Multi-GCN: Graph Convolutional Networks for Multi-View Networks, with Applications to Global Poverty

Muhammad Raza Khan, Joshua E. Blumenstock
University of California, Berkeley
mraza@bekeley.edu, jblumenstock@berkeley.edu

Abstract

With the rapid expansion of mobile phone networks in developing countries, large-scale graph machine learning has gained sudden relevance in the study of global poverty. Recent applications range from humanitarian response and poverty estimation to urban planning and epidemic containment. Yet the vast majority of computational tools and algorithms used in these applications do not account for the multi-view nature of social networks: people are related in myriad ways, but most graph learning models treat relations as binary. In this paper, we develop a graph-based convolutional network for learning on multi-view networks. We show that this method outperforms state-of-the-art semi-supervised learning algorithms on three different prediction tasks using mobile phone datasets from three different developing countries. We also show that, while designed specifically for use in poverty research, the algorithm also outperforms existing benchmarks on a broader set of learning tasks on multi-view networks, including node labelling in citation networks.

1 Introduction

Over the past several years, large-scale graph machine learning has gained increasing relevance in the domain of international poverty research (Blumenstock 2016). Driven largely by the expansion of mobile phone networks throughout developing countries – roughly 95% of the world population now has mobile phone coverage (GSMA 2016) – vast quantities of network data are constantly being generated by people living in even extremely poor and marginalized communities. Recent work has shown how such data can be used to inform critical policy decisions, including the measurement of living conditions (Blumenstock, Cadamuro, and On 2015), the spread of infectious diseases (Wesolowski et al. 2015), and the management of humanitarian crises (Lu, Bengtsson, and Holme 2012). Private companies are also taking advantage of this new source of data, for instance by using data from mobile phones to generate credit scores that can expand credit to millions of people historically shut out of the formal banking ecosystem (Francis, Blumenstock, and Robinson 2017).

However, a critical constraint to the use of these data in settings related to economic development is the lack of scalable algorithms for performing prediction tasks on sparse multi-view networks. Multi-view networks (also referred to as multiplex and multi-modal networks), are networks in which nodes can be related in multiple ways, and are the natural abstraction for mobile phone networks, where different individuals have different types of relationships and can interact using different modalities (such as phone calls, text messages, money transfers, and app-based activity). Yet, the vast majority of applied research using mobile phone data — in developing and developed countries alike — ignores the multi-view nature of phone networks.

This paper develops a novel approach for learning on multi-view networks, which bridges two different strands in the research literature. The first strand involves methods for efficient analysis of multi-view networks; the second explores algorithms for semi-supervised graph learning (see Related Work, below). The method we develop provides an efficient approach for applying convolutional neural networks to multi-view graph-structured data. We benchmark this new method, which we call Multi-GCN (short for Multi-View Graph Convolutional Networks), on three different mobile network datasets, on three different prediction tasks relevant to the international development community: (1) predicting the adoption of a new "financial inclusion" technology in a West African country; (2) predicting whether an individual is living below the poverty line in an East African country; (3) predicting the gender of mobile phone subscribers in a South Asian country. In all cases, we find that Multi-GCN outperforms state-of-the-art benchmarks, including standard Graph Convolutional Networks (Kipf and Welling 2017), Node2Vec (Grover and Leskovec 2016), Deepwalk (Perozzi, Al-Rfou, and Skiena 2014), and LINE (Tang et al. 2015).

While designed specifically with the developing-country context in mind (where the sparsity and multi-view properties of networks are very salient), we show that Multi-GCN can be more generally applied to a wide range of problems involving multi-view networks. Indeed, most real-world networks are multi-view, including the network data most frequently used by AI researchers (e.g., data from Twitter, Amazon, Netflix, etc.). Our second set of results shows that Multi-GCN can improve upon state-of-the-art algorithms not just in poverty-related contexts, but also in traditional classification problems. In particular, we show that

Multi-GCN outperforms competing algorithms on citation labeling tasks (using benchmark datasets from Citeseer and Cora) that have been studied extensively in prior work.

2 Related Work

2.1 Technical Related Work

Our goal is to develop an efficient method for node-level transductive semi-supervised learning over multi-view graphs. Here, we begin with a general overview of semi-supervised learning, then focus on various approaches to graph-based semi-supervised learning, and finally discuss related work on multi-view networks.

Graph-Based Semi-Supervised Learning One of the biggest issue with applying supervised learning algorithms in a developing country is that it is often costly to collect labels for training. For instance, when using mobile phone data to predict the wealth of subscribers, Blumenstock, Cadamuro, and On (2015) manually conducted a survey of roughly 1,000 subscribers. Semi-supervised learning tries to solve this problem by using unlabeled data along with the labeled data to train better classifiers (see (Zhu 2005) for a survey). Our focus is on transductive semi-supervised learning, which assumes that all the unlabeled data is available at the training time and does not attempt to generalize to data unseen during training.

Graph-based semi-supervised learning (GSSL) is a popular approach for semi-supervised learning that treats labeled and unlabeled instances as graph vertices, and relationships between instances as edges (Liu, Wang, and Chang 2012). GSSL algorithms try to learn a classifier that is consistent with the labeled data while making sure that the prediction for similar nodes is also similar. This is achieved by minimizing a loss function with two factors: a) supervised loss over the labeled instances, and b) a graph-based regularization term. Different GSSL algorithms use different functions for graph regularization. Label propagation-based approaches, for instance, use a constrained label lookup function (e.g., Zhou et al. (2004a)). Related, kernel-based approaches parameterize regularization term in the Reproducing Kernel Hilbert Space (RKHS).

Learning Over Graphs The success of word embedding algorithms like Word2Vec (Mikolov et al. 2013) has inspired similar algorithms for graphs. For instance, DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) learns embeddings by predicting the neighborhood of nodes based on random walks over the graphs, while LINE (Tang et al. 2015) and Node2vec (Grover and Leskovec 2016) allow for advanced sampling schemes. More recently, neural network-based approaches have been proposed to perform learning over graphs. These have been extended to the task of semi-supervised learning (Bruna et al. 2013; Defferrard, Bresson, and Vandergheynst 2016), including recent work by Kipf and Welling (2017) that proposes a Graph Convolutional Network (GCN), which we take as a starting point for our approach.

Learning Over Multi-View Graphs The key distinction between our approach and prior work is our desire to handle graphs with multiple views, i.e., graphs where vertices can be connected in more than one way. In recent years, many different algorithms have been proposed for learning on multi-view graphs. These algorithms can be broadly divided into three main categories: 1) co-training algorithms, 2) learning with multiple kernels, and 3) subspace learning (See Xu, Tao, and Xu (2013) for a survey). Recent work by Dong et al. (2014) show that subspace approaches — which find a latent subspace shared by multiple views — perform well relative to co-training and kernelized approaches on a range of tasks. We therefore focus our attention on integrating subspace learning approaches with recent innovations in graph convolutional networks.

Comparison with existing work Our main contribution is to propose an efficient method for adapting GSSL to multi-view contexts. Existing approaches to GSSL cannot be readily implemented on such data; those algorithms that do handle multiple views generally treat views and vertices equally. We show that current "state of the art" methods like Graph Convolutional Networks (Kipf and Welling 2017) can be enhanced by augmenting the input graph using subspace analysis over Grassman manifolds. Farseev et al. (2017) have demonstrated that subspace merging approach can be quite accurate for the problem of cross-domain recommendation which is different from our experimental settings and context as described in the section 4.

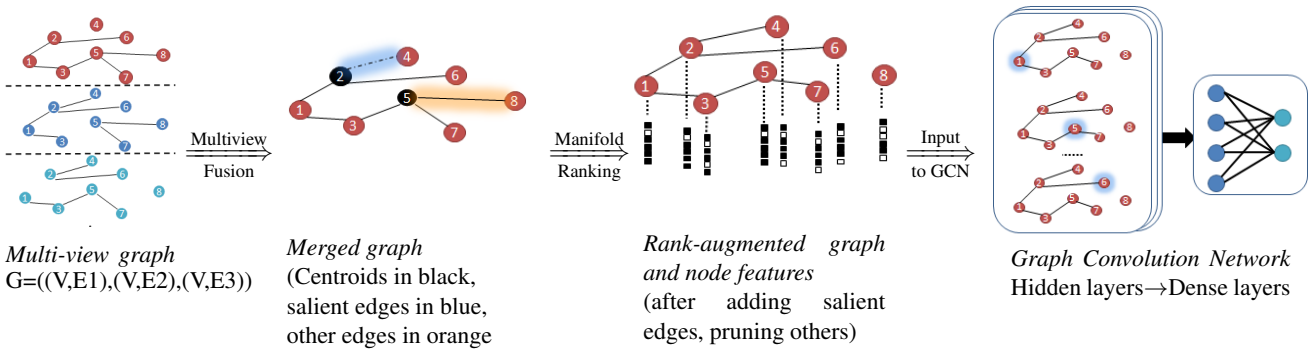
2.2 Empirical Related Work

Our experimental results focus on three prediction tasks of relevance to the international development community:

Predicting poverty. A large number of humanitarian applications — from poverty targeting to program monitoring — require accurate estimates of the welfare for beneficiary populations. Recently, several papers have shown how digital trace data can be used to estimate the socioeconomic status of individuals, households, and villages. For instance, Jean et al. (2016) show that daytime satellite imagery can be used to estimate village wealth; Quercia et al. (2012) find that Twitter data can be used to estimate levels of deprivation, and Blumenstock (2015) shows that mobile phone metadata can be used to estimate the welfare of individuals and regions.

Product adoption. We focus on the adoption of "mobile money", a suite of phone-based financial services that are designed to promote financial inclusion among those traditionally shut out of the formal banking ecosystem (Suri 2017). Within this literature, our work relates most closely to Khan and Blumenstock (2016), who analyze the predictors of mobile money adoption in three different developing countries.

Gender prediction. Gender equality and women's empowerment are one of the Sustainable Development Goals, and recent work explores how digital trace data can be used to assess progress toward this goal (Fatehkia, Kashyap, and Weber 2018). Mislove et al. (2011) and Frias-Martinez, Frias-Martinez, and Oliver (2010) show that gender can be predicted from social media and mobile phone data.



Multi-view graph
G=((V,E1),(V,E2),(V,E3))

Merged graph
(Centroids in black, salient edges in blue, other edges in orange

Rank-augmented graph and node features
(after adding salient edges, pruning others)

Graph Convolution Network
Hidden layers→Dense layers

Figure 1: Overview of the Multi-view Graph Convolutional Network (Multi-GCN)

Broadly, these prior studies demonstrate a proof of concept: that digital trace data can be used to predict the characteristics and outcomes of individuals. However, such analysis rely on off-the-shelf algorithms that rarely, if ever, account for the multi-view nature of real-world social networks. This paper shows that a simple approach to multi-view learning can yield substantial improvements on these real-world prediction tasks.

3 *Multi-GCN*: Multi-View Graph Convolutional Networks

Our approach to semi-supervised learning on multi-view graphs integrates three steps, depicted in Figure 1. First, we use methods from subspace analysis to efficiently merge multiple views of the same graph. Second, we use a manifold ranking procedure to identify the most informative sub-components of the graph and to prune the graph upon which learning is performed. Finally, we apply a convolutional neural network, adapted to graph-structured data, to allow for semi-supervised node classification.

3.1 Merging Subspace Representations

Given an undirected multilayer graph with M layers $G = G_{i}{}_{i=1}^{M}$ such that each layer G_i has the same vertex set V but same or different edges set E_i, we first calculate the graph Laplacian for each of the individual layers. If D_i and W_i represent the degree matrix and the adjacency matrix for the i^{th} view of the graph, then the normalized graph Laplacian is defined as

$$L_i = D_i^{-1/2}(D_i - W_i)D_i^{-1/2} \tag{1}$$

Given the graph Laplacian L_i for each layer of the graph, we calculate the spectral embedding matrix U_i through trace minimization:

$$\min_{U_i \in \mathbb{R}^{n*k}} tr(U_i' L_i U_i), \quad \text{s.t. } U_i' U_i = 1 \tag{2}$$

This trace minimization problem can be solved by the Rayleigh-Ritz theorem. The solution U_i contains the first k eigenvectors corresponding to the k smallest eigenvalues of L_i. The spectral embedding embeds nodes of the original graph to a low dimensional spectral domain (See Von Luxburg (2007) for details).

A Grassman manifold $\mathcal{G}(k, n)$ can be considered as a set of k-dimensional linear subspaces in $\mathbb{R}^n$ where each unique subspace is mapped to a unique point on the manifold. Each point on the manifold can be represented by an orthonormal matrix $Y \in \mathbb{R}^{n*k}$ whose columns span the corresponding k-dimensional subspace in $\mathbb{R}^{n*k}$ and the distance between the subspaces can be calculated as a set of principal angles $\{\theta_i\}_{i=1}^{k}$ between these subspaces. Dong et al. (2014) show that the projection distance between two subspaces Y_1 and Y_2 can be represented as a separate trace minimization problem:

$$d^2_{proj}(Y_1, Y_2) = \sum_{i=1}^{k} \sin^2 \theta_i = k - tr(Y_1 Y_1' Y_2 Y_2') \tag{3}$$

where, based on Eq. 3, the projection distance between the target representative subspace U and the individual subspaces $U_{i}{}_{i=1}^{M}$ can be calculated as:

$$\begin{aligned} d^2_{proj}(U, \{U_i\}_{i=1}^{M}) = & \sum_{i=1}^{M} d^2_{proj}(U, U_i) \\ = \; & kM - \sum_{i=1}^{M} tr(UU' U_i U_i') \end{aligned} \tag{4}$$

Minimization of Eq. 4 ensures that individual subspaces are close to the final representative subspace U.

Finally, to ensure that the original vertex connectivity in each graph layer is preserved, we include a separate term that minimizes the quadratic-form Laplacian (evaluated on the columns of U):

$$\min_{U \in \mathbb{R}^{n*k}} \sum_{i=1}^{M} tr(U' L_i U) + \alpha_i [kM - tr(UU' U_i U_i')], \quad \text{s.t. } U_i' U = 1 \tag{5}$$

In Eq 5, α is the regularization parameter that balances the trade-off between the two terms in the objective function. Rearranging Eq. 5 and ignoring the constant terms yields

$$\min_{U \in \mathbb{R}^{n*k}} tr[U'(\sum_{i=1}^{M} L_i - \sum_{i=1}^{M} \alpha_i U_i U_i')U], \tag{6}$$

As before, the Rayleigh-Ritz theorem can be used to solve Eq 5. The solution is given by the fist k eigenvectors of the modified Laplacian:

$$L_{mod} = \sum_{i=1}^{M} L_i - \sum_{i=1}^{M} \alpha_i U_i U_i' \tag{7}$$

3.2 Graph-Based Manifold Ranking

Though the modified Laplacian calculated above can be fed directly to the downstream graph convolutional networks, model performance can be increased by ranking the nodes in the manifold based on their saliency with respect to some critical nodes (Zhou et al. 2004b). To rank points on the manifold, we use the closed form function,

$$f^* = (I - \beta * L_{mod})^{-1} q \tag{8}$$

Here, I represents the identity matrix, L_{mod} is the normalized Laplacian as calculated in Eq. 7, and β is the regularization parameter. Given a vector q containing the indices of the query nodes, Eq. 8 calculates the saliency of the other nodes with respect to the query nodes; the saliency of these nodes can then be used to add or prune edges from the induced underlying graph. The use of manifold-based ranking suits our approach as the modified Laplacian representing merged subspaces can be used directly for saliency detection. The query nodes can be selected as the centroids determined by any clustering algorithm over the manifold.

The algorithm for the subspace merging and subsequent manifold ranking is shown in Algorithm 1. The time complexity of Algorithm 1 for a graph with M layers with N users per layer is $O(MN^3 + MN^2K + N^2C^2 + tN)$ where K represents the number of eigenvectors to be calculated and C is the number of centroids $O(MN^3)$ is the cost of computing Laplacians and Eigenvector matrix for all the M layers ; $O(MN^2K)$ is the cost of computing modified Laplacian; $O(N^2C^2)$ is the cost of computing C clusters using k-means clustering; $O(tN)$ is the cost of manifold ranking. using the iterative version described by (Zhou et al. 2004b).

3.3 Graph Convolution Networks

The application of convolutional neural networks to irregular or non-Euclidean grids, such as graphs, is based on the fact that convolutions are multiplications in the Fourier domain, which implies that graph convolutions can be expressed as the multiplication of a signal $x \in \mathbb{R}^N$ with a filter $g(\theta)$ (see Bruna et al. (2013)):

$$g_\theta * x = g_\theta(L)x = U g_\theta U^T x \tag{9}$$

Here, U represents the eigen-decomposition of the normalized graph Laplacian $L = I - D^{-1/2}AD^{-1/2}$ and I, D, A represent the identity, degree and the adjacency matrix, respectively. Graph convolutions can be further expressed in terms of Chebyshev polynomials as

$$g_{\theta'} * x = \sum_{k=0}^{K} \theta_k' T_k(\tilde{L})x \tag{10}$$

Algorithm 1: Fusion of multiple views of a graph

Input: $\{A_i\}_{i=1}^M$: $n \times n$ adjacency matrices of individual graph layers $\{G_i\}_{i=1}^M$, with G_1 being the most informative layer
Input: $\{\alpha_i\}_{i=1}^M$,regularization parameters per subspace to be merged
Input: K, salient query points
Input: Y, number of salient edges per centroid to add
Input: Z, number of non-salient edges per centroid to prune
Input: β, manifold ranking regularizer
Output: L_{mod}:Merged Laplacian,A_{mod}: Merged Adjacency matrix, E_s:Salient Edges, E_{ns}: Non salient edges

Step 1: Compute normalized Laplacian matrix L_i for each layer of the graph
Step 2: Compute subspace representation U_i for each layer of the graph
Step 3: Compute the modified Laplacian matrix $L_{mod} = \sum_{i=1}^{M} L_i - \sum_{i=1}^{M} \alpha_i U_i U_i'$
Step 4: Perform clustering on the modified Laplacian to identify K salient points i.e. centroids $\{q_i\}_{i=1}^K$
Step 5: For each of the centroid rank other edges on the manifold $f^* = (\mathcal{I} - \beta * L_{mod})^{-1} q$
Step 6: For each centroid q_i add Y salient edges to the E_s and Z non-salient edges to the E_{ns}
Step 7: Add E_s to A_1 to form A_{mod}
Step 8: Remove E_{ns} from A_{mod}

where $\tilde{L}$ is the rescaled Laplacian, T_k represents the Chebyshev polynomials, and θ' represents the vector of Chebyshev coefficients. Following Kipf and Welling (2017), by approximating the maximum value of the largest eigenvalue and constraining the number of free parameters, the convolution operation can be represented as

$$g_\theta * x = \theta(I + \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2})x \tag{11}$$

where $\tilde{A} = A + I$ and $\tilde{D} = \sum \tilde{A}$ are the renormalized versions of A and D. This renormalization avoids numerical instabilities resulting from exploding/vanishing gradients (Defferrard, Bresson, and Vandergheynst 2016).

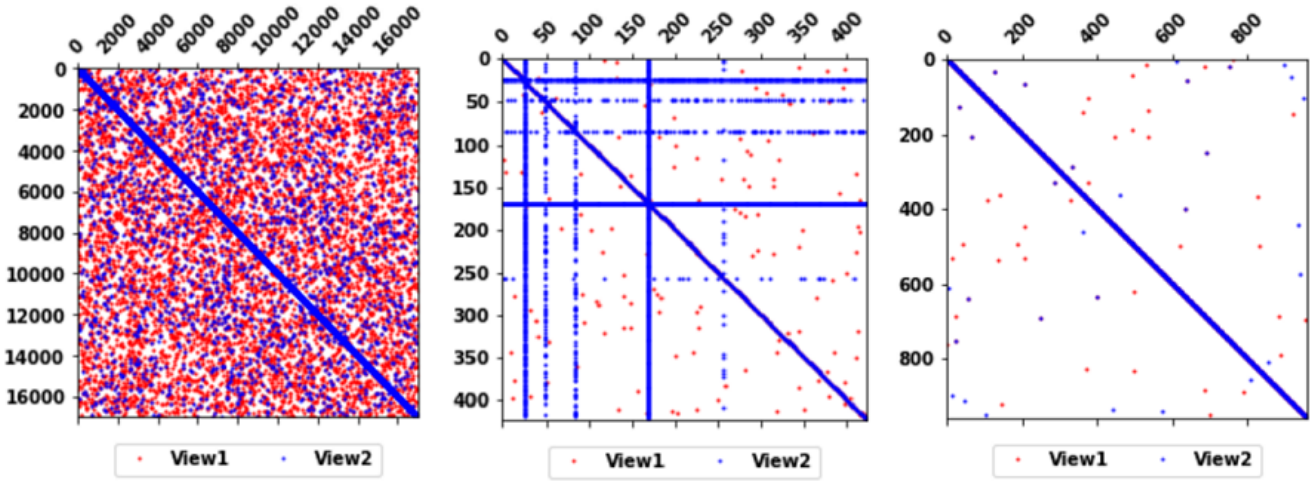
The modified graph (A_{mod} in Algorithm 1) resulting from the merger of Laplacians using the subspace analysis and manifold ranking can be fed directly into the graph convolution networks defined above. The forward propagation model for a two layer network can then be represented as

$$Z = F(X, A) = softmax(\hat{A}\, ReLU(\hat{A}XW^0)W^1) \tag{12}$$

Here, $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is calculated as a preprocessing step before giving the input to the neural network. W^0 and W^1 represent the input-to-hidden-layer and hidden-layer-to-output weight matrices for a two layer neural network, and can be trained using gradient descent. ReLU and Softmax represent the activation functions in the hidden and output layers.

Dataset	Data Type	Nodes	Edges (view 1)	Edges (view 2)	Classes	Features	Label Rate
Product Adoption	Phone logs (West Africa)	17,000	23,032	18,371	2	132	0.002
Poverty Prediction	Phone logs (East Africa)	422	544	1,799	2	1,709	0.094
Gender Prediction	Phone logs (South Asia)	958	992	978	2	821	0.042
Citeseer	Citation network	3,327	4,732	3,492	6	3,703	0.036
Cora	Citation network	2,708	5,429	2,846	7	1,433	0.052

Table 1: Summary statistics. The Label Rate indicates the fraction of instances that are labeled.



(a) Product Adoption (b) Wealth Prediction (c) Gender Prediction

Figure 2: Mobile phone spy plots. Dots indicate that two individuals have communicated by voice (red) or SMS (blue).

4 Experiments and Data

4.1 Datasets

Our first set of experiments test Multi-GCN on three prediction tasks relevant to international development. Each one uses a different dataset of mobile phone Call Detail Records (CDR), obtained from three different developing countries with GDP per capita less than $1,600 USD. These datasets contain detailed metadata on all communication events (calls, messages) that occur on the mobile phone network. Each CDR dataset contains multiple possible relationships between nodes (views); we extract one view corresponding to phone calls between users, and another corresponding to text messages. We separately construct a large set of features of each user (such as total call volume and degree centrality), using the combinatoric approach described in Khan and Blumenstock (2016).

Table 1 presents summary statistics for each of these datasets. The connections and sparsity of each network are shown in Figure 2. These spy plots help visualize the structure of the adjacency matrices for each graph view, where a dot indicates that an edge exists between those two individuals on the corresponding view.

Product adoption dataset
The first dataset that we use is a sample of a dataset of mobile phone activity from a West African country. Here, the classification of interest is whether or not the user eventually adopts a new financial inclusion product. There are two possible classifications: (1) Did not adopt; (2) Adopted and used the product. Following the experimental setup described in Kipf and Welling (2017), we randomly selected 20 users from each category (40 total) for the training dataset; the validation and the testing dataset consist of 500 and 1000 randomly selected users, respectively.

Poverty prediction dataset
The wealth prediction dataset consists of several thousand transactions of different mobile phone users from an East African country. We attempt to classify users as poor or non-poor, where labels were obtained by Blumenstock, Cadamuro, and On (2015) through a small set of phone surveys that were conducted with mobile phone subscribers. Again, we randomly selected 20 users from each category as the training dataset, while the size of the validation dataset and the testing dataset is 100 and 200 respectively.

Gender prediction dataset
The gender prediction dataset originates from a developing country in South Asia. Here, the classification task is to predict the gender of the mobile phone users, where gender labels are provided by the operator for a small number of la-

Method	Product Adoption	Poverty Prediction	Gender Prediction
DeepWalk (first view)	56.43±0.187	51.91±0.62	53.18± 0.55
DeepWalk (second view)	51.97±0.112	50.34±0.36	50.84±0.64
DeepWalk (view union)	56.81± 0.114	50.87±0.95	52.34±0.50
Node2vec (first view)	53.87±0.20	52.26±0.58	50.12± 0.40
Node2vec (second view)	50.50±0.11	49.70±0.23	51.68±0.40
Node2vec (view union)	54.50±0.11	50.52±0.63	51.64±0.53
LINE (first view)	51.11±0.01	50.15±0.02	51.56± 0.001
LINE (second view)	50.83±0.01	52.29±0.001	50.00±0.001
LINE (view union)	56.26±0.003	50.18±0.001	51.33±0.002
GCN (first view)	70.74±2.2	55.19±2.33	63.97± 1.29
GCN (second view)	71.40±1.81	50.06±0.81	63.01±0.013
GCN (view union)	71.90±0.9	50.22±0.56	63.90±1.32
Multi-GCN (this paper)	**73.47±0.91**	**59.23±0.20**	**66.34± 1.03**

Table 2: Classification accuracy on mobile phone data. Numbers indicate mean classification accuracy (percentage) and standard error over 10 randomly selected dataset splits of equal size.

beled instances. We randomly select 20 users from each category for training; the size of the validation and the testing datasets are 100 and 800, respectively.

Citation classification datasets
A final set of experiments replicates the experimental design of Kipf and Welling (2017) to test Multi-GCN on more standard node labelling tasks. In these datasets, nodes are documents and the first view corresponds to the citation links between the research papers. We construct the second view from the textual similarity of the papers. Specifically, if the normalized cosine similarity between documents is greater than 0.8, then we create an edge in the second view of the citation network.

4.2 Experimental setup

In general, our goal is to correctly classify nodes in a network, where only a very small fraction of nodes are labeled. In the experiments, we start from a small sample of labeled nodes and test the ability of Multi-GCN, as well as several state-of-the-art algorithms, to correctly classify unlabeled nodes in the validation and testing sets. We use three popular node embedding algorithms (Node2vec, Deepwalk, and LINE) as a first set of baselines. In addition, we provide three baselines based on graph convolutional networks (Kipf and Welling 2017). The first two, *GCN (first view)* and *GCN (second view)*, apply GCN over the two respective adjacency matrices from phone and text message activity. The third, *GCN (view union)*, operates on the union of the adjacency matrices of the first view and the second view. In each GCN baseline, the node features are constructed from the adjacency matrix of the first view.

After merging different views, we rank the interaction between nodes using Eq. 8 based on their salience with respect to the query points. The value of the regularization parameter α (see Eq. 7) is selected through 10-fold cross-validation. We similarly tune the hyper-parameters β to 0.99 and set the number of query points to ten times the number of classes.

After adding salient edges and eliminating non-salient edges through the ranking process, both the adjacency matrix of the modified graph and the node features are passed as input to a two-layer graph convolutional network as described in Section 3. All of the GCN-based models, including Multi-GCN, are trained for a maximum of 200 iterations, using *Adam* (Adaptive moment estimation extension to stochastic gradient descent – see Kingma and Ba (2014)) and a learning rate of 0.01. Other GCN hyperparameters are set using the same values reported in Kipf and Welling (2017).

5 Results

Experimental results for the three developing-country datasets are shown in Table 2. Each row in this table indicates the average and standard error of the classification accuracy over 10 randomly drawn train-test splits of the same size for each dataset, constructed as described in Section 4. The last row in Table 2 shows the performance of Multi-GCN. In all four datasets, Multi-GCN outperforms existing state-of-the-art benchmarks, with the margin of improvement greatest in the poverty prediction task and smallest in the gender prediction task.

The second set of experimental results, comparing Multi-GCN to recent benchmarks on a more standard node classification task, are shown in Table 3. In addition to performing a comparison over randomly drawn train-test splits, we also compare the performance of Multi-GCN against a different set of randomized test-train splits, as used in the original tests by Kipf and Welling (2017), with an additional validation set of 500 instances used for hyper-parameter tuning. In all cases, we observe improvements in predictive accuracy of Multi-GCN relative to existing approaches.

6 Discussion

This paper proposes a new approach to semi-supervised learning on multi-view graphs. Through a series of exper-

Predefined train-test splits		
Method	**Citeseer**	**Cora**
ManiReg (first view) - Yang, Cohen, and Salakhutdinov (2016)	60.1	59.5
DeepWalk (first view) - Perozzi, Al-Rfou, and Skiena (2014)	43.2	67.2
Planetoid (first view) - Yang, Cohen, and Salakhutdinov (2016)	64.7	75.7
GCN (first view)	70.3	81.5
GCN (second view)	50.7	53.6
GCN (view union)	70.7	80.4
Multi-GCN (this paper)	**71.3**	**82.5**
Randomized train-test splits		
GCN (first view)	67.9± 0.5	80.1±0.5
GCN (second view)	53.6±0.1	56.9±0.3
GCN (view union)	67.9±0.3	78.5±0.1
Multi-GCN (this paper)	**70.5± 0.2**	**81.1±0.2**

Table 3: Classification accuracy on citation networks. Top panel shows the mean classification accuracy (percentage) for the pre-defined test-train splits as described by Yang, Cohen, and Salakhutdinov (2016). Bottom panel shows the classification accuracy (percentage) and standard error over 10 randomly selected dataset splits of equal size.

iments, we show that this approach improves upon state-of-the-art embedding- and convolution-based algorithms on a variety of prediction tasks related to both poverty research and to node labelling in general.

Relative to single-view learning algorithms, the main value of the multi-GCN approach is that it incorporates non-redundant information from multiple views into the learning process. Thus, the gains from multi-GCN depend on the prediction task, and the importance of multi-view graph structure to that task. Intuitively, this depends on the mutual information between. This intuition is also supported by a closer look at the results in Table 2. Here, we observe that while Multi-GCN provides the biggest gains relative to Deepwalk, Node2vec and LINE in the case of product adoption, the gains relative to single-view GCN are more modest. By contrast, the performance gain on the poverty and gender prediction tasks is significantly higher for Multi-GCN, even relative to the other single-view GCN benchmarks. The spy plots in Figures 2a-2c help explain this pattern. In particular, we can see that different views in the product adoption setting appear somewhat redundant, whereas for poverty and gender prediction the views appear more independent.

We believe future work should explore several limitations of the current analysis. In particular, there is much to be learned from a more systematic exploration of the value of additional views, and for different methods for merging views (beyond the subspace learning approach developed in Section 3.1). We are also exploring how graphs with varying degrees of sparsity and a different fraction of labeled nodes can impact the performance of Multi-GCN relative to alternative approaches.

7 Conclusion

Graph convolutional networks have recently achieved considerable success in a variety of learning tasks on irregular, graph-structured data. Leveraging insights from spectral graph theory, GCN's are beginning to replicate the success that CNN's have seen on more regular image and text data. For a wide variety of learning tasks relevant to graph-structured data — in contexts ranging from advertising in online networks to intervening in the spread of a contagious disease — this is a promising development.

In this paper, we have shown that state-of-the-art GCNs can achieve even greater performance on a variety of classification tasks when the multi-view nature of the underlying network is incorporated into the learning process. While motivated by three applications in global poverty research, the performance gains appear to generalize to other graph-based classification problems. We therefore view Multi-GCN as an important first step in adapting neural network-based approaches to multi-view networks and hope that it provides a foundation for future work in this space.

8 Acknowledgements

This research was supported by the National Science Foundation Grant under award #CCF - 1637360 (Algorithms in the Field) and by the Office of Naval Research (Minerva Initiative) under award N00014-17-1-2313.

References

Blumenstock, J.; Cadamuro, G.; and On, R. 2015. Predicting poverty and wealth from mobile phone metadata. *Science* 350(6264):1073–1076.

Blumenstock, J. E. 2014. Calling for Better Measurement: Estimating an Individual's Wealth and Well-Being from Mobile Phone Transaction Records. In *The 20th ACM Conference on Knowledge Discovery and Mining (KDD '14), Workshop on Data Science for Social Good.*

Blumenstock, J. E. 2016. Fighting poverty with data. *Science* 353(6301):753–754.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 3844–3852.

Dong, X.; Frossard, P.; Vandergheynst, P.; and Nefedov, N. 2014. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on signal processing* 62(4):905–918.

Farseev, A.; Samborskii, I.; Filchenkov, A.; and Chua, T.-S. 2017. Cross-domain recommendation via clustering on multi-layer graphs. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 195–204. ACM.

Fatehkia, M.; Kashyap, R.; and Weber, I. 2018. Using facebook ad data to track the global digital gender gap. *World Development* 107:189–209.

Francis, E.; Blumenstock, J.; and Robinson, J. 2017. Digital credit: A snapshot of the current landscape and open research questions. *CEGA White Paper*.

Frias-Martinez, V.; Frias-Martinez, E.; and Oliver, N. 2010. A gender-centric analysis of calling behavior in a developing economy using call detail records. In *AAAI spring symposium: artificial intelligence for development*.

Grover, A., and Leskovec, J. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

GSMA. 2016. Unlocking rural coverage: Enablers for commercially sustainable mobile network expansion. Technical report.

Jean, N.; Burke, M.; Xie, M.; Davis, W. M.; Lobell, D. B.; and Ermon, S. 2016. Combining satellite imagery and machine learning to predict poverty. *Science* 353(6301).

Khan, M. R., and Blumenstock, J. E. 2016. Predictors without borders: Behavioral modeling of product adoption in three developing countries. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Liu, W.; Wang, J.; and Chang, S.-F. 2012. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE* 100(9):2624–2638.

Lu, X.; Bengtsson, L.; and Holme, P. 2012. Predictability of population displacement after the 2010 haiti earthquake. *Proceedings of the National Academy of Sciences* 109(29):11576–11581.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Mislove, A.; Lehmann, S.; Ahn, Y.-Y.; Onnela, J.-P.; and Rosenquist, J. N. 2011. Understanding the demographics of twitter users. *ICWSM* 11(5th):25.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14. ACM.

Quercia, D.; Ellis, J.; Capra, L.; and Crowcroft, J. 2012. Tracking gross community happiness from tweets. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, 965–968. ACM.

Suri, T. 2017. Mobile money. *Annual Review of Economics* 9(1):497–520.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077. International World Wide Web Conferences Steering Committee.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing* 17(4):395–416.

Wesolowski, A.; Qureshi, T.; Boni, M. F.; Sundsøy, P. R.; Johansson, M. A.; Rasheed, S. B.; Engø-Monsen, K.; and Buckee, C. O. 2015. Impact of human mobility on the emergence of dengue epidemics in pakistan. *Proceedings of the National Academy of Sciences* 112(38):11887–11892.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, 40–48. JMLR. org.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004a. Learning with local and global consistency. In *Advances in neural information processing systems*, 321–328.

Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; and Schölkopf, B. 2004b. Ranking on data manifolds. In *Advances in neural information processing systems*, 169–176.

Zhu, X. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.