[1]University of Sydney; [2]University of Technology (Sydney)
*mili7522@uni.sydney.edu.au; elija.t.perrier@student.uts.edu.au; c.xu@sydney.edu.au*

# Deep Hierarchical Graph Convolution for Election Prediction from Geospatial Census Data

Mike Li[1], Elija Perrier[2], Chang Xu[1]

March 31, 2019

# Overview

# How can this...

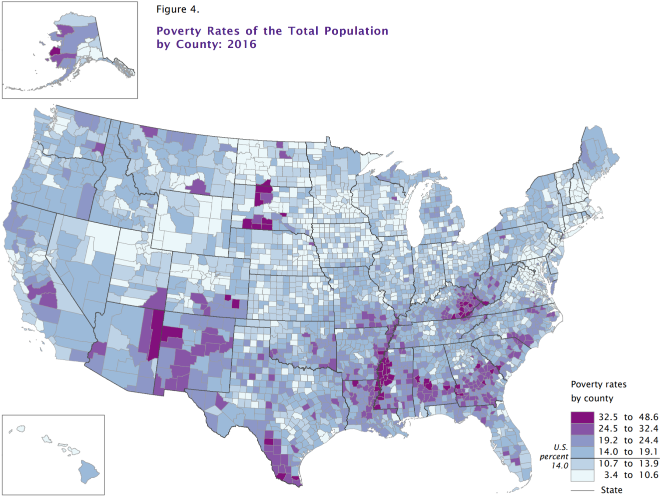



Figure: US Poverty Rates by County 2016

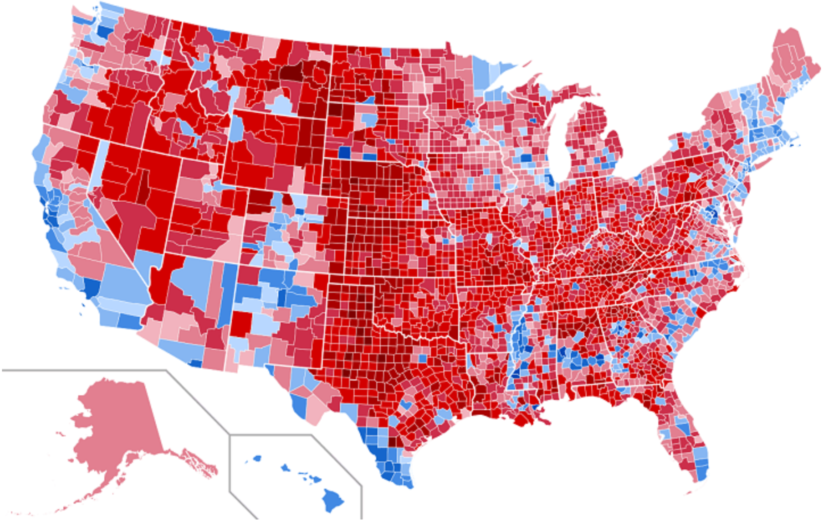


Figure: US 2016 Presidential Results by County (Red: Trump; Blue: Clinton)

# Or this...?

# Context: Hierarchical Graphs and GIS

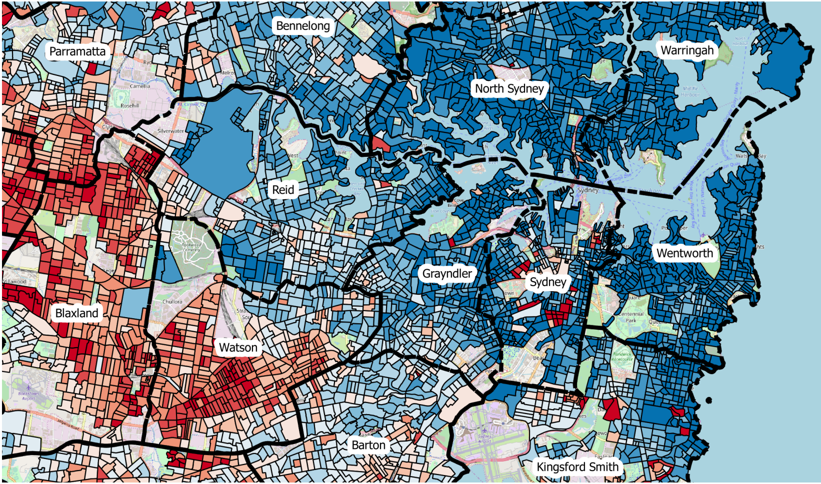



Figure: ABS SEIFA Index 2016 v Federal Electorates

# Predicting elections from GIS data

**Problem:**

- Predict two-party preferred vote at Australia federal 2016 election using Australian 2016 Census data
- Both GIS datasets and Census data organised into hierarchical aggregates
- More than 15,000 SA1s in New South Wales alone
- Census not very predictive of vote using standard GLMs
- Mapping all graph connections intractable

# Predicting elections from GIS data

**Problem:**

- Predict two-party preferred vote at Australia federal 2016 election using Australian 2016 Census data
- Both GIS datasets and Census data organised into hierarchical aggregates
- More than 15,000 SA1s in New South Wales alone
- Census not very predictive of vote using standard GLMs
- Mapping all graph connections intractable

**Solution:**

- Leverage intrinsic graph structure of GIS data
- Generate one-hop graph
- Extend GCNNs to handle hierarchical/auxiliary graphs
- Introduce adaptive learning on adjacency operators

# Convolutional Neural Networks

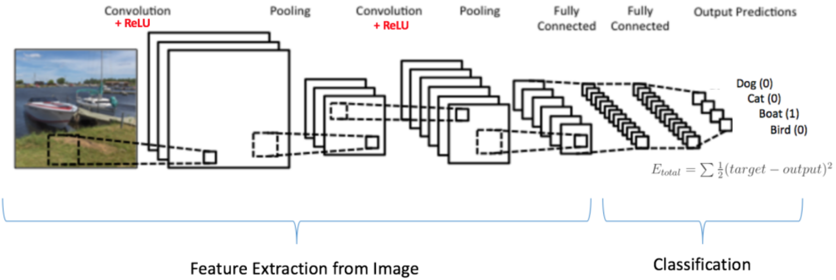



Figure: Convolutional Neural Networks (source: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/)

# Graphs

- Graphs are composed of:
  - Vertices (V)
  - Edges (E)
    - Directed or undirected
    - Can be weighted
- Represents entities and relationships, eg:
  - Brain regions
  - Citations
  - Social network

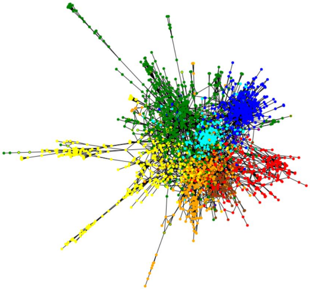


Figure: CORA Dataset - Citation Network

# Graph Convolutional Network

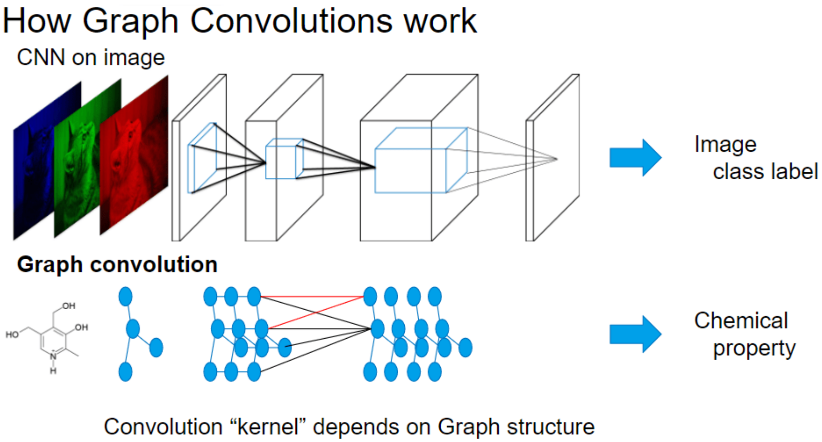



Figure: CNN vs Graph-CNN (source: https://preferredresearch.jp/2017/12/18/chainer-chemistry-beta-release/)

# Experiment: predicting Australian election results

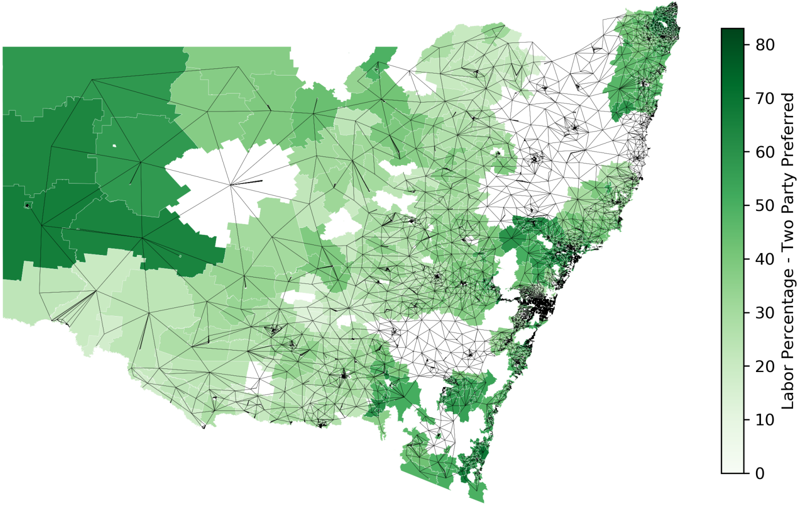



Figure: Labor party two-party preferred by Statistical Area 1 with graph overlay

# Multi-layered GIS hierarchy and Hierarchical GCNNs

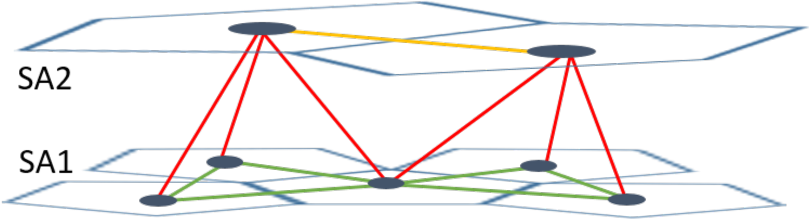



Figure: Schema of Census SA1-SA2 hierarchy: SA1 graph (green), SA2 graph (yellow), inter-graph adjacency relations (red). Projection operations transmit information to from SA1 to SA2, embeddings from SA2 to SA1

# Adding Hierarchy to Graph CNNs

**Hierarchical GCNNs**:

- Include hierarchical structure in GCNN
- Adjacency matrix for each graph level e.g. $\mathbf{A}$, SA1 level features $\mathbf{V} \in \mathbb{R}^n$
- SA2 level features $\mathbf{V} \in \mathbb{R}^m$
- avoid complexity of one big adjacency matrix
- Vertices $\mathbf{V}$ are convolved with $\mathbf{F}$ via matrix multiplication $\mathbf{V}_{\text{out}} = \mathbf{F}\mathbf{V}_{\text{in}} \in \mathbb{R}^n$.

# Filters, Projections and Embeddings

- Each hierarchical graph level's convolution can then be approximated as:

$$\mathbf{F}_1^{(n\times n)} \approx h_1 \mathbf{A}_1^{(n\times n)} \qquad \mathbf{F}_2^{(m\times m)} \approx h_2 \mathbf{A}_2^{(m\times m)} \tag{1}$$

- The hierarchical model deploys parallel filter-learning at different graph levels with intermediate transfer of information between graphs using the learnt embedding operator filters $\mathbf{E} = h_3\mathbf{A}_3$ and learnt projection operator filters $\mathbf{P} = h_4\mathbf{A}_4$:

$$\mathbf{E}^{(n\times m)}\mathbf{V}^{(m\times 1)} \rightarrow \mathbf{V}^{(n\times 1)} \tag{2}$$

$$\mathbf{P}^{(m\times n)}\mathbf{V}^{(n\times 1)} \rightarrow \mathbf{V}^{(m\times 1)} \tag{3}$$

- Learnt embedding operator filters $\mathbf{E} = h_3\mathbf{A}_3$ and projection operator filters $\mathbf{P} = h_4\mathbf{A}_4$, these encode adjacency relations between graph levels

# Output

**Hierarchical GCNNs**:

- The feature tensor output for $j$ filters for both the SA1 (with $c$ features) and SA2 (with $d$ features) levels is then:

$$\mathbf{V}_{1,\text{out}}^{(n\times j)} = \underbrace{\left(\mathbf{F}_{1}^{(n\times n\times c\times j)}\mathbf{V}_{\text{in}}^{(n\times c)}\right)}_{n\times j} + \mathbf{b}_{1}^{(n\times j)} \tag{4}$$

$$\mathbf{V}_{2,\text{out}}^{(m\times j)} = \underbrace{\left(\mathbf{F}_{2}^{(m\times m\times d\times j)}\mathbf{V}_{\text{in}}^{(m\times d)}\right)}_{m\times j} + \mathbf{b}_{1}^{(m\times j)} \tag{5}$$

- Biases $\mathbf{b}^{n\times j}$ (SA1) and $\mathbf{b}^{m\times j}$ (SA2)

# Method

**Target**: predict Australian Labor Party two-party preferred vote at SA1 level using 2016 Census features:

- *labels*: publicly available 2PP result by SA1
- *features*: 81 socio-economic features (income, education etc) - select hierarchical Census features levels (SA1, SA2)
- generate one-hop nearest neighbour graph for each level, centroid to centroid
- construct embedding and projection operators
- train on common socio-demographic features (income, education etc)
- benchmark against generalised linear models (GLMS), multi-layer perceptron and standard GCNN
- trial multiple variants of HGCNN structure
- trial on 80/20, 20/80 and 10/90 train/test splits

# Hierarchical GCNN Variants

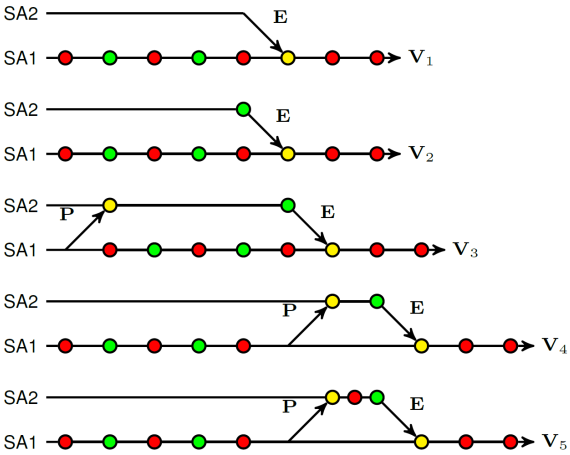


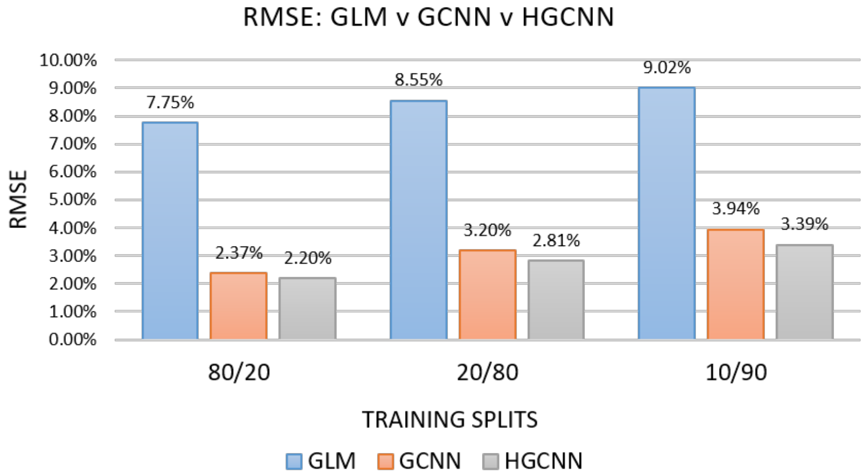

Figure 3: Hierarchical GCNN variants: green = graph convolution; red = graph embedding; yellow = composition of SA1 and SA2 features (via projection **P** or embedding **E**).

# Results: Table

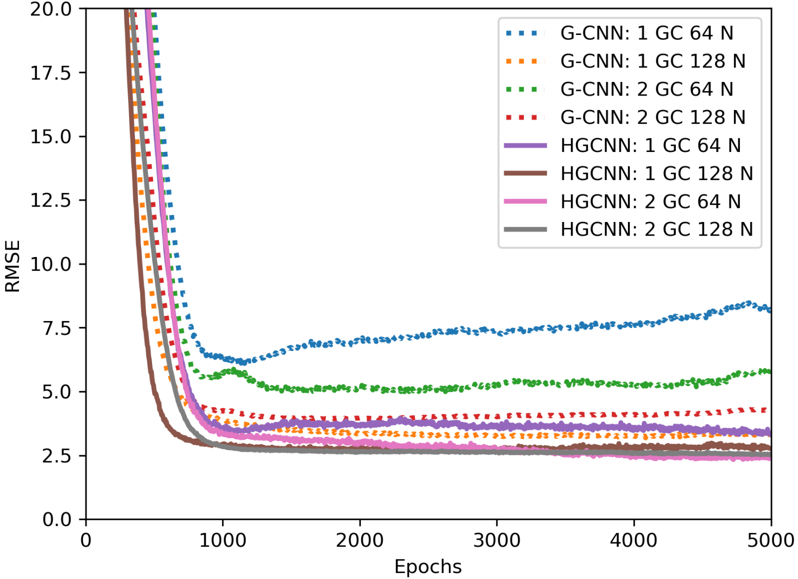
| Model | 80% Training Data | | 20% Training Data | | 10% Training Data | |
|---|---|---|---|---|---|---|
| | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ |
| Linear Regression | 9.35 *(0.07)* | 0.58 *(0.01)* | 9.55 *(0.12)* | 0.56 *(0.01)* | 10.07 *(0.82)* | 0.51 *(0.08)* |
| Ridge Regression | 9.34 *(0.07)* | 0.58 *(0.01)* | 9.53 *(0.11)* | 0.57 *(0.01)* | 9.86 *(0.48)* | 0.53 *(0.05)* |
| LASSO | 10.13 *(0.06)* | 0.51 *(0.01)* | 10.16 *(0.06)* | 0.51 *(0.01)* | 9.02 *(0.08)* | 0.61 *(0.01)* |
| LASSO Lars | 11.72 *(0.13)* | 0.34 *(0.02)* | 12.28 *(0.23)* | 0.28 *(0.03)* | 11.95 *(0.11)* | 0.32 *(0.01)* |
| Random Forest | 7.75 *(0.09)* | 0.71 *(0.01)* | 8.55 *(0.07)* | 0.65 *(0.01)* | 9.02 *(0.08)* | 0.61 *(0.01)* |
| MLP | 6.58 *(0.06)* | 0.66 *(0.01)* | 7.56 *(0.07)* | 0.55 *(0.01)* | 8.19 *(0.14)* | 0.44 *(0.01)* |
| Graph-CNN | 2.37 *(0.13)* | 0.97 *(0.01)* | 3.20 *(0.09)* | 0.94 *(0.01)* | 3.94 *(0.15)* | 0.91 *(0.01)* |
| Hierachical GCNN (V1) | 2.24 *(0.12)* | 0.97 *(0.01)* | 3.01 *(0.10)* | 0.95 *(0.01)* | 3.79 *(0.16)* | 0.92 *(0.01)* |
| Hierachical GCNN (V2) | **2.20** *(0.12)* | 0.97 *(0.01)* | **2.81** *(0.10)* | 0.96 *(0.01)* | 3.45 *(0.13)* | 0.94 *(0.01)* |
| Hierachical GCNN (V3) | 2.27 *(0.18)* | 0.97 *(0.01)* | 2.85 *(0.10)* | 0.95 *(0.01)* | 3.52 *(0.16)* | 0.92 *(0.01)* |
| Hierachical GCNN (V4) | 2.59 *(0.17)* | 0.96 *(0.01)* | 2.91 *(0.12)* | 0.95 *(0.01)* | 3.44 *(0.14)* | 0.94 *(0.01)* |
| Hierachical GCNN (V5) | 2.69 *(0.20)* | 0.96 *(0.01)* | 2.93 *(0.12)* | 0.95 *(0.01)* | **3.39** *(0.12)* | 0.94 *(0.01)* |

Figure: Comparison of GCNN variants with traditional benchmarks, using RMSE and $R^2$ as measures. The five variants of the Hierarchical GCNN sequentially increase in complexity (standard deviations close to zero were rounded up to 0.01).

# Results: Comparison



# Results: Loss v Epochs



# Results: HGCNN v GCNN Map

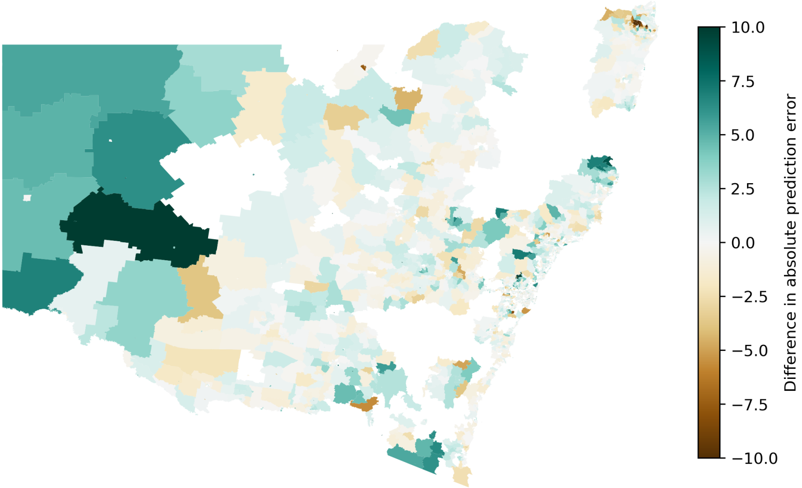


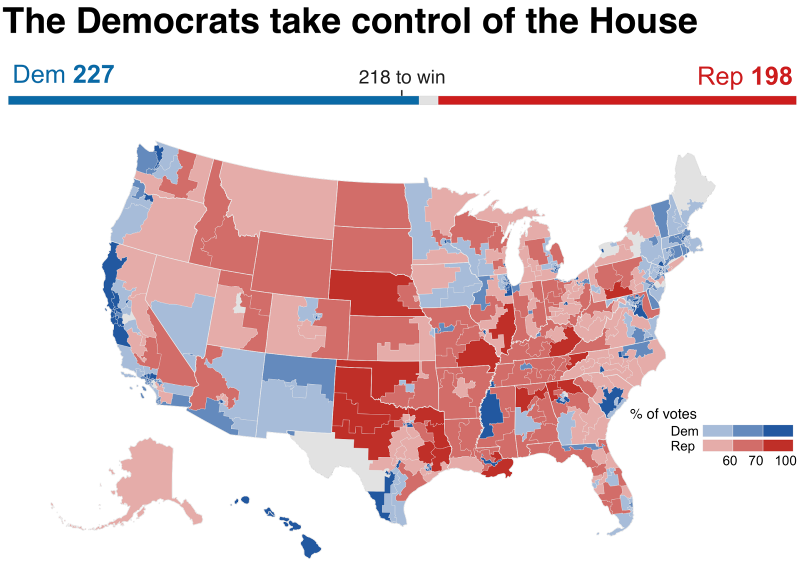

Figure: GCNN Prediction Error - Hierarchical GCNN Prediction Error

# Conclusion

**Conclusions**:

- Useful for applying CNN methods to multi-layered and sparse graph-structured heterogeneous datasets
- Enriches predictive power of GIS data and other multi-level hierarchical graph data
- Efficiently captures neighbourhood effects/interactions across graph and across hierarchy
- Wide application to social science/economics and to geospatial datasets
- Paper: https://www.aaai.org/Papers/AAAI/2019/AAAI-LiM.4653.pdf

# Future Work




Source: AP, 12/11/2018. Grey districts are undeclared

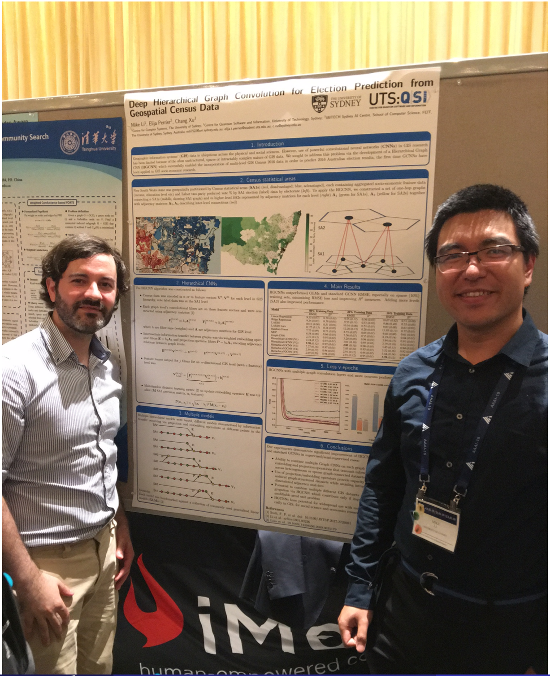
BBC

# Select References

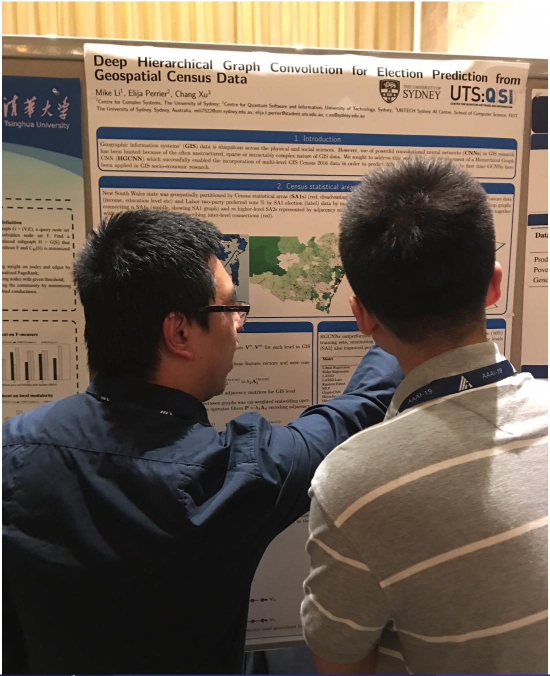

- Such, Felipe Petroski, et al. "Robust spatial filtering with graph convolutional neural networks." IEEE Journal of Selected Topics in Signal Processing 11.6 (2017): 884-896.
- Sandryhaila, Aliaksei, and Jos MF Moura. "Discrete signal processing on graphs." IEEE transactions on signal processing 61.7 (2013): 1644-1656.
- Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018. Adaptive Graph Convolutional Neural Networks. arXiv:1801.03226 [cs, stat]. arXiv: 1801.03226.

# Deep Hierarchical Graph Convolution for Election Prediction from Geospatial Census Data

**Mike Li**[1] **Elija Perrier**[2] **Chang Xu**[3]
[1] Centre for Complex Systems, The University of Sydney, Sydney, Australia
[2] School of Quantum Software and Information, The University of Technology, Sydney, Australia
[3] UBTECH Sydney AI Centre, School of Computer Science, FEIT, University of Sydney, Australia
mili7522@uni.sydney.edu.au, elija.t.perrier@student.uts.edu.au, c.xu@sydney.edu.au


Deep Hierarchical Graph Convolution for Election Prediction from Geospatial Census Data
THE UNIVERSITY OF SYDNEY
UTS:QSI
1. Introduction
2. Census statistical areas
2. Hierarchical CNNs
3. Multiple models
4. Main Results
5. Loss v epochs
6. Conclusions
Tsinghua University
mmunity Search
human-empowered


MAKE AMERICA
GREAT AGAIN
MINTS
Net Wt .4oz (12g)


Deep Hierarchical Graph Convolution for Election Prediction from Geospatial Census Data
Mike Li[1], Elija Perrier[2], Chang Xu[3]
THE UNIVERSITY OF SYDNEY
UTS:QSI
Tsinghua University
1. Introduction
2. Census statistical areas
AAAI-19


A Cambrian Explosion of Machine
Learning Research Topics
Generative modeling
Security
accountability

# Graph CNNs

**Graph GCNNs**:

- Method for applying CNNs to data in graph-structure but not traditionally grid-like. Two approaches: *spectral* v *spatial*
- *Spectral* filtering - spectral eigendecomposition of adjacency matrix;
- *Spatial filtering* - Filters for each graph level $\mathbf{F}$ are linear approximations to Chebyshev polynomials (Such et al., 2017; Sandryhaila, Aliaksei, and Jos MF Moura, 2013)

$$\mathbf{F} = \sum_{i=0}^{k} h_k \mathbf{A}^k$$

  (we set $k = 1$ for simplicity)
- Curse of dimensionality as graph sizes extent can restrict input of auxiliary graph info

# Adjacency learning

- Inclusion of generalised Mahalanobis metric[1] to update embedding operator weights

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \tag{6}$$

- $L_2$-norm with adjustable weights in which the distance $\mathcal{D}$ between two feature vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^c$ in $c$-dimensional features space
- $\mathbf{M}$ is positive semi-definite precision matrix of weights of dimension $c^2$
- Hadamard (elementwise) product such that the embedding filter becomes:

$$\mathcal{G} \circ \mathbf{E}^{n \times m} = h_3 \mathcal{G} \circ \mathbf{A}_3 \tag{7}$$

[1] Li, Ruoyu, et al. "Adaptive Graph Convolutional Neural Networks." arXiv preprint arXiv:1801.03226 (2018).

# Adjacency learning

| Variants | 80% | 20% | 10% |
|---|---|---|---|
| V1 | **2.22** | 3.02 | 3.81 |
| V2 | **2.15** | 2.81 | 3.43 |
| V3 | **2.20** | 2.86 | 3.52 |
| V4 | **2.50** | 2.92 | 3.45 |
| V5 | **2.64** | 2.95 | 3.39 |

Figure: RMSE with Adjacency learning. Indicative improvement but results within uncertainties.

# Additional hierarchical level

| **Variants** | **80%** | **20%** | **10%** |
|---|---|---|---|
| V2 with SA3 | 2.52 | 2.97 | 3.60 |
| V2 with SA3 (with GC) | 2.29 | **2.77** | **3.37** |
| V5 with SA3 | **2.66** | **2.90** | **3.28** |
| V5 with SA3 (with GC) | **2.66** | **2.90** | **3.35** |

Figure: RMSE with addition of SA3 level. Improved performance v less complex hierarchies in certain cases.