Semi-Supervised Classification with Graph Convolutional Networks

Hyungu Kahng

2018.04.13

Contents

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Thomas N. Kipf
University of Amsterdam
T.N.Kipf@uva.nl

Max Welling
University of Amsterdam
Canadian Institute for Advanced Research (CIFAR)
M.Welling@uva.nl

ABSTRACT

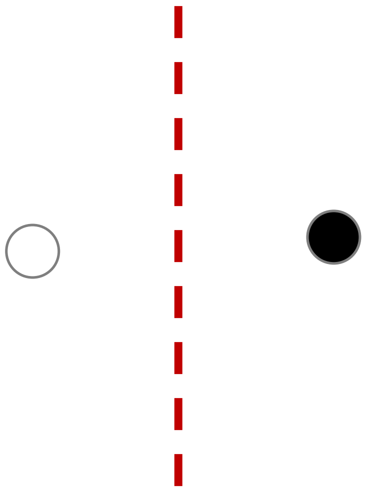
We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on graphs. We motivate the choice of our convolutional architecture via a localized first-order approximation of spectral graph convolutions. Our model scales linearly in the number of graph edges and learns hidden layer representations that encode both local graph structure and features of nodes. In a number of experiments on citation networks and on a knowledge graph dataset we demonstrate that our approach outperforms related methods by a significant margin.

Introduction

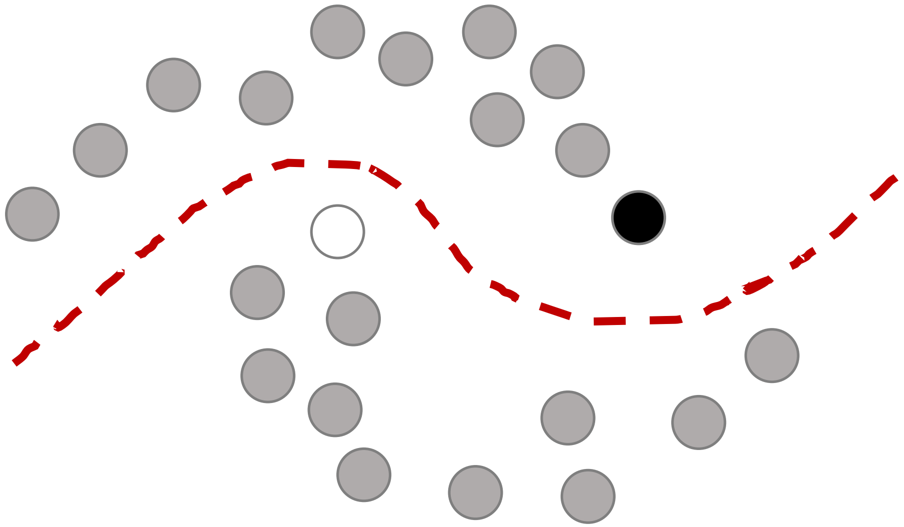
Introduction

Objective of semi-supervised learning

❖ Exploit the properties of unlabeled data to better understand the population structure



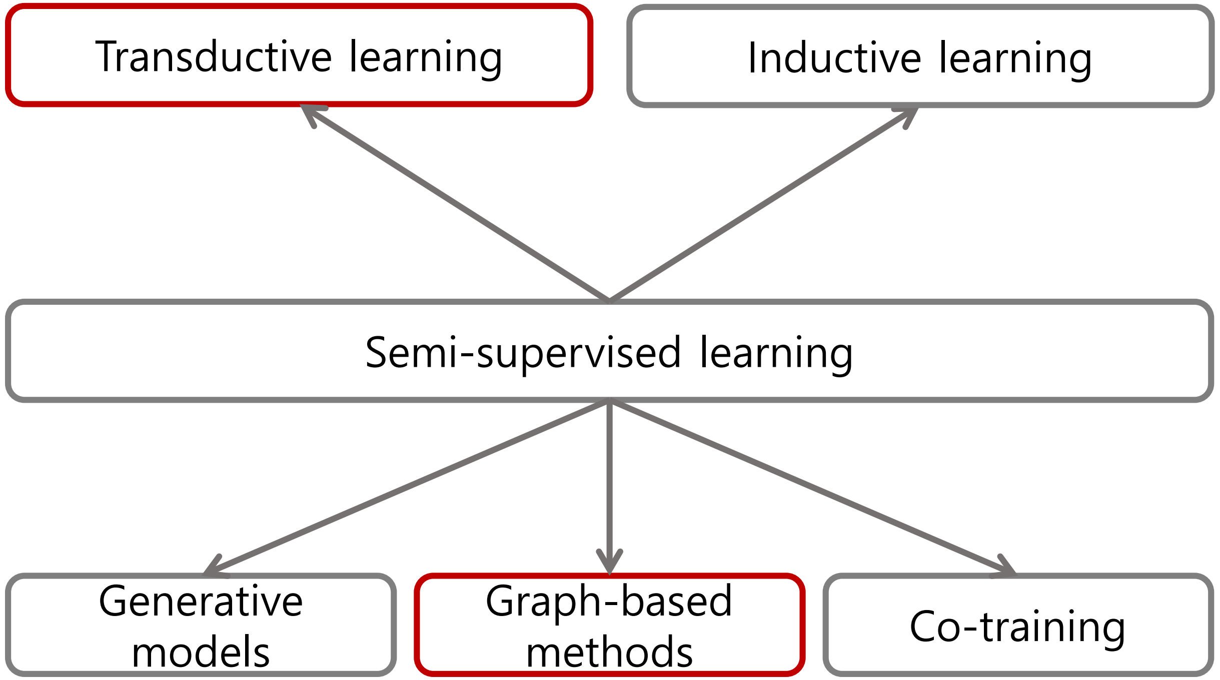
Supervised learning



Semi-supervised learning

Introduction

Categorization of semi-supervised learning

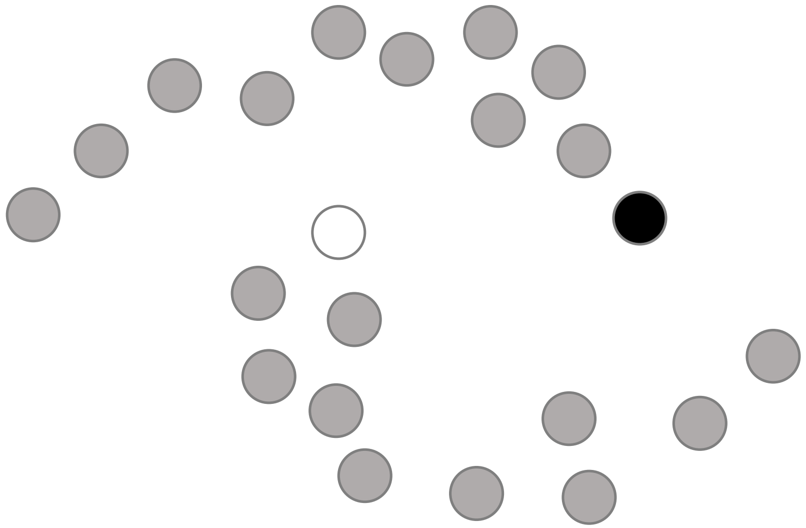


Introduction

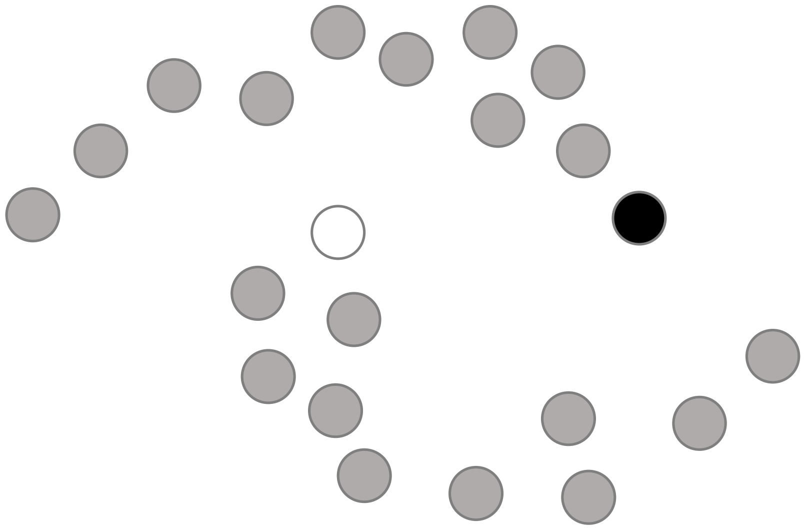
Categorization of semi-supervised learning

❖ Transductive learning aims to infer the correct labels for given unlabeled data only

❖ Inductive learning learns the correct mapping from features to class labels



Transductive learning

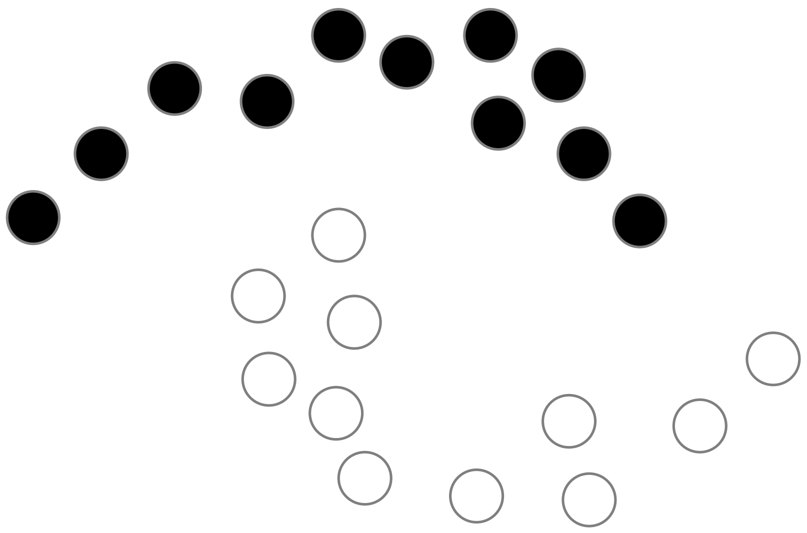


Inductive learning

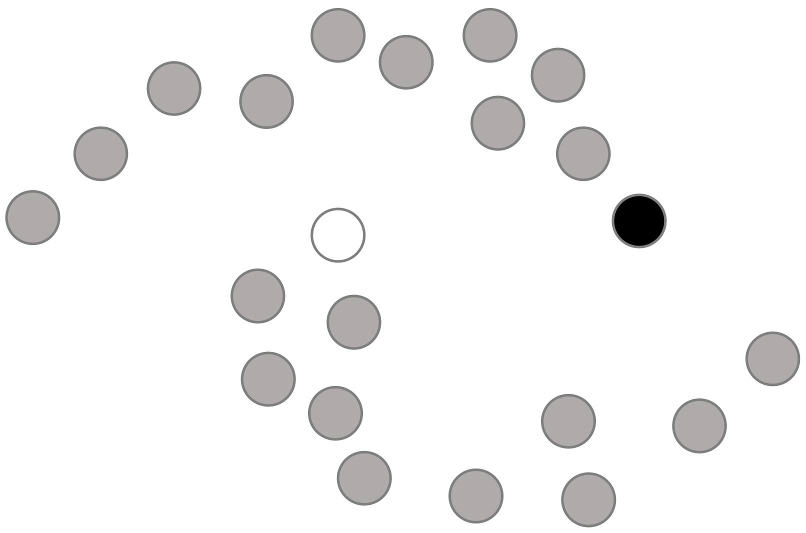
Introduction

Categorization of semi-supervised learning

- ❖ Transductive learning aims to infer the correct labels for given unlabeled data only
- ❖ Inductive learning learns the correct mapping from features to class labels



Transductive learning

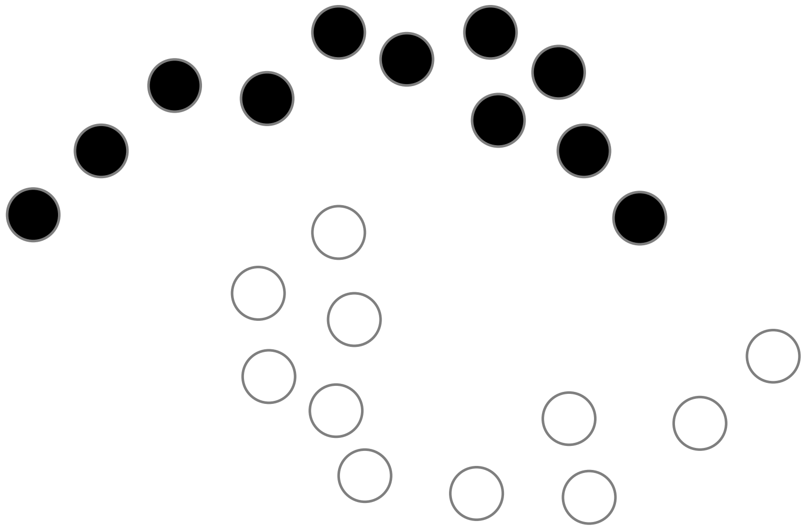


Inductive learning

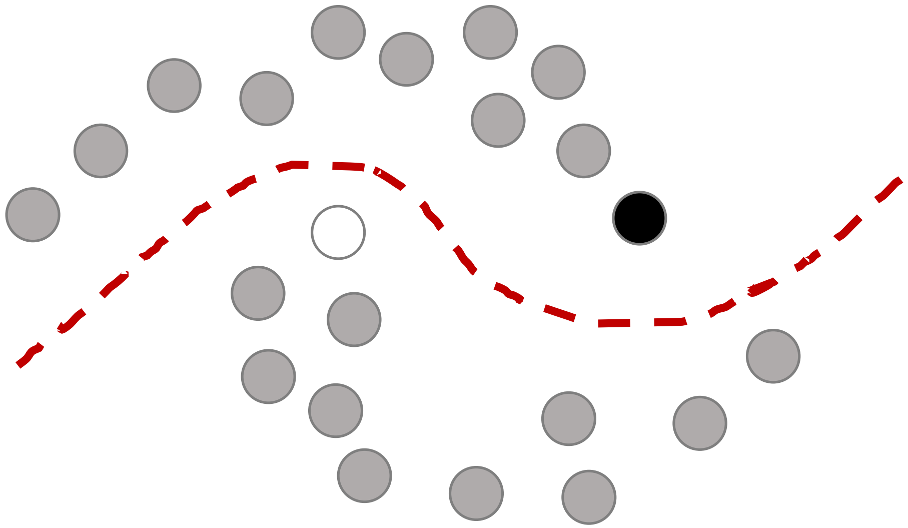
Introduction

Categorization of semi-supervised learning

- ❖ Transductive learning aims to infer the correct labels for given unlabeled data only
- ❖ Inductive learning learns the correct mapping from features to class labels



Transductive learning



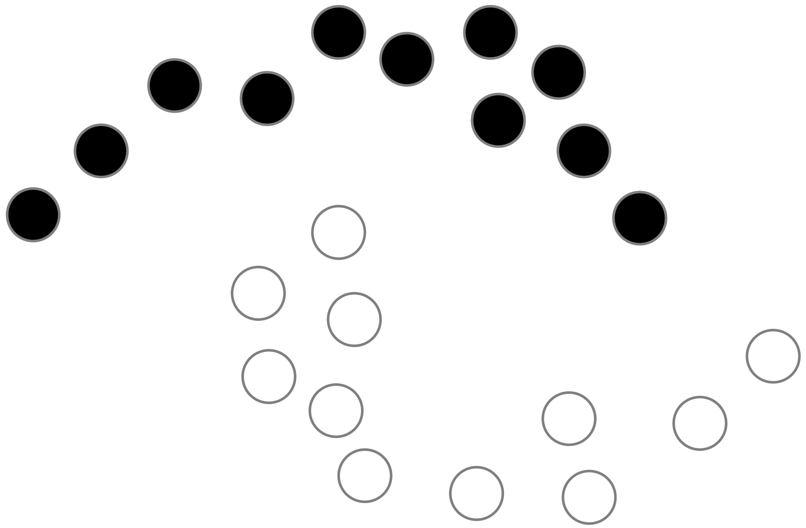
Inductive learning

Introduction

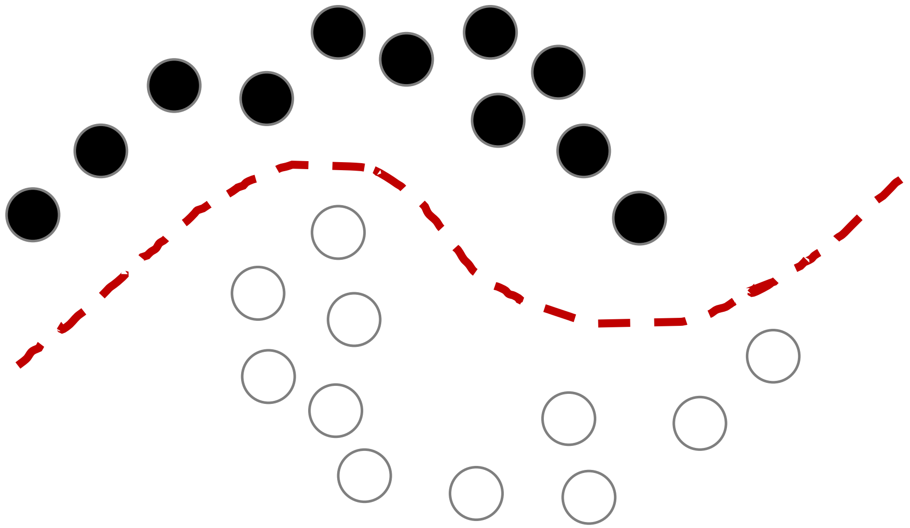
Categorization of semi-supervised learning

❖ Transductive learning aims to infer the correct labels for given unlabeled data only

❖ Inductive learning learns the correct mapping from features to class labels



Transductive learning

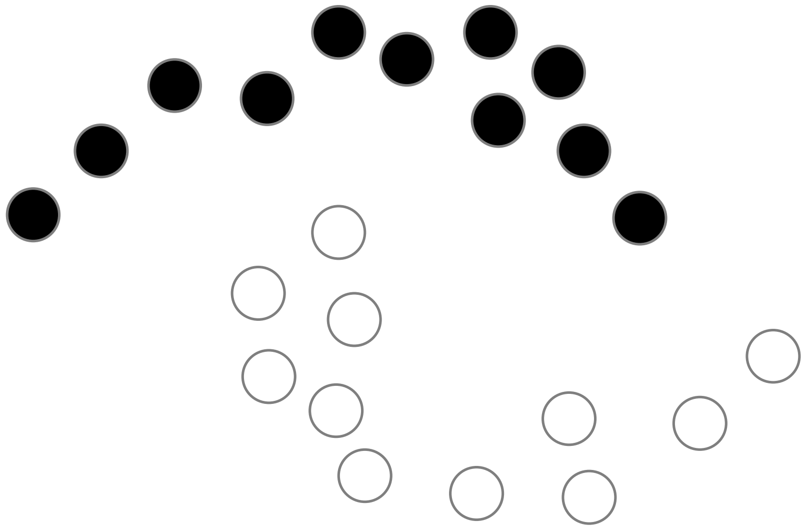


Inductive learning

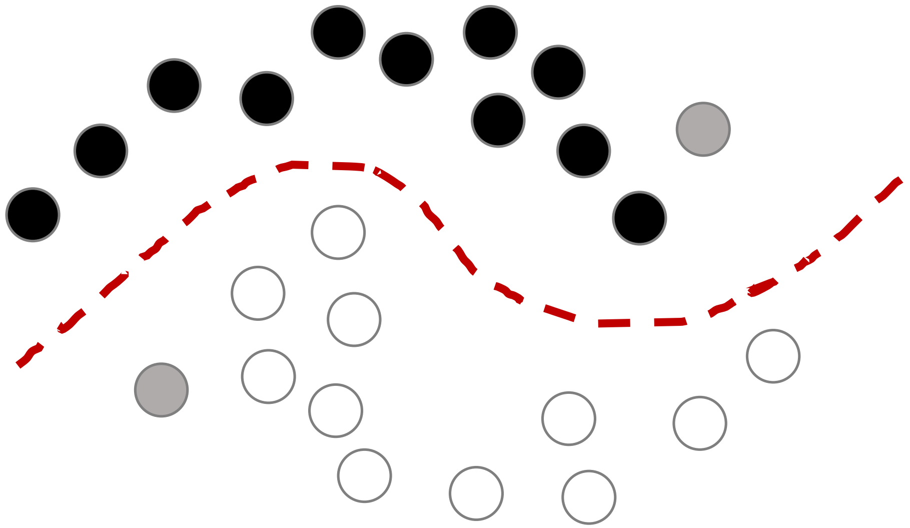
Introduction

Categorization of semi-supervised learning

- Transductive learning aims to infer the correct labels for given unlabeled data only
- Inductive learning learns the correct mapping from features to class labels



Transductive learning



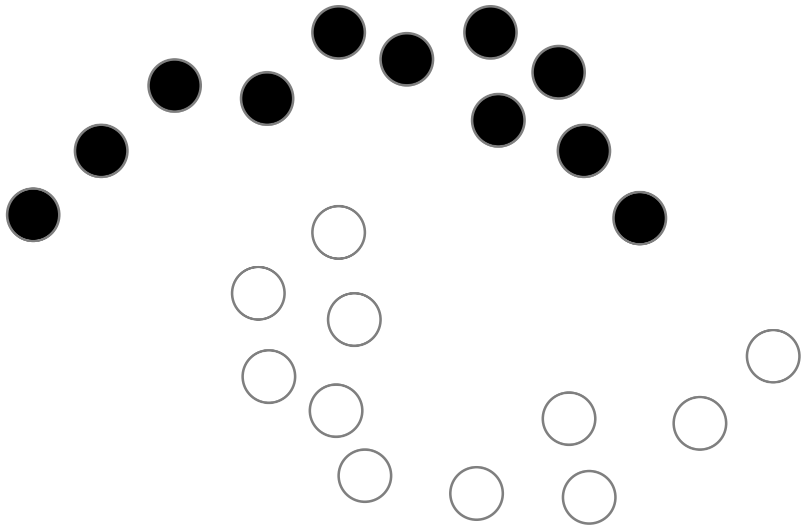
Inductive learning

Introduction

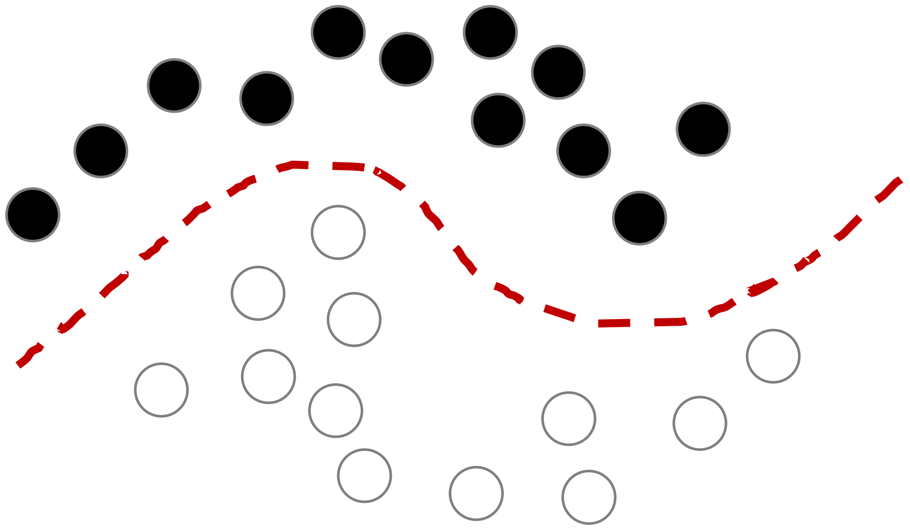
Categorization of semi-supervised learning

❖ Transductive learning aims to infer the correct labels for given unlabeled data only

❖ Inductive learning learns the correct mapping from features to class labels



Transductive learning

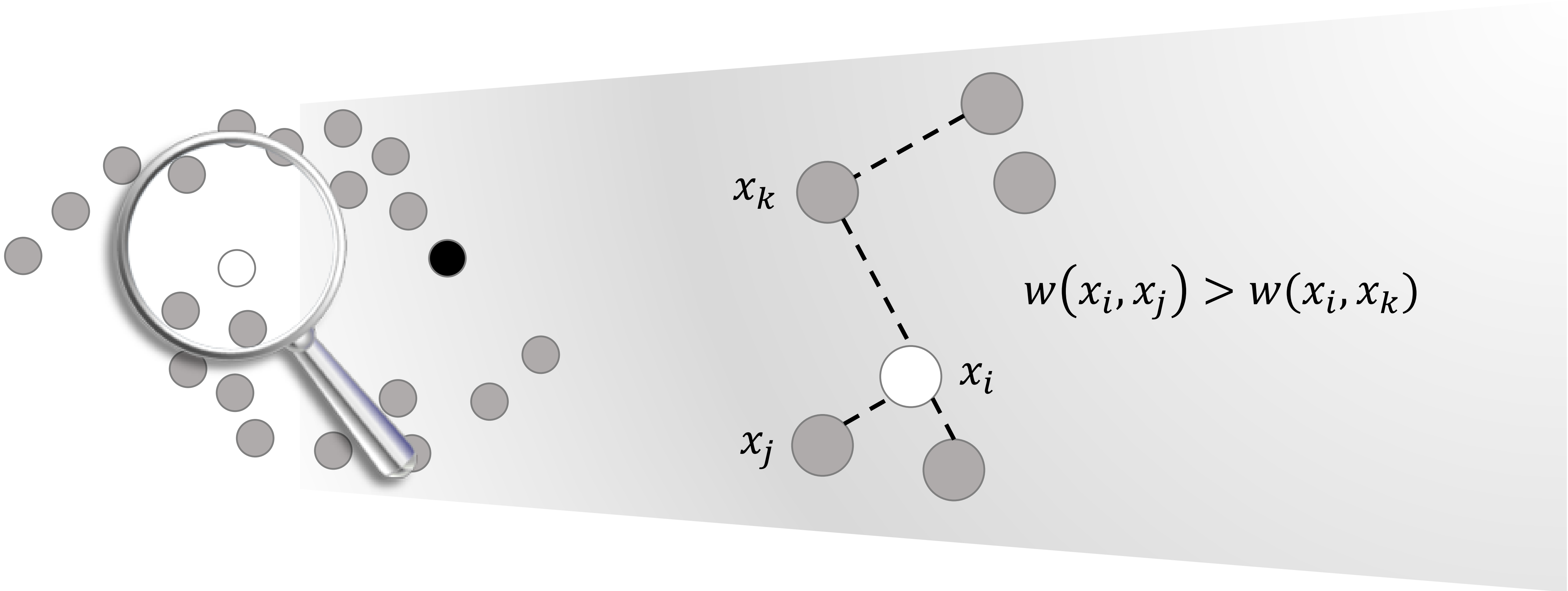


Inductive learning

Introduction

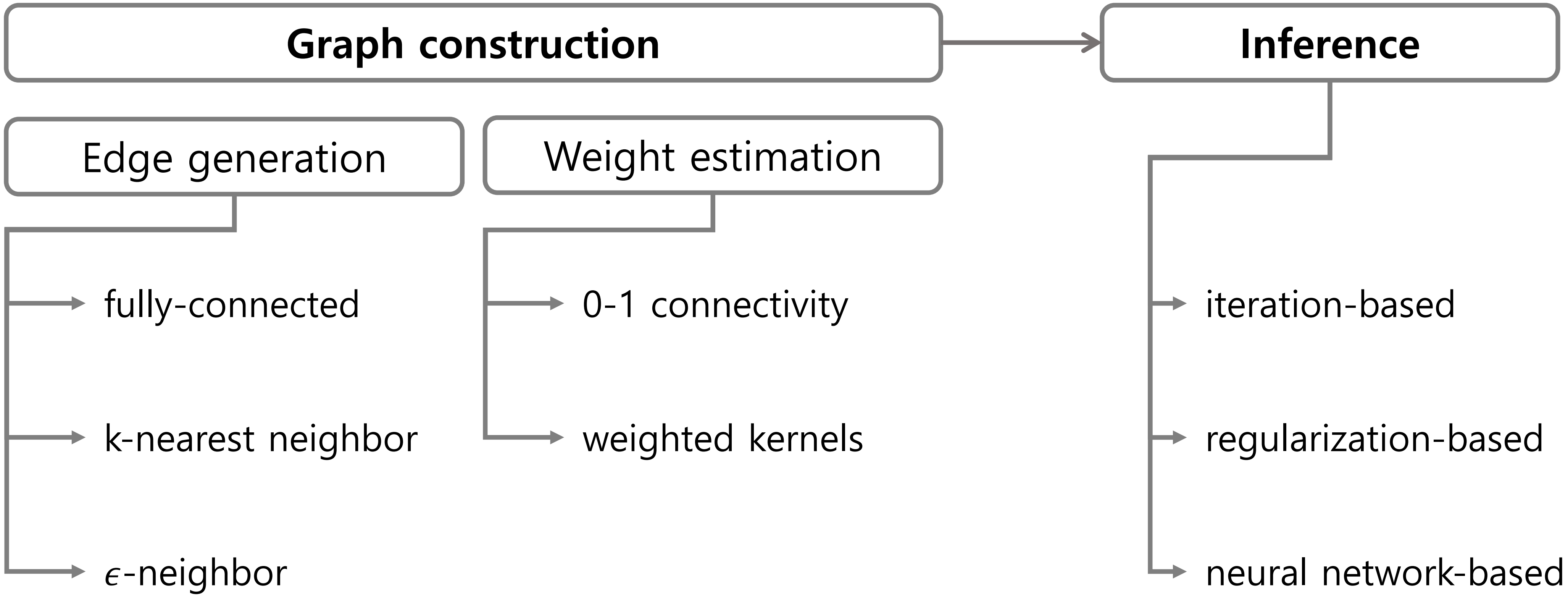
Graph-based semi-supervised learning

- Utilize the graph representation of data, where labeled and unlabeled instances are represented as vertices, and edges encode the similarity between instances



Introduction

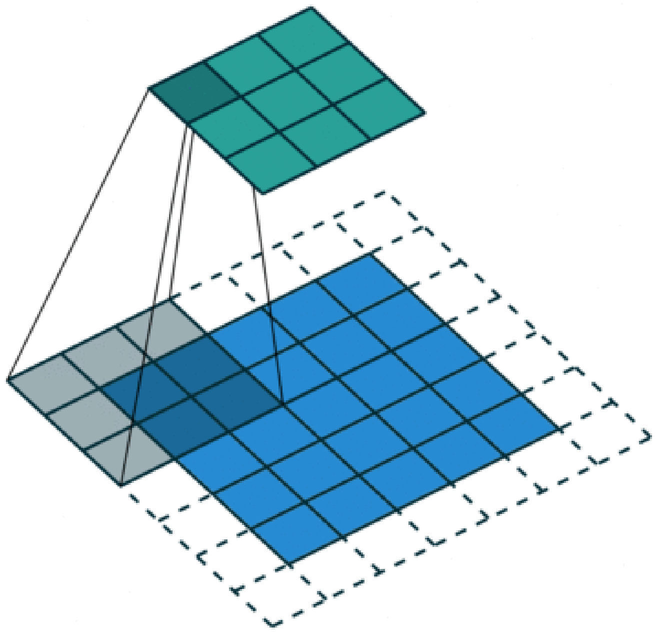
Graph-based semi-supervised learning

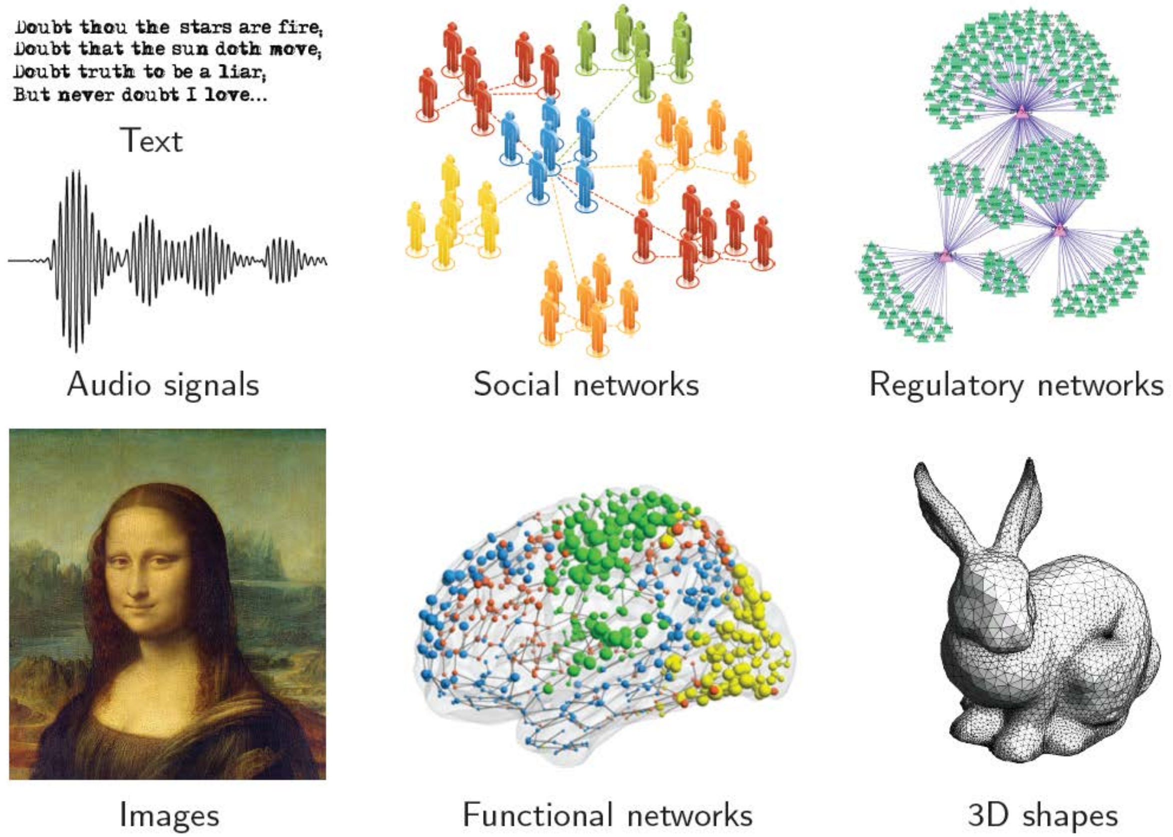


Introduction

Convolutional neural networks

❖ Localized convolutional filters recognize identical features independently of their spatial locations





What geometric structure of images, speech, video, and text is exploited by CNNs?

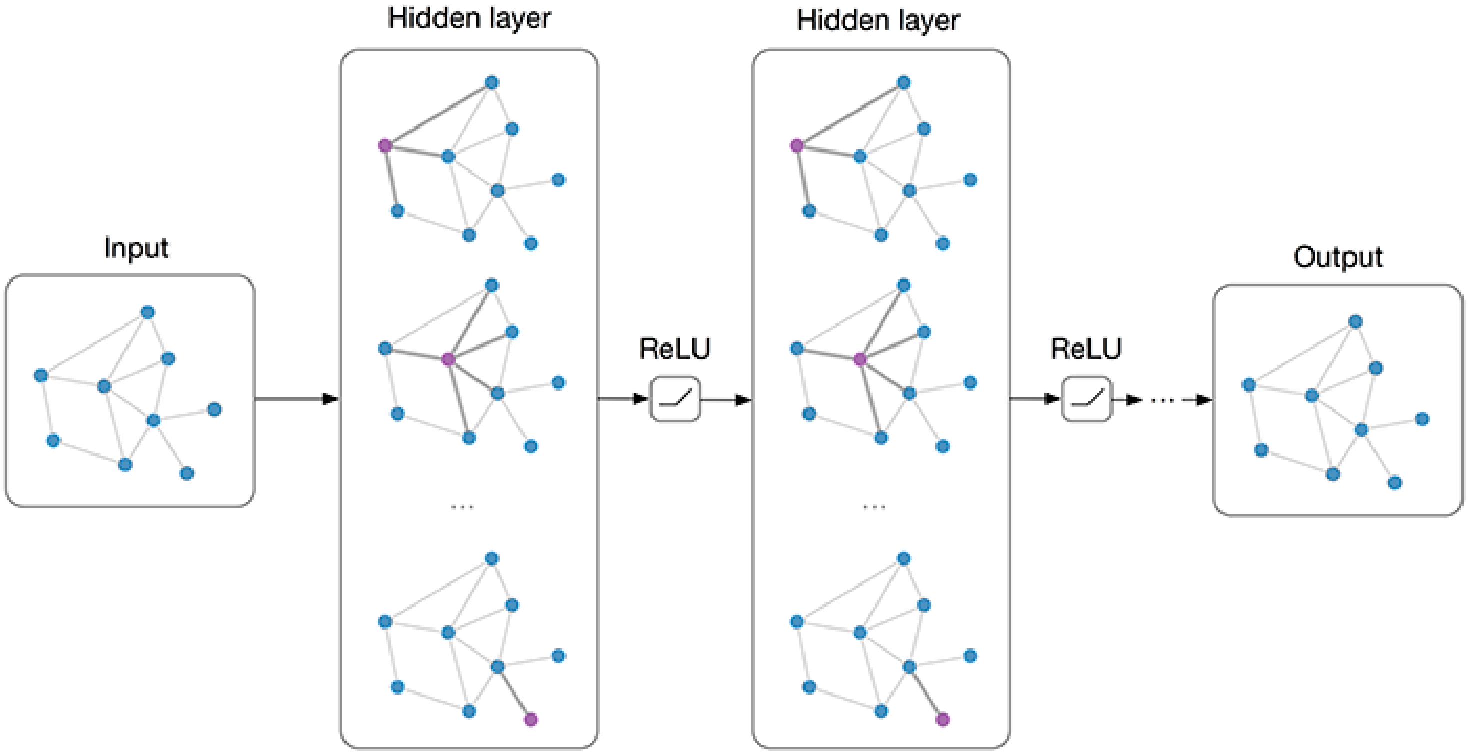
How to leverage such structure on non-Euclidean domains?

Method

Method

Graph convolution

- Extension of spatial convolutional neural networks to graph-structured data
- Spectral graph convolutions can be approximated via a localized first-order approximation*

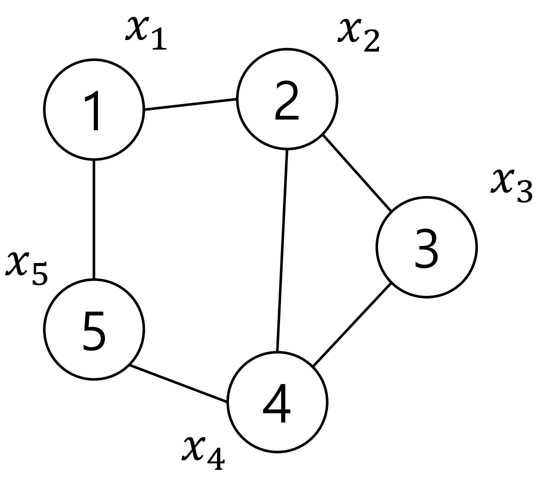


*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V,E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} \mathbf{1} & 1 & 0 & 0 & 1 \\ 1 & \mathbf{1} & 1 & 1 & 0 \\ 0 & 1 & \mathbf{1} & 1 & 0 \\ 0 & 1 & 1 & \mathbf{1} & 1 \\ 1 & 0 & 0 & 1 & \mathbf{1} \end{bmatrix}, \qquad \widetilde{D} = \begin{bmatrix} \mathbf{3} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{4} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{3} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{4} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{3} \end{bmatrix}$$

$$\tilde{A} = A + I_N,$$

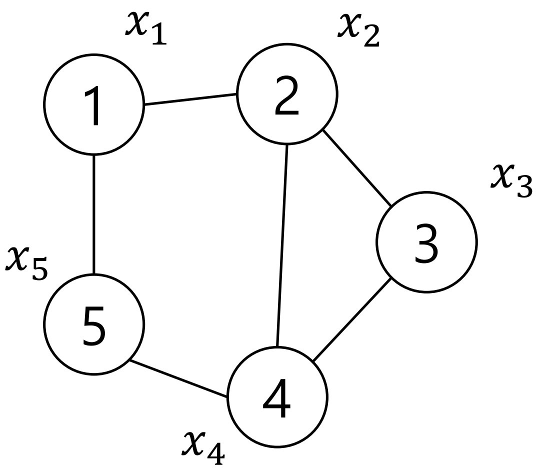
$$\widetilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij},$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V,E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \cdot H^l W^l\right),$$

$$\begin{aligned} where \quad & W^l = trainable\ weight\ matrix\ of\ layer\ l, \\ & \sigma(\cdot) = activation\ function, \\ & H^l \in \mathcal{R}^{N\times D_l}, H^0 = X \in \mathcal{R}^{N\times D_0} \end{aligned}$$

$$\tilde{A} = A + I_N,$$

$$\tilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij},$$

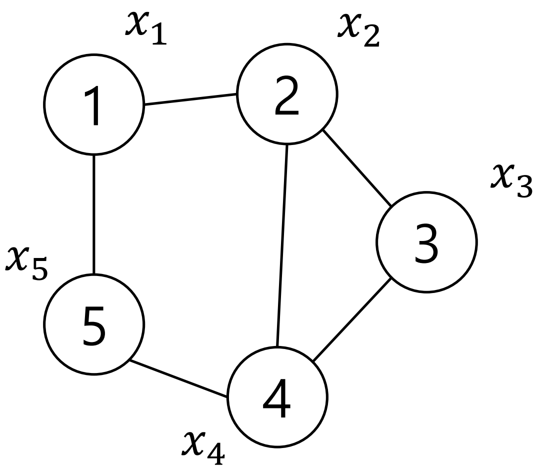
$$\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \tilde{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V,E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$\tilde{A} = A + I_N,$$

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij},$$

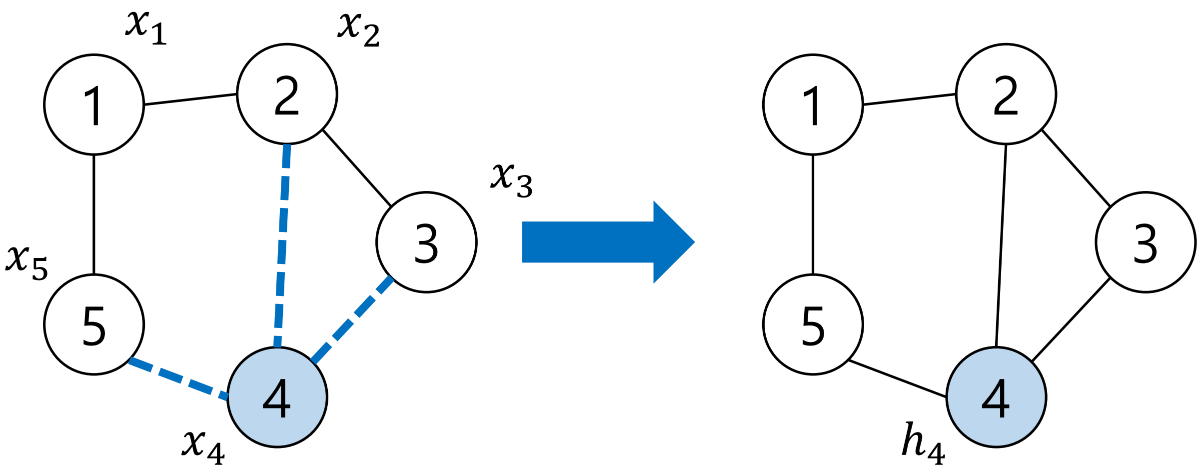
$$H^1 = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}} \cdot H^0 W^0\right)$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V, E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$\tilde{A} = A + I_N,$$
$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij},$$

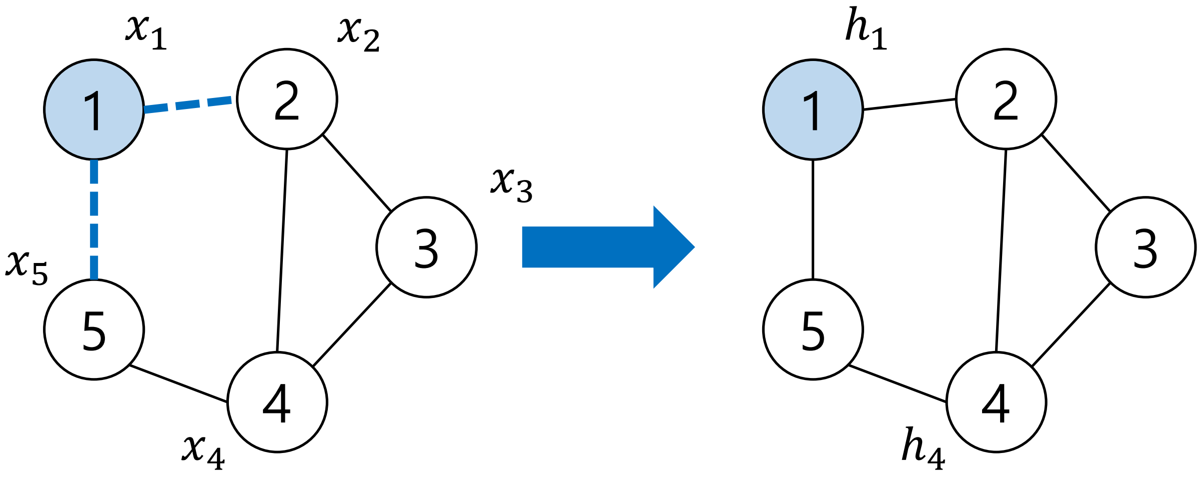
$$H^1 = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \cdot H^0 W^0\right)$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V,E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$\tilde{A} = A + I_N,$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij},$$

$$H^1 = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \cdot H^0 W^0\right)$$

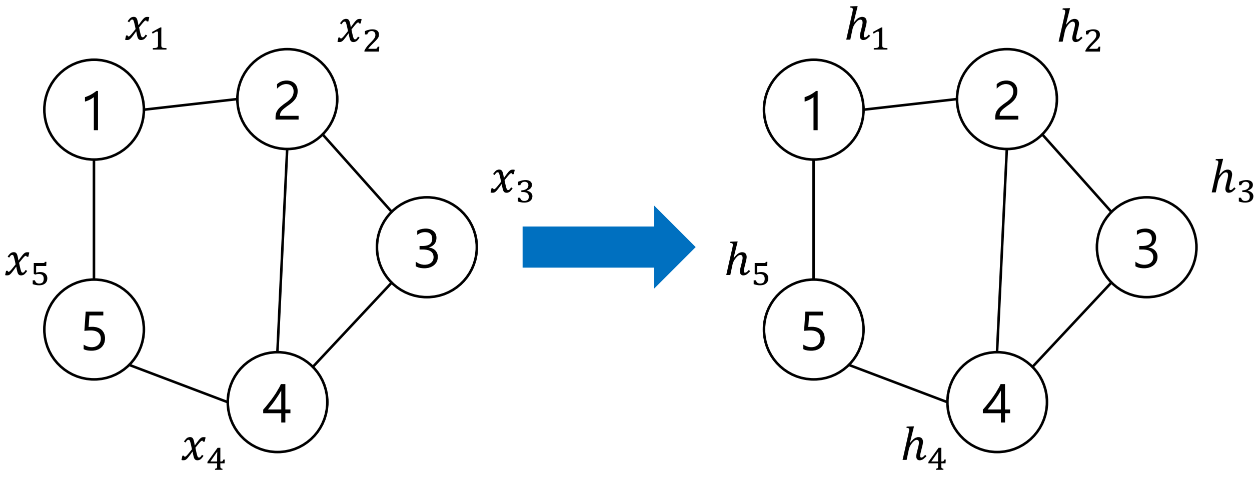
*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

❖ An undirected graph $G(V,E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.

❖ A graph convolutional layer* is defined as follows:



$\tilde{A} = A + I_N,$

$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij},$

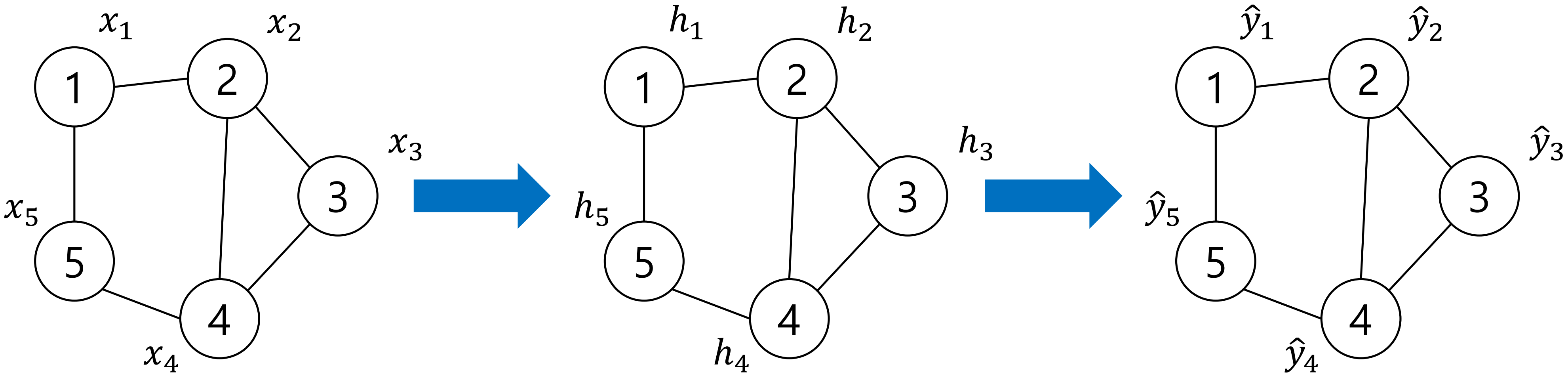
$$H^1 = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \cdot H^0 W^0\right)$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

- An undirected graph $G(V, E)$ is given, where each vertex corresponds to a instance and each edge weights encode the similarities between instances.
- A graph convolutional layer* is defined as follows:



$$\tilde{A} = A + I_N,$$
$$\widetilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij},$$

$$H^1 = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}} \cdot H^0 W^0\right)$$

$$\hat{Y} = softmax\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}} \cdot H^1 W^1\right)$$

*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution

❖ In a semi-supervised classification setting, the graph convolutional network is trained to minimize the categorical cross-entropy loss **over only the labeled data**

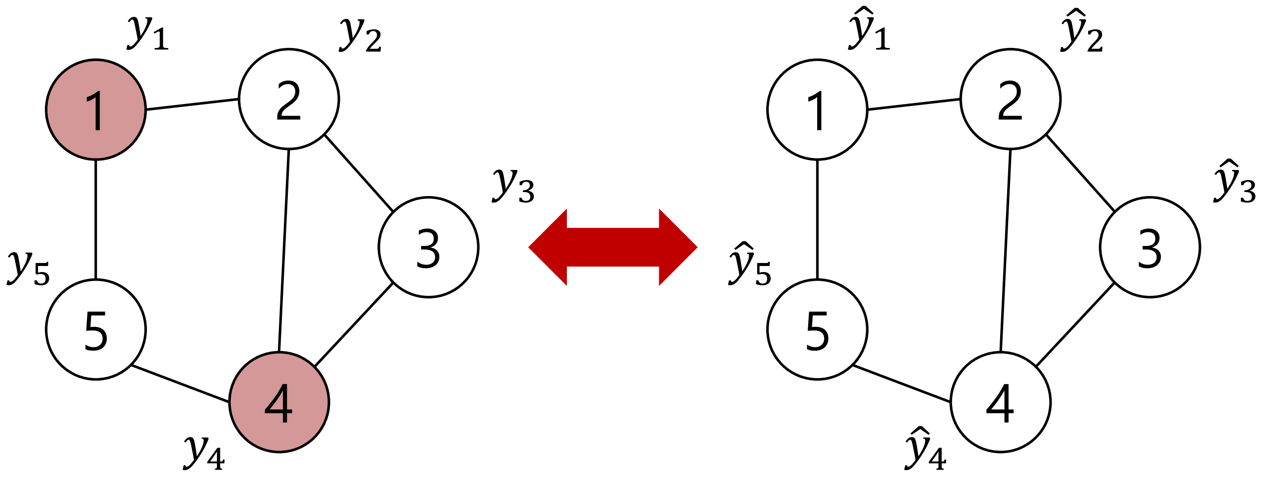
$$minimize \sum_{i=1}^{L} \sum_{c=1}^{C} -y_{ic} \cdot log\hat{y}_{ic}$$

$y_i \in \mathcal{R}^C,$

$N = L + U,$

$L \in \{1,4\},$

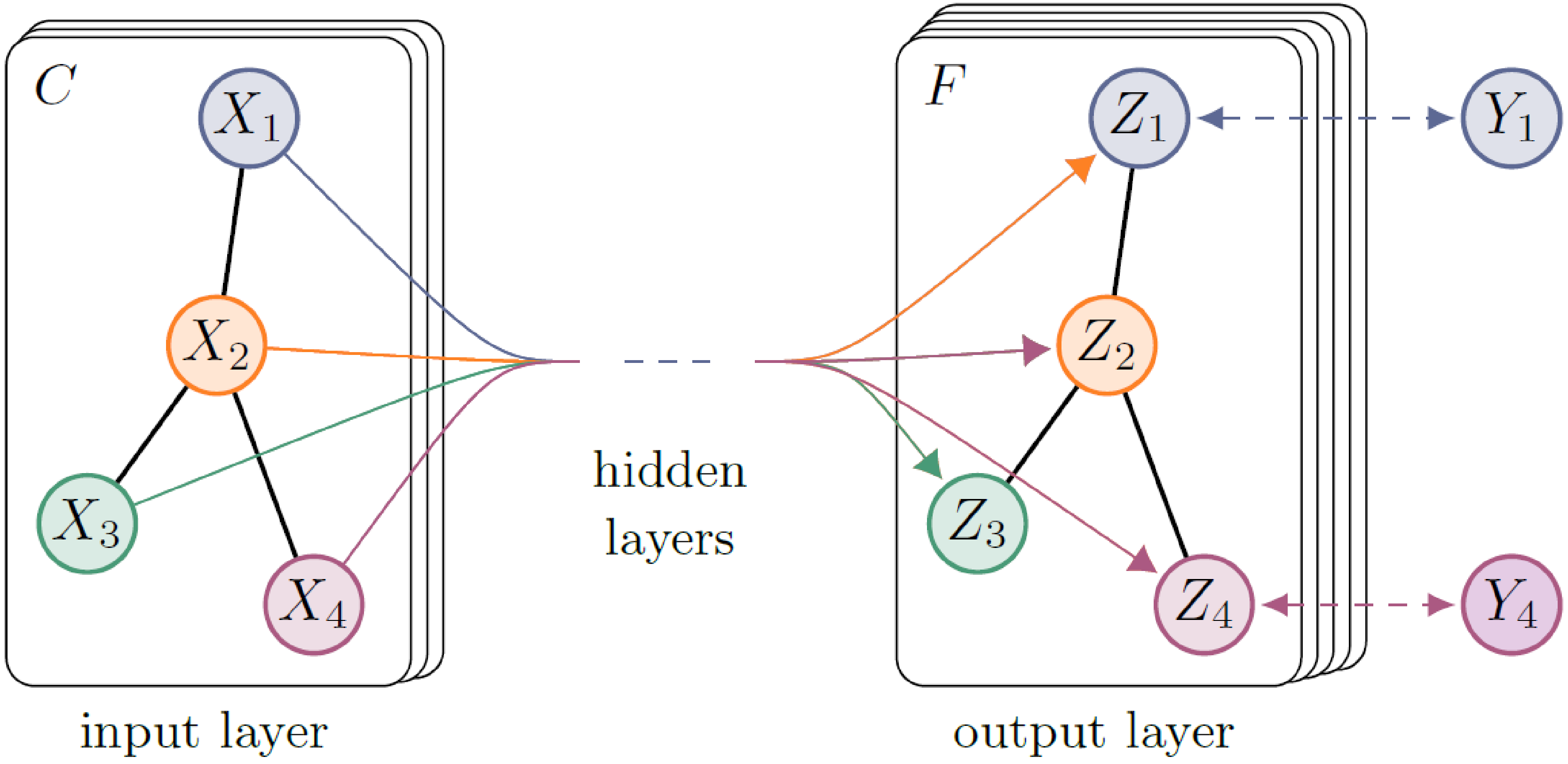
$\mathrm{U} \in \{2,3,5\}.$



*: Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Method

Graph convolution for SSL



Experiments

Experiments

Datasets

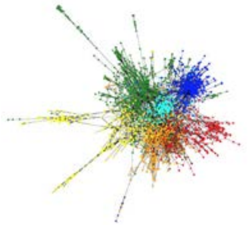


Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Experiments

Comparison with other graph SSL algorithms

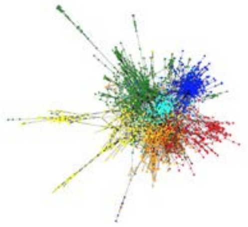


Table 2: Summary of results in terms of classification accuracy (in percent).

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Experiments

Comparison of propagation models

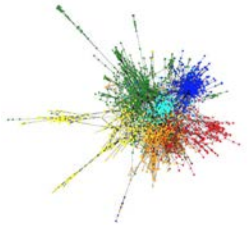
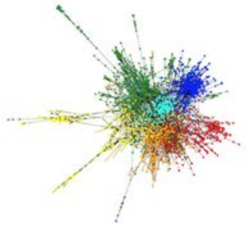


Table 3: Comparison of propagation models.

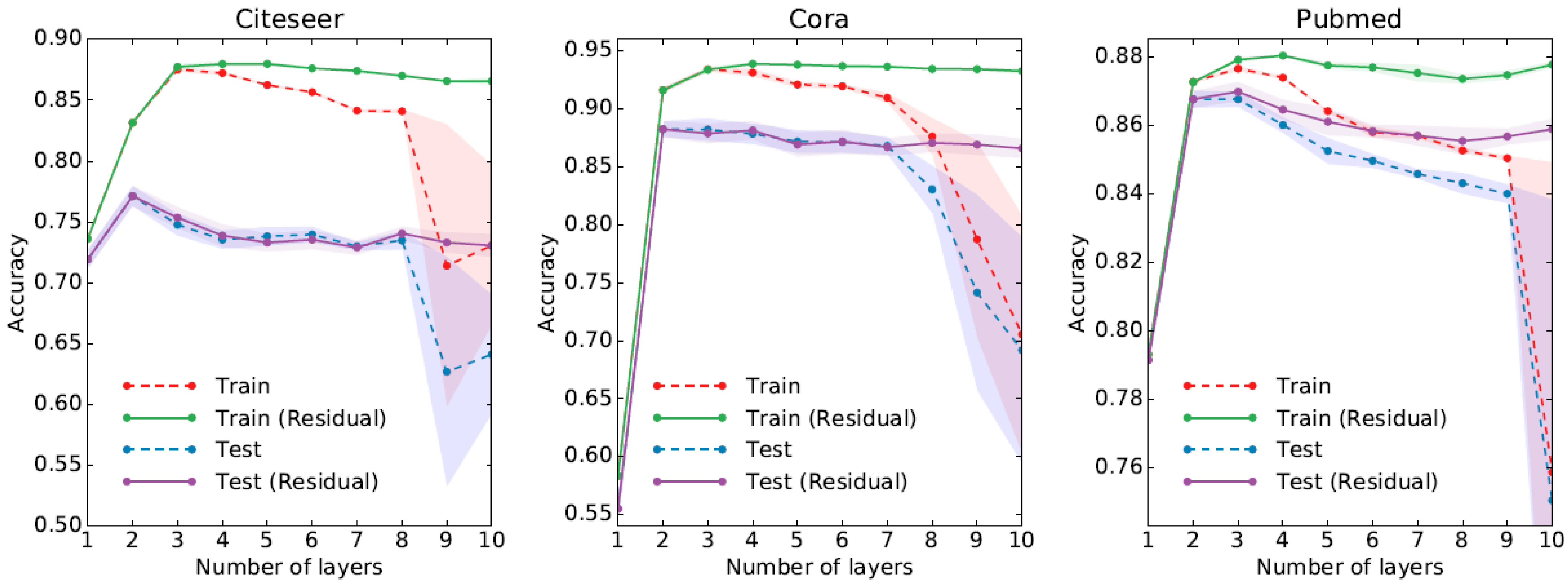
Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5)	$K=3$	$\sum_{k=0}^{K} T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
	$K=2$		69.6	81.2	73.8
1st-order model (Eq. 6)		$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)		$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	**70.3**	**81.5**	**79.0**
1st-order term only		$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Experiments

Model depth with skip connections



$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) + H^{(l)}$$



Conclusion

Conclusion

- A new neural network model which operates directly on graphs, motivated from a first order approximation of spectral graph convolutions
- Outperforms other graph-based SSL methods by a significant margin
- Geometric Deep Learning on Graphs and Manifolds, NIPS 2017 tutorial, video