Discovering functionality of urban regions by learning low-dimensional representations of a spatial multiplex network

Seyedsaeed Hajiseyedjavadi
University of Pittsburgh
seh138@pitt.edu

Yu-Ru Lin
University of Pittsburgh
yurulin@pitt.edu

Konstantinos Pelechrinis
University of Pittsburgh
kpele@pitt.edu

ABSTRACT

The complex relationships in an urban environment can be captured through multiple inter-related sources of data. These relationships form multilayer networks, that are also spatially embedded in an area, could be used to identify latent patterns. In this work, we propose a low-dimensional representation learning approach that considers multiple layers of a multiplex network simultaneously and is able to encode similarities between nodes across different layers. In particular, we introduce a novel neural network architecture to jointly learn low-dimensional representations of each network node from multiple layers of a network. This process simultaneously fuses knowledge of various data sources to better capture the characteristics of the nodes. To showcase the proposed method we focus on the problem of identifying the functionality of an urban region. Using a variety of public data sources for New York City, we design a multilayer network and evaluate our approach. Our results indicate that our proposed approach can improve the accuracy of traditional approached in an unsupervised task.

KEYWORDS

Urban computing, Multiplex network, Network embedding

ACM Reference Format:
Seyedsaeed Hajiseyedjavadi, Yu-Ru Lin, and Konstantinos Pelechrinis. 2018. Discovering functionality of urban regions by learning low-dimensional representations of a spatial multiplex network . In *Proceedings of the 3rd Mining Urban Data Workshop (MUD3).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

Advancements in sensing technologies have enabled the collection of a vast and diverse amount of information for our cities. For example, the ever-increasing popularity of location-aware social media like Foursquare, Yelp, Twitter, Instagram etc. provide an excellent source of data that provide information about the mobility and activities of people in the city in the form of *check-ins* or geo-tagged user generated content (e.g., text, image, video etc.). In addition to this human-based urban sensing media other hardware-based sensing systems such as, GPS-equipped vehicles [17, 29, 30] and ticketing system of public transportation [32] provide complementary information for the behavior, mobility [9] and interactions recorded in a city.

All of these different information represent different types of relations between entities (them being city-dwellers, urban areas etc.). These relationships essentially form a multi-layer network[1], that is, a network where there are multiple different connections (i.e., type of edges) between the nodes. For example, considering as the nodes of a multiplex network being city-dwellers we can have two different layers, the one representing "who-is-friends-with-whom" and the other representing "who-has-been-colocated-with-whom".

Given the prevalence of networked data there has been an increasing interest in learning a low-dimensional representation of graphs, which further allows to use traditional machine learning methods for tackling various problems such as classification and prediction. The objective of graph embedding techniques is to learn a low-dimensional representation of the graph nodes such that the obtained vectors preserve some network property. The latter can refer to the network structure [10, 26] or the attributes of the original graph [12] to name a few possibilities. As alluded to above, the benefit of having low-dimensional representations is that we can apply existing state-of-art machine learning algorithms to solve network-related problems like link prediction [25], community detection [7] and graph visualization [27]. To the best of our knowledge, existing graph embedding methods are limited in the sense that they are mostly focused on learning an embedding space for simple graphs. In this work, we present a novel neural network architecture that is able to map nodes based on their relationships in different layers into a low-dimensional embedding space. While we elaborate on our approach later, in brief, the high level idea of our approach is the constraint that the representation we obtain for the same node from different layers must be close, while the vector representations of different nodes must be distant to each other. Furthermore, our architecture is flexible, and can also support a semi-supervised approach, where we utilize a small portion of node labels available.

In order to introduce and evaluate our multiplex network embedding approach we focus on the problem of identifying the functionality of an urban region. Discovering functional regions is not a trivial task. Until recently, labor-intensive surveys had been the primary source of decision making of urban planners. Furthermore, most of the time, the land use of a region could be a mixture of diverse functions which may not easily distinguishable [31]. On the other hand, as aforementioned the variety of sensors available today allows us to obtain a large and diverse dataset with information about activities and interactions of people in urban areas. Thus, the availability of these data sources provides an opportunity to fuse them and obtain a holistic view through a multiplex network, where nodes are urban areas and edges represent relationships

[1] We will use the terms multi-layer, multiplex and composite network interchangeably for the rest of the text.

MUD3, August 20th, 2018, London, UK

between the areas. In particular, we collect a number of different relationships between different areas of New York City including, mobility (e.g., trips originating/destined from two areas during the same time), context (e.g., type of venues within the two areas) etc. We further apply our multiplex network embedding to obtain a low-dimensional representation of the nodes based on all the layers, and we finally use this representation for classifying the nodes into different types of functions.

In summary, the contribution of our work is twofold:

(1) We present a novel neural network framework that jointly learns low-dimensional node representation from a multilayer network.
(2) We tackle the important problem of discovering the functionality of urban regions by using multiple human-generated data sources and our multilayer network embedding.

The remainder of this paper is organized as follows: In the following section we review relevant to our study literature. In section 3, we present our proposed architecture in more details, while Section 4 is dedicated to the evaluation of our method. Finally, Section 4.3 we present conclusions and future directions of our research.

2 RELATED WORK

In this section, we briefly describe several notable existing methodologies addressing the methodological problem of learning graph representation as well as our application of land use detection in an urban area.

2.1 Network embedding

A network embedding learns a low-dimensional representations of nodes in a graph, that is, nodes can be represented through a latent vector space. There are different objectives used to learn a network embedding. The most widely used ones force the representation to be learned to accurately preserve the graph structure [10, 26], and its attributes[12]. These embeddings allow us to apply traditional machine learning techniques, such as clustering [7] and classification [16] on complex networks.

In recent years, numerous approaches have been proposed to solve this task. Among the first works we can refer to [24] in which the authors propose a node embedding framework named Deep Walk. The framework is similar to what was presented as word embedding model in [21]. Considering a node as word, and each sentence as a sequence of nodes that a random walker visits during its walk, one can learn representation of a node based on its local neighbors that frequently appear before or after visiting it. Furthermore, Leskovec et al. [10] design a more flexible notion of a node's network neighborhood, named node2vec, by introducing a biased node sampling strategy, which allows the random walker to traverse various neighborhood to learn richer representations of nodes. More recently, Chang et al. [6] propose and examine a graph embedding method on heterogeneous networks. They use a deep neural network architecture to learn representations of nodes of different types, while in [28] the authors highlight the importance of using a joint latent space to embed nodes of two different but related networks into a common low-dimensional space.

2.2 Multilayer network

Many real-world systems are characterized by different views and properties that can not be modeled by single network. Ignoring these characteristics or aggregating them into a single layered network usually ends up losing valuable information [1]. Therefore, to model these systems, multilayer (or multiplex) networks are introduced. Multilayer networks are designed as more advanced network structures to capture various existing interactions or relationship between entities in a complex system. In a multilayer network, each type of relationship between nodes is represented through set of edges connecting nodes in a layer. These layers can be interconnected, thus, capturing interdependencies between the different layers. It is worth mentioning that multiplex network is a well-known structure to model temporal complex networks in a way that snapshots of the network taken in a specific time-window is considered as layers of the multilayer network [4, 13].

2.3 Spatial network

In a spatial network its nodes and edges are embedded in the geographical space. Typically nodes correspond to fixed regions, while (weighted) edges reflect connections between these regions, usually based on some underlying process (e.g., mobility) [3]. Complex spatial network representations and frameworks have helped researchers to devise realistic models and tackle various problems in transportation [14], navigation [5] and human mobility pattern modeling [2, 19].

As an example, in [15] the authors record the average number of phone calls made in each tower cell during different time of a day and construct a spatial network where nodes represents cell towers and the edges represent the (thresholded) correlation between the cell phone connections observed in the corresponding pair of cell towers. They further used this network to identify communities of regions that represent similar land use. As another example, Liu *et al.* [18] built a spatially-embedded network using taxi trip data from Shanghai. Every region corresponds to a node in the network, and the origin-destination pairs observed in the taxi trips correspond to the network edges. Using this network the authors further divide the urban area into several clusters, where regions with high intra-city flows are highlighted. This can potentially help urban planners to redefine administrative boundaries based on daily human mobility patterns.

2.4 Discovering urban land use

Identifying functionality of different regions of a city has been one the most significant problem in an urban studies. Pei et al. [23] were one of the first researchers tackled the problem by using mobile phone data collected in the city of Singapore. They claim that pervasive usage of mobile phones provides a valuable human generated data source that could be leveraged to reveal social function of urban land use. They cluster regions based on feature vector created on the relative calling pattern and the total calling volume of each region. Similar to [23] but with more publicly available dataset, Frias-Martinez et al. [8] represent each region of a city by tweeting activity patterns. Assuming regions with similar functionality have similar tweeting activity patterns during weekdays and weekends, they apply spectral clustering to assemble those regions having similar land use together. Addressing the importance of using multiple data sources to infer functionality of region, Yuan et al. [30][29] employ topic-based inference model based on both type of venues located in each region and human mobility pattern between regions. To do so, each region could be considered as a document made of words generated by a triple of time, volume of incoming, and volume of outgoing trips to and from the region. Furthermore, they consider the number of different venues located in that region as metadata of that document (i.e. region) and apply Dirichlet Multinomial Regression (DMR)[22].As a result, this method produces a distribution of topics (i.e. functions). Again region with similar distribution of functions could be clustered and be considered having similar land use.

3 PROPOSED METHODOLOGY

In this section, we present our proposed framework. First, we introduce the notations used throughout the paper. Then we describe our multilayer network feature learning algorithm and the proposed neural network architecture in more details.

3.1 Notations and Definitions

We model a multilayer network as a set of l network layers $\{g^1, g^2, ..., g^L\}$, where $g^l = (V, E^l)$ denotes a specific layer consisting of V nodes and E^l intra-layer edges. Simply put, the set of nodes are identical in all layers, while the set of intra-layer edges are different and represent different types of relationships. To reflect the significance of relationship between entities captured through edge set E^r of layer r, these edges could be weighted.

In this study, our objective is to learn low-dimensional representations of nodes of each layer. We use notion of w_i^l to denote the learned embedding space of node i on layer l.

3.2 Preliminary

Our frame is based on the the Skip-gram model was introduced by Mikolov *et al.* [20, 21]. It was introduced to learn a word embedding that can effectively predict its context (i.e., surrounding words). In particular they introduce two models, namely, the Continuous Bag of Words (CBOW) and the Skip-gram. In the Skip-gram model, the input is an one-hot vector representing the target word, and the objective function to be minimized is the log loss of the probability. Mathematically, the objective function of a Skip-gram model is:

$$\text{minimize } J = -logP(N(w_t)|w_t; \theta) \tag{1}$$

where θ are the model parameters that need to be learned and $P(N(w_t))$ is the probability of the context of the word with embedding w_t being $N(w_t)$. Considering a Naive assumption that the probability of words appearing in the context of the target word are independent of each others, we can rewrite the formula as:

$$\text{minimize } J = -log \prod_{w_c \in N(w_t)} P(w_c|w_t; \theta) \tag{2}$$

Finally, by assuming symmetry in the feature space the conditional probability can be modeled using the Softmax function. Hence, with w_t and w_c being the embedding vectors of the target and context words of a document with vocabulary set of V we have:

$$\text{minimize } J = -log \prod_{c \in C_t} \frac{\exp(w_c^T w_t)}{\sum_{i=1}^{|V|} \exp(w_i^T w_t)} \tag{3}$$

where C_t is the set of all words existing in the context window of the target word w_t. With this objective function in hand, one can simply compute the gradient and update the cost function at each iteration via Stochastic Gradient Descent.
Simply put, the goal of skip-gram model is to learn a representation w_t for each word such that its inner product with the corresponding representation of its context words w_c will be much higher as compared to that with words that do not appear in its context.

The Skip-gram model was further used by Perozzi *et al.* [24] in Deepwalk to learn node representation in a network. The authors sample the graph by utilizing random walks on the graph [11] and consider each node as a word in the Skip-gram mode and each sentence as a sequence of nodes that a random walker visits.

3.3 Proposed Framework

In this work, we propose an approach to learn an embedding for a multilayer network extending/using the Skip-gram model. In particular, we simultaneously learn a low-dimensional space of nodes based on their relationships in different layers. As alluded to above, in order to present and evaluate our approach, we will consider the application of discovering the land use of an urban region. This setting is ideal for considering a multilayer network, that is able to represent urban region (nodes) and different relationships (edges) between them on different network layers. By learning an embedding space of nodes in this graph, we expect nodes with similar functionality to obtain similar/closer representation.

For our approach we will also rely on graph sampling through random walks. Each layer is considered as a regular single network and thus, we will have multiple sequences of nodes for each layer.

Objective: Our objective is to learn a low-dimensional representation that not only preserves the structure of each layer, but also *forces* similar nodes in different layers have close representation. To learn the model, our framework optimizes the sum of multiple loss functions as follows:

$$\text{minimize } J = \sum_{l=1}^{L} (J_l) + J_p \tag{4}$$

where J_l is the loss function from learning the embedding space of nodes in layer l and J_p corresponds to objective function of pairing the same nodes of different layers (it can be thought of as a type of regularization). Pairing nodes is the process of enforcing representations of a node on multiple layers be similar. Conversely, representation of dissimilar nodes on different layers will become more distant to each other.

3.4 Learning process

To learn the embedding space of each layer, we use the negative sampling introduced in [21]. Negative sampling is an effective approach proposed to reduce the computational cost of the inner product of $|V|$ embedding vectors at each iteration (Eq. 3). Instead of computing inner products of the embedding of the target node with every other nodes of the network, we only sample a small set

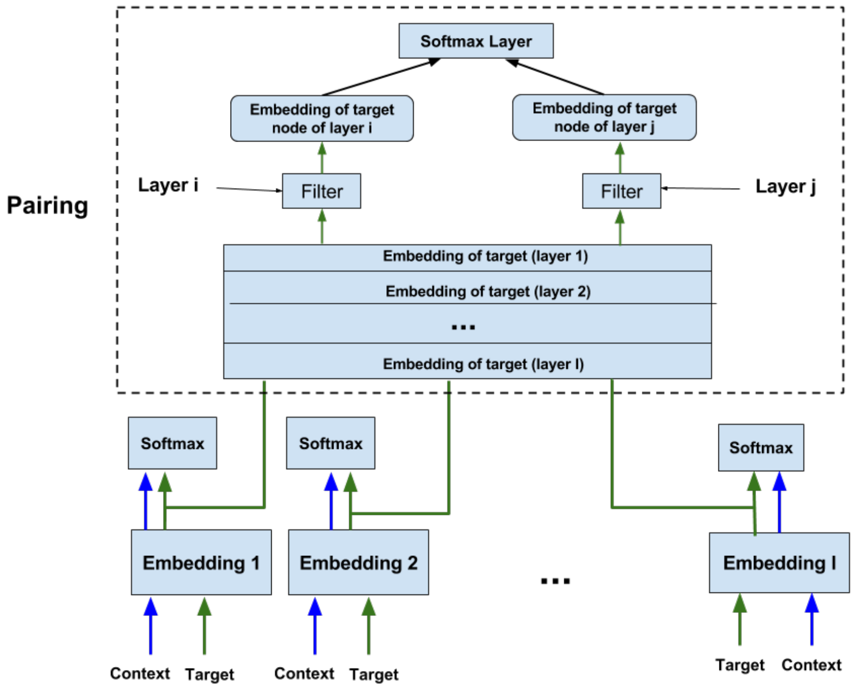

Figure 1: The proposed neural network architecture to learn low-dimensional representation of nodes in a multilayer network

of nodes that did not appear in the target's context. Assuming C_t as a set of visited nodes in the context of the target node V_t and $\hat{C}_t$ as a set of negative examples. With respect to Negative Sampling strategy, we can rewrite equation 3 as follows:

$$\underset{\theta}{\operatorname{argmax}} \prod_{c \in C_t} P(1|w_c, w_t, \theta) \prod_{c \in \hat{C}_t} P(0|w_c, w_t, \theta) \tag{5}$$

Here, we use one and zero for positive and negative samples respectively. With replacing probability with sigmoid function, we will have

$$\underset{\theta}{\operatorname{argmax}} \sum_{c \in C_t} log(\frac{1}{1 + exp(-w_c^T w_t)}) + \sum_{c \in \hat{C}_t} log(\frac{1}{1 + exp(w_c^T w_t)}) \tag{6}$$

Therefore, the loss function for learning embedding vector of nodes on layer l will be:

$$J_l = -\Big[\sum_{c \in C_t^l} log\sigma((w_c^l)^T . w_t^l) + \sum_{c \in \hat{C_t}^l} log\sigma(-(w_c^l)^T . w_t^l) \Big] \tag{7}$$

So far, the embedding space of each node is learned independently. However, as aforementioned, our framework enforces same nodes on different layers to have similar representations, and likewise, different nodes should have distant embedding spaces, through the pairing nodes process. This essentially allows us to fuse information obtained from different sources. We achieve this by setting:

$$J_p = -\Big[\sum_{t = t'} log\sigma((w_t^i)^T . w_{t'}^j) + \sum_{t \neq t'} log\sigma(-(w_t^i)^T . w_{t'}^j) \Big] \tag{8}$$

This function implies that during the learning process if the selected nodes of layers i and j are identical, then they are considered as positive sample between layers. Conversely, for non-identical nodes, they are considered as a negative sample. To pair nodes of more than two layers, we need to randomly choose i and j in the sampling phase which will be discussed in more details in the following section.

3.5 Framework architecture

Figure 1 illustrates our proposed neural network architecture. The first layer of the neural network, includes l modules that correspond to the number of network layers we have. The input to each module is a pair of context w_c^l and target nodes w_t^l. For the softmax layer, we need to know whether these w_c^l belongs to C_t^l (i.e., positive sample) or $\hat{C}_t^l$ (i.e., negative sample). The process of learning the embedding spaces of this layer is done by minimizing the loss function introduced in the previous section (Eq. 7). In each layer, the representation of each node is learned by looking at its context nodes. The choice of context nodes is defined by the sequence of nodes that the random-walker visits. If the walker tries to stay in the neighborhood of the previously visited node, the representation of each node will preserve the local structure of its neighborhood. Otherwise, the random-walker can go deeper in the network, and consequently, the embedding space will capture the richer structure of the network.

The next hidden layer of the neural network is our node pairing module. In this module, the target node's representation of each

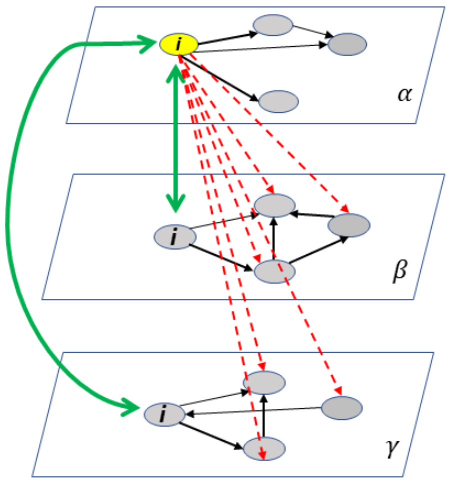

Figure 2: An example of positive and negative pairs of the node i from layer α and the nodes from the other layers. Green lines represent positive pairs and red dashed line represent negative pairs

layer will be stacked over each others. Simply put, for each iteration, a matrix $S_{l \times k}$ is constructed where l is the number of layers, and k is the dimension of the embedding space. If the number of layers exceeds two, then at each iteration, two layers i and j are randomly selected. The target nodes of the selected layers, w_t^i and $w_{t'}^j$, would be compared in the softmax layer based on the Eq. 8. The key insight of this module is that if these two target nodes are representing the same entity (i.e. $t = t'$) their embedding space need to be close (Fig. 2). Otherwise, if $t \neq t'$ then the inner product of their embedding spaces need to be near zero to minimize the objective function. This leads to learn dissimilar representations of those nodes.
In this work, we use the flexible approach introduced in [10] to learn representations to preserve a trade-off between local and global structure of the network. As it is discussed in this paper, the time-complexity of each sample of a single network is $O(\frac{l}{k(l-k)})$ in which l is the length of a random walk and k is the number of generated samples. In our work, the process of sampling of each layer is independent of the other layers which could be computed in parallel.

4 EXPERIMENTAL EVALUATION

In this section, we evaluate our model through its ability to discover the functionality of urban regions in NYC.

4.1 Datasets

In order to build the spatial multiplex network required we use various data sources that cover activities in NYC. In particular we use the following sources: 1) **Yellow taxi trip records** [2]. This dataset includes latitude and longitude of passenger pick-ups and drop-offs, dates, times, Taxi IDs and trip IDs. After removing trips with missing pick-up or drop-off coordinates, we have a total of 53,324,684 records collected between May to September 2016. 2) **Points of Interest (POI)**[3]. This dataset was collected using the Foursquare API and includes information about 181,208 POIs located in Manhattan, New York. Alongside the location of POIs, we have categories of them which falls into 9 major groups. 3) **User-generated content on Twitter**. This dataset includes more than 27 million anonymized geo-tagged tweets. Each tweet has a unique ID, anonymized user ID, time and the location of the posted tweet.

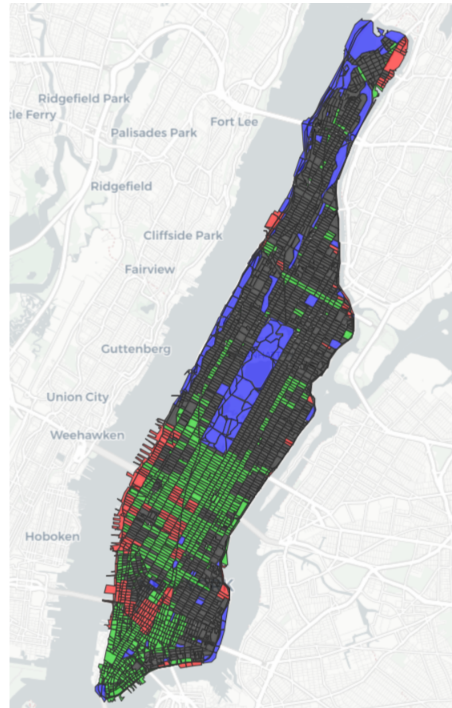

Figure 3: The Land Use of Manhattan. Red, blue, grey, and green areas respectively represents manufacturing, parks, residential, and commercial zones.

For our urban region functionality ground truth we use NYC ZOLA (the Zoning and Land Use Application)[4]. Each plot of land within the city's jurisdiction has been categorized into 4 major functions; residence, commercial, or manufacturing and parks (Fig. 3). For each region, we assign the the functionality of a zone that it has the greatest spatial overlap with. Except some exceptions, each region entirely falls inside a single zone.

4.2 Building the multilayer network

In our setting, we consider each block as our urban unit that corresponds a network node.
Using the datasets aforementioned we build the following 4 layers.

[2] http://www.nyc.gov/

[3] https://developer.foursquare.com/

[4] http://www1.nyc.gov/site/planning/zoning/about-zoning.page

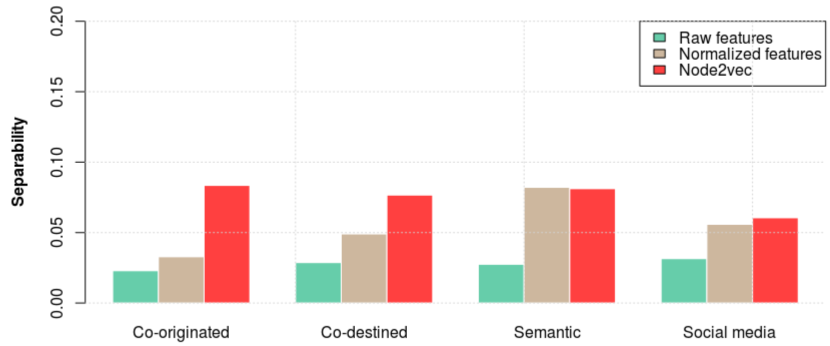

Figure 4: The comparison of baselines by the measure of Separability

Co-originated layer: We start by creating a feature vector $U^{(l)}$ for each node l (i.e., block). Each element of $U^{(l)}$ represents the average number of trips that are originated from the block l. To differentiate human mobility pattern on working days and holidays, $U^{(l)}$ is made by concatenation of average number of outgoing trips happened on weekdays and weekends. In other words, by using hourly granularity, as we choose to, we would have $U^{(l)}$ with the length of 48. Then, we standardize feature vectors $U^{(l)}$ into $\widehat{U^{(l)}}$ and calculate cosine similarities between the $\widehat{U^{(i)}}$ and $\widehat{U^{(j)}}$ for each pairs of i and j. The obtained cosine similarities are used as the weights of the link between the corresponding pair of nodes. In our final network layer we keep the top- k links (based on their weight value). In our experiments, we set $k = 10$ for all layers.

Co-destined layer: This layer is built in a similar to previous layer manner. In this case the feature for each node is based on the average number of incoming trips to a region.

Social media layer: Similar to the previous layers, the feature vectors are built based on the average number of geo-tagged tweets posted in each city block.

Semantic layer: For this layer, the feature vector of each node captures the distribution of the POI categories from Foursquare. Edges connect nodes based on the similarity of the feature vectors. Unlike the previous layers, temporality does not have any effect on the categories of venues.

The multiplex network encodes various relationship aspects of a region. Regions with similar mobility pattern, social media usage pattern, and venues are strongly inter connected to each other in this network. Intuitively connections with such similarities will have similar land use. Consequently, the learned space of nodes with similar characteristics needs to be close to each other, so that we can assert they have similar land use pattern.

4.3 Evaluation metric

The output of our framework is the node representations of different layers. In other words, we learn a feature vector representing each region of the city. To reiterate, the intuition is regions with close feature space belongs to the same functional zones. For our evaluations, we define an evaluation metric named *separability*, which is independent of any clustering algorithm applied:

Intracluster distances : the average of distances between representations of all pair of nodes i and j where they both have identical labels

Intercluster distances: the average of distances between representations of all pair of nodes i and j where they both have non-identical labels

$$Separability = \frac{Intercluster - Intracluster}{max(Intercluster, Intracluster)} \tag{9}$$

The intuition behind this metric is that if nodes of the same labels have close representations and nodes with different labels have far representations, the separability value would be high.

4.4 Baselines

In order to evaluate the goodness of the proposed framework we compare its performance with several baselines as follows:

Raw features: For each region l, its corresponding unnormalized feature vector is used (i.e. $U^{(l)}$).

Normalized features: For each region l, its corresponding normalized feature vector is used (i.e. $\widehat{U^{(l)}}$).

Independent embedding: After constructing each network layer based on steps described in section 4.2, we apply node2vec [10] on each layer to independently learn representations of each layer's nodes. The obtained representations will be used for the evaluation.

4.5 Results

We evaluate the performance of introduced baseline by the measure of Separability (Fig. 4). As it can be seen, applying single network embedding on each layer almost outperforms using raw and normalized feature vectors. One reason behind that must be the way network is built and used for the embedding. As it was described earlier, nodes (i.e. regions) with similar patterns are closely interconnected together and thus, during the sampling process, it is

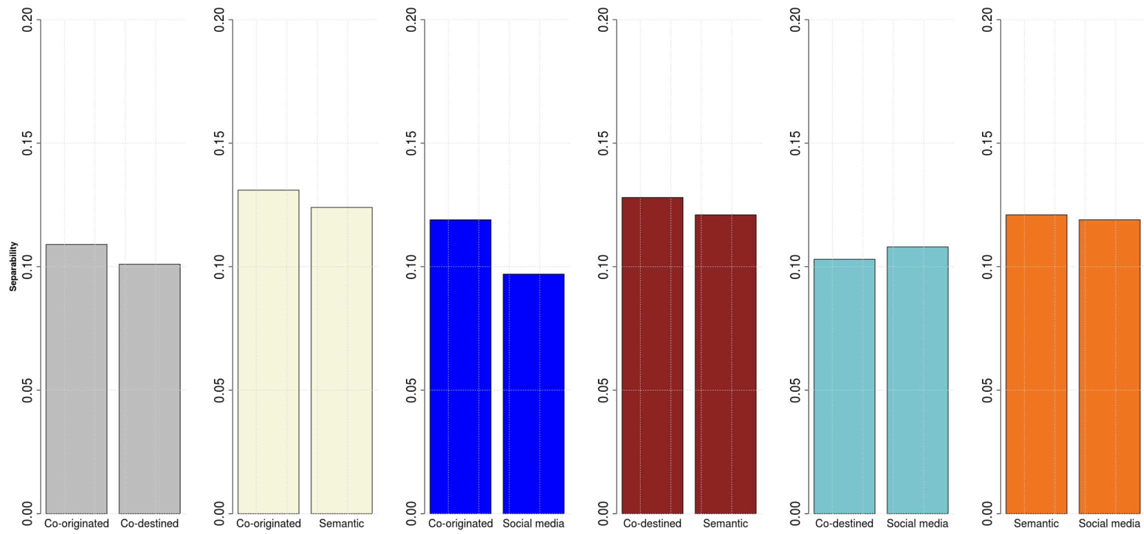

Figure 5: The performance of the proposed framework on every pair of layers

more likely the random walker visits densely connected nodes during its traverse. Using nodes appeared in the same visits as positive samples for learning phase, their representations would be learned in close to each others and far from other non-similar nodes. Therefore, as we expected the Separability value of the node representations would be higher.

We use our framework to learn node representations of all possible pair of layers together. Although the proposed framework is capable of learning node representations of more than two layers, for the sake of interpretability of the results, we only examine a pair of layers for each experiment and keep more advanced experiments and analysis for future works. In Fig. 5, we demonstrate the performance of the proposed framework by the measure of Separability. As it can be observed, the process of co-learning, enhanced the goodness of results. This must have come from the fact that in our framework, the representation of each node on each layer not only is learned by its context nodes, but also during pairing process we assure that the embedding space should not be different to the very same node on the other layer. In other words, this process share views between multiple layers to learn richer and more comprehensive representations for each nodes.

The other observation is the effect of semantic layer on the improvement of the results. Semantic layer, which is constructed based on the categories of venues location in each region, implies the type of activities that could be done in each region. During our experiments, we observed that this layer has a significant impact on improving the accuracy of representations of other layers, especially co-originated and co-destined layers. On the other hand, we can see that using two layers of co-originated and co-destined together did not end up a significant increase of the performance. We can argue that these two layers are sharing similar mobility patterns and could be combined into one for reducing input data.

5 CONCLUSION AND FUTURE WORK

In this work, we present a novel neural network architecture that aims to jointly learn from multiple layers of the network to find low-dimensional representations of each node. We learn low-dimensional embedding vectors of nodes in a way that the representation of identical nodes belonging to different layers would be similar, and embedding spaces of different nodes would be distant to each other. To measure its effectiveness we map information of publicly available datasets collected from New York City as a multilayer network and applied our framework to discover the functionality of its regions. For future direction, we plan to examine the effectiveness of the framework by adding labels during the learning process to improve the accuracy of the model. Also, it needs more experiment and analysis to reveal the hidden interaction of different layers and the contribution of each one on the result.

6 ACKNOWLEDGMENTS

This work is part of the PittSmartLiving[5] project, which is supported in part by National Science Foundation Award CNS-1739413.

REFERENCES

[1] Alberto Aleta and Yamir Moreno. 2018. Multilayer Networks in a Nutshell. *arXiv preprint arXiv:1804.03488* (2018).
[2] Duygu Balcan, Vittoria Colizza, Bruno Gonçalves, Hao Hu, José J Ramasco, and Alessandro Vespignani. 2009. Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences* 106, 51 (2009), 21484–21489.
[3] Marc Barthélemy. 2011. Spatial networks. *Physics Reports* 499, 1 (2011), 1–101.
[4] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. 2014. The structure and dynamics of multilayer networks. *Physics Reports* 544, 1 (2014), 1–122.
[5] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. 2010. Catastrophic cascade of failures in interdependent networks. *Nature* 464, 7291 (2010), 1025–1028.

[5]https://pittsmartliving.org/

[6] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 119–128.

[7] Weicong Ding, Christy Lin, and Prakash Ishwar. 2017. Node Embedding via Word Embedding for Network Community Discovery. *IEEE Transactions on Signal and Information Processing over Networks* 3, 3 (2017), 539–552.

[8] Vanessa Frias-Martinez and Enrique Frias-Martinez. 2014. Spectral clustering for sensing urban land use using Twitter activity. *Engineering Applications of Artificial Intelligence* 35 (2014), 237–245.

[9] Pedram Gharani, Kenrick Fernande, and Vineet Raghu. 2018. TRAJEDI: Trajectory Dissimilarity. *arXiv preprint arXiv:1803.03716* (2018).

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[11] Pili Hu and Wing Cheong Lau. 2013. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865* (2013).

[12] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 731–739.

[13] Saeed Haji Seyed Javadi, Pedram Gharani, and Shahram Khadivi. 2018. Detecting Community Structure in Dynamic Social Networks Using the Concept of Leadership. In *Sustainable Interdependent Networks*. Springer, 97–118.

[14] Stefan Lämmer, Björn Gehlsen, and Dirk Helbing. 2006. Scaling laws in the spatial structure of urban road networks. *Physica A: Statistical Mechanics and its Applications* 363, 1 (2006), 89–95.

[15] Maxime Lenormand, Miguel Picornell, Oliva G Cantú-Ros, Thomas Louail, Ricardo Herranz, Marc Barthelemy, Enrique Frías-Martínez, Maxi San Miguel, and José J Ramasco. 2015. Comparing and modelling land use organization in cities. *Royal Society open science* 2, 12 (2015), 150449.

[16] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. 2017. Attributed Network Embedding for Learning in a Dynamic Environment. *arXiv preprint arXiv:1706.01860* (2017).

[17] Antonio Lima, Rade Stanojevic, Dina Papagiannaki, Pablo Rodriguez, and Marta C González. 2016. Understanding individual routing behaviour. *Journal of The Royal Society Interface* 13, 116 (2016), 20160021.

[18] Xi Liu, Li Gong, Yongxi Gong, and Yu Liu. 2015. Revealing travel patterns and city structure with taxi trip data. *Journal of Transport Geography* 43 (2015), 78–90.

[19] Thomas Louail, Maxime Lenormand, Miguel Picornell, Oliva García Cantú, Ricardo Herranz, Enrique Frias-Martinez, José Javier Ramasco, and Marc Barthelemy. 2015. Uncovering the spatial structure of mobility networks. *Nature Communications* 6 (2015), 6007.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[22] David Mimno and Andrew McCallum. 2012. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *arXiv preprint arXiv:1206.3278* (2012).

[23] Tao Pei, Stanislav Sobolevsky, Carlo Ratti, Shih-Lung Shaw, Ting Li, and Chenghu Zhou. 2014. A new insight into land use classification based on aggregated mobile phone data. *International Journal of Geographical Information Science* 28, 9 (2014), 1988–2007.

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

[25] E Shijia, Shengbin Jia, Yang Xiang, and Zilian Ji. 2016. Knowledge Graph Embedding for Link Prediction and Triplet Classification. In *Knowledge Graph and Semantic Computing: Semantic, Knowledge, and Linked Big Data*. Springer, 228–232.

[26] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[27] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[28] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Embedding of Embedding (EOE): Joint Embedding for Coupled Heterogeneous Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 741–749.

[29] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 186–194.

[30] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2015. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2015), 712–725.

[31] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38.

[32] Chen Zhong, Michael Batty, Ed Manley, Jiaqiu Wang, Zijia Wang, Feng Chen, and Gerhard Schmitt. 2016. Variability in regularity: Mining temporal mobility patterns in london, singapore and beijing using smart-card data. *PloS one* 11, 2 (2016), e0149222.