

Link Activation using Variational Graph Autoencoders

Saeed Jamshidiha, Vahid Pourahmadi, Abbas Mohammadi, Mehdi Bennis

Abstract—An unsupervised method is proposed for link activation in wireless networks by identifying clusters of interfering users. A k-nearest neighbors interference graph is first defined for the wireless network which is then mapped to a stochastic latent space. The users are then clustered in the latent space using a Gaussian mixture model, and one user from each interfering cluster is activated while the rest of the users in that cluster remain idle. The proposed framework is scalable, works across several network topologies such as device to device (D2D), and is close to the optimal solution in performance.

Index Terms—Graph Embedding, Variational Graph Autoencoder, Wireless Networks, Bayesian Deep Learning

I. INTRODUCTION

GRAPH-based deep learning methods have recently attracted great attention, since there are many applications in which data are represented in the form of graphs, such as social networks, wireless networks and so on. [1].

The problem of resource allocation in a wireless network is an application in which data can be naturally represented by a graph. This problem is considered as a model-free statistical learning problem in [2, 3], where a graph neural network with lower computational complexity compared to a fully connected neural network has been utilized. The authors have expanded their work in [4] and have analyzed the performance of their proposed graph neural network under fairness constraints. The authors of [5] have proposed a graph neural network-based power allocation method considering an interference channel. An equivalent graph is defined for a D2D wireless network in [6], which is then mapped to an embedding space using the structure2vec algorithm [7] and the users are classified into active and idle classes in the embedding space using a fully connected neural network. The definition of the equivalent graph in [6] is only applicable to D2D networks and cannot be generalized to other network types.

In this letter, a fully unsupervised framework is proposed for solving the problem of link activation in an arbitrary wireless network, comprised of users with similar transmission powers. To achieve this goal, an interference graph is first defined, which is then mapped to a low dimensional stochastic latent representation using a variational graph autoencoder (VGAE), resulting in a much more scalable algorithm, compared to traditional graph-based algorithms. The users are then clustered in the representation domain using a Gaussian mixture model (GMM). After identifying the clusters of interfering users, a

random user from each cluster is activated in each time slot, while the rest of the users are idle.

The main contribution of this letter is to present a novel link activation method. A graph based Bayesian deep learning method (VGAE) is utilized to map the channel gain matrix into a low dimensional representation space. A GMM is utilized to cluster the users. As a result of using a low dimensional representation, the proposed method is scalable. As a result of using the Bayesian deep learning framework, the proposed method is more efficient in dimensionality reduction, compared to traditional methods like the PCA, since it has been trained on numerous networks in the training phase. The proposed method is not dependent on network type. The transmitters, however, are assumed to be similar in transmission power.

The rest of this paper is organized as follows. The system model is defined in section II. The proposed framework for link activation is presented in section III. Simulation results are presented in section IV, and section V concludes the paper.

II. SYSTEM MODEL

A wireless network of N users is considered in an a by b meter office building-type environment. Path loss, log-normal shadowing and Rayleigh fading result in channel gain $h_{ij}(t)$ between users i and j in time slot t . These gains are summarized in the channel gain matrix $\mathbf{H}(t)$. The users are randomly placed in the environment during each simulation episode, and their positions are assumed to be fixed during each episode. The channel gains, however, can change due to fading, and this results in changes in the channel gains from one time slot to the next.

Communication is half-duplex, and each user could transmit and receive data in each episode, but not simultaneously. There are no limits on how the users are paired, and multiple transmitters could be paired to a receiver (multiple access scenario) and multiple receivers could be paired to a transmitter (broadband scenario).

Defining $P_i(t)$ as the power allocated to user i in time slot t , the allocated power matrix $\mathbf{P}(t)$ can be defined as:

$$\mathbf{P}(t) = [P_1(t), P_2(t), \dots, P_N(t)]^T \quad (1)$$

where \top denotes vector transposition. The signal to interference and noise ratio (SINR) of the link between transmitter i and receiver j in time slot t is then defined as:

$$\gamma_{ij}(t) = \frac{P_i(t) \cdot h_{ij}(t)}{\eta + \sum_{i=1, i \neq j}^N P_i(t) \cdot h_{ij}(t)}, \quad (2)$$

where η is the additive white Gaussian noise power.

The data rate of the link between transmitter i and receiver j in time slot t can be defined as:

S. Jamshidiha, V. Pourahmadi and A. Mohammadi are with the Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran, (E-Mail: s_jamshidiha@aut.ac.ir, v.pourahmadi@aut.ac.ir, abm125@aut.ac.ir)

M. Bennis is with the Centre for Wireless Communications, University of Oulu, Finland, (E-Mail: mehdi.bennis@oulu.fi)

$$r_{ij}(t) = W \cdot \log(1 + \gamma_{ij}(t)), \quad (3)$$

where W is the bandwidth of the communication channel.

To indicate whether a user is actively transmitting or idle, $\beta(t) = [\beta_i(t)]$ is defined as follows:

$$\beta_i(t) = \begin{cases} 1 & \text{if user } i \text{ is actively transmitting in time slot } t \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The link activation policies are formulated as:

$$\mathbf{P}(\mathbf{H}(t), \beta(t)) = [\beta_1(t) \cdot P_{max}, \beta_2(t) \cdot P_{max}, \dots, \beta_N(t) \cdot P_{max}]^T \quad (5)$$

where P_{max} is the maximum allowable transmission power.

Given the described system model, the goal of this letter is to find a link activation policy that can maximize the sum rate of the network. This task is formulated as the following optimization problem:

$$\begin{aligned} \max_{\mathbf{P}(\mathbf{H}(t), \beta(t))} & \sum_{t=1}^T \sum_{j=1}^N \sum_{i=1}^N r_{ij}(t) \\ \text{s.t.} & \mathbf{P}(\mathbf{H}(t), \beta(t)) \in \{0, P_{max}\}^{N_T} \\ & r_{min} \leq \frac{1}{T} \sum_{t=1}^T r_{ij}(t) \end{aligned} \quad (6)$$

where T is the number of time slots in each simulation episode, N_T is the number of transmitters, and r_{min} is the minimum acceptable average rate on each link. The first constraint limits the transmission powers, and the second constraint guarantees an average rate requirement.

III. THE PROPOSED LINK ACTIVATION FRAMEWORK

The exact solution of the link activation problem (6) is NP-hard, and several methods have been proposed in the literature to find an approximate solution. In this section, we first present an approximation to this problem in subsection III-A, and then proceed to solve the proposed approximated problem in subsection III-B.

A. The Approximate Problem

Since interference (the summation in the denominator of (2)) is the major contributing factor in the achievable rates in (6), the active transmitters (users with $\beta_i(t) = 1$) should be selected in such a way that interference between them is minimized. In order to do so, we group interfering users in the same cluster, and activate one user from each cluster at any given time. These clusters are defined by means of $\alpha = [\alpha_{ik}]$, where $\alpha_{ik} = 1$ signifies membership of user i in cluster k . The clusters are obtained by solving the following problem:

$$\begin{aligned} \alpha &= \operatorname{argmax}_{\alpha} \sum_{k=1}^{N_c} \sum_{i=1}^N \sum_{j=1}^N h_{ij}(t) \alpha_{ik} \alpha_{jk} \\ \text{s.t.} & \alpha_{ik} \in \{0, 1\} \\ & \sum_{k=1}^{N_c} \alpha_{ik} = 1 \end{aligned} \quad (7)$$

where N_c is the number of clusters. The solution of the optimization problem (6) is approximated by:

$$\begin{aligned} \beta(t) &= \operatorname{argmax}_{\beta(t)} \sum_{j=1}^N \sum_{i=1}^N r_{ij}(t) \\ \text{s.t.} & \beta_i(t) \in \{0, 1\} \\ & \sum_{i=1}^N \alpha_{ik} \beta_i = 1 \\ & r_{min} \leq \frac{1}{T} \sum_{t=1}^T r_{ij}(t) \end{aligned} \quad (8)$$

In practice, the value of α has a much greater impact on the sum rate of the network than the value of β given a fixed value of α . Also, the feasible set of (8) is tightly limited by the constraints. As a result, by picking a random user in each cluster, and assigning $\beta_i = 1$ to that user, and $\beta_i = 0$ to all other users in that cluster, a very fast suboptimal solution could be obtained without much degradation of performance. Since the active users are selected randomly in each time slot, the third constraint is satisfied.

Problem (7) could be solved in the graph domain as well as the vector space domain. A VGAE is utilized in this letter to perform the mapping in a scalable way, and a GMM is used in the resulting vector space to obtain the suitable clusters for the link activation task. The proposed method is much faster than both solving the problem on the original channel gain matrix, and in the graph domain using traditional graph clustering methods.

B. The Proposed Solution

The proposed method aims to identify clusters of interfering users, and activate only one user from each cluster in each time slot. The block diagram of the proposed framework is presented in Fig. 1. Following the order of the block diagram, an interference graph is defined in subsection III-B1. This graph is then mapped to a stochastic latent space in subsection III-B2. The users are then clustered in the latent domain in subsection III-B3.

1) *K-Nearest Neighbors Interference Graph*: A wireless communication network could be thought of as a fully connected graph with edge features equal to the channel gains. To reduce computational complexity, instead of using the fully connected graph, we define a k-nearest neighbors interference graph by linking each node to the k nodes ($k < N$) with which it has the strongest channels (largest channel gains).

To formally define this graph, the feature vector for node i is first defined as:

$$\mathbf{x}_i = [h_{ij}, h_{il}, \dots, h_{im}]^T, \quad (9)$$

where $h_{ij}, h_{il}, \dots, h_{im}$ are the k largest elements in row i of the channel gain matrix \mathbf{H} . These features could be represented as an $N \times k$ matrix \mathbf{X} .

The adjacency matrix $\mathbf{A} = [a_{ij}]$ of the k-nearest neighbors interference graph is then obtained by:

$$a_{ij} = \begin{cases} 1 & \text{if } h_{ij} \in \mathbf{x}_i \\ 0 & \text{if } h_{ij} \notin \mathbf{x}_i \end{cases} \quad (10)$$

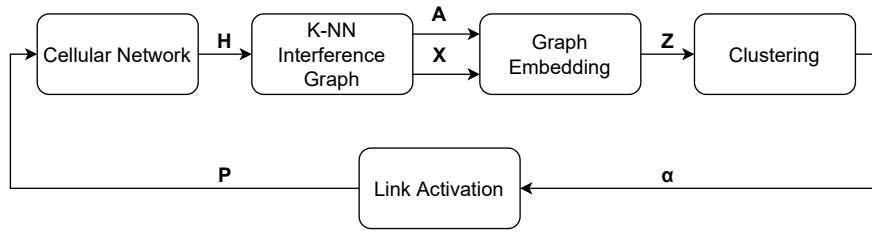


Fig. 1: Block Diagram of the Proposed Framework

2) *Graph Embedding*: The goal is to assign a low dimensional vector to each user to facilitate the clustering of the users in that vector space. An inference model is trained to map the wireless network into this latent variable space, and a generative model is trained to recover the network from the latent variable representation, in the form of a VGAE [8]. Formally, the nodes of the interference graph are mapped to a latent space:

$$f : v_i \in G \rightarrow \mathbf{z}_i \in \mathbb{R}^d \quad (11)$$

where v_i represents node i in graph G , \mathbf{z}_i represents the vector representation assigned to node i , which is a stochastic latent variable, and $d \ll N$ where N is the number of vertices in G .

In this letter, the VGAE introduced in [8] has been utilized. This architecture is comprised of an inference model and a generative model [8].

Inference Model [8]:

$$q(\mathbf{Z} | \mathbf{X}, \mathbf{A}) = \prod_{i=1}^N q(\mathbf{z}_i | \mathbf{X}, \mathbf{A}), \quad (12)$$

where \mathbf{Z} is the matrix of stochastic latent variables \mathbf{z}_i , and:

$$q(\mathbf{z}_i | \mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{z}_i | \boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)), \quad (13)$$

where $\mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2))$ is the normal distribution with mean $\boldsymbol{\mu}_i$ and standard deviation $\boldsymbol{\sigma}_i$, obtained by applying a graph convolutional network on (\mathbf{X}, \mathbf{A}) [8]:

$$\boldsymbol{\mu} = GCN_{\boldsymbol{\mu}}(\mathbf{X}, \mathbf{A}) \quad (14a)$$

$$\log \boldsymbol{\sigma} = GCN_{\boldsymbol{\sigma}}(\mathbf{X}, \mathbf{A}) \quad (14b)$$

where $GCN()$ is the graph convolutional neural network defined in [9]. After calculating $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ using (14), \mathbf{Z} could be obtained by sampling the distribution in (12).

Generative Model [8]:

$$p(\mathbf{A} | \mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(a_{ij} | \mathbf{z}_i, \mathbf{z}_j), \quad (15)$$

$$p(a_{ij} | \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^T \mathbf{z}_j), \quad (16)$$

where $\sigma()$ is the sigmoid function.

3) *Clustering*: So far, we have assigned a latent representation \mathbf{z}_i to every user. The optimization problem in (7) is solved by clustering the users in the representation domain using a Gaussian mixture model (GMM):

$$\begin{aligned} \max_k \quad & \pi_k \cdot Pr\{\mathbf{z}_i | \alpha_{ik} = 1\} \\ \text{subject to} \quad & Pr\{\mathbf{z}_i | \alpha_{ik} = 1\} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2) \\ & 0 \leq \pi_k \leq 1 \\ & \sum_{i=k}^{N_c} \pi_k = 1 \end{aligned} \quad (17)$$

where k is the cluster to which user i is assigned ($\alpha_{ik} = 1$), $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ is the normal distribution with mean $\boldsymbol{\mu}_k$ and standard deviation $\boldsymbol{\sigma}_k$ for cluster k , π_k is the weight associated to cluster k in the GMM, and N_c is the number of clusters.

Up to this point, the value of α has been obtained by mapping the users to the latent representation space and clustering them in that space using (17). Now, by assigning β the solution of (6) can be approximated. As explained earlier, we simply activate one random user in each cluster.

IV. SIMULATION RESULTS

In order to verify the claims made earlier, that (6) can be approximated by (8), and that instead of actually solving (8), β could be assigned randomly, the solutions of (6) and (8) are obtained by exhaustively searching all possible configurations, and are then compared to the proposed method. The results obtained by link activation based on spectrum sensing, and random link activation are provided for comparison. Since the figures are obtained by simulation, they are not perfectly smooth, and even after averaging many randomly initiated cases, they still exhibit some variations.

A. D2D Network

The following results are obtained from a D2D network operating at 2.4 GHz frequency in a 500m by 500m environment, with the power of each active link equal to -20 dBm (100mW). 1000 randomly initiated networks have been simulated, each for 10 time slots, resulting in 10000 simulated time slots.

As evident in Fig. 2 the performance of the proposed method is very similar to the optimal solutions of (6) and (8).

To explore the effect of the difference between measured SINRs and captured SINRs, the case of noisy CSI has been simulated, which arises as a result of the difference between measured and captured SINRs. Fig. 3 presents this case.

Fig. 4 shows the performance of the proposed method for larger values of N . Since the optimal solutions of (6) and (8) are obtained by an exhaustive search, these solutions take prohibitively long times to calculate. As a result, they are omitted from this comparison.

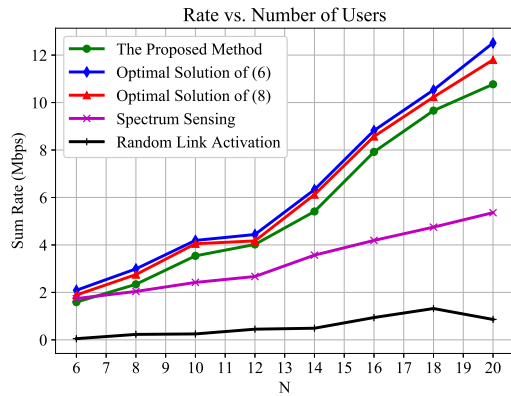


Fig. 2: The Proposed Method vs. the Optimal Solutions

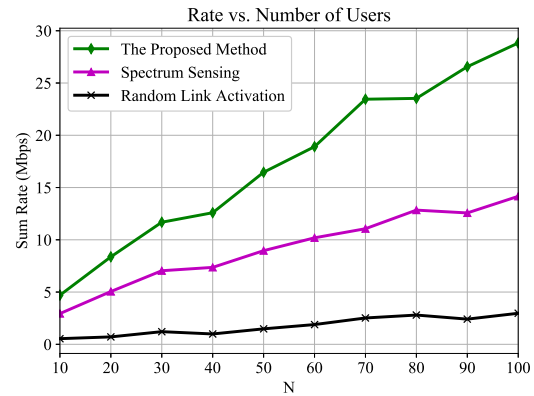


Fig. 4: Performance for Large Values of N

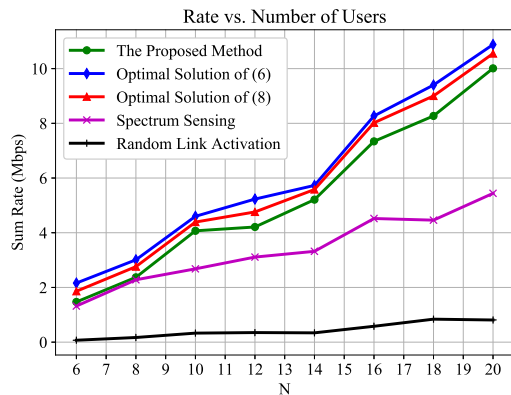


Fig. 3: Noisy CSI

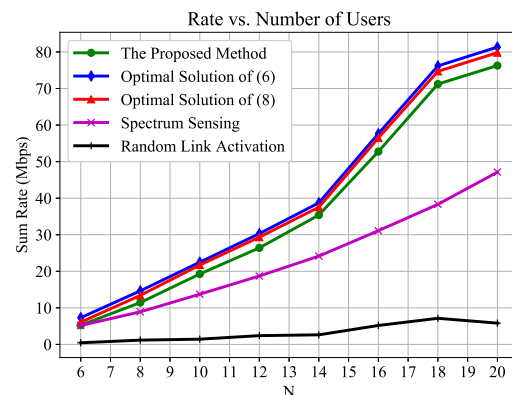


Fig. 5: Fully Random Ad-hoc Network

B. Fully Random Ad-hoc Network

The structure of the network is fully random in this scenario, i.e. any transmitter could transmit to any number of receivers in each time slot, and the associations are random.

Fig. 5 compares the proposed method, the optimal solutions of (6) and (8), link activation by spectrum sensing, and random link activation in a fully random network.

V. CONCLUSION

In this letter, an unsupervised link activation method was presented for any arbitrary wireless network structure comprised of similar users, using a VGAE and a GMM, where the users are clustered by the GMM in the latent representation domain obtained from the VGAE, and one transmitter from each cluster is activated in each time slot. Link activation in a D2D network and an ad-hoc network was simulated, demonstrating the slight difference between the proposed method and the optimal solution.

REFERENCES

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
 [2] M. Eisen and A. R. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Transactions on Signal Processing*, 2020.

[3] M. Eisen and A. Ribeiro, "Large scale wireless power allocation with graph neural networks," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2019.
 [4] N. Naderializadeh, M. Eisen, and A. Ribeiro, "Wireless power control via counterfactual optimization of graph neural networks," *arXiv preprint arXiv:2002.07631*, 02 2020.
 [5] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," *arXiv preprint arXiv:1907.08487*, 2019.
 [6] M. Lee, G. Yu, and G. Y. Li, "Graph embedding based wireless link scheduling with few training samples," *IEEE Transactions on Wireless Communications*, 2020.
 [7] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *International conference on machine learning*, pp. 2702–2711, 2016.
 [8] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *NIPS Workshop on Bayesian Deep Learning*, 2016.
 [9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.