Geometric Deep Learning Techniques on Graphs

Asiri Wijesinghe, Qing Wang

Research School of Computer Science
Australian National University
asiri.wijesinghe@anu.edu.au, qing.wang@anu.edu.au

June 17, 2021

Agenda

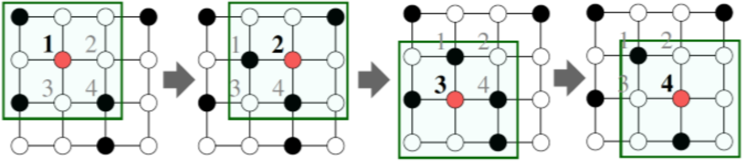
Agenda

Introduction

- Geometric deep learning techniques generalize the deep neural models on non-Euclidean domain such as graphs and manifolds.
- CNNs are a powerful deep learning approach which has been widely applied in various fields:
  - Object recognition
  - Image classification
  - Semantic segmentation
- Traditionally, CNNs only deal with data that has a regular Euclidean structure, such as images, videos and text.

Introduction: Convolution Operation

- Number of parameters are independent of the input size.
- Parameter sharing and sparsity of connections.
- Filters are localized and learnable.
- A filter can perform shift-invariance directly on an Euclidean domain.

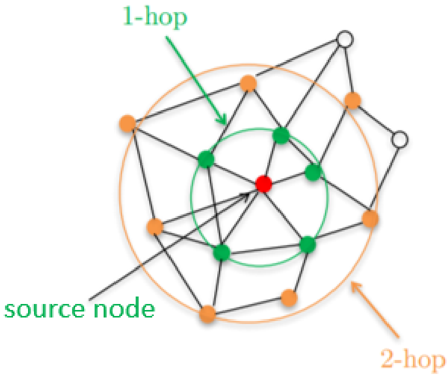


Introduction: Convolution on Graphs

- Graph convolution is more challenging due to its irregular non-Euclidean structure.
- The notion of shift-invariance cannot be applied directly on a non-Euclidean domain.
- Graph convolution aggregates the auxiliary information from localized neighborhood entities to generate intermediate feature maps.
- Convolution operation mainly depends on the graph filter.
- Redefine graph filters to adhere the localization w.r.t. neighborhood structure of the given node.

Introduction: Graph Filters

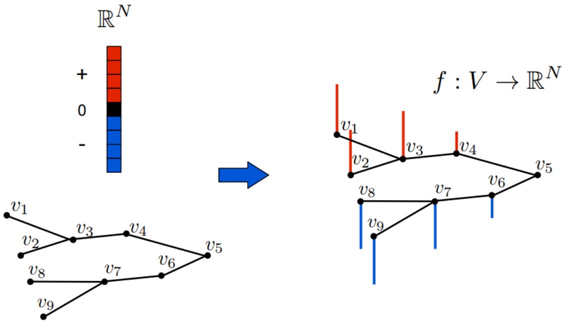
- There are two categories of graph filters:
  - Spatial graph filters: convolutions directly defined on graphs through neighbors that are spatially close to a current vertex.
  - Spectral graph filters: convolutions indirectly defined on graphs through spectral representations.



- Let $G = (V, E, A)$ be an undirected and weighted graph, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, and $A \in \mathbb{R}^{n \times n}$ is an adjacency matrix which encodes the weights of edges.

Definition (Graph Signal)

A *graph signal* is represented as a vector $x \in \mathbb{R}^n$ whose i^{th} component x_i is the value of x at the i^{th} vertex in V.



Introduction: Definitions

Definition (Graph Laplacian)

The graph Laplacian is defined as $L = I - D^{-1/2}AD^{-1/2}$, where $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $D_{ii} = \sum_j A_{ij}$ and I is an identity matrix.

Definition (Spectral Decomposition)

L is diagonalizable by the eigendecomposition such that $L = U \Lambda U^H$, where $\Lambda = diag\left([\lambda_0, \ldots, \lambda_{n-1}]\right) \in \mathbb{R}^{n \times n}$ and U^H is a hermitian transpose of U.

Introduction: Spectral Graph Filters

Definition (Spectral Filters)

Given a graph signal $x \in \mathbb{R}^n$, the *graph Fourier transform* of x is $\hat{x} = U^H x \in \mathbb{R}^n$ and its inverse is $x = U\hat{x}$. A *graph filter* h can filter x by altering the graph frequencies as,

$$h(L)x = h(U\Lambda U^H)x = Uh(\Lambda)U^H x, \tag{1}$$

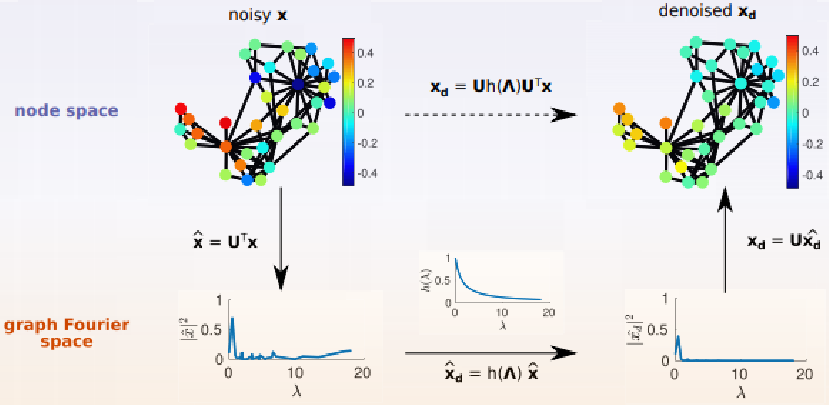
where $h(\Lambda)$ is a approximation of polynomial function. The K-hop localized polynomial filter is defined as,

$$h(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k, \tag{2}$$

where $\theta \in \mathbb{R}^n$ is a learnable filter coefficients.

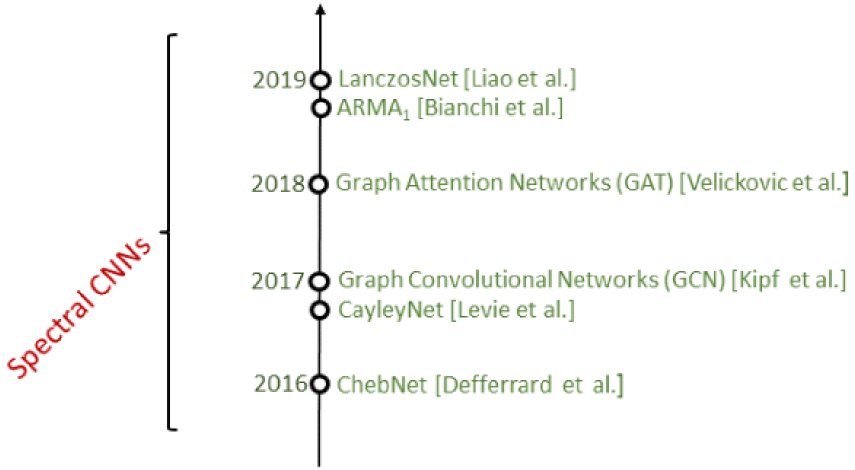
Introduction: Spectral Graph Filters

- Graphical illustration of a spectral filter operation on graph.



Agenda

History of Related Works



Limitations of Existing Works

- Most of the existing filters are basis-dependent.
- Polynomial filters:
  - Hard to specialize in narrow frequency bands.
  - Sensitive to changes in the underlying graph structure.
  - Very smooth and can hardly model sharp changes.
- Rational polynomial filters:
  - Accept a narrow band of frequencies.
  - Higher learning and computational complexities, as well as numerical instabilities.

Agenda

Research Problem

Can we build a geometric deep learning model for graphs that can meet the following requirements?

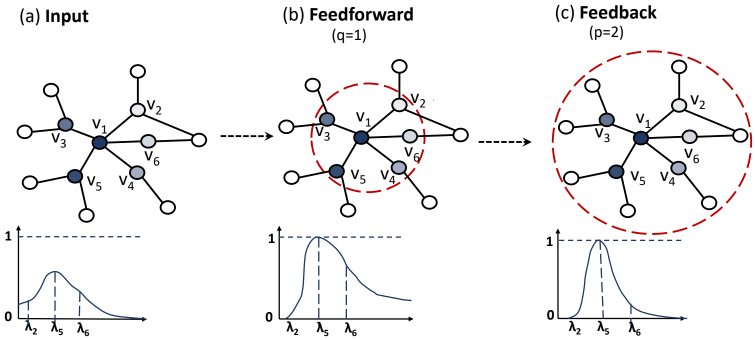
1. Improve k-hop localization filter operations on graphs using effective and efficient graph filters.
2. Define a spectral convolutional propagation rule using the proposed graph filters to perform a semi-supervised classification task on graph.

To answer this question, we propose the Distributed Feedback-Looped Networks (DFNets).

Agenda

DFNets: Overview

- We introduce a new class of spectral graph filters, called *feedback-looped* filters.
- A simplified example of illustrating feedback-looped filters.



DFNets: Key Contributions

- We aim to develop a new class of spectral graph filters that can overcome the limitations of prior works.
- We propose a novel spectral CNNs architecture to incorporate with feedback-looped graph filters:
  - Improved localization due to its rational polynomial form.
  - Efficient computation - linear convergence time and linear memory requirements w.r.t. the number of edges.
  - Theoretical properties - theoretically guaranteed convergence w.r.t. a specified error bound.
  - Dense architecture - layer-wise propagation rule with densely connects layers.
  - Layer-wise regularization term to prevent the generation of spurious features.

DFNets: Feedback-Looped Filters

- Feedback-looped filters belong to a class of Auto Regressive Moving Average (ARMA) filters.

Definition (Feedback-Looped Filters)

$$h_{\psi,\phi}(L)x = \left(I + \sum_{j=1}^{p} \psi_j L^j\right)^{-1} \left(\sum_{j=0}^{q} \phi_j L^j\right) x, \tag{3}$$

where parameters p and q refer to the *feedback* and *feedforward* degrees, respectively. $\psi \in \mathbb{C}^p$ and $\phi \in \mathbb{C}^{q+1}$ are two vectors of complex coefficients. The frequency response of feedback-looped filters is defined as:

$$h(\lambda_i) = \frac{\sum_{j=0}^{q} \phi_j \lambda_i^j}{1 + \sum_{j=1}^{p} \psi_j \lambda_i^j}. \tag{4}$$

DFNets: Feedback-Looped Filters

- To circumvent the issue of matrix inversion for large graphs, feedback-looped filters use the following approximation:

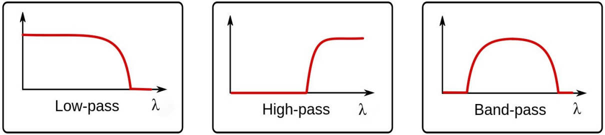
Recursive Approximation of Feedback-Looped Filters

$$\bar{x}^{(0)} = x \text{ and } \bar{x}^{(t)} = -\sum_{j=1}^{p} \psi_j \tilde{L}^j \bar{x}^{(t-1)} + \sum_{j=0}^{q} \phi_j \tilde{L}^j x, \qquad (5)$$

where $\tilde{L} = \hat{L} - (\frac{\hat{\lambda}_{max}}{2})I$, $\hat{L} = I - \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}$, $\hat{A} = A + I$, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and $\hat{\lambda}_{max}$ is the largest eigenvalue of $\hat{L}$.

DFNets: Feedback-Looped Filters

- Two techniques to alleviate the issues of gradient vanishing/ exploding and numerical instabilities:
  - Scaled-normalization technique:
    (a) $\tilde{L} = \hat{L} - (\frac{\hat{\lambda}_{max}}{2})I$ - centralize the eigenvalues of the Laplacian $\hat{L}$ and reduce its spectral radius bound.
  - Cut-off frequency technique:
    (a) Map graph frequencies to a uniform discrete distribution, we define a *cut-off frequency* $\lambda_{cut} = (\frac{\lambda_{max}}{2} - \eta)$, where $\eta \in [0, 1]$.
    (b) This trick allows the generation of ideal high-pass filters so as to sharpen a signal by amplifying its graph Fourier coefficients.



DFNets: Coefficient Optimization

- We aim to find the optimal coefficients ψ and ϕ that make the frequency response as close as possible to the desired frequency response, i.e. to minimize the following error:

Frequency Response Error

$$\acute{e}(\tilde{\lambda}_i) = \hat{h}(\tilde{\lambda}_i) - \frac{\sum_{j=0}^{q} \phi_j \tilde{\lambda}_i^j}{1 + \sum_{j=1}^{p} \psi_j \tilde{\lambda}_i^j} \quad (6)$$

Linear Approximation of the Error (w.r.t. the coefficients ψ and ϕ)

$$e(\tilde{\lambda}_i) = \hat{h}(\tilde{\lambda}_i) + \hat{h}(\tilde{\lambda}_i) \sum_{j=1}^{p} \psi_j \tilde{\lambda}_i^j - \sum_{j=0}^{q} \phi_j \tilde{\lambda}_i^j. \quad (7)$$

DFNets: Coefficient Optimization

- The stable coefficients ψ and ϕ can be learned by minimizing e as a convex constrained least-squares optimization problem:

Objective Function

Let $e = [e(\tilde{\lambda}_0), \ldots, e(\tilde{\lambda}_{n-1})]^T$, $\hat{h} = [\hat{h}(\tilde{\lambda}_0), \ldots, \hat{h}(\tilde{\lambda}_{n-1})]^T$, $\alpha \in \mathbb{R}^{n \times p}$ with $\alpha_{ij} = \tilde{\lambda}_i^j$ and $\beta \in \mathbb{R}^{n \times (q+1)}$ with $\beta_{ij} = \tilde{\lambda}_i^{j-1}$ are two Vandermonde-like matrices. Then, we have $e = \hat{h} + diag(\hat{h})\alpha\psi - \beta\phi$.

$$\textbf{minimize}_{\psi,\phi} \; ||\hat{h} + diag(\hat{h})\alpha\psi - \beta\phi||_2 \qquad (8)$$

$$\textbf{subject to} \;\; ||\alpha\psi||_\infty \leq \gamma \text{ and } \gamma < 1$$

DFNets: Spectral Convolutional Layer

- The propagation rule of a spectral convolutional layer is defined as:

Spectral CNNs Layer

$$\bar{X}^{(t+1)} = \sigma(\mathbf{P}\bar{X}^{(t)}\theta_1^{(t)} + \mathbf{Q}X\theta_2^{(t)} + \mu(\theta_1^{(t)};\theta_2^{(t)}) + b), \tag{9}$$

where σ refers to a non-linear activation function such as $ReLU$. $\mathbf{P} = -\sum_{j=1}^{p} \psi_j \tilde{L}^j$ and $\mathbf{Q} = \sum_{j=0}^{q} \phi_j \tilde{L}^j$. $\bar{X}^{(0)} = X \in \mathbb{R}^{n \times f}$ is a graph signal matrix where f refers to the number of features. $\bar{X}^{(t)}$ is a matrix of activations in the t^{th} layer.

DFNets: Spectral Convolutional Layer

- In our model, each layer is directly connected to all subsequent layers in a feed-forward manner.
- We concatenate multiple preceding feature maps column-wise into a single tensor.
- Densely connected CNN architecture has several compelling benefits:
  - Reduce the vanishing-gradient issue.
  - Increase feature propagation and reuse.
  - Refine information flow between layers.

DFNets: Theoretical Analysis

- Feedback-looped filters have several nice properties:
  - Guaranteed convergence and stability - pole of rational polynomial filters should be in the unit circle of the z-plane to guarantee the stability.
  - Universal design - corresponding filter coefficients can be learned independently of the underlying graph and are universally applicable.
  - Complexity analysis:

Spectral Graph Filter	Type	Learning Complexity	Time Complexity	Memory Complexity
Chebyshev filters [1]	Polynomial	$O(k)$	$O(km)$	$O(m)$
Lanczos filters [2]		$O(k)$	$O(km^2)$	$O(m^2)$
Cayley filters [3]	Rational polynomial	$O((r+1)k)$	$O((r+1)km)$	$O(m)$
ARMA_1 filters [4]		$O(t)$	$O(tm)$	$O(m)$
d parallel ARMA_1 filters [4]		$O(t)$	$O(tm)$	$O(dm)$
Feedback-looped filters (ours)		$O(tp+q)$	$O((tp+q)m)$	$O(m)$

Agenda

Numerical Experiments

- We evaluate our models on two benchmark tasks:
  - Semi-supervised document classification.
  - Semi-supervised entity classification.
- Datasets:

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

- Baseline methods:
  - Five methods using spatial graph filters and seven methods using spectral graph filters.

Numerical Experiments

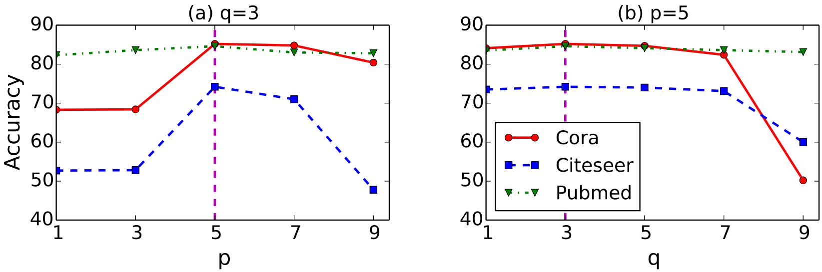
- Our spectral CNN models:
  - DFNet: a densely connected spectral CNN with feedback-looped filters.
  - DFNet-ATT: a self-attention based densely connected spectral CNN with feedback-looped filter.
  - DF-ATT: a self-attention based spectral CNN model with feedback-looped filters.
- Hyperparameter settings:

Model	L2 reg.	#Layers	#Units	Dropout	[p, q]	λ_{cut}
DFNet	9e-2	5	[8, 16, 32, 64, 128]	0.9	[5, 3]	0.5
DFNet-ATT	9e-4	4	[8, 16, 32, 64]	0.9	[5, 3]	0.5
DF-ATT	9e-3	2	[32, 64]	[0.1, 0.9]	[5, 3]	0.5

Numerical Experiments

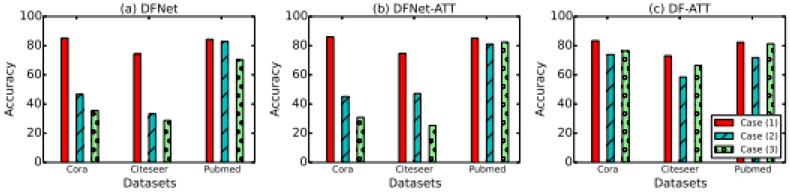
Model	Cora	Citeseer	Pubmed	NELL
SemiEmb	59.0	59.6	71.1	26.7
LP	68.0	45.3	63.0	26.5
DeepWalk	67.2	43.2	65.3	58.1
ICA	75.1	69.1	73.9	23.1
Planetoid*	64.7	75.7	77.2	61.9
Chebyshev	81.2	69.8	74.4	-
GCN	81.5	70.3	79.0	66.0
LNet	79.5	66.2	78.3	-
AdaLNet	80.4	68.7	78.1	-
CayleyNet	81.9*	-	-	-
$ARMA_1$	84.7	73.8	81.4	-
GAT	83.0	72.5	79.0	-
GCN + DenseBlock	82.7 ± 0.5	71.3 ± 0.3	81.5 ± 0.5	66.4 ± 0.3
GAT + Dense Block	83.8 ± 0.3	73.1 ± 0.3	81.8 ± 0.3	-
DFNet (ours)	**85.2 ± 0.5**	**74.2 ± 0.3**	**84.3 ± 0.4**	**68.3 ± 0.4**
DFNet-ATT (ours)	**86.0 ± 0.4**	**74.7 ± 0.4**	**85.2 ± 0.3**	**68.8 ± 0.3**
DF-ATT (ours)	83.4 ± 0.5	73.1 ± 0.4	**82.3 ± 0.3**	**67.6 ± 0.3**

- Comparison under different polynomial orders:
  - Evaluate DFNet on three citation network datasets using different polynomial orders p= [1,3,5,7,9] and q= [1,3,5,7,9].



Numerical Experiments

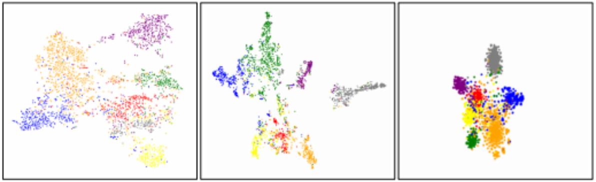
- How effectively the scaled-normalisation and cut-off frequency techniques can help to learn graph representations?



  - Case (1): using both scaled-normalization and cut-off frequency.
  - Case (2): using only cut-off frequency.
  - Case (3): using only scaled-normalization.

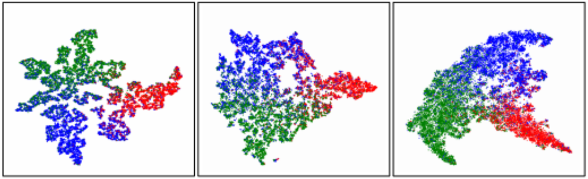
Numerical Experiments

- We analyze the node embeddings by DFNets over two datasets.
  - Cora embeddings in a 2-D space.



(a) GCN (b) GAT (c) DFNet (ours)

  - Pubmed embeddings in a 2-D space.



(a) GCN (b) GAT (c) DFNet (ours)

Agenda

Conclusion

- We have proposed a novel spectral CNNs method with Feedback-Looped filters for graphs called DFNets.
  - Improved localization due to its rational polynomial form.
  - Efficient computation - linear convergence time and linear memory requirements w.r.t. the number of edges.
  - Theoretical properties - theoretically guaranteed convergence w.r.t. a specified error bound.
  - Dense architecture - layer-wise propagation rule with densely connects layers.
  - Layer-wise regularization term to prevent the generation of spurious features.
- DFNets outperformed state-of-the-art methods in both benchmark tasks over all datasets.

References I

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst.
Convolutional neural networks on graphs with fast localized spectral filtering.
In *Advances in neural information processing systems (NeurIPS)*, pages 3844–3852, 2016.

Thomas N Kipf and Max Welling.
Semi-supervised classification with graph convolutional networks.
In *International Conference on Learning Representations (ICLR)*, 2017.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein.
Cayleynets: Graph convolutional neural networks with complex rational spectral filters.
IEEE Transactions on Signal Processing, 67(1):97–109, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio.
Graph attention networks.
International Conference on Learning Representations (ICLR), 2017.

Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel.
LanczosNet: Multi-scale deep graph convolutional networks.
In *Proceedings of the seventh International Conference on Learning Representation (ICLR)*, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec.
Inductive representation learning on large graphs.
In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1024–1034, 2017.

References III

Elvin Isufi, Andreas Loukas, Andrea Simonetto, and Geert Leus.
Autoregressive moving average graph filtering.
IEEE Transactions on Signal Processing, 65(2):274–288, 2017.

Filippo Maria Bianchi, Daniele Grattarola, L Livi, and C Alippi.
Graph neural networks with convolutional ARMA filters.
arXiv preprint arXiv:1901.01343, 2019.

Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert.
Deep learning via semi-supervised embedding.
In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty.
Semi-supervised learning using gaussian fields and harmonic functions.
In *Proceedings of the 20th International conference on Machine learning (ICML)*, pages 912–919, 2003.

References IV

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena.
Deepwalk: Online learning of social representations.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD)*, pages 701–710, 2014.

Qing Lu and Lise Getoor.
Link-based classification.
In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 496–503, 2003.

Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov.
Revisiting semi-supervised learning with graph embeddings.
In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 40–48, 2016.