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] Centre for 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

Abstract

Geographic information systems' (**GIS**) research is widely used within the social and physical sciences and plays a crucial role in the development and implementation by governments of economic, education, environment and transportation policy. While machine learning methods have been applied to GIS datasets, the uptake of powerful deep learning CNN methodologies has been limited in part due to challenges posed by the complex and often poorly structured nature of the data. In this paper, we demonstrate the utility of GCNNs for GIS analysis via a multi-graph hierarchical spatial-filter GCNN network model in the context of GIS systems to predict election outcomes using socio-economic features drawn from the 2016 Australian Census. We report a marked improvement in performance accuracy of Hierarchical GCNNs over benchmark generalised linear models and standard GCNNs, especially in semi-supervised tasks. These results indicate the widespread potential for GIS-GCNN research methods to enrich socio-economic GIS analysis, aiding the social sciences and policy development.

Introduction

Geographical information systems (**GIS**) are frameworks for the storage, ordering and representation of geospatial data. They are integral to many social and physical sciences and to the functioning of modern economies and governments, playing an important role in infrastructure, agriculture, transport, logistics, urban management and economic management policy design and implementation (Management Association 2013). Complementary GIS data, such as Census and other demographic surveys in particular, constitute important datasets upon which researchers and policymakers rely. Despite their ubiquity and utility, the size and complexity of many GIS datasets can present tractability challenges that limit the capacity of traditional models to fully leverage the relationships between geospatial features. Many current techniques ignore the geography inherent in GIS data (including spatial boundaries, clustering effects and distance measures), opting to treat the separate regions as independent and identically distributed. In the context of deep learning, the heterogeneous nature of the intrinsic graphs constructible from GIS data render traditional neural-network and convolutional neural network (**CNN**) approaches problematic. Recent developments in spectral and spatial filtering graph CNN (**GCNN**) methodologies (Such et al. 2017) together with novel adaptive graph representation learning, provide a potential means of overcoming these limitations by opening up heterogeneous data-structures to CNN-driven analysis.

In this paper, we build upon state-of-the art GCNN approaches to develop a hierarchical spatial-filtering GCNN (**Hierarchical GCNN**) model (equipped with an adaptive distance learning metric) that significantly enhances the predictive power of socio-economic Census data (from the Australian Bureau of Statistics (**ABS**)) in the prediction of Australian election results (from the Australian Electoral Commission (**AEC**)). By connecting the neighbouring statistical regions into a planar graph and performing graph convolutions along these edges, our experiments demonstrate a root mean-squared error (**RMSE**) measures of 2.20% compared with optimal standard generalised linear model (**GLM**) benchmark RMSEs of 7.75%, a decrease of nearly 70%. We additionally demonstrate that defining auxiliary networks and embedding operations based upon a Hierarchical GCNN model to feed in residual information from higher levels of statistical agglomeration outperforms standard GCNN approaches by up to an additional 14% in semi-supervised tasks. Code for the models is provided at https://github.com/mili7522/Hierarchical-GCNN.

Related Work

Machine Learning with GIS

GIS data is analysed in a rich variety of ways, incorporating tools from topology, graph theory and linear algebra. It is characterised by process modelling (Heywood, Cornelius, and Carver 2011) which seeks to model complex behaviour of spatial systems (Jakeman, Beck, and McAleer 1995) via *a priori* models (that seek explanatory theories) and *a posteriori* models to test and explore the domain of theories (Heywood, Cornelius, and Carver 2011). GIS approaches utilise diverse mathematical models depending upon the research questions (usually classification or regression-based) of interest: GLMs, such as linear or logistic regression, are common where the input features are known. This is espe-

cially the case in quantitative social sciences assessing geographic drivers of inequality and the social impact of policy in which survey data (e.g. Census or HILDA datasets) has a geographical component (Baum, Bill, and Mitchell 2008; Baker et al. 2016). Where input features are uncertain, stochastic methods are also commonly used (Heywood, Cornelius, and Carver 2011). In most cases, the research objective is usually the probabilistic prediction of a geospatially-situated outcome of interest.
Machine learning and neural networks have been used with GIS since the 1990s (Hewitson and Crane 2012; Openshaw and Openshaw 1997). More recently, deep learning based methods have been used for flood-route modelling (Peters, Schmitz, and Cullmann 2006), the analysis of soil erosion and mineral deposit identification (Noack et al. 2012), traffic flow prediction problems (Polson and Sokolov 2017) and even forest-fire modelling in concert with gradient boosting (Sachdeva, Bhatia, and Verma 2018).

The application of powerful CNN methods to GIS data is, however, underdeveloped. This is in part because while spatial ordering of GIS data provides a degree of intrinsic structure for GIS data, graph relations between geospatial features can often be heterogeneous or otherwise highly complex. Graphs where vertex number and edge connection distribution differ across a geospatial manifold cannot be easily adapted for use with conventional CNNs (Defferrard, Bresson, and Vandergheynst 2016) because they lack the grid-like translational structure needed for traditional CNN, metrics and dyadic clustering (Bruna et al. 2014).

For these reasons, the application of CNN methods to GIS has involved the imposition of homogeneous (grid) structure on features (e.g. satellite image classification analysis (Albert, Kaur, and Gonzalez 2017)) or remote sensing (Nogueira, Penatti, and Santos 2017). Developing means of applying CNN methods to GIS therefore has potentially wide application in geospatial sciences in general and socio-economic research in particular.

Graph-CNNs and Spatial Filtering

A potential means of overcoming barriers to the application of CNN methods to unstructured GIS datasets are GCNNs, which utilise graph relations of underlying features to impose structure on datasets necessary for the application of CNN filtering (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016; Seo et al. 2018; Kipf and Welling 2016; Henaff, Bruna, and LeCun 2015).

Early GCNNs relied predominantly on spectral filtering in which learning occurs in the frequency (or Fourier) domain via harmonic analysis of adjacency matrices. Even though these methods could only be applied to fixed graph structures (since the eigendecomposition of the graph Laplacian is unique for each graph structure (Monti, Otness, and Bronstein 2018)), such methods have been able to be used to adapt the features of compositionality in order to solve higher-dimensional learning problems in a non-euclidean setting (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016). An example of a classification task applied to a fixed network is the integration of fMRI data over a functional brain network to predict disease. In contrast, the graph structure represented in a GIS context just depends on some arbitrary parcellation of the underlying area.

One means of handling the heterogeneity of GIS datasets is provided by alternative spatial filtering of GCNNs in which filters are framed as polynomial functions of the usual adjacency matrix (Bruna et al. 2014; Sandryhaila and Moura 2013; Such et al. 2017) and learning occurs in the spatial domain. Spatial filtering has the additional advantage of overcoming the non-localised filter problem of some spectral GCNN algorithms, since it acts per node by construction. These localised interactions allow well-defined connections between graphs in a multi-graph context and this extension enables spatial GCNNs to tackle the hierarchical nature of many GIS datasets.

Graph Data Generation and Analysis

GIS Data Analysis Our experiments tested the efficacy of GCNNs and Hierarchical GCNNs (explained below) for modelling geographically-specific electoral results from the 2016 Australian federal election. Our features were selected from the 2016 Australian Census, a half-decadal national descriptive statistical survey of the Australia's population held on a specific night at the dwelling they are located at the time (ABS 2016).

Census data is considered among the highest quality socio-economic data available. It is provided to the public via anonymised aggregated features whereby populations are grouped into geospatial areas according to the Australian Statistical Geography Standard - essentially tiling the nation into polygonal areas. The smallest of such areas for which the suite of Census socioeconomic data is available is the Statistical Area Level 1 (**SA1**) level, followed by a hierarchy of successively larger regional aggregations (SA2, SA3 and so on). The SA1s are constructed as polygons using GIS ESRI shape files (provided by both the ABS and the AEC) which are two-dimensional maps geocoded according to a coordinate reference system. SA1 population counts vary between around 200 to 800 persons for most SA1s, with an average size of 400 persons (ABS 2016). GIS software was used to determine which SA1s neighboured each other from which a one-hop SA1 graph was constructed (see Figure (1)). This process was repeated for the SA2 level (and SA3 level in one experiment). Links between SA1 and SA2-nodes were constructed using an ABS correspondence file connecting the levels.

GIS datasets have been used previously within political science literature (Schram 1992) to assess spatial patterns of voter behaviour. In particular, (Stimson and Shyy 2013) developed spatial typologies of voter support combining 2007 Australian federal election with the then-most recent 2006 Census. The model developed in this paper builds upon and expands the Census-derived SA1 and SA2 features used in the literature (Stimson and Shyy 2013; Liao, Shyy, and Stimson 2009) such as: age and sex, family and household structure, housing tenure, ethnicity/ancestry, residential mobility, digital accessibility, labour force data, industry and occupation, education, religious affiliation and income. The benchmark models used in (Stimson and Shyy 2013) are primarily GLMs including linear regression, least

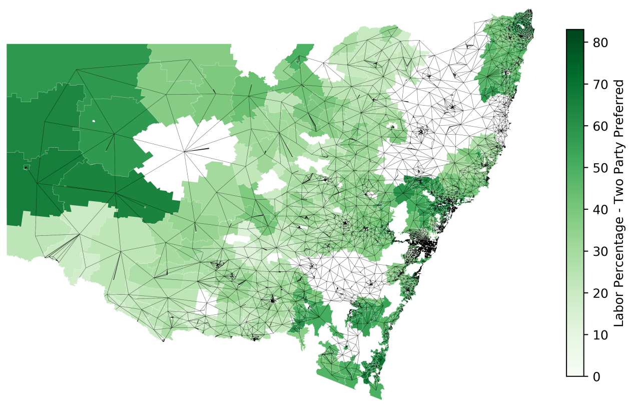

Figure 1: Network of neighbouring regions constructed by finding polygon neighbours using GIS software. The edges connect the centroids of neighbouring regions. Colour indicates the two party preferred values used as the training signal. Some areas of the network span regions with missing data from the AEC source.

squares/robust line-fitting, MANOVA clustering and discriminant analysis. The experiments in this paper tested the efficacy of similar GLMs against standard and Hierarchical GCNN architectures.

Election data was sourced from the AEC, which provides publicly available datapacks containing results from the 2016 election (AEC 2016). This includes electorate, SA1 and polling booth data available for download from their website. Our models were trained on the two-party preferred (**2PP**) vote which gives results give the results of ballots after preferences have been distributed, usually between the Australian Labor Party and the Liberal/National Coalition. Adapting (Stimson and Shyy 2013), the 2PP results by booth were mapped onto the AEC's publicly provided dataset detailing the number of voters from each SA1 at each respective polling booth for the 2016 election, providing an estimate of 2PP at the SA1 level.

Hierarchical GCNN Much GIS data, especially Census data, falls within a natural hierarchical structure given by successively larger regional agglomerations. Hierarchical socio-economic GIS data can be decomposed into distinct heterogeneous graphs where lower-scale individual GIS feature maps and graphs may differ from high-level regional or catchment-area data. A primary motivation of exploring the efficacy of GCNN models was the desire to capture the hierarchical nature of heterogeneous GIS networks. However, simply combining multi-level GIS graphs into one single large graph can become computationally intractable or lead to overfitting during model training and filter learning due to expanded parametrisation that comes from consolidation of adjacency matrices into an expanded adjacency relation. Therefore the capacity to run parallel GCNNs across networked but distinct graphs while controlling when to transmit information from one graph to another is important for GCNN GIS analysis. In this Hierarchical GCNN scenario, the training and prediction is still made at the lowest level (SA1) of aggregation, but the higher level hierarchies provide auxiliary information that is integrated using a residual connection near the end of the network.

Our starting point is to consider a two-level GIS hierarchy comprising lower-level statistical areas (SA1s) (around 300 households each) and higher-level regional (suburban) statistical areas (SA2s) built from SA1 aggregations. Each level has its own graph generated via the GIS centroidal coordinates, with adjacency restricted to nearest-neighbours. For an SA1 graph of n nodes and SA2 graph of m nodes ($m < n$) we have an SA1 adjacency matrix $\mathbf{A}_1 \in \mathbb{R}^{n\times n}$ and an SA2 adjacency matrix $\mathbf{A}_2 \in \mathbb{R}^{m\times m}$. Matrices $\mathbf{A}_3 \in \mathbb{R}^{n\times m}$ and $\mathbf{A}_4 \in \mathbb{R}^{m\times n}$ are linear operators that facilitate information-sharing between the different-dimensional feature spaces of each level. The embedding

$$\mathbf{A}_3^{(n\times m)} : \mathbb{R}^{m\times 1} \to \mathbb{R}^{n\times 1}, \qquad \mathbf{V}^{(m\times 1)} \mapsto \mathbf{V}^{(n\times 1)} \tag{1}$$

transfers information from the SA2 level graph to the SA1 level graph by acting on SA2 vertex vectors $\mathbf{V}^{(m\times 1)}$ (one for each SA1 feature), while the projection

$$\mathbf{A}_4^{(m\times n)} : \mathbb{R}^{n\times 1} \to \mathbb{R}^{m\times 1}, \qquad \mathbf{V}^{(n\times 1)} \mapsto \mathbf{V}^{(m\times 1)} \tag{2}$$

transfers information from the SA1 to SA2 level by acting on SA1 vertex vectors $\mathbf{V}^{(n\times 1)}$ (one for each SA2 feature). Here superscript terms in brackets indicate the dimension of the operator or vector. Filter-learning is conducted on both levels separately using a combination of graph convolution and graph pool embedding (Such et al. 2017) layers, with intermittent projections and embeddings of features across the graph levels before combining to output an n-dimensional SA1-level output estimate for our predictor, 2PP vote by SA1.

Spatial domain convolutional filters for the SA1 and SA2 graphs are given by linear approximations to Chebyshev polynomials (Such et al. 2017; Sandryhaila and Moura 2013) $\mathbf{F} = \sum_{i=0}^{k} h_k \mathbf{A}^k$. Here the filter $\mathbf{F}$ is a kth-degree polynomial of the graph adjacency matrix $\mathbf{A}^k$ where the order k of each polynomial term $\mathbf{A}^k$ encodes the number (k) of hops from a given vertex multiplied by the given filter tap (we set $k = 1$ for simplicity). Vertices $\mathbf{V}$ are convolved with $\mathbf{F}$ via matrix multiplication $\mathbf{V}_{\text{out}} = \mathbf{F}\mathbf{V}_{\text{in}} \in \mathbb{R}^n$. Each hierarchical graph level's convolution can then be approximated as:

$$\mathbf{F}_1^{(n\times n)} \approx h_1 \mathbf{A}_1^{(n\times n)} \qquad \mathbf{F}_2^{(m\times m)} \approx h_2 \mathbf{A}_2^{(m\times m)} \tag{3}$$

where there are several convolutional layers for each graph level, depending on the depth of the network. 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 $\mathbf{P} = h_4\mathbf{A}_4$:

$$\mathbf{E}^{(n\times m)}\mathbf{V}^{(m\times 1)} \to \mathbf{V}^{(n\times 1)} \tag{4}$$

$$\mathbf{P}^{(m\times n)}\mathbf{V}^{(n\times 1)} \to \mathbf{V}^{(m\times 1)} \tag{5}$$

This approach is generalisable for c multiple vertex features, the effect being a stack of $n\times n$ filter matrices for each of the

c vertex filter slices (one for each feature) $\mathbf{F}_1^{(c)}$ (the parentheses index the specific feature) for the SA1 level and d lots of $m \times m$ vertex filter slices $\mathbf{F}_1^{(d)}$ for the SA2 level (in our case with only single adjacency matrix for each graph rather than multiple matrices for each edge feature as per (Such et al. 2017)). The choice of features for the SA2 level in experiments varied between a feature set identical to that of the SA1 nodes and smaller feature sets for different runs. There may be multiple (j) such filters bound up in a tensor form: $\mathbf{F} \in \mathbb{R}^{n \times n \times c \times j}$. The action of these operators on an input feature vector $V_{\text{in}}^{(n \times c)}$ leads to in an output vector $V_{\text{out}} \in \mathbb{R}^{n \times j}$ (and analogously for the SA2 case). The feature tensor output for j filters for both the SA1 (with c features) and SA2 (with d features) levels is then:

$$\mathbf{V}_{1,\text{out}}^{(n \times j)} = \underbrace{\left(\mathbf{F}_1^{(n \times n \times c \times j)} \mathbf{V}_{\text{in}}^{(n \times c)}\right)}_{n \times j} + \mathbf{b}_1^{(n \times j)} \tag{6}$$

$$\mathbf{V}_{2,\text{out}}^{(m \times j)} = \underbrace{\left(\mathbf{F}_2^{(m \times m \times d \times j)} \mathbf{V}_{\text{in}}^{(m \times d)}\right)}_{m \times j} + \mathbf{b}_1^{(m \times j)} \tag{7}$$

with biases $\mathbf{b}^{n \times j}$ (SA1) and $\mathbf{b}^{m \times j}$ (SA2). The final output prediction vector is at the SA1 level and is given by $\mathbf{V}_{1,\text{out}}^{(n \times 1)}$, obtained by further contractions.

Hierarchical GCNN Variants To explore the effects of different layer-configurations and information-sharing across the SA1 and SA2 graphs, five variants (indicated by V1,...,V5) of Hierarchical GCNNs (shown in Figure (3)) were tested during the experiments. These are characterised by different arrangements of operations (either graph convolution layers shown by green nodes or zero-hop convolutions shown by red nodes) and sequences of projection and embedding operations (yellow nodes). All networks were initialised with two parallel graph channels. V1 architecture only included an embedding $\mathbf{E}^{(n \times m)}$ (indicated by a yellow node) from SA2 to SA1 feature space. This connection is a residual sum on each vertex vector, given by:

$$\mathbf{V}_{SA1}^{(n \times 1)} = \mathbf{E}^{(n \times m)} \mathbf{V}_{SA2}^{(m \times 1)} \mathbf{W} + \mathbf{V}_{SA1}^{(n \times 1)} \in \mathbb{R}^n \tag{8}$$

The best performance is found when no batch normalisation and no activation function is performed after the addition, which allows the smoothing of the direct path (He et al. 2016). The residual nature of the link means that information from the higher hierarchical layers are only incorporated if useful for the task, which is performed solely on the SA1 nodes.

V2 architecture was identical to V1 except for an additional graph convolution layer (indicated by a green node) at the SA2 level. V3 architecture contained both a projection from the SA1 level to SA2 level, an additional graph convolution layer with the first node with feature vector and a projection $\mathbf{P}^{(m \times n)}$ from SA1 to SA2 feature space:

$$\mathbf{V}_{SA2}^{(m \times 1)} = \mathbf{P}^{(m \times n)} \mathbf{V}_{SA1}^{(n \times 1)} \mathbf{W} + \mathbf{V}_{SA2}^{(m \times 1)} \in \mathbb{R}^m \tag{9}$$

Variants V4 and V5 in Figure (3) incorporated additional projections and embeddings at later stages in the sequence.

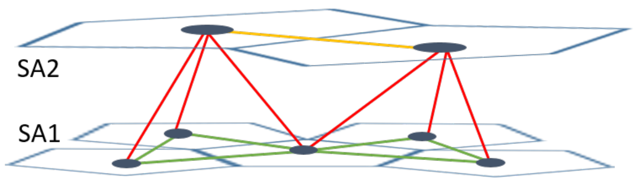

Figure 2: Schema of SA1 and SA2 level graphs with links between them.

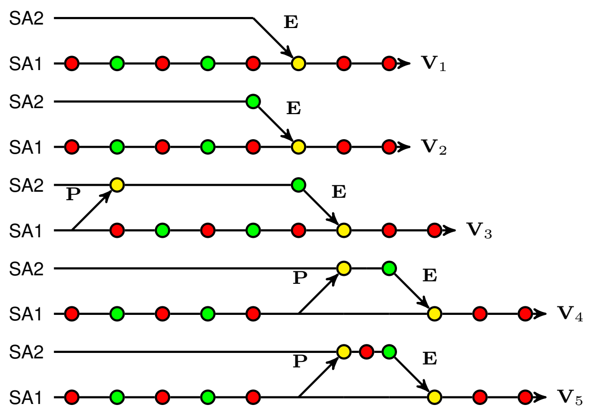

Figure 3: Hierarchical GCNN variants: green = graph convolution; red = graph embedding; yellow = composition of SA1 and SA2 features (via projection **P** or embedding **E**).

Adaptive learning Despite being useful starting points, intrinsic GIS graphs may be sub-optimal for learning tasks. Because unsupervised learning of full optimal graph structure from unstructured data is often intractable or complex (Hamilton, Ying, and Leskovec 2017), a 'half-way' point between fully unsupervised graph learning and fixed adjacency is provided by semi-supervised adaptive graph learning in which input adjacency matrices are updated via a distance learning metric drawn from features related to the intrinsic GIS graph. Such methods have improved classification tasks and have been used in spectral convolutional settings.

In experiments below, we experimented with using a generic distance learning measure known within statistics as the generalised Mahalanobis metric (Li et al. 2018; Wang and Sun 2015; Perez, Ribeiro, and Perez 2016) novelly applying it in a spatial-filtering context. We used an 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 is given as:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \tag{10}$$

where $\mathbf{M}$ is positive semi-definite precision matrix of weights of dimension c^2. Here index i runs over the SA1 dimensions (n) and index j runs over the SA2 dimensions (m). Only the embedding operator was subject to the distance learning adaptation (the same set of feature categories enabled the difference vector between the SA1 feature set and SA2 equivalent feature to be is calculated), being updated according to a Gaussian $\mathcal{G}(\mathbf{x}_i, \mathbf{x}_j) = \exp(\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)/2\sigma^2)$ (assuming normalcy with $\sigma = 1$) using the Hadamard (elementwise) product such that the embedding filter becomes: $\mathcal{G} \circ \mathbf{E}^{n \times m} = h_3 \mathcal{G} \circ \mathbf{A}_3$.

Experiments and Results

Methods Our experiments were designed to test the performance of the Hierarchical GCNN's predictions of Labor 2PP by SA1 using Census features by comparison with a standard GCNN, multi-layer perceptron network (**MLP**) and the GLMs. Each experiment included both a standard supervised training problem with five-fold cross validation (80% training, 20% test) and two semi-supervised tasks with (i) 20% training, 80% test and (ii) 10% training, 90% test. The full graph structure remains available despite the small size of the training set. The semi-supervised task is valuable as it is often desirable to extract information in the case where more training data is unavailable, such as observing overall patterns from a limited number of survey results.

Centroid-to-centroid one-hop graphs for the SA1s and SA2s were constructed by finding polygon neighbours in open source QGIS software (Figure (2)). The resulting network for the whole state is shown in Figure (1). The GLMs chosen were: linear regression, ridge regression with cross-validation (**CV**), LASSO, LASSO Lars CV and Random Forest algorithms from Sci-Kit Learn. We also benchmarked against a densely-connected neural MLP (in which each SA1 as treated independently and not connected to any other) and the standard non-hierarchical GCNN (acting only on the SA1 graph). As the output is continuous, our primary metrics for comparison were RMSE and R^2 for both the neural networks and GLMs.

A standard setting of two graph convolutional layers with 128 neurons for each except the last layer was used throughout the tests, although Fig 4 explores the performance under different settings for these hyperparameters. The ReLU activation function and the ADAM optimiser were used. Table (1) sets-out RMSE and R^2 values (and standard deviations) for each training/test split. One challenge when using GIS data challenge is the 'modifiable areal unit problem', namely whether zonal/aggregation or scale selection systematically biases aggregate results reliant upon statistics drawn from such aggregations (e.g. higher level geospatial aggregations). The Hierarchical GCNN model mitigates its potential impact by down-weighting the contribution of higher-level aggregated features if they do not contribute to model optimisation such that at worst the performance of the model will not decrease.

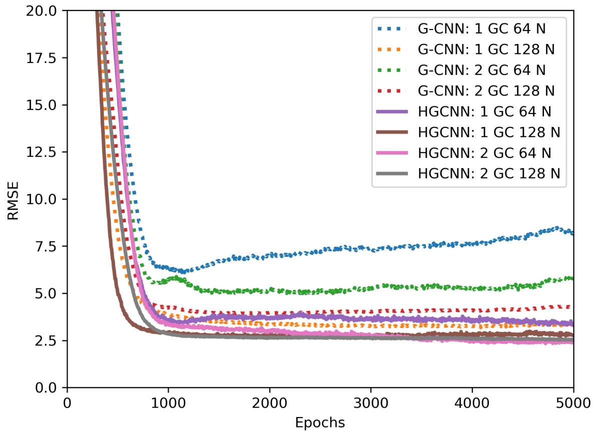

Figure 4: RMSE of test set of standard GCNN vs Hierarchical GCNN (V2) for one or two graph convolution (GC) layers and 64 or 128 neurons per layer

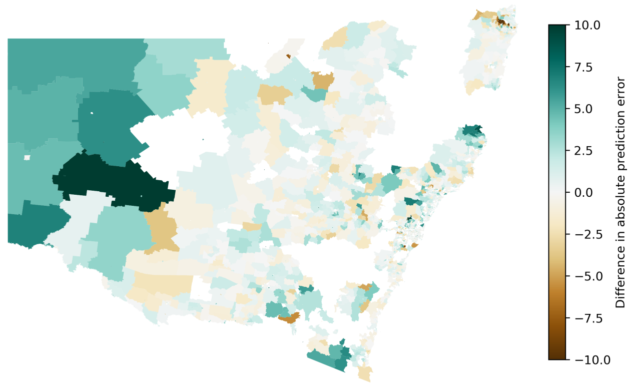

Figure 5: Absolute prediction error of Hierarchical GCNN subtracted from the absolute prediction error of the standard GCNN (difference clipped between 10 and -10). The geographic spread shows that the Hierarchical GCNN clearly performs better (fewer brown areas in compared to blue), and seem to work especially well in regional areas.

Discussion

GLMs and MLP As evident in Table (1), the benchmark GLMs performed significantly worse across all training/test partitions of the datasets by comparison with the standard GCNN and the Hierarchical GCNNs. The lower bounds RMSE for standard linear regression and ridge regression with CV were around 9.35% with a maximum R^2 of 0.58. In the 10/90% semi-supervised case, R^2 and RMSE deteriorated. LASSO models performed worst, with RMSEs of just above 10% though interestingly they performed marginally better on the 10/90% training/test split. The best performing GLM was the Random Forest with a RMSE around 7.75% and R^2 of 0.71. The basic MLP model outperformed each of the GLMs, achieving lower-bound RMSE of 6.58% and R^2 of 0.66.

Hierarchical GCNN The main results of the experiments evident in Table (1) are: (i) the significant increase in performance of GCNNs over GLMs and MLP and (ii) a further improvement in performance over standard GCNNs via the adoption of a Hierarchical GCNN model for the semi-supervised tasks. The standard GCNN's performance (RMSE of 2.37% and R^2 of 0.97) for the 80/20% train/test split represents a decrease in RMSE of over 70% by comparison with the best performing Random Forest benchmark

Table 1: 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).

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*)
Hierarchical 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*)
Hierarchical 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*)
Hierarchical 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*)
Hierarchical 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*)
Hierarchical 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*)

and of over 75% by comparison with benchmark linear regression RMSE - representing a nearly three-fold improvement in RMSE and nearly two-fold improvement in R^2 over the best GLM. For the semi-supervised cases, the standard GCNN also significantly outperformed all GLMs on both RMSE and R^2.

Table (1) also demonstrates the improvement to the standard GCNN architecture achieved via the novel introduction of a hierarchical approach to GCNNs. The Hierarchical GCNN improved RMSE over the standard GCNN over both the supervised and semi-supervised tasks. The simple variant V2 was able to decrease the RMSE to 2.20% with 80% of the data in the training set and reduce RMSE to 2.81% with 20% of the data in the training set. For both semi-supervised tasks, all variants outperformed the standard CNN on both RMSE and R^2 measures.

Figure (4) compares the performance over training epochs of the GCNN and Hierarchical GCNN for different neuron/layer number parametrisations. We can see that even with 80% of the data in the training set, the standard GCNN has a tendency to overfit with 64 neurons per layer. The test performance decreases as the number of epochs continues from 1000 to 5000. By comparison, we see that Hierarchical GCNN achieves good performance with fewer epochs, is more stable and also has less performance variation between different number of layers and number of neurons.

By comparing the different variants of the Hierarchical GCNN, it can be seen that the specific structure of the Hierarchical GCNN impacts the performance depending on the nature of the task. Table (1) shows that a simple projection operation with one graph convolution at the SA2 level works well in most cases and lead to the best performance when training on 80% of the data. Variations of additional complexity tend to increase training difficulty or increase overfitting in these circumstances. However for the smallest training/test partition $10/90\%$, increasing the number of operations at the SA2 level, including utilising an additional embedding operator led to the best performance. Variants V4 and V5 perform their embedding from SA1 to SA2 near the end of the network instead of the start, which allows the embedded features to have undergone transformation and convolutions within the SA1 branch.

In this case, the best performing Hierarchical GCNN V5 registered an RMSE of 3.39% and R^2 of 0.94, an decline in RMSE of 0.55% and improvement in R^2 of 0.03 over the standard GCNN. Having one graph convolution at the SA2 level seemed to always be beneficial compared to directly performing the projection operator filters. This suggests that the Hierarchical GCNN increases the receptive field of the SA1 nodes, with the residual link allowing only the useful information to be incorporated.

In all test/training partitions, the GCNNs outperformed even the best-performing GLM model. This is especially notable when comparing the $10/90\%$ train/test split, in which both the standard GCNN (with RMSE of 3.94% and R^2 of 0.91) and Hierarchical GCNN (V5) (RMSE of 3.39% and R^2 of 0.94) trained on 10% of the data outperformed GLMs that were trained on 80% of the data. Hierarchical GCNNs thus can outperform standard GLMs and a standard GCNN approach. Given the hierarchical and often sparse nature of socio-economic GIS training sets, the ability to utilise hierarchical graph structure to improve overall performance is potentially quite useful for researchers.

To understand the reason for the better performance of the Hierarchical GCNN, the maps in Fig. (5) visualise the difference in RMSE in the standard GCNN vs the Hierarchical GCNN (averaged over the 10 repetitions of the semi-supervised task with 10% of the data in the training set). The Hierarchical GCNN outperforms the standard GCNN in many of the regional areas where the graph structure differs substantially from the more geometrically regular city regions.

This suggests that when data is sparse, the standard GCNN network may not be able to learn a filter that applies equally to all neighbour types. However, the relationship between the different levels of the geographical hierarchy is more consistent and so it is able to provide residual information where the links from the neighbours themselves are insufficient. The ability of the Hierarchical GCNN to capture such 'neighbourhood effects' is therefore an im-

Table 2: RMSE of experiments using distance learning to adapt the weight of the projection edges. Bold indicates improved performance compared to the same variant in Table (1). Utility of these results was uncertain as standard deviations (suppressed) were in the order of 0.1%.

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

portant feature of its performance, providing an additional means by which information can be transmitted throughout the graph network between nodes not connected within their own graph level.

Table 3: RMSE of experiments including SA3 regions as an additional third hierarchy in the network. Bold indicates improved performance compared to the same variant in Table (1). Rows labelled '(with GC)' include a graph convolution at the SA3 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**

The distance learning results for the Hierarchical GCNN variants were mixed and are presented in Table (2). All variants with distance learning achieved lower five-fold RMSEs than the equivalent variants without distance learning for the supervised case. However, the variance in these results was similar in magnitude to any benefits gained, so the overall utility is uncertain. The addition of distance learning for the semi-supervised cases seemed to have little effect or slightly lower performance due to increased risk of overfitting.

Table (3) tests a further extension of the hierarchical application of our network, adding a third level (of SA3s) on top of the SA1s and SA2s. Doing so can produce further improvements, especially in the semi-supervised case as it further widens the receptive field of possible graph convolutions. Most models variants saw an improvement with the addition of the SA3 layer, including V5 on the 10%/90% dataset with an RMSE of 3.28%, making it the best performing of the sparsely-trained models. This again suggests the beneficial impact of incorporating auxiliary hierarchical graphs in the case of limited training data or sparse data sets.

Although it wasn't explored here, a key benefit of including networks at different aggregations is that it allows the use of data sources which are aggregated at different statistical levels (e.g. data sources from the tax office reported at the SA3 level). The projection and embedding operations described are general enough to allow the use of distinct node features at each hierarchical level.

Conclusion and future work

The results of the experiments detailed in this paper demonstrate the utility of applying Hierarchical GCNN methods to GIS social-sciences problems, such as estimation of aggregate characteristics of populations within geographic localities, and the improved performance of Hierarchical GCNNs over standard GCNNs, particularly in cases of minimal training data. As demonstrated via their high RMSE and comparatively low R^2 values, standard GLM methods and MLP networks were not especially predictive of 2PP by SA1 on such GIS Census data.

The inclusion of both standard and Hierarchical GCNNs clearly has an effect, both in the fully supervised context (in which the majority of the data is used in the training set) and even moreso in semi-supervised contexts, precisely due to their capacity to leverage geographical connections. For Hierarchical GCNNs, the capacity to leverage multi-level geographic connections via higher-level graph convolutions while transmitting information between different graph levels appears to have played a role in its superior performance.

In this paper, we have demonstrated the utility of GCNN methods for GIS in a social science context and the improved performance over standard GCNNs of novel Hierarchical GCNN methods described above. Hierarchical GCNNs, which incorporate projections and embeddings between different multi-level graphs, allow incorporation of multi-level socio-economic GIS data in a way that enriches the predictive insights from Census and other GIS datasets due in part to the Hierarchical GCNN's capacity for incorporation of neighbourhood features.

The significant performance improvement of Hierarchical GCNNs over GLMs offers considerable potential for enhancing modelling of voter behaviour along with the design and implementation of policy decisions across government industry, which often rely upon standard econometric techniques such as generalised equilibrium analysis to ascertain the impact of policy interventions on metrics, such as social welfare or GDP.

References

ABS. 2016. Census. *Australian Bureau of Statistics.*

AEC. 2016. 2016 federal election results. *Australian Electoral Commission.*

Albert, A.; Kaur, J.; and Gonzalez, M. C. 2017. Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Halifax, CANADA, Aug 13-17, 2017*, 1357–1366.

Baker, E.; Bentley, R.; Lester, L.; and Beer, A. 2016. Housing affordability and residential mobility as drivers of locational inequality. *Applied Geography* 72:65–75.

Baum, S.; Bill, A.; and Mitchell, W. 2008. Unemployment in Non-Metropolitan Australia: integrating geography, social and individual contexts. *Australian Geographer* 39(2):193–210.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *arXiv:1606.09375 [cs, stat]*.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Representation Learning on Graphs: Methods and Applications. *arXiv:1709.05584 [cs]*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity Mappings in Deep Residual Networks. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 630–645. Cham: Springer International Publishing.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep Convolutional Networks on Graph-Structured Data. *arXiv:1506.05163 [cs]*.

Hewitson, B., and Crane, R. 2012. *Neural Nets: Applications in Geography*. GeoJournal Library. Springer Netherlands.

Heywood, I.; Cornelius, S.; and Carver, S. 2011. *An Introduction to Geographical Information Systems*. Pearson Education Limited.

Jakeman, A.; Beck, M.; and McAleer, M. 1995. *Modelling Change in Environmental Systems*. Wiley.

Kipf, T. N., and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*.

Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018. Adaptive Graph Convolutional Neural Networks. *arXiv:1801.03226 [cs, stat]*.

Liao, E.; Shyy, T.-K.; and Stimson, R. J. 2009. Developing a webbased eresearch facility for sociospatial analysis to investigate relationships between voting patterns and local population characteristics. *Journal of Spatial Science* 54(2):63–88.

Management Association, I. R., ed. 2013. *Geographic Information Systems: Concepts, Methodologies, Tools, and Applications*. IGI Global.

Monti, F.; Otness, K.; and Bronstein, M. M. 2018. Motifnet: a motif-based graph convolutional network for directed graphs. *arXiv:1802.01572 [cs]*.

Noack, S.; Barth, A.; Irkhin, A.; Bennewitz, E.; and Schmidt, F. 2012. Spatial Modeling of Natural Phenomena and Events with Artificial Neural Networks and GIS. *International Journal of Applied Geospatial Research (IJAGR)* 3(1):1–20.

Nogueira, K.; Penatti, O. A. B.; and Santos, J. A. d. 2017. Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. *Pattern Recognition* 61:539–556.

Openshaw, S., and Openshaw, C. 1997. *Artificial intelligence in geography*. Wiley.

Perez, J. L. R.; Ribeiro, B.; and Perez, C. M. 2016. Mahalanobis distance metric learning algorithm for instance-based data stream classification. In *2016 International Joint Conference on Neural Networks (IJCNN)*, 1857–1862.

Peters, R.; Schmitz, G.; and Cullmann, J. 2006. Flood routing modelling with Artificial Neural Networks. *Adv. Geosci.* 9:131–136.

Polson, N. G., and Sokolov, V. O. 2017. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies* 79:117.

Sachdeva, S.; Bhatia, T.; and Verma, A. K. 2018. GIS-based evolutionary optimized Gradient Boosted Decision Trees for forest fire susceptibility mapping. *Natural Hazards* 92(3):1399–1418.

Sandryhaila, A., and Moura, J. M. F. 2013. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing* 61(7):16441656.

Schram, A. 1992. Testing economic theories of voter behaviour using micro-data. *Applied Economics* 24(4):419–428.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2018. Structured sequence modeling with graph convolutional recurrent networks. In Cheng, L.; Leung, A. C. S.; and Ozawa, S., eds., *Neural Information Processing*, 362–373. Cham: Springer International Publishing.

Stimson, R., and Shyy, T.-K. 2013. And now for something different: modelling socio-political landscapes. *The Annals of Regional Science* 50(2):623–643.

Such, F. P.; Sah, S.; Dominguez, M. A.; Pillai, S.; Zhang, C.; Michael, A.; Cahill, N. D.; and Ptucha, R. 2017. Robust Spatial Filtering With Graph Convolutional Neural Networks. *IEEE Journal of Selected Topics in Signal Processing* 11(6):884–896.

Wang, F., and Sun, J. 2015. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery* 29(2):534–564.