
Caltech101	Object	9144	102	6	Gabor (48)	WM (40)	CENT (254)	HOG (1984)	GIST (512)	LBP (928)	

Experiments and Results (cont.)

TABLE IV
AVERAGE CLUSTERING PERFORMANCE (NMI, ACC) AND STANDARD DEVIATION (%) OVER 10 RUNS BY DIFFERENT MULTI-VIEW SPECTRAL CLUSTERING METHODS.

Metric	Dataset	MVGL [13]	AWP [23]	MCGC [52]	ETLMSC [5]	DGF [29]	FPMVS [53]	SC (best)	MVD (ours)
NMI	ORL	83.79±.00	87.88±.00	89.39±.00	89.03±.22	91.80±.17	85.61±.18	90.78±.50	**95.37**±.25
	Yale	68.03±.00	69.42±.00	68.39±.00	69.74±.82	73.90±.00	70.75±.92	71.14±.76	**75.56**±1.2
	Reuters	9.61±.00	11.66±.00	9.38±.00	28.64±.12	16.04±.06	30.51±.36	19.73±.06	**32.45**±.21
	BBCSport	92.95±.00	91.49±.00	82.02±.00	**97.25**±.09	92.68±.00	93.25±.61	87.11±.00	94.46±.00
	CiteSeer	4.78±.00	37.14±.00	18.06±.00	36.53±.22	38.10±.07	42.81±.13	17.63±1.1	**44.23**±.15
	Reuters-21578	8.63±.00	28.67±.00	11.43±.00	25.58±.72	30.94±.06	32.17±.58	20.26±.11	**35.53**±.44
	Flower17	22.51±.00	46.58±.00	44.38±.00	61.17±.53	**64.92**±.37	49.23±.36	47.34±.45	62.50±.07
	UCI-digits	88.10±.00	93.02±.00	83.71±.00	92.33±.06	**95.90**±.08	91.73±.20	92.52±.00	93.30±.13
	NUS-WIDE	7.21±.00	16.28±.00	14.55±.00	15.12±.32	19.86±.26	20.12±.46	17.33±.33	**21.72**±.53
	MSRC-v1	72.80±.00	76.61±.00	71.80±.00	79.54±.34	80.98±.00	78.57±.52	69.07±.78	**89.96**±.73
	ALOI	69.75±.00	73.60±.00	82.45±.00	88.63±.64	91.08±.33	87.51±.60	80.18±.45	**92.47**±.42
	Caltech20	41.74±.00	58.28±.00	57.25±.00	67.18±.23	65.37±.08	65.47±.68	52.74±.20	**72.30**±.13
	Caltech101	41.97±.00	14.13±.00	41.52±.00	49.72±.64	46.76±.08	42.83±.35	48.41±.20	**54.92**±.49
ACC	ORL	71.25±.00	76.50±.00	78.25±.00	81.33±.52	84.20±.62	82.63±.76	80.88±.99	**88.60**±1.4
	Yale	70.30±.00	67.27±.00	63.64±.00	65.91±.34	70.91±.00	68.25±.58	69.21±.75	**76.00**±1.4
	Reuters	21.35±.00	25.44±.00	23.92±.00	38.62±.37	31.64±.06	44.53±.12	27.27±.26	**49.70**±.00
	BBCSport	97.98±.00	97.43±.00	91.18±.00	95.94±.06	97.98±.00	96.17±.05	95.96±.00	**98.35**±.00
	CiteSeer	25.91±.00	64.07±.00	43.72±.00	54.21±.22	63.50±.21	58.63±.07	40.64±1.1	**69.14**±.00
	Reuters-21578	30.33±.00	49.40±.00	32.80±.00	43.68±.37	50.77±.07	49.23±.48	42.36±.27	**51.56**±.53
	Flower17	25.00±.00	44.85±.00	43.90±.00	60.28±.44	**67.03**±.97	51.85±.14	43.56±.75	64.40±.07
	UCI-digits	86.05±.00	96.75±.00	82.40±.00	92.64±.16	**98.25**±.08	92.70±.25	96.59±.20	96.75±.13
	NUS-WIDE	14.85±.00	14.20±.00	15.35±.00	15.57±.86	16.80±.12	16.34±.66	13.86±.47	**18.65**±.72
	MSRC-v1	75.24±.00	87.14±.00	74.29±.00	83.76±.35	87.14±.00	85.42±.76	67.29±.84	**94.70**±.57
	ALOI	56.62±.00	61.43±.00	77.04±.00	81.42±.75	84.51±1.0	81.47±.54	68.65±1.3	**87.38**±.52
	Caltech20	49.37±.00	55.87±.00	58.89±.00	58.42±.36	59.67±.08	**63.26**±.82	38.52±.20	61.24±.49
	Caltech101	13.44±.00	26.22±.00	23.00±.00	29.77±.41	23.53±.39	31.49±.59	26.74±.54	**34.26**±.44

Summary of diffusion process

- A generic tool to learn pairwise affinity in various settings, unsupervised, semi-supervised, or supervised, using neighborhood information
- A message passing or neighbor aggregation framework on graphs that can propagate various information, such as edge weight, vertex label, vertex representation
- A simple yet effective iterative formula backed up by mathematical justification
- Future works could focus on improving computational efficiency, integration with representation learning

The END

Thank you!

Bibliography

[1] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[2] X. Yang, L. Prasad, and L. J. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 28–38, 2012.

[3] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1213–1226, 2018.

[4] Q. Li, W. Liu, and L. Li, "Affinity learning via a diffusion process for subspace clustering," *Pattern Recognition*, vol. 84, pp. 39–50, 2018.

Bibliography (cont.)

[5] Q. Li, W. Liu, and L. Li, "Self-reinforced diffusion for graph-based semi-supervised learning," *Pattern Recognition Letters*, vol. 125, pp. 439–445, 2019.

[6] Q. Li, S. An, W. Liu, and L. Li, "Semisupervised learning on graphs with an alternating diffusion process," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[7] Q. Li, S. An, L. Li, W. Liu, and Y. Shao, "Multi-view diffusion process for spectral clustering and image retrieval," *IEEE Transactions on Image Processing*, 2023.