MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing

Sami Abu-El-Haija\(^1\), Bryan Perozzi\(^2\), Amol Kapoor\(^2\), Nazanin Alipourfard\(^1\), Kristina Lerman\(^1\), Hrayr Harutyunyan\(^1\), Greg Ver Steeg\(^1\), Aram Galstyan\(^1\)

Code: [http://github.com/samihaija/mixhop](http://github.com/samihaija/mixhop)

Slides: [http://sami.haija.org/icml19](http://sami.haija.org/icml19)
Agenda

- Review Graph Convolutional Networks (GCN)
  - Application Semi-Supervised Node Classification (SSNC)
  - Shortcoming of GCN
- MixHop: Higher-Order GCN
  - Sparsification
- MixHop Results on SSNC
Agenda

- Review Graph Convolutional Networks (GCN)
  - Application Semi-Supervised Node Classification (SSNC)
  - Shortcoming of GCN
- MixHop: Higher-Order GCN
  - Sparsification
- MixHop Results on SSNC
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017

Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML’19

Poster #88
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017

Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML’19

Poster #88
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

Input Features → GC Layer 1 → Latent Features → GC Layer L → Output Features

[1] Kipf & Welling, ICLR 2017
Information Sciences Institute
Abu-El-Haija et al, MixHop, ICML'19
Graph Convolutional Network (GCN) [1]

Train on semi-supervised node classification:

- measure **Loss** on labeled nodes \((y_4, y_2)\)

[1] Kipf & Welling, ICLR 2017

Graph Convolutional Network (GCN) [1]

Train on semi-supervised node classification:
- measure **Loss** on labeled nodes \((y_4, y_2)\)
- Backprop to learn GC layers.

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017

Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML’19

Poster #88
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML'19
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML'19
Graph Convolutional Network (GCN) [1]

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

[1] Kipf & Welling, ICLR 2017
Graph Convolutional Network (GCN) [1]

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML'19
Graph Convolutional Network (GCN) [1]

Tensor Graph

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017

Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML'19
Graph Convolutional Network (GCN) [1]

Tensor Graph

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML'19
Shortcoming of Vanilla GCN

Vanilla GC Layer

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML'19
Shortcoming of Vanilla GCN

😊 fc is shared ⇒ inductive

Vanilla GC Layer

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]
Shortcoming of Vanilla GCN

😊 fc is shared ⇒ inductive
😢 Appendix Experiments of [1] shows no gains beyond 2 layers

Vanilla GC Layer

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]

[1] Kipf & Welling, ICLR 2017

Abu-El-Haija et al, MixHop, ICML’19
Shortcoming of Vanilla GCN

😊 fc is shared ⇒ inductive
😢 Appendix Experiments of [1] shows no gains beyond 2 layers
😢 cannot mix neighbors from various distances in arbitrary linear combinations

Vanilla GC Layer

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]
Shortcoming of Vanilla GCN

😊 fc is shared ⇒ inductive
😢 Appendix Experiments of [1] shows no gains beyond 2 layers
😢 cannot mix neighbors from various distances in arbitrary linear combinations e.g. cannot learn Gabor Filters!

Vanilla GC Layer

\[ H^{(1)} = \sigma( \hat{A} X W^{(1)} ) \]

[1] Kipf & Welling, ICLR 2017

Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML'19

Poster #88
Shortcoming of Vanilla GCN

😊 fc is shared ⇒ inductive
😢 Appendix Experiments of [1] shows no gains beyond 2 layers
😢 cannot mix neighbors from various distances in arbitrary linear combinations e.g. cannot learn **Gabor Filters**!

[1] Kipf & Welling, ICLR 2017

**Vanilla GC Layer**

\[ H^{(1)} = \sigma(\hat{A}XW^{(1)}) \]
Detour: Review Gabor Filters
Neuroscientists discover their importance in the primate visual cortex [2, 3]:

Detour: Review Gabor Filters

Neuroscientists discover their importance in the primate visual cortex [2, 3]:

Further, they are automatically recovered by training CNNs on images [4, 5]

Main Motivation
Main Motivation

Extend the class of representations realizable by GCNs e.g. to learn Gabor Filters
Agenda

- Review Graph Convolutional Networks (GCN)
  - Application Semi-Supervised Node Classification (SSNC)
  - Shortcoming of GCN
- MixHop: Higher-Order GCN
  - Sparsification
- MixHop Results on SSNC
Our Model: MixHop

MixHop GC Layer

Vanilla GC Layer

\[ H^{(1)} = \sigma (\hat{A} X W^{(1)}) \]
Our Model: MixHop

MixHop GC Layer

Vanilla GC Layer

$$H^{(1)} = \sigma(\hat{A}XW^{(1)})$$

Couple of code lines implements concatenation
Our Model: MixHop

MixHop GC Layer

\[ H^{(1)} = \sigma \left( \widehat{A}^j X W_j^{(1)} \right) \]

\( j \in P \)

Vanilla GC Layer

\[ H^{(1)} = \sigma(\widehat{A} X W^{(1)}) \]

Couple of code lines implements concatenation
Our Model: MixHop

MixHop GC Layer

\[ H^{(1)} = \bigg\| \sigma \left( \hat{A}^j X W^{(1)}_j \right) \bigg\|_{j \in P} \]

Abu-El-Haija et al, *MixHop*, ICML'19
Our Model: MixHop

MixHop GC Layer

\[ H^{(1)} = \bigoplus_{j \in P} \sigma \left( \hat{A}^j X W_j^{(1)} \right) \]

Abu-El-Haija et al, MixHop, ICML’19
Our Model: MixHop

MixHop GC Layer

$$H^{(1)} = \sigma \left( \hat{A}^j X W_j^{(1)} \right)$$

Inductive
Can incorporate distant nodes
Our Model: MixHop

MixHop GC Layer

\[
H^{(1)} = \bigg\|_{j \in P} \sigma \left( \hat{A}^j X W_j^{(1)} \right)
\]

- Inductive
- Can incorporate distant nodes
- Can mix neighbors across distances in arbitrary linear combinations
Our Model: MixHop

MixHop GC Layer

\[ H^{(1)} = \sigma \left( \hat{A}^j X W_j^{(1)} \right) \]

- Inductive
- Can incorporate distant nodes
- Can mix neighbors across distances in arbitrary linear combinations
  i.e. can learn Gabor Filters!
Sparsification

We add group L2-Lasso Regularization to drop-out columns feature matrices, similar to [6]


Information Sciences Institute

Abu-El-Haija et al, MixHop, ICML'19

Poster #88
Sparsification

We add group L2-Lasso Regularization to drop-out columns feature matrices, similar to [6]

2\textsuperscript{nd} layer of Cora drops-out zeroth-power completely.

[images are rotated space]

Agenda

- Review Graph Convolutional Networks (GCN)
  - Application Semi-Supervised Node Classification (SSNC)
  - Shortcoming of GCN
- MixHop: Higher-Order GCN
  - Sparsification
- MixHop Results on SSNC
## Results on Citation Datasets

<table>
<thead>
<tr>
<th>Model</th>
<th>Citeseer</th>
<th>Cora</th>
<th>Pubmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-Layer MLP</td>
<td>70.6±1</td>
<td>69.0±1.1</td>
<td>78.3±0.54</td>
</tr>
<tr>
<td>Chebyshev (<a href="#">Defferrard et al., 2016</a>)</td>
<td>74.2±0.5</td>
<td>85.5±0.4</td>
<td>81.8±0.5</td>
</tr>
<tr>
<td>Vanilla GCN (<a href="#">Kipf &amp; Welling, 2017</a>)</td>
<td>76.7±0.43</td>
<td>86.1±0.34</td>
<td>82.2±0.29</td>
</tr>
<tr>
<td>GAT (<a href="#">Velickovic et al., 2018</a>)</td>
<td>74.8±0.42</td>
<td>83.0±1.1</td>
<td>81.8±0.18</td>
</tr>
<tr>
<td>MixHop: default architecture (ours)</td>
<td>76.3±0.41</td>
<td>87.0±0.51</td>
<td>83.6±0.68</td>
</tr>
<tr>
<td>MixHop: learned architecture (ours)</td>
<td><strong>77.0±0.54</strong></td>
<td><strong>87.2±0.32</strong></td>
<td><strong>83.8±0.44</strong></td>
</tr>
</tbody>
</table>

Table 3: Classification results on random partitions of ([Yang et al., 2016](#)) datasets.
Results on (Synthetic) Homophily Datasets

With less homophily, our performance gap increases

[Graph showing test accuracy vs. homophily for different models: MixHop, GAT, Chebyshev, Features (2-Layer MLP), and Vanilla GCN.]

Abu-El-Haija et al, MixHop, ICML'19
Results on (Synthetic) Homophily Datasets

With less homophily, our method learns more feature differences (i.e. Gabor-like Filters)

With less homophily, our performance gap increases
References

Conclusion

- With just a couple of lines, Kipf’s model can be extended to incorporate powers of (normalized) adjacency matrix
- Allowing it to learn general neighborhood mixing, and its special cases: Gabor-like Filter and Delta Ops
- Inspection shows Delta Ops are indeed learned with lower levels of homophily.

Thank you for listening!

Poster #88

Slides at: http://sami.haija.org/icml19