Dataset Recommendation via Variational Graph Autoencoder

Basmah Altaf, Uchenna Akujuobi, Lu Yu, Xiangliang Zhang
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{basmah.altaf, uchenna.akujuobi, lu.yu, xiangliang.zhang}@kaust.edu.sa

Abstract—This paper targets on designing a query-based dataset recommendation system, which accepts a query denoting a user’s research interest as a set of research papers and returns a list of recommended datasets that are ranked by the potential usefulness for the user’s research need. The motivation of building such a system is to save users from spending time on heavy literature review work to find usable datasets. We start by constructing a two-layer network: one layer of citation network, and the other layer of datasets, connected to the first-layer papers in which they were used. A query highlights a set of papers in the citation layer. However, answering the query as a naive retrieval of datasets linked with these highlighted papers excludes other semantically relevant datasets, which widely exist several hops away from the queried papers. We propose to learn representations of research papers and datasets in the two-layer network using heterogeneous variational graph autoencoder, and then compute the relevance of the query to the dataset candidates based on the learned representations. Our ranked datasets shown in extensive evaluation results are validated to be more truly relevant than those obtained by naive retrieval methods and adoptions of existing related solutions.

Keywords—dataset recommendation; query-based recommendation; heterogeneous variational graph autoencoder

I. INTRODUCTION

Data availability is crucial in quantitative and qualitative analysis, and verification of scientific hypothesis and proposals. Researchers often need to evaluate the efficiency and performance of their solutions on benchmark datasets, or to find new datasets to explore new application domains. Usually, researchers find datasets by two means. First, they extensively read a number of papers and then know what datasets were used in their interested papers. This time-consuming way is difficult and not efficient for researchers who are at the entry-level of their research domains. Even several datasets can be found after spending time on reading several papers; there may exist other usable data resources because exhaustive search is almost impossible. The second way is to use search engines, e.g., Google dataset search1 or Delve2. However, the returned results highly depend on the specified query words. It is often necessary to go several rounds by verifying the search results and revising query words [1]. Motivated by the difficulties of finding usable and relevant datasets, we aim at developing a dataset recommendation model, which assists users to efficiently find more truly relevant and usable datasets than current time-demanding ways. This is not discouraging researchers from reading papers, because in order to grasp the latest trends in academia, there is a burden of reading a large number of research papers that are necessary for starting new research. In contrast, the system can suggest additional usable datasets, which direct researchers to read more relevant papers and make exploration of new domains than manually searching.

Our studied problem can be formally described as: given a bipartite graph 𝐻𝑃,𝐷 linking papers 𝑃 and datasets 𝐷, a citation graph 𝐺𝑃, and the content of papers 𝑋𝑃, we target on answering a query 𝑄 ≜ {𝑝𝑖}𝑖=1,...,|𝑄|, which is composed of a few papers in 𝑃 indicating a user’s research interests. The answer is the top-𝑘 datasets {𝑑1, 𝑑2, ..., 𝑑𝑘} selected from 𝐷, ranked according to the relevance to the query 𝑄.

Intuitively, one can think of a naive way to answer the query. That is, retrieve all 𝑑𝑗 that are linked with papers 𝑝𝑖 in 𝑄, and then rank them by their degree in 𝐻𝑃,𝐷. Even more, extend 𝑄 by including 𝑝𝑖’s first or second order neighbors on citation graph 𝐺𝑃, and then retrieve all 𝑑𝑗 linked with papers in extended 𝑄. These naive solutions do not find a set of relevant datasets but exclude other semantically relevant datasets that widely exist in several hops away from the queried papers. Also, ranking by degree considers only the overall popularity of dataset, and ignores the semantic meaning of the query 𝑄. Our evaluation results presented in Section V also verify that such naive solutions perform worse than other methods.

We thus address this problem by mapping papers and datasets to the same low-dimension semantic space, and then score the candidate datasets by their relevance to the query papers, which is measured based on the mapped representations. Such representation-based ideas have been popularly used in the recent study of recommendation problems [2], [3]. However, we face a query-based recommendation problem that needs a new design to shape the general representation learning idea for the following special challenges.

First, paper and dataset recommendation should be learned by jointly considering the bipartite graph 𝐻𝑃,𝐷, the citation graph 𝐺𝑃, and the content of papers 𝑋𝑃, as shown in Figure 1. However, current representation learning models for addressing recommendation problems are mainly designed for learning user and item representation from only the user-item bipartite graph [2], [3]. Even when

1https://toolbox.google.com/datasetsearch
2http://delve.kaust.edu.sa
leveraging social relationships to simulate user-item interaction [4], [5], the model is not designed to jointly learn from an attributed heterogeneous graph like what we have. Such social recommendation systems focus on cross-domain knowledge transferring [6], because user representations are separately learned from social links and interactions with items as two tasks in different domains. Considering cross-domain is not necessary for our problem as our bipartite graph, citation graph, and paper content are in the same domain where there exist common semantics of research topics. Recent embedding models for homogenous plain or attributed graph such as DeepWalk [7], node2vec [8], GraphRec [5], DANSER [9], TADW [10], ASNE [11] and ANRL [12] are adoptable, however by sacrificing the information of paper content, or the roles played by different types of nodes. Other heterogeneous network embedding models work on specified meta-paths [13], [14], which capture well the similarity between heterogeneous nodes, but ignore the nodes’ homogeneous neighborhood or nodes’ content. In our case, neighboring papers in the citation graph have strong correlations and thus must be used for learning paper representations. Papers’ content is also indispensable because it already forms a representation, e.g., by bag-of-words, although not compact and rich enough. We discuss in details the related approaches in Section II.

Second, query-based recommendation is relevant to context-aware recommendation, the goal of which is to incorporate the contextual information of a user to recommend specific items [15]. However, most of the existing context-aware recommender systems mainly focus on how to collect or utilize certain contextual information such as time [16] and location [17] and are not able to model stand-alone recommendation based on query only. Recently, interesting query based recommendation is studied for music, citations, and image retrieval [18]–[20]. These proposed models are designed to answer the query of a single entity rather than a list of entities. Moreover, these methods are domain specific and are not applicable to model heterogeneous relations and node attributes as needed in our case.

To facilitate our study, we build a Heterogeneous Variational Graph AutoEncoder (HVGAE) that jointly learns representations for papers and datasets in low-dimensional latent space from \( H_{P,D}, G_P, \) and \( X_P \). More specifically, two autoencoders take respectively the paper-dataset bipartite network \( (H_{P,D}) \), and the attributed paper-paper citation network \( (G_P + X_P) \), to jointly compress the rich information into latent embeddings of datasets and papers, which can reconstruct \( H_{P,D} \) and \( G_P \). The model architecture is shown in Figure 2. The query (list of papers) representation is then obtained from paper representation by simply taking the element-wise mean of papers’ representation vectors. Finally, we employ similarity score function for query and all datasets to find the top-k datasets that are highly relevant to the given query.

The proposed HVGAE is an appropriate solution for our problem and has the following advantages comparing to other representation learning models. 1) It integrates seamlessly \( H_{P,D}, G_P, \) and \( X_P \) to represent papers and datasets in the same semantic space regarding the latent research topics they are relevant to; 2) It does not rely on a “sampling” strategy to extract the evidence of paper-dataset relevance, which has a significant impact on the representation results; 3) It can capture the uncertainty of paper and dataset representation, and thus support the relevance measure between them be based on the comparison of two distributions, rather than on two single vectors. Both papers and datasets are likely related to several latent research topics to different extents. HVGAE allows the comparison of them on all related aspects.

We implement HVGAE model and evaluate its effectiveness on the query-based dataset recommendation problem, which to the best of our knowledge has not been studied before. On a real dataset including 8519 papers and 5610 datasets, we show HVGAE can effectively answer queries in different length (from one paper to more than six papers). When comparing with the baselines, including naive solutions and those adopting other representation learning methods, HVGAE shows better performance on different ranking metrics.

II. RELATED WORK

Our study is relevant to the following five lines of work.

Social Recommender Systems: Social recommendation leverages social information to solve data sparsity and cold-start problems in traditional collaborative filtering methods. Our problem is analogous to incorporating social relations in recommendation; the papers are equivalent to users and datasets are analogous to items while the paper citation network can be viewed as the social relationship among users.
Recent works [5], [9] employ convolutional techniques to address the multi-hop problem for learning from both user-user and user-item relations. However, they only incorporate the social graph into recommendation, while many real-world applications are associated with rich node attributes. To solve the heterogeneity of social relations, we learn semantics of paper nodes from their associated rich text information and their neighboring nodes associated text.

Query based Recommendation: In this direction, there is a query-based music recommendation via preference embedding [18] that allows a user to explore new song tracks by specifying either a track, an album, or an artist. Besides this, citation recommendation methods accept a given manuscript as the query, and return a list of reference papers that can be cited by this new draft [19], [20]. These approaches are mostly designed for single content based retrieval only, and meet the needs of specific domain. In our study, we need to model jointly the relationship between homogeneous nodes (paper) and heterogeneous nodes (paper and dataset) besides the content of paper nodes for retrieving datasets based on the query specified as a list of papers.

Homogeneous Graph Embedding: In recent years, there has been a surge in approaches that automatically learn to encode graph structure into low-dimensional embeddings. There are embedding models that only consider the graph structure, such as DeepWalk [7], node2vec [8]. Also, there are attributed graph embedding models incorporating the node contents, including Graph2Gauss [21], GraRep [22], TADW[10], Planetoid [23], ASNE [11], ANRL [12], GCN [24], column networks [25], GraphSAGE algorithm [26], and GAT [27]. Although these attributed graph embedding models can be applied to our paper-dataset heterogeneous network, they treat paper and dataset without distinguishing their nature.

Heterogeneous Information Network (HIN) Embedding, which handles different types of nodes in networks: Metapath2vec proposed in [13] obtains the neighbors of a node via meta-paths and learns the HIN embedding by skip-gram [41] with negative sampling. A possible limitation of metapath2vec is that the learned embeddings mainly reflect the structural features of HIN, rather than path features, and may not be directly useful for the recommendation task. Besides, it does not consider node attributes. In the context of recommender systems, task-guided, path-augmented, and semantic-aware HIN embedding methods have been proposed in [4], [14], [28], [29]. However, these approaches are designed to model the relationship between heterogeneous nodes using meta-paths and thus unable to emphasize the social links dependency in learning representation, which in our case is the paper-paper citation relationship playing important roles in the inference of paper-dataset relevance. In addition, these methods typically consider a node’s local neighborhood within the same path and thus overlook the global structure information. Moreover, these models rely on a sampling strategy to collect paths in HIN. However, it is difficult to find an “optimal” sampling strategy as the embedding and path sampling are not jointly optimized in a unified framework. Other deep neural models exempt HIN embedding from sampling paths by random walk, e.g. HNE [30], TransE [31], ProxEmbed [32], and HIN2Vec [33]. However, all these methods fail to consider the attributes of nodes in HIN. In our case, paper nodes content is indispensable for HIN embedding because the content itself already forms a representation e.g., by bag-of-words, although not compact and rich enough.

Variational Graph Autoencoders (VGAE): From methodology perspective, our work is related to variational graph autoencoders [34], which is a framework for unsupervised learning on graph-structured data based on the variational autoencoder (VAE) [35]. An attributed network embedding model using VGAE is proposed in [36], for learning both node and attribute representations in the same space. However, none of these work can consider collectively the bipartite links between heterogeneous nodes (paper and dataset nodes in our case), citations between paper-paper, and paper content. VGAE was also introduced to learn item rating embeddings, by considering priors for users or items, and the user review texts [37]. However, the main contribution of our work lies in the employment of VGAE to learn node representations, rather focusing on collaborative filtering.

III. PROPOSED MODEL FOR QUERY-BASED DATASET RECOMMENDATION

We address the query-based dataset recommendation problem by three steps:

1) Learn the representation of paper and dataset nodes based on HVGAE (Section III-A). Formally, from the bipartite graph \( H_{p,d} \) (where \( H_{i,j} = 1 \) if paper \( p_i \) used dataset \( d_j \); \( H_{i,j} = 0 \) otherwise), citation graph among papers \( G_P \) with adjacency matrix \( A_{p,p} \), and content of paper nodes \( X_p \), we aim to learn embeddings for paper \( p \) and dataset \( d \) as \( \tilde{Z}_p \) and \( \tilde{Z}_d \), respectively.

2) Learn representation for query \( Q \) as \( \tilde{z}_Q \) from representation \( \tilde{Z}_p \) of papers in \( Q \) (Section III-B).

3) Make dataset recommendation by measuring the relevance of query \( Q \) and dataset candidates \( d \) such that \( s(Q,d) = f(\tilde{z}_Q, \tilde{z}_d) \) (Section III-C).

A. Learning \( \tilde{Z}_p \) and \( \tilde{Z}_d \) (Step 1)

The framework of learning paper and dataset representation vector \( \tilde{z}_p \) and \( \tilde{z}_d \) is shown in Figure 2. Notations used in our model are summarized in Table I.

As shown in Figure 2, we encode the attributed citation network (described as \( A_{p,p}, \) and \( X_P \)) by GraphSAGE [26], which is a state-of-the-art homogeneous attributed graph
embedding method. It is selected because of its capability of doing inductive learning (inerring new node’s representation by defining neighborhood edges), which allows us to infer the representation for a query. The dataset encoder is a densely connected multi-layer perceptron (MLP), taking the representation for a query. The dataset encoder is a densely connected two-layer MLP, taking input of \( \phi \) for modeling the posterior distribution \( q_\phi (\mathbf{z} | x) \) assumed to be Gaussian with a diagonal covariance matrix. Thus, given input \( x_i \), the encoder outputs the mean and variance in \( N(\mu_{z_i}, \text{diag}(\sigma_{z_i}^2)) \), from which \( z_i \) is sampled.

**Encoder setting:** In our case, two inference networks (encoders) are parametrized by \( \phi_1 \) and \( \phi_2 \) for modeling the posterior distribution \( q_\phi (\mathbf{z} | x, A_{P,D}) \) and \( q_\phi (\mathbf{z} | H^T_{P,D}) \), respectively, as shown in Figure 3 (a). Here \( \mathbf{Z}_P \in \mathbb{R}^{(|P|) \times L} \) and \( \mathbf{Z}_D \in \mathbb{R}^{(|D|) \times L} \) is the collection of \( \mathbf{z}_p \) for all \( p \in P \) (\( \mathbf{z}_d \) for all \( d \in D \)). More specifically, by assuming the independence of latent variable \( \mathbf{z}_p \) among papers,

\[
q_{\phi_1} (\mathbf{z}_p | X_P, A_{P,D}) = \prod_{p \in P} q_{\phi_1} (\mathbf{z}_p | X_P, A_{P,D})
\]

(1)

with \( q_{\phi_1} (\mathbf{z}_p | X_P, A_{P,D}) \) assumed to be \( N(\mathbf{z}_p | \mu_p, \text{diag}(\sigma^2_p)) \). Here, \( \mu_p \) and \( \sigma^2_p \) are the mean and variance for paper \( p \), obtained by sending \( h_p = \text{GraphSAGE}(X_P, A_{P,D}) \) separately to a one-layer neural network with activation function \( \text{ReLU}(x) = \max(0, x) \),

\[
\mu_p = \text{ReLU}(W h_p + b), \quad \sigma^2_p = \exp(\text{ReLU}(W h_p + b))
\]

(2)

In our setting, \( h_p \) is the GraphSAGE embedding of paper \( p \) after \( k \) iterations (default set as 2):

(3)

where \( N(p) \) is a set of neighbors of paper node \( p \). We set \( \text{Combine} \) function to be \( \text{sum} \) and \( \text{Aggregation} \) function to be \( \text{mean} \), following [26].

Similarly, by assuming the independence of latent variable \( \mathbf{z}_d \) among datasets,

\[
q_{\phi_2} (\mathbf{z}_d | H^T_{P,D}) = \prod_{d \in D} q_{\phi_2} (\mathbf{z}_d | H^T_{P,D})
\]

(4)

with \( q_{\phi_2} (\mathbf{z}_d | H^T_{P,D}) \) assumed to be \( N(\mathbf{z}_d | \mu_d, \text{diag}(\sigma^2_d)) \).
two separate one-layer neural networks to obtain $\mu_d$ and $\sigma_d^2$ respectively. Thus, $\mu_d$ and $\sigma_d^2$ share the parameters of the two-layer MLP encoder, and then are produced by two different networks with ReLU activation function.

$$h_d = \text{MLP}(H_p^T, D)$$
$$\mu_d = \text{ReLU}(W h_d + b), \quad \sigma_d^2 = \exp(\text{ReLU}(W h_d + b)) \tag{5}$$

**Decoder setting:** The two generative networks (decoders) are parametrised by $\theta_1$ and $\theta_2$ for modeling the generation (reconstruction) of $A_{p, p}$ and $H_{p, p}$, by $p_\theta(A_{p, p}|Z_p)$ and $p_\theta(H_{p, p}|Z_p, \tilde{Z}_p)$, shown in Figure 3(b). Due to the conditional independence of $A_{p, p}$ and $H_{p, p}$ given $Z_p$, we have

$$p_\theta(A_{p, p}, H_{p, p}|Z_p, \tilde{Z}_p) = p_\theta(A_{p, p}|Z_p) p_\theta(H_{p, p}|Z_p, \tilde{Z}_p) \tag{6}$$

We set the generative model by an inner product between latent variables:

$$p_\theta(A_{p, p}|Z_p) = \prod_{p_i \in P} \prod_{j \in P} p_\theta(A_{p_i, p_j}|z_{p_i}, z_{p_j}) \tag{7}$$

with $p_\theta(A_{p_i, p_j}|z_{p_i}, z_{p_j}) = \sigma(z_{p_i} \cdot z_{p_j})$ where $\sigma(.)$ is the logistic sigmoid function. And,

$$p_\theta(H_{p, p}|Z_p, \tilde{Z}_p) = \prod_{p_i \in P} \prod_{d \in D} p_\theta(H_{p, d}|z_{p_i}, \tilde{z}_d) \tag{8}$$

with $p_\theta(H_{p, d}|z_{p_i}, \tilde{z}_d) = \sigma(\tilde{z}_d \cdot \tilde{z}_d)$ where $\sigma(.)$ is the same logistic sigmoid function. The intuition here is for one pair of paper and dataset node in $H_{p, p}$, their embeddings should be close if $H_{p, d} = 1$. Otherwise, their embeddings should separate them away since $H_{p, d} = 0$.

**Objective Function:** We define an objective function to maximize the likelihood of the observed adjacency matrix $A_{p, p}$ and bipartite graph $H_{p, D}$.

$$p(A_{p, p}, H_{p, D}) = \int_{Z_p} \int_{\tilde{Z}_D} p(A_{p, p}, H_{p, p}, \tilde{Z}_p, \tilde{Z}_D) d\tilde{Z}_p d\tilde{Z}_D$$

$$= \mathbb{E}_{q_\phi} \left[ \log p_\theta(A_{p, p}, H_{p, p}, \tilde{Z}_p, \tilde{Z}_D) \right] - \mathbb{E}_{q_\phi} \left[ \log q_\phi(\tilde{Z}_p, \tilde{Z}_D|A_{p, p}, H_{p, p}, X_p) \right] \tag{9}$$

By taking log on both left and right, and using Jensen’s inequality $f(E[x]) \geq E[f(x)]$ for the concave log function, we have the lowerbound of the log likelihood as

$$\log p(A_{p, p}, H_{p, p}, X_p) \geq \mathbb{E}_{q_\phi} \left[ \log p_\theta(A_{p, p}, H_{p, p}, \tilde{Z}_p, \tilde{Z}_D) \right] - \mathbb{E}_{q_\phi} \left[ \log q_\phi(\tilde{Z}_p, \tilde{Z}_D|A_{p, p}, H_{p, p}, X_p) \right] \tag{10}$$

Here $q_\phi(\tilde{Z}_p, \tilde{Z}_D|A_{p, p}, H_{p, p}, X_p)$ is a variational distribution that approximates the true posterior distribution (inference network) $p(\tilde{Z}_p, \tilde{Z}_D|A_{p, p}, H_{p, p}, X_p)$ in Variational Bayes, and $\phi$ is a set of the variational parameters learnt when approximating $p_\theta$ with $q_\phi$.

Given the inference model in Figure 3(a), the variational distribution $q_\phi$ can be factorized as:

$$q_\phi(\tilde{Z}_p, \tilde{Z}_D|A_{p, p}, H_{p, p}, X_p) = \prod_{p \in P} q_{\phi_1}(\tilde{z}_p|A_{p, p}, X_p) \prod_{d \in D} q_{\phi_2}(\tilde{z}_d|H_p^T, D) \tag{11}$$

where $\phi_1$ and $\phi_2$ are the variational parameters learnt for approximating $p(\tilde{Z}_p)$ and $p(\tilde{Z}_D)$, respectively.

Having the generative model in Figure 3(b), we can write $p_\theta(A_{p, p}, H_{p, p}, \tilde{Z}_p, \tilde{Z}_D)$ as:

$$p_\theta(A_{p, p}, H_{p, p}, \tilde{Z}_p, \tilde{Z}_D) = p_\theta(A_{p, p}|Z_p) p_\theta(H_{p, p}|Z_p, \tilde{Z}_p) p(\tilde{Z}_p) p(\tilde{Z}_D)$$

$$= \prod_{p_i \in P} p_\theta(A_{p_i, p_j}|z_{p_i}, z_{p_j}) \prod_{d \in D} p_\theta(H_{p, d}|z_{p_i}, \tilde{z}_d) p(\tilde{Z}_p) p(\tilde{Z}_D) \tag{12}$$

Plugging (11) and (12) into (10), we get

$$\mathbb{E}_{q_\phi} \left[ \sum_{p \in P} \log p_\theta(A_{p, p}|Z_p) + \sum_{d \in D} \log p_\theta(H_{p, d}|Z_p, \tilde{z}_d) \right]$$

$$- \mathbb{KL}[q_\phi(\tilde{Z}_p|A_{p, p}, X_p)||p(\tilde{Z}_p)] - \mathbb{KL}[q_\phi(\tilde{Z}_D|H_p^T, D)||p(\tilde{Z}_D)]$$

$$\geq \mathcal{L}(\theta_1, \theta_2, \phi_1, \phi_2; A_{p, p}, H_{p, p}, X_p) \tag{13}$$

where $\mathcal{L}(\theta_1, \theta_2, \phi_1, \phi_2; A_{p, p}, H_{p, p}, X_p)$ is the variational lower bound also known as evidence lower bound (ELBO) on the marginal likelihood of the observed variables $(A_{p, p}, H_{p, p})$, and $\mathbb{KL}(Q||P)$ is KL-divergence between two distributions $Q$ and $P$. We then address the problem of maximizing ELBO.

**Optimization:** The first and second term in Eq. (13), can be approximated by classic Monte Carlo approximation through random sampling $\tilde{Z}_p \approx q_{\phi_1}(\tilde{z}_p|A_{p, p}, X_p)$, and $\tilde{Z}_D \approx q_{\phi_2}(\tilde{z}_D|H_p^T, D)$ respectively, while the KL has convenient analytical solutions. With sampling involved for nodes $\tilde{Z}_p$ and $\tilde{Z}_D$, the gradients do not propagate through $q_{\phi_1}(\tilde{Z}_p)$ and $q_{\phi_2}(\tilde{Z}_D)$. Therefore, Kingma and Welling [35] proposed reparameterization trick (RT) using a differentiable transformation. Instead of sampling $\tilde{z}_p$ from $N(\mu_p, \sigma_p^2)$ directly, we sample $\epsilon_p$ from a fixed distribution $\epsilon \sim N(0, I)$ and construct $\tilde{z}_p = \mu_p + \sigma_p^2 \epsilon_p$. Similarly, $\tilde{z}_d = \mu_d + \sigma_d^2 \epsilon_d.$
B. Inference of Query $Q$ (Step 2)

We infer the representation of query $Q = \{p_i\}_{i=1, \ldots, \vert Q \vert}$ as a Gaussian distribution $\tilde{z}_Q \sim \mathcal{N}(\mu_Q, \Sigma_Q)$, where:

$$
\mu_Q = \text{MEAN}(\mu_p), \quad \Sigma_Q = \text{MEAN}(\Sigma_p)^2 \quad \forall p \in Q
$$

(14)

where $\text{MEAN}(\cdot)$ is the element-wise mean of paper vectors in $\{\tilde{z}_p \sim \mathcal{N}(\mu_p, \Sigma_p)^2, \forall p \in Q\}$. This operation is inspired by the inductive GraphSAGE model. A query can be considered as a new node, which is connected to already existing nodes in the graph. The $\mu$ function used here acts like the $\text{Aggregate}$ function in Eq. (3) for aggregating the neighboring nodes. $\tilde{z}_Q$ thus encodes the links between paper nodes (if any) in the query, and thus strengthens the representation of topic if repeated in many papers in the query.

C. Dataset Recommendation for Query $Q$ (Step 3)

Now with the learned representation vector for datasets $\tilde{z}_d$ and for the query $\tilde{z}_Q$, we can evaluate the relevance of dataset candidates to the query. Since the representations are learnt in form of Gaussian distribution, we use KL divergence to measure their relevance:

$$
s(Q, d) = \text{KL}(\tilde{z}_Q || \tilde{z}_d) = \log \frac{\Sigma_d}{\sigma_Q} + \frac{\Sigma_Q}{2\sigma_d} + \frac{(\mu_Q - \mu_d)^2}{2\sigma_d} - \frac{1}{2}
$$

(15)

The final recommendation is made by selecting the top-$k$ datasets, ranked according to the values $s(Q, d)$, from the smallest (most relevant) to the largest (least relevant).

IV. Experiment Setup

A. Evaluation Data Collection

We collect academic network data from Delve system [38]. We remove all isolated papers that have no links to any other papers or datasets. After preprocessing, 8519 papers and 5610 datasets form the evaluation dataset in our study.

Training data: We used 8093 papers published from 1991 to 2014, and 5610 datasets referenced for training the representation models (i.e., the paper and dataset representations). This heterogeneous network has 360K citations among papers and 14K bipartite links between papers and datasets.

Query $Q$: For evaluating dataset recommendation results, we define a query corresponding to each of the 426 papers (published in 2015-2016) that were not used for training. The set of papers in one query are selected from the reference list of one paper if they are available in our preprocessed academic network. Thus, we can form 426 queries, with ground-truth datasets that were selected by authors in the corresponding paper, confirming the strong relevance of these used datasets to the query (reference papers).

B. Baselines and Evaluation Metrics

We compare our approach with the following baseline methods that span different categories as shown in Table II. These baselines obtain paper and dataset representations in different ways, most of which are single vectors, rather than distributions. We thus calculate the paper-dataset relevance score by inner product of their vectors. For those representations with a distribution, we also measure the relevance by KL-divergence. The score of one dataset to a query is obtained by average aggregation of all papers in $Q$, similar to the setting in our model. The baselines include:

- A naive way retrieves all datasets that are directly linked with papers $p_i$ in $Q$, and then ranks them by their degree in $H_{P,D}$ (Naive Retrieval 1). Another naive way extends $Q$ by including the first order neighbors of $p_i$ in $Q$, and then retrieves all datasets linked with papers in the extended $Q$ (Naive Retrieval 2).
- Pointwise Regression-lMF(implicit Matrix Factorization) [39] is a point-wise approach for collaborative filtering. It takes bipartite graph $H_{P,D}$ as input and produces paper and dataset latent factor vectors $\tilde{z}_p, \tilde{z}_d \in \mathbb{R}^L$ for competing with our learned vectors.
- Bayesian Pairwise Ranking (BPR) [40] forms an objective function targeting on ranking positive samples (observed paper-dataset pairs linked in $H_{P,D}$) higher than negative samples (unobserved pairs). It can be used as a standalone method to learn $\tilde{z}_p, \tilde{z}_d \in \mathbb{R}^L$ that minimizes the ranking loss, and can also be used as an optimization method by taking pretrained representation vectors from other models and maximize the likelihood of paper-dataset pairs linked in $H_{P,D}$.
- DeepWalk (DW) [7] is designed for node embedding of homogeneous graph and used here by treating our heterogeneous graph as a homogeneous one (for evaluating the importance of learning paper and dataset regarding their own nature).
- node2vec (n2v) [8] is an extension of DeepWalk, and used similarly here as DeepWalk.
- metaPath2vec [13] learns representations of heterogeneous nodes by sampling random walks transitioning between different type of nodes, thus preserving the heterogeneous neighborhood of a node. It works on $H_{P,D} + A_{P,P}$, but does not integrate $X_P$.
- Network Embeddings + Latent Semantic Indexing (LSI). Since DW [7], n2v [8], and metapath2vec[13] only leverage structural information of networks; for a fair comparison, we concatenate content vectors of paper (after applying LSI) to the vectors learned by DW, n2v and metapath2vec as new baselines.
- Network Embeddings + doc2vec is similar to the previous baseline by just replacing LSI by doc2vec [41], which learns vector representations for variable length texts such as sentences and documents.
- TADW [10] is an attributed homogeneous graph embedding model, which takes inputs of all $H_{P,D} + A_{P,P} + X_P$. However, it doesn’t distinguish paper and dataset nodes.
we set parameters value of maximum node degree to 134 for paper-dataset pairwise ranking. Methods including node2vec, doc2vec, metapath2vec, and paper. For the unsupervised network embedding baselines abstract (if available), and keywords (if available) for each the same paper, and PPD walk represents the dataset used the same paper, PD walk represents two papers sharing the same paper, and window size=5. For node2vec sampling strategy, we set parameters \( p \) and window size=5. For node2vec, we used paper-paper (PP) walk, paper-dataset (PD) walk, paper-paper (PP) walk, and paper-paper-dataset (PPD) walk, and choose window size={5, 5, 5, 7}. PP walk represents two papers citing the same paper; PD walk represents two papers sharing the same dataset; DP walks represent two datasets being referred by the same paper, and PPD walk represents the dataset used by the cited paper. For doc2vec and LSI, we used the title, abstract (if available), and keywords (if available) for each paper. For the unsupervised network embedding baselines methods including node2vec, doc2vec, metapath2vec, and TADW, for a fair comparison with our model, we use BPR as the objective function to improve these embeddings w.r.t. paper-dataset pairwise ranking.

For our method using GraphSAGE as paper encoder, we set parameters value of maximum node degree to 134, number of nodes sampled in layer 1 and layer 2 to 81 and 22 respectively. For initialization of weight matrices, we applied Glorot initialization strategy. The whole training process lasts 110 epochs for our model, using Adam optimizer (Kingma and Ba 2014) with learning rate of \( 10^{-4} \), \( L_2 \) regularization for all the parameters with \( \lambda = 10^{-3} \) and early stopping to accelerate the training process and to prevent overfitting.

V. Experimental Results

A. Dataset Recommendation

The evaluation results are reported in Table II. We summarize the observations as follows.

1) HVGAE outperforms the baselines, which shows that the learned embeddings are useful for query based dataset prediction, by considering both the heterogeneity and attributes, and learning embeddings as distributions.

2) CDL, VGAE and CAN are the second best shown by Precision, Recall and AUC. However, these methods perform second best on ranking relevant datasets as shown by NDCG and MRR.

3) BPR can help on improving the recommendation results, although it alone is not a good solution. However, we observe that initializing embeddings with pre-trained network embeddings and then optimizing using BPR helps to achieve better performance (e.g., TADW+BPR). We do not employ BPR as loss function for auto-encoders based methods as their embedding process is already guided w.r.t. reconstruction of edges.

4) Simple concatenating network embedding with paper content has limited improvement, e.g., combining doc2vec or LSI with network embedding vectors. Therefore, the way of integrating paper content in heterogeneous graph has significant influence on the recommendation.

5) Naive methods perform worse than other representation-based methods, showing the necessity of measuring paper-dataset relevance in a semantic space.

B. Paper-Dataset Representation Visualization

To further analyze the paper and dataset representation learned by HVGAE, we visualize them in 2-D space with t-SNE in Fig. 4. We observe the following:

1) Similar papers are mapped close to each other. For example, paper p12 and p4183 are close, and they both study data privacy. Another close pair p75 and p42 are about Transfer learning with regularization.

2) Datasets used under similar research topics are mapped close. For example, d3, d5 and d29 are close, and they were commonly used for evaluation and applications in SVM, semi-supervised learning, decision trees. Datasets d4913, d4914, d4915, d4916 and d3579 form a cluster, and have commonly been used for text-analysis, latent topics, classification and clustering.

3) Papers and their used datasets are mapped close, supporting our decoding stage: reconstructing the bipartite network \( H_{P,D} \) from \( \hat{Z}_P \) and \( \hat{Z}_D \). For example, paper p7309 and its used dataset d8 are close. Paper p69 and its used dataset d17 are close.
Table II
BENCHMARKING DATASET RECOMMENDATION

<table>
<thead>
<tr>
<th>Information Source</th>
<th>Algorithms</th>
<th>L</th>
<th>Pre@5</th>
<th>Recall@5</th>
<th>NDCG@5</th>
<th>MRR@5</th>
<th>AUC@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{p,d}$</td>
<td>Naive Retrieval 1</td>
<td>NA</td>
<td>0.0761</td>
<td>0.3803</td>
<td>0.3384</td>
<td>0.3264</td>
<td>0.6986</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>Naive Retrieval 2</td>
<td>NA</td>
<td>0.1179</td>
<td>0.5896</td>
<td>0.4659</td>
<td>0.4277</td>
<td>0.8293</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>PR-3MF</td>
<td>256</td>
<td>0.0793</td>
<td>0.3967</td>
<td>0.3659</td>
<td>0.357</td>
<td>0.7041</td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>128</td>
<td>0.1214</td>
<td>0.6072</td>
<td>0.4598</td>
<td>0.4133</td>
<td>0.8999</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>DeepWalk</td>
<td>128</td>
<td>0.0033</td>
<td>0.0164</td>
<td>0.0085</td>
<td>0.006</td>
<td>0.7913</td>
</tr>
<tr>
<td></td>
<td>node2vec</td>
<td>128</td>
<td>0.1714</td>
<td>0.8568</td>
<td>0.5912</td>
<td>0.5118</td>
<td>0.9722</td>
</tr>
<tr>
<td></td>
<td>metapath2vec</td>
<td>128</td>
<td>0.1742</td>
<td>0.8709</td>
<td>0.5870</td>
<td>0.4914</td>
<td>0.9765</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>DeepWalk+BPR</td>
<td>64</td>
<td>0.1476</td>
<td>0.7379</td>
<td>0.5377</td>
<td>0.4626</td>
<td>0.9673</td>
</tr>
<tr>
<td></td>
<td>node2vec+BPR</td>
<td>128</td>
<td>0.0793</td>
<td>0.3967</td>
<td>0.3659</td>
<td>0.357</td>
<td>0.7041</td>
</tr>
<tr>
<td></td>
<td>metapath2vec+BPR</td>
<td>128</td>
<td>0.1714</td>
<td>0.8568</td>
<td>0.5912</td>
<td>0.5118</td>
<td>0.9722</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>DeepWalk+LSI+BPR</td>
<td>256</td>
<td>0.1700</td>
<td>0.8498</td>
<td>0.5978</td>
<td>0.5163</td>
<td>0.9720</td>
</tr>
<tr>
<td></td>
<td>node2vec+LSI+BPR</td>
<td>256</td>
<td>0.1751</td>
<td>0.8754</td>
<td>0.6155</td>
<td>0.5284</td>
<td>0.9741</td>
</tr>
<tr>
<td></td>
<td>Node2vec+Doc2vec+BPR</td>
<td>256</td>
<td>0.1566</td>
<td>0.7829</td>
<td>0.5816</td>
<td>0.4910</td>
<td>0.9775</td>
</tr>
<tr>
<td></td>
<td>node2vec+Doc2vec+BPR</td>
<td>256</td>
<td>0.1735</td>
<td>0.8674</td>
<td>0.6289</td>
<td>0.5382</td>
<td>0.9745</td>
</tr>
<tr>
<td></td>
<td>MetaPath2vec+LSI+BPR</td>
<td>256</td>
<td>0.1340</td>
<td>0.6098</td>
<td>0.4978</td>
<td>0.4433</td>
<td>0.9605</td>
</tr>
<tr>
<td></td>
<td>MetaPath2vec+Doc2vec+BPR</td>
<td>256</td>
<td>0.1513</td>
<td>0.7567</td>
<td>0.5512</td>
<td>0.4862</td>
<td>0.9699</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>TADW +BPR</td>
<td>16</td>
<td>0.1814</td>
<td>0.9092</td>
<td>0.5523</td>
<td>0.4869</td>
<td>0.9635</td>
</tr>
<tr>
<td></td>
<td>CVAE-CF</td>
<td>16</td>
<td>0.0934</td>
<td>0.4672</td>
<td>0.3572</td>
<td>0.3214</td>
<td>0.9100</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>DCDL</td>
<td>256</td>
<td>0.1790</td>
<td>0.8498</td>
<td>0.5978</td>
<td>0.5163</td>
<td>0.9720</td>
</tr>
<tr>
<td></td>
<td>CV AE-CF</td>
<td>256</td>
<td>0.1751</td>
<td>0.8754</td>
<td>0.6155</td>
<td>0.5284</td>
<td>0.9741</td>
</tr>
<tr>
<td></td>
<td>Node2vec+LSI+BPR</td>
<td>256</td>
<td>0.1566</td>
<td>0.7829</td>
<td>0.5816</td>
<td>0.4910</td>
<td>0.9775</td>
</tr>
<tr>
<td></td>
<td>Node2vec+Doc2vec+BPR</td>
<td>256</td>
<td>0.1735</td>
<td>0.8674</td>
<td>0.6289</td>
<td>0.5382</td>
<td>0.9745</td>
</tr>
<tr>
<td></td>
<td>MetaPath2vec+LSI+BPR</td>
<td>256</td>
<td>0.1340</td>
<td>0.6098</td>
<td>0.4978</td>
<td>0.4433</td>
<td>0.9605</td>
</tr>
<tr>
<td></td>
<td>MetaPath2vec+Doc2vec+BPR</td>
<td>256</td>
<td>0.1513</td>
<td>0.7567</td>
<td>0.5512</td>
<td>0.4862</td>
<td>0.9699</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>TADW +BPR</td>
<td>512</td>
<td>0.1536</td>
<td>0.7680</td>
<td>0.6473</td>
<td>0.6087</td>
<td>0.9607</td>
</tr>
<tr>
<td></td>
<td>GAE</td>
<td>16</td>
<td>0.1794</td>
<td>0.8993</td>
<td>0.5478</td>
<td>0.4681</td>
<td>0.9772</td>
</tr>
<tr>
<td></td>
<td>VGAE</td>
<td>16</td>
<td>0.1796</td>
<td>0.8995</td>
<td>0.5653</td>
<td>0.4876</td>
<td>0.9789</td>
</tr>
<tr>
<td></td>
<td>CAN</td>
<td>16</td>
<td>0.1793</td>
<td>0.8967</td>
<td>0.5754</td>
<td>0.4908</td>
<td>0.9792</td>
</tr>
<tr>
<td>$H_{p,d}$ + $A_{p,p}$</td>
<td>Our Method (HVGAE)</td>
<td>16</td>
<td>0.1845</td>
<td>0.9225</td>
<td>0.7153</td>
<td>0.6536</td>
<td>0.9876</td>
</tr>
</tbody>
</table>

Figure 4. Visualization of the learned paper (blue) and dataset (red) embeddings (better viewed on pdf by zooming in).

Figure 5. Impact of the number of papers specified in a query.

C. Impact of Query Length

The 426 queries we used for evaluation are from 426 papers (published in 2015-2016) that were not seen for training. These queries can be grouped into, Group 1 with length 1-3 (317 queries), Group 2 with length 4-6 (86 queries), and Group 3 with length > 6 (23 queries). We study the sensitivity of HVGAE and three baselines to the number of papers specified in a query. Figure 5 shows:

1) consistently HVGAE has the best performance in all groups with different lengths w.r.t. all metrics.
2) generally, when more paper are specified in the query, more accurate recommendation can be made. That is naturally because a better query representation can be determined when the query is specified with more interested papers.
3) content and structure augmented methods (HVGAE and CAN) perform better than structure only models (node2vec+BPR, and bipartite BPR).

D. Case Studies

Case Study 1: Naive Retrieval v.s. HVGAE

We take an example query $Q = \{ p_{84}, p_{207}, p_{433}, p_{5729} \}$, $\{ p_{84}, p_{207} \}$ are about sequential pattern mining using prefix span, or bitmap, while $\{ p_{433}, p_{5729} \}$ study clustering in high dimensional space and deep-order preserving submatrix problem. From Table III we see, Naive Retrieval 1 recommends $\{ d_{31}, d_{32}, d_{33} \}$ which are relevant only for sequential pattern mining, $d_{326}$ is a drug activity data used by $p_{433}$ and $d_{3876}$ is from $p_{5729}$. However, none of them is a correct hitting. HVGAE is able to learn the semantics of query from the joint semantics of papers, which is submatrix problem based on sequential pattern mining, and returns the top-5 ranked datasets that are all relevant to clustering, feature-selection, classification and matrix decomposition.

To verify our results, we checked from which paper this query was generated and what is the dataset $d_{29}$. The paper
is a new approach for the deep order preserving submatrix problem based on sequential pattern mining and the dataset is gene’ dataset commonly used for clustering and margin-analysis performance.

Similarly, in query $Q = \{p7187, p7214\}$, $p7187$ studies object detection and semantic segmentation, and $p7214$ is about deep features for text spotting. Naive Retrieval 1 is not able to find any relevant datasets while HVGA recommends datasets $\{d0, d42, d11, d727, d168\}$ which are all about finding text in images, object recognition.

Note that the ground truth datasets here are those used in one new paper from which a query is defined. It is possible that several different new papers cite the same queried papers. That is saying, one query may have several sets of ground truth datasets, each corresponding to a new paper that includes queried papers in its reference list. Although we cannot deny the interestingness of the recommended datasets if they are different from the ground truth, we can still evaluate the recommendation model to see if a correct hitting is found in the result.

**Case Study 2: Other baselines vs. HVGA**

We send a query $Q = \{p51, p1938\}$ including two papers studying Rule Extraction from Support Vector Machines (SVM). The relevant datasets determined by authors who wrote a paper citing $\{p51, p1938\}$ are $\{D23 = Glass04v5, D292 = Lcd7digit, D88 = Sonar\}$. We observe in Table IV that HVGA has correct hitting on all the three datasets. However, the baseline methods have only maximum 2 correct recommendations.

**E. Component Analysis of HVGA**

HVGA learns $H_{P,D}$, $A_{P,D}$, $H_P$, and $X_P$. We present the results when HVGA taking different inputs in Table V. When considering $H_{P,D}$ only, our objective functions is reduced to

$$ \log p(H_{P,D}) \geq \mathbb{E}_{\theta_\phi} \left[ \sum_{d \in D} \log p_\theta(H_{P,d}|\tilde{Z}_P, \tilde{Z}_d) \right] -$$

Table III: Top-5 Dataset ranking by Naive Retrieval 1 and HVGA. The ground truth and correct hitting is highlighted in blue.

<table>
<thead>
<tr>
<th>Query</th>
<th>Ground Truth</th>
<th>Naive Technique (Top-5 ranked datasets)</th>
<th>HVGA (Top-5 ranked datasets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>${p84, p207, p433, p5729}$</td>
<td>${d29}$</td>
<td>${d31, d33, d32, d326, d3876}$</td>
<td>${d3, d29, d5, d4, d11}$</td>
</tr>
<tr>
<td>${p7187, p7214}$</td>
<td>${d0}$</td>
<td>${d5, d29, d3462, d3963, d5034}$</td>
<td>${d0, d42, d11, d727, d168}$</td>
</tr>
</tbody>
</table>

Table IV: Case Study 2: Answering $Q = \{p51, p1938\}$

<table>
<thead>
<tr>
<th>Methods</th>
<th>Top-5 ranked datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Retrieval 1</td>
<td>${d29, d57}$</td>
</tr>
<tr>
<td>m2v+bpr</td>
<td>${d1160, d2815, d29, d4335, d44}$</td>
</tr>
<tr>
<td>m2v+d2v+bpr</td>
<td>${d29, d922, d649, d1220, d0}$</td>
</tr>
<tr>
<td>TADW</td>
<td>${d135, d2000, d58, d1432, d7}$</td>
</tr>
<tr>
<td>VGA</td>
<td>${d29, d44, d88, d10, d5}$</td>
</tr>
<tr>
<td>CAN</td>
<td>${d292, d88, d13, d6, d31}$</td>
</tr>
<tr>
<td>Our Method</td>
<td>${d29, d88, d54, d292, d23}$</td>
</tr>
</tbody>
</table>

Table V: Component Analysis of HVGA

<table>
<thead>
<tr>
<th>Source</th>
<th>Pre@5</th>
<th>Recall@5</th>
<th>NDCG@5</th>
<th>MRR@5</th>
<th>AUC@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{P,D}$</td>
<td>0.1071</td>
<td>0.5472</td>
<td>0.5346</td>
<td>0.5249</td>
<td>0.9661</td>
</tr>
<tr>
<td>$A_{P,D}, H_P, X_P$</td>
<td>0.1828</td>
<td>0.9305</td>
<td>0.6258</td>
<td>0.5829</td>
<td>0.9694</td>
</tr>
<tr>
<td>$A_{P,D}, H_P, D, X_P$</td>
<td>0.1847</td>
<td>0.9225</td>
<td>0.7153</td>
<td>0.6556</td>
<td>0.9876</td>
</tr>
</tbody>
</table>

![Figure 6](image_url) Parameter sensitivity w.r.t. representation dimensions L.
VI. CONCLUSION AND FUTURE WORK

In this work, we tackled a novel problem of query-based dataset recommendation by learning paper and dataset representations jointly from paper-paper citation graph, paper-dataset bipartite graph and paper content. The proposed HVGAE model is able to effectively rank relevant datasets given a query referencing list of papers. An online system is under-development based on the proposed model to help researchers find more usable dataset resources than what they are aware of now. In future, we want to also consider semantics of citations between paper nodes and to present an explainable recommendation model.

ACKNOWLEDGMENT

This work is supported by King Abdullah University of Science and Technology (KAUST), Saudi Arabia.

REFERENCES