THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學
Department of Computing
電子計算學系

Label Efficient Semi-Supervised Learning via Graph Filtering

Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, Zhichao Guan

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

1. Graph-Based Semi-Supervised Learning

Input:

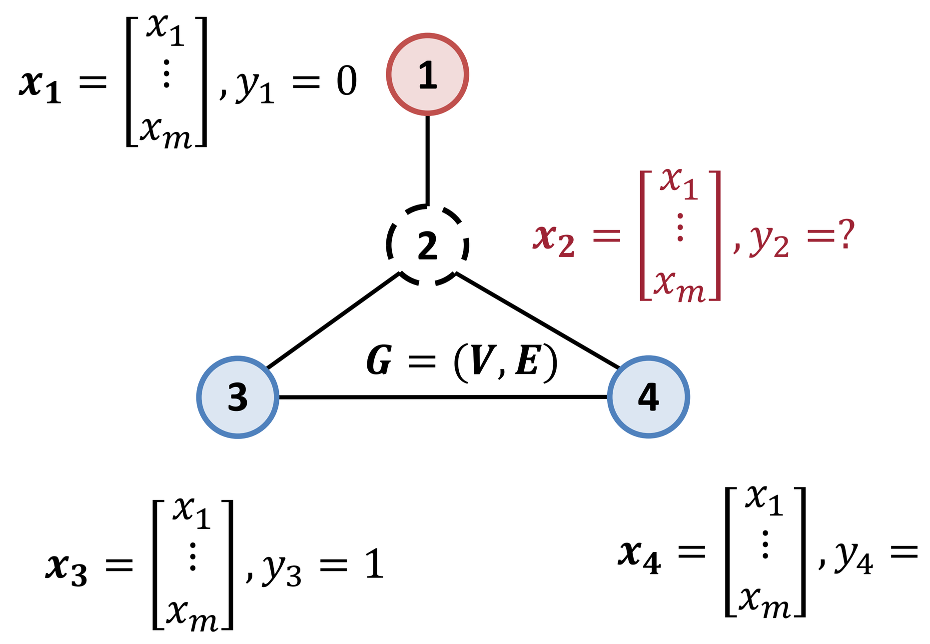
$G = (V, E)$

X, features matrix

Y, a part of labels

Output:

Labels of unlabeled nodes



2. Graph Signal Processing (GSP) [1]

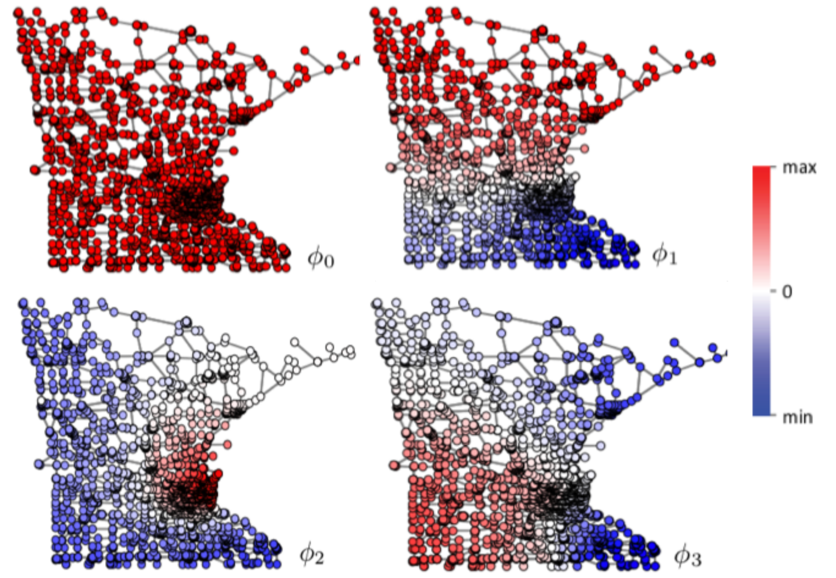
Graph Laplacian:

$$L = D - W$$

where $D = \mathrm{diag}(d_i)$ are degrees of nodes.

Eigen Decomposition:

$$L = \underbrace{\Phi}_{\text{Fourier basis}} \overbrace{\Lambda}^{\text{frequences}} \Phi^{-1}$$



Fourier Basis in Graph Domain with different frequencies [2]

Convolutional Filter:

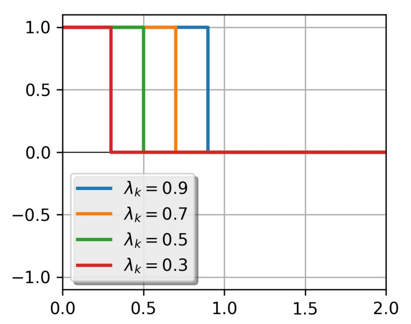
$$G = \Phi p(\Lambda) \Phi^{-1} = p(L)$$

Convolution:

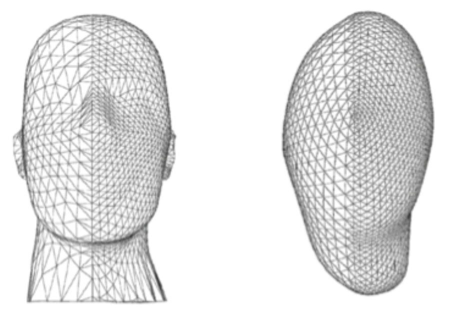
$$Z = GX$$

Low-pass Filter:

$p(\lambda)$ reserve low frequency signals and remove high frequency ones.



Ideal low-pass Frequency response



Before and after low-pass filtering. (Take vertex coordinates as signals) [3]

3. Revisit and Generalize Label Propagation (GLP)

Dissecting LP [4,5] into Signals, Filters, and Classifiers

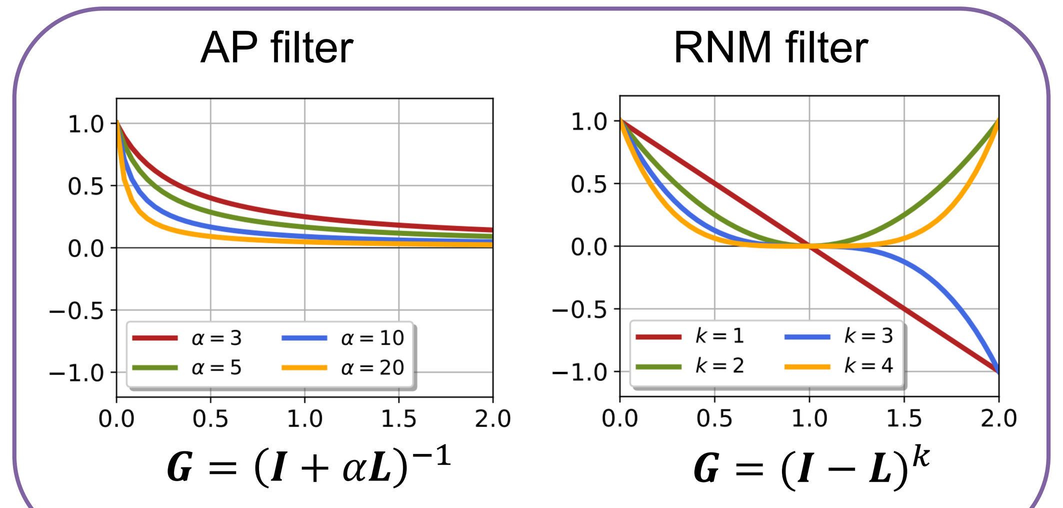
$$\underset{Z}{\mathrm{argmin}} \parallel Z - Y \parallel_2^2 + \alpha \, \mathrm{Tr}(Z^\top L Z)$$

$$Z = \underbrace{(I + \alpha L)^{-1}}_{\text{A fixed low-pass filter}} \underbrace{Y}_{\text{Label matrix as graph signals}}$$

$$\underbrace{\mathrm{label}_i = \underset{j}{\mathrm{argmax}} \, Z_{ij}}_{\text{Classifier}}$$

GLP: X → Any low-pass graph convolutional filter G → $Z = GX$ → Any supervised classifier (e.g. MLP) → prediction

$$Z = \underbrace{G}_{\text{Any low-pass graph Filter}} \underbrace{X}_{\text{Feature matrix as graph signals}}$$



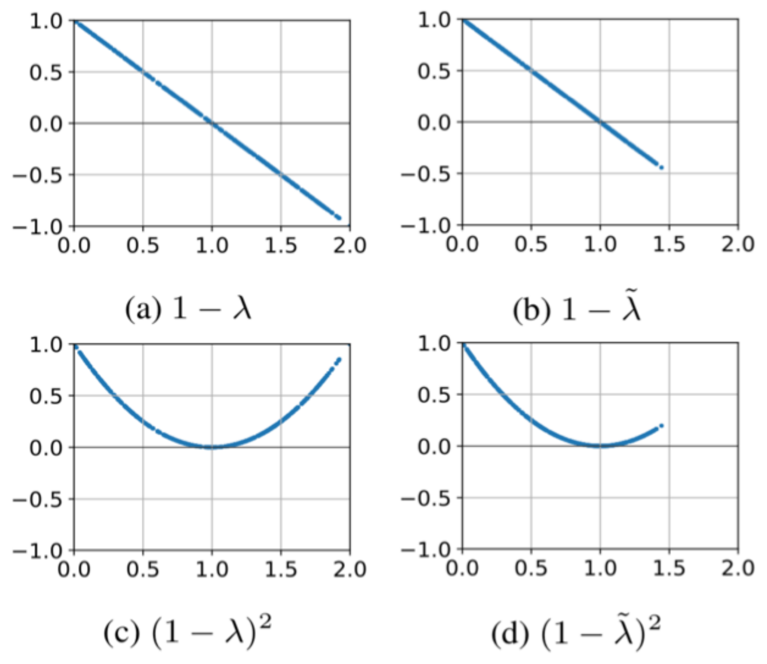
low-pass graph filters

4. Revisit and Improve Graph Convolutional Networks

Revisit GCN[6] from the Perspective of GSP :

$$Z = \mathrm{softmax}\big(\widetilde{W}_s ReLU\big(\widetilde{W}_s X \Theta^{(0)}\big)\Theta^{(1)}\big)$$

- **Normalized Graph Laplacian**
  Restrict the eigenvalues within [0, 2], so the filter is close to a low-pass filter.
- ***K*-Layer Structure**
  Stacking more layers makes GCN more low-pass.
- **Renormalization Trick**
  Adding self-loops (the renormalization trick) shrinks the eigenvalues of $\tilde{L}_s$ from $[0, \lambda_m]$ to $\left[0, \frac{(d_m)}{(d_m+1)}\lambda_m\right]$. It compresses the range of eigenvalues and makes the filter more low-pass.

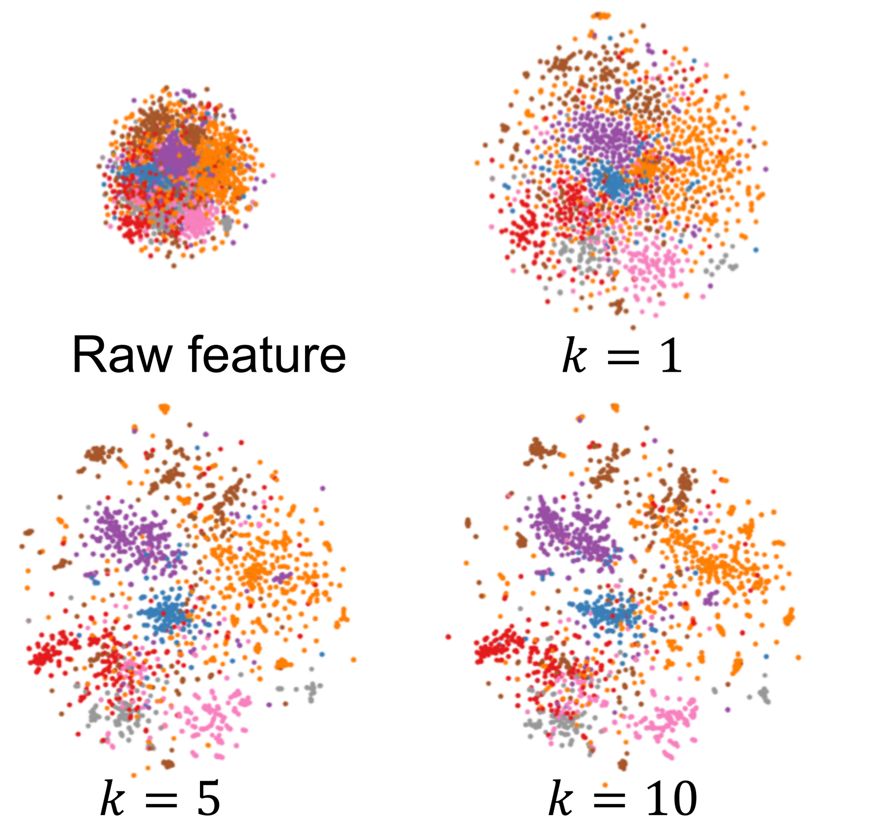


Eigenvalue compression effect of the renormalization trick. (a) & (c): Frequency response without self-loops. (b) & (d) : frequency response with self-loops.

Improved GCN (IGCN):

$$Z = \mathrm{softmax}\big(\underbrace{\widetilde{W}_s^k}_{k\text{-order RNM filter}} ReLU\big(\widetilde{W}_s^k X \Theta^{(0)}\big)\Theta^{(1)}\big)$$

IGCN can achieve label efficiency by using the exponent k to conveniently adjust the filter strength. In this way, it can maintain a shallow structure with a reasonable number of trainable parameters to avoid overfitting.



Visualization of raw and filtered Cora features (by the RNM filter with different k)

5. Experiments

Table 1. Classification accuracy and running time on citation networks and NELL.

Label rate	20 labels per class				4 labels per class				10%	1%	0.1%
	Cora	CiteSeer	PubMed	Large Cora	Cora	CiteSeer	PubMed	Large Cora	NELL		
ManiReg	59.5	60.1	70.7	-	-	-	-	-	63.4	41.3	21.8
SemiEmb	59.0	59.6	71.7	-	-	-	-	-	65.4	43.8	26.7
DeepWalk	67.2	43.2	65.3	-	-	-	-	-	79.5	72.5	58.1
ICA	75.1	69.1	73.9	-	62.2	49.6	57.4	-	-	-	-
Planetoid	75.7	64.7	77.2	-	43.2	47.8	64.0	-	84.5	75.7	61.9
GAT	79.5	68.2	76.2	67.4	66.6	55.0	64.6	46.4	-	-	-
MLP	55.1 (0.6s)	55.4 (0.6s)	69.5 (0.6s)	48.0 (0.8s)	36.4 (0.6s)	38.0 (0.5s)	57.0 (0.6s)	30.8 (0.6s)	63.6 (2.1s)	41.6 (1.1s)	16.7 (1.0s)
LP	68.8 (0.1s)	48.0 (0.1s)	72.6 (0.1s)	52.5 (0.1s)	56.6 (0.1s)	39.5 (0.1s)	61.0 (0.1s)	37.0 (0.1s)	84.5 (0.7s)	75.1 (1.8s)	65.9 (1.9s)
GCN	79.9 (1.3s)	68.6 (1.7s)	77.6 (9.6s)	67.7 (7.5s)	65.2 (1.3s)	55.5 (1.7s)	67.7 (9.8s)	48.3 (7.4s)	81.6 (33.5s)	63.9 (33.5s)	40.7 (33.2s)
IGCN(RNM)	80.9 (1.2s)	69.0 (1.7s)	77.3 (10.0s)	68.9 (7.9s)	70.3 (1.3s)	57.4 (1.7s)	69.3 (10.3s)	52.1 (8.1s)	85.9 (42.4s)	76.7 (44.0s)	66.0 (46.6s)
IGCN(AR)	81.1 (2.2s)	69.3 (2.6s)	78.2 (11.9s)	69.2 (11.0s)	70.3 (3.0s)	58.0 (3.4s)	70.1 (13.6s)	52.5 (13.6s)	85.4 (77.9s)	75.7 (116.0s)	67.4 (116.0s)
GLP(RNM)	80.3 (0.9s)	68.8 (1.0s)	77.1 (0.6s)	68.4 (1.8s)	68.0 (0.7s)	56.7 (0.8s)	68.7 (0.6s)	51.1 (1.1s)	86.0 (35.9s)	76.1 (37.3s)	65.4 (38.5s)
GLP(AR)	80.8 (1.0s)	69.3 (1.2s)	78.1 (0.7s)	69.0 (2.4s)	67.5 (0.8s)	57.3 (1.1s)	69.7 (0.8s)	51.6 (2.3s)	80.3 (57.4s)	67.4 (76.6s)	55.2 (78.6s)

Table 2. Zero-Shot Image Recognition

Method	Devise	SYNC	GCNZ	GPM	DGPM	ADGPM	IGCN(RNM)			GLP(RNM)		
							k=1	k=2	k=3	k=2	k=4	k=6
Accuracy	59.7	46.6	68.0 (1840s)	**77.3** (864s)	67.2 (932s)	76.0 (3527s)	**77.9** (864s)	**77.7** (1583s)	73.1 (2122s)	76.0 (12s)	75.0 (13s)	73.0 (11s)

References

[1] Aliaksei Sandryhaila and Jose M. F. Moura, "Discrete signal processing on graphs".
[2] Bronstein et al., "Geometric deep learning: going beyond euclidean data".
[3] Desbrun et al., "Implicit fairing of irregular meshes using diffusion and curvature flow".
[4] Zhu, Xiaojin et al., "Semi-supervised learning using gaussian fields and harmonic functions",
[5] Zhou, Denny et al., "Learning with local and global consistency"
[6] Kipf and Welling, "Semi-Supervised Classification with Graph Convolutional Networks"

Project Homepage