Mining top-k Popular Datasets via a Deep Generative Model

Uchenna Akujuobi
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia
Email:uchenna.akujuobi@kaust.edu.sa

Ke Sun
Data61, CSIRO
Sydney, Australia
Email: sunk@ieee.org

Xiangliang Zhang
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia
Email:xiangliang.zhang@kaust.edu.sa

***Abstract*—Finding popular datasets to work on is essential for data-driven research domains. In this paper, we focus on the problem of extracting top-k popular datasets that have been used in data mining, machine learning, and artificial intelligence fields. We solve this problem on an attributed citation network, which includes node content information (text of published papers) and paper citation relations. By formulating the problem as a semi-supervised multi-label classification one, we develop an efficient deep generative model for learning from both the document content and citation relations. The evaluation on a real-world dataset shows that our proposed model outperforms baseline methods. We then apply the model further to reveal the top-k frequently cited datasets in selected areas and report interesting findings.**

***Keywords*-Deep Generative Models, Semi-supervised Learning, Document Classification, Multi-label Classification, Citation Network**

I. Introduction

In this paper, we target on addressing a real application problem: *what are the top-k popular datasets used for evaluation in a given research field*? The knowledge of the top-k popular dataset used in any given research field provides a better understanding of the popular datasets used in that field; which will provide more insights on the topics and also, hints on datasets to look into when working on topics under/related to the field. Although there has been no prior research paper on the extraction of popular datasets for given topics given a citation network, making an internet search for "top datasets" yields a search result page with lots of blogs and write-ups of the top datasets used in research which is often based on personal opinion. However, this fails to show the use or usefulness of the reported datasets in different fields of research. This paper is based on the analysis of academic papers and thus, provides a narrower and more realistic report.

We formalize this practical problem as a *semi-supervised multi-label learning in attributed graph* problem. We use the available resources in a data-driven search engine called Delve[1], which provides what datasets were used for evaluation in more than 2 million papers including those published at prestigious venues in the broad area covering data mining, machine learning, computer vision, and others [1]. To aggregate the evaluation datasets in a given research field, we are left with the problem to find out the published papers in this field. Due to the overlapping nature of research areas, one academic paper usually can be labeled with multiple topic tags. For example, a paper can fit into both *graph mining* and *neural networks* subfield category if it applies neural networks to graph problems. The labels are mainly determined by the paper content and supplemented by the related papers that it cited in the reference and those that are citing it (utilizing the additional topological information has been observed to lead to better document classification models [13] [30]). Therefore, we have a *multi-label classification problem to address given the paper content and its citation relations (as known as attributed citation graph)*. With the help of partially available label information, a *semi-supervised learning* approach is desired, to provide more accurate classification results than unsupervised ways.

To address our semi-supervised multi-label learning problem on attributed citation graph, we need effective new solutions. There exist some approaches for semi-supervised multi-label classification that can be directly applied on paper content [10], [14], [23]. However, citation graph information cannot be easily adopted in them. A potential solution is to employ attributed graph embedding that represents each node by a low-dimensional vector [3], [18], and then apply semi-supervised multi-label classification on the embedding vectors. However, the weakness is the independent process of embedding and semi-supervised learning, which limits classification accuracy. One can also apply semi-supervised attributed graph embedding [8], [13], [30] to use the partially available labels to guide the attributed graph embedding. However, these approaches work in multi-class scenarios, rather than multi-label cases. Last but not least, some of these approaches (e.g., [13]) do not scale well to our problem as it is only scalable to the number of edges.

In this paper, we investigate the use of deep generative models (DGM) for solving our problem because of its scalable and expressive nature, which allows for more complex latent distributions to be learned. In order to learn from both the text and graph information, the frequently used naive approach is the concatenation of both input features. How-

[1]https://delve.kaust.edu.sa

ever, this limits the expressive nature of DGM especially when the inputs are generated by different distributions (e.g., Gaussian and Bernoulli). We increase the flexibility of deep generative models by modeling two input layers, to better capture the intrinsic information from the graph topology and node content information. The unified model is trained from end to end and produces accurate labels for documents in different domains, which allows us to aggregate the documents and their used evaluation datasets further, and ultimately report the top-k popular datasets in different domains.

Our contributions in this work are summarized as follows:

- We propose a deep generative model for the semi-supervised multi-label learning problem in attributed graphs.
- We validate the proposed model on the real-world attributed citation graph in Delve system and show that it outperforms the state-of-the-art approaches.
- We classify 886,109 documents and extract the top-k popular dataset resources in 20 subfields.

II. Related Works

A. Attributed Graph Mining

Attributed graphs are graphs in which nodes are associated with attributes (in our case text). Many real-world network data exist in attributed form. For this reason, there has been a rise in the demand and development of efficient algorithms that can handle attributed graphs [4], [29], [33]. A citation network is said to be attributed if its nodes and/or edges bear some additional information like the document texts or citation contexts. Some recent works, including Planetoid [30] and Graph Convolutional networks (GCN) [13] etc., experimented on attributed citation networks, where each document in the citation network has text information. Based on early works on graph embeddings (see for example [24]), Yang et al. [30] introduced node labels to obtain a semi-supervised embedding and applied a feed-forward neural network to extend to an inductive setting. Kipf and Welling [13] took a first-order approximation of the graph spectral convolution [6] for semi-supervised node classification and obtained state-of-the-art performance. The proposed GCN has a computational complexity scaling with the number of edges and is limited to transductive learning. These limitations are tackled through sampling methods [8]. In this work, we build a semi-supervised model that can deal with moderately dense graphs with millions of nodes, without sophisticated graph convolutions. In contrast, our model is simple to implement: we first learn a traditional network embedding [7], and then utilizing both labeled and unlabeled samples to fuse this embedding with node attributes which have text and multiple labels.

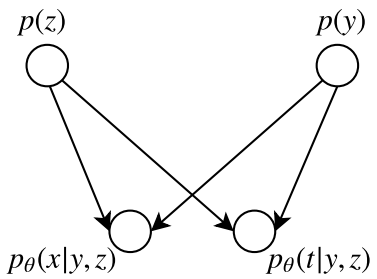
B. Deep Generative Models for Semi-supervised Learning

Deep generative models [5], [12], [20] are powerful deep neural network models to learn data distributions based on prior parametric assumptions. This framework was applied to a semi-supervised setting by Kingma et al. [12]. In a simplified scenario, an observed data sample x is assumed to be generated by $p(x \mid z)$ that is parameterized by a neural network, where z acts as a latent representation distributed according to some simple parameter-free distribution $p(z)$. By assuming a corresponding inference model given by $q(z \mid x)$ that approximates the posterior $p(z \mid x)$ and is parameterized by another neural network, all model parameters can be learned by variational inference. As noted by Maaloe et al. [20], the proposed model is not easy to be trained end-to-end with more than one layer of stochastic latent variables. Recent works such as the Ladder network [26] have improved on the performance with end-to-end training.

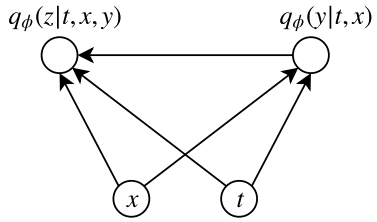
The auxiliary deep generative approach [20] enriches the flexibility of the generative model by adding a latent auxiliary variables a, so that the generative model is given by $p(x, z, a) = p(x \mid z, a)p(z, a)$. By assuming a parametric inference model $q(a, z \mid x)$, the marginal $q(z \mid x) = \int q(a, z \mid x)da$ can be a non-Gaussian distribution and therefore can fit better the true posterior $p(z \mid x)$. If the data is (partially) labeled, a discrete latent variable y can be introduced so that the generative model is specified as $p(x, y, z, a) = p(y)p(z)p(a \mid y, z)p(x \mid y, z, a)$. Our proposition is based on this auxiliary approach while specifically designed for graph data sets to incorporate both link structure information and node attribute information.

III. Deep Generative Model for Semi-Supervised Multi-label Document Classification in Attributed Graphs

We first formally define our problem. Given an attributed graph, e.g., a citation network, $G = \{V, E\}$, node set V is a set of documents including a small subset V_l having known labels, while the remaining documents V_u are unlabeled. Note that we focus on the multi-label problem, where each document can belong to more than one class, i.e., $y_i \in 2^\ell$, where ℓ is the number of classes. The set of edges E are citation links between the documents. Each document in the network contains text information such as document title, abstract, keywords, and full body text. With the intuition that connected documents with similar text contents are likely to share the same labels, a model $f(T, X)$ conditioned on both the topological structure X and the text information T is expected to capture the intrinsic correlations between the documents better, comparing to using only the topological structure X [13]. Our problem is then defined as: given a training set $S = \{(t_i, x_i, y_i) : 1 \leq i \leq |V_l|\}$, the goal is to produce a multi-label classifier that infers labels for V_u with minimized errors [31].



(a) Our generative model.



(b) Our inference model.

Figure 1: Probabilistic graphical model. The variables x and t are the graph topology and text inputs respectively, z is a latent variable, and y is the label variable. Each incoming arrow to the variables is a deep neural network with parameters θ and ϕ.

Due to the limited number of labeled samples, we model our classification problem using generative models to learn from both the labeled and unlabeled samples efficiently. Our graphical model is shown in Figure 1. The variables x and t are the graph topology and text inputs respectively, z is a latent variable, and y is the label variable. In this work, we obtain x of a node by applying node2vec [7], due to its superior performance on representing the graph topology information. In the generative model, both x and t jointly depend on latent variable z and label variable y. In the inference model, label variable y is determined by node topology x and node content t, while the variable z captures the intrinsic relationship among x, t and y.

By assumption, the generative model has a joint distribution $p(t, x, y, z) = p(y)p(z)p_\theta(x \mid y, z)p_\theta(t \mid y, z)$, with

$$\begin{aligned} p(y) &= \mathrm{Ber}(y \mid \gamma), \\ p(z) &= \mathcal{N}(z \mid 0, I), \\ p_\theta(x \mid y, z) &= \mathcal{N}\left(x \mid \mu_{\theta_1}(y, z),\, \mathrm{diag}(\sigma^2_{\theta_1}(y, z))\right), \\ p_\theta(t \mid y, z) &= \mathcal{N}\left(t \mid \mu_{\theta_2}(y, z),\, \mathrm{diag}(\sigma^2_{\theta_2}(y, z))\right), \end{aligned} \quad (1)$$

where y is a random binary vector of length ℓ, $\mathrm{Ber}(.\mid \gamma)$ is a multivariate Bernoulli distribution with independent random bits with activation probabilities specified by the vector γ so that $p(y_i = 1) = \gamma_i$ (that is the probability for the i'th bit activated). $p(z)$, $p(x \mid y, z)$ and $p(t \mid y, z)$ are all assumed to be Gaussian distributions, with $\mathcal{N}(\cdot \mid \mu, \Sigma)$ denoting a multivariate Gaussian distribution with mean μ and covariance matrix Σ.

To infer the latent parameters, the model is optimized by maximizing the lower bound on the likelihood of both labeled and unlabeled samples. For a labeled sample (t, x, y), its log likelihood

$$\log p_\theta(t, x, y) = \log \int_z p(t, x, y, z) dz$$

has the variational lower bound

$$\mathcal{L}(t, x, y) = \mathbb{E}_{q_\phi(z \mid t, x, y)} \left[\log \frac{p_\theta(t, x, y, z)}{q_\phi(z \mid t, x, y)} \right]. \quad (2)$$

For an unlabeled sample x and t, its log likelihood

$$\log p_\theta(t, x) = \log \int_y \int_z p(t, x, y, z) dz dy$$

has the lower bound

$$\mathcal{U}(x) = \mathbb{E}_{q_\phi(y, z \mid t, x)} \left[\log \frac{p_\theta(t, x, y, z)}{q_\phi(y, z \mid t, x)} \right], \quad (3)$$

where $q_\phi(y, z \mid t, x) = q_\phi(z \mid y, t, x) q_\phi(y \mid t, x)$, including

$$\begin{aligned} q_\phi(y \mid t, x) &= \mathrm{Ber}(y \mid \gamma_\phi(t, x)), \\ q_\phi(z \mid y, t, x) &= \mathcal{N}(z \mid \mu_\phi(t, y, x), \mathrm{diag}(\sigma^2_\phi(t, y, x))). \end{aligned} \quad (4)$$

The overall objective function is defined to maximize the lower bound and meanwhile minimize the classification error in labeled data. That is, to maximize

$$\begin{aligned} E = &\sum_{t_l, x_l, y_l} \mathcal{L}(t_l, x_l, y_l) + \sum_{t_u, x_u} \mathcal{U}(t_u, x_u) \\ &+ \beta \frac{N_l + N_u}{N_l} \sum_{t_l, x_l, y_l} \log q(y_l \mid t_l, x_l), \end{aligned} \quad (5)$$

where $\beta > 0$ is a hyper-parameter for weighting the classification error. N_l and N_u are the numbers of labeled and unlabeled samples respectively. More details of the derivations can be found in the appendix. The Adam optimizer is applied to minimize the cost function. The classifier in Eq. (4) $q_\phi(y \mid t, x)$ is then used to calculate probabilities for unlabeled (t_u, x_u) belonging to each class. $\gamma(t)$ and $\gamma(x)$ are activation vectors for multivariate Bernoulli distributions given t and x respectively. Although we focus on multi-label problems with continuous inputs, variables x, t, and y could be of any distribution. The computational complexity of the labeled and unlabeled cost functions in equations 2 and 3 is $\mathcal{O}(2^l)$, where $l = \dim(y)$. We assume l is small enough and tackling this complexity is left as a future work.

IV. Evaluation on Citation Graph in Delve

A. Delve Citation Graph

We first evaluate the effectiveness of the proposed model on classifying documents in Delve system. Note that this is the only publicly available attributed graph with nodes

explicitly given in multi-label setting. The widely used datasets for attributed academic document categorization are the Cora, Citeseer, and the Pubmed dataset [27]. These three datasets [2] are multi-class datasets (each paper having a single class) and compose of 2708, 3327, and 19717 papers, respectively. The full Cora[3] is a multi-label attributed graph dataset. However, the labels are organized in a hierarchical tree structure. For example, placing a paper X under *C++*, in turn places it under *programming*, and placed under *computer science*. In this case, paper X will have labels *C++, programming*, and *computer science*. Our application problem is to find popular evaluation datasets in given fields, which may have overlaps, but not in hierarchical structures. In some papers [19], [28], DBLP four area dataset [9] is used in a multi-label setting, by converting the keywords used in each author's publication or author's publication venues as the labels. However, since the aim is to extract information from papers in different topic categories, using a dataset like this might be redundant as publication venues often cut across several topic ranges.

At the moment of writing this paper, the Delve database is composed of more than 2 million scholarly publications from more than 1,000 different conferences and journals in various categories/domains. These scholarly publications are linked together to form a citation network. Table I shows the full graph statistics of the Delve citation network. For each document (node in the citation graph), the available content information includes *title, author(s), abstract, publication venue, keywords, and full body text* when present.

A document can belong to more than one category. The initial document labeling in the Delve citation graph was conducted as a crowd-sourced project where participants were asked to manually assign papers to one or more of 20 predefined categories gathered from the fields of machine learning, data mining, computer vision, and robotics. These categories were hand-picked and agreed by domain experts to represent trending topics in these fields. See Table II for the category list and the number of documents in each category. Since the documents are extracted from different conferences in these areas, they provide a diverse set of citation information and semantics. There are 4477 labeled documents in total. Each document on average has two labels (max/min no. of labels in these papers are six/one).

To prepare the dataset from Delve, a preprocessing step is applied in building the text features includes removal of stop words, converting letters to lower case, and stemming using the porter-stemming algorithm [25]. Then node text features t are extracted by applying the latent semantic analysis method on the document-term matrix features, resulting in features vectors of 300-dimension. Node topological features x are obtained by applying Node2Vec [7], which is a recently proposed skip-gram based graph embedding methods that map each node to a d-dimensional vector, to the full Delve citation graph (which is composed of over 2 million nodes). The parameters of Node2Vec are set as $p = 4$ and $q = 1$ to keep in line with the typical values used in Node2Vec [7]. We leave the default values of the other Node2Vec parameters of $d = 128, r = 10, l = 80, k = 10$, where d is the feature dimension, r is the number of walks per source, l is the length of walk per source, and k is the context size for optimization.

We train our model finally on $821,976$ papers after excluding papers that have no outlink in the Delve citation graph. We use the random hyper-parameter search [2] to determine the best latent layers dimension to use. All implementations are conducted in Python using Tensorflow libraries and run on GPU workstations.

[2]http://linqs.umiacs.umd.edu/projects//projects/lbc/

[3]https://people.cs.umass.edu/∼mccallum/data.html

Table I: Graph statistics of the Delve citation network

No. of nodes	2,116,429
No. of edges	9,434,474
No. of closed triangles	6,032,686
No. of open triangles	1,201,828,844
Fraction of closed triads	0.004995
Fraction of largest connected component	0.999662
Approximate full diameter	12

Table II: Mullti-label documents from Delve system

	Category name	Papers assigned
1	Information retrieval	922
2	Natural language processing	624
3	Clustering	301
4	Optimization methods	302
5	Gene and cancer (bioinformatics)	165
6	Tracking (computer vision)	478
7	Security and privacy	494
8	Time series	119
9	Graph mining & social network	295
10	Supervised learning	290
11	Feature selection & extraction	150
12	Rule learning	332
13	Semi-supervised & active learning	144
14	Agent systems (AI)	469
15	Recommendation	97
16	Unsupervised learning	83
17	Dimensionality reduction	58
18	Neural networks	164
19	Online learning	26
20	Multi-label classification	16

B. Baseline Methods to Compare

For comparison, we evaluated the performance of several supervised and semi-supervised algorithms on the Delve multi-label dataset, and here report the performance of two supervised and semi-supervised algorithms that gave the best result, in the binary relevance framework.

Supervised Methods: Linear SVM (LSVM) is an implementation of SVM using a linear kernel, and Gaussian Naive Bayes (GNB) is an extension of the Naive Bayes algorithm

commonly by assuming a Gaussian distribution.

Semi-Supervised Methods: Label propagation (LProp) [34] and Label Spreading (LSpread) [32] are semi-supervised algorithms where labels are propagated from labeled to unlabeled nodes. The main difference between the two algorithms is that Label propagation uses the graph Laplacian while Label spreading uses the normalized graph Laplacian in the design of the transition matrix.

C. Evaluation Metrics and Results

Various metrics can be used in evaluating multi-label classifier models. In our experiment, we measure the classifiers using the precision, recall, F1-scores, and subset accuracy. We show both the micro and macro averaging methods. The expressions of the performance metrics used in the experiments are

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}),$$

$$\text{Micro-Precision} = \sum_{i=1}^{L} \text{TP}_i / \sum_{i=1}^{L} (\text{TP}_i + \text{FP}_i),$$

$$\text{Macro-Precision} = \sum_{i=1}^{L} \text{Precision}_i / L,$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}),$$

$$\text{Micro-Recall} = \sum_{i=1}^{L} \text{TP}_i / \sum_{i=1}^{L} (\text{TP}_i + \text{FN}_i),$$

$$\text{Macro-Recall} = \sum_{i=1}^{L} \text{Recall}_i / L,$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

$$\text{Micro-F1} = 2 \times \frac{\text{Micro-Precision} \times \text{Micro-Recall}}{\text{Micro-Precision} + \text{Micro-Recall}},$$

$$\text{Macro-F1} = 2 \times \frac{\text{Macro-Precision} \times \text{Macro-Recall}}{\text{Macro-Precision} + \text{Macro-Recall}},$$

$$\text{Subset-Acc} = \frac{1}{N} \sum_{N}^{n=1} \mathbb{I}[y_n = t_n],$$

where TP, FP, FN are the true positives, false positives, and false negatives given a positive and negative class set respectively. TP_i, FP_i, FN_i are the true positives, false positives, and false negatives given a positive class i respectively. Precision_i and Recall_i are the precision and recall score for class i. N is the number of test samples, y_n and t_n are the predicted and target labels respectively. $\mathbb{I}$ is an identity function that outputs 1 only when the prediction matches the true subset exactly, and 0 otherwise.

We report the average F1-score obtained after 5-fold cross-validation. We compare it against several supervised methods adopted for multi-label learning; however, we report result from Gaussian Naive-Bayes [11] and LSVM - the baseline methods which gave the best results for the multi-label tasks. Table III shows the performance result obtained from the multi-label evaluation. Figure 2 shows the classification accuracy of each class independently when using Naive Bayes, LSVM and the proposed DGM. We also show the results obtained using the only the graph, text, and a concatenation of graph and text information as input to the model in table IV. It can be observed that our proposed model has the best performance over other baselines.

V. Application of Extracting Top-k Popular Datasets in 20 Fields

We then apply the proposed model to classify all unlabeled papers in the real multi-label Delve dataset, and then we extract and report top-k popular data sources from the documents in each class. The general algorithm is provided in Algorithm 1.

Algorithm 1: Full algorithm for top-k extraction

Input: Citation graph, document text embeddings, and a set of datasets used by each documents
Output: Top-k datasets mentioned in papers in each class ranked according to number of citation

```
Initialization;
Feed graph and text embeddings to the DGM architecture;
Train DGM using learned hyperparameters from CV;
Predict class labels of unlabeled samples;
foreach class label c ∈ C do
    Select papers assigned to class c and extract the datasets;
    Rank extracted datasets by citation count;
    Select top-k dataset
end
```

For this task, we train our DGM models with the full $821,976$ papers (i.e., 4477 labeled and $817,499$ unlabeled set), using the best configurations gotten from the multi-labeled document experiment (see section IV-A) for each class. Then using the trained model, we predict the classes of the papers. From the output, we extract the URLs mentioned in the papers assigned to each class. We manually analyze the obtained URLs; selecting the valid dataset related resources in each class. Due to space constraints, we picked the classes with F1-Score 0.79 or higher and show the top ten datasets resources in table V.

Analyzing the results obtained for the *Agent* class, we observe that the top results compose of modeling and simulation tools. We attribute this to the fact that the verification and validation of AI agent systems are more complicated than the traditional evaluation method of giving a dataset as input and testing against an expected value [21]. Thus, we conclude that in this research area, the evaluations are

Table III: Result summary of the multi-label evaluations on the Delve dataset

Methods	Recall		Precision		F1-score		Subset accuracy
	Macro	Micro	Macro	Micro	Macro	Micro	
GNB	0.61	0.68	0.42	0.44	0.48	0.53	0.26
LSVM	0.43	0.54	0.66	0.73	0.51	0.62	0.44
LProp	0.40	0.50	0.43	0.52	0.41	0.51	0.38
LSpread	0.40	0.50	0.44	0.53	0.41	0.51	0.38
DGM (BR)	0.51	0.6	0.56	0.62	0.52	0.61	0.42
DGM	0.45	0.55	0.62	0.71	**0.52**	**0.62**	**0.48**

Table IV: Result summary showing the performance of different inputs to the DGM Model

Methods	Recall		Precision		F1-score		Subset accuracy
	Macro	Micro	Macro	Micro	Macro	Micro	
Graph	0.19	0.30	0.52	0.77	0.26	0.43	0.29
Text	0.41	0.50	0.64	0.73	0.49	0.59	0.43
Graph and Text (concat)	0.41	0.51	0.67	0.75	0.49	0.61	0.44
Graph and Text (seperate)	0.45	0.55	0.62	0.71	**0.52**	**0.62**	**0.48**

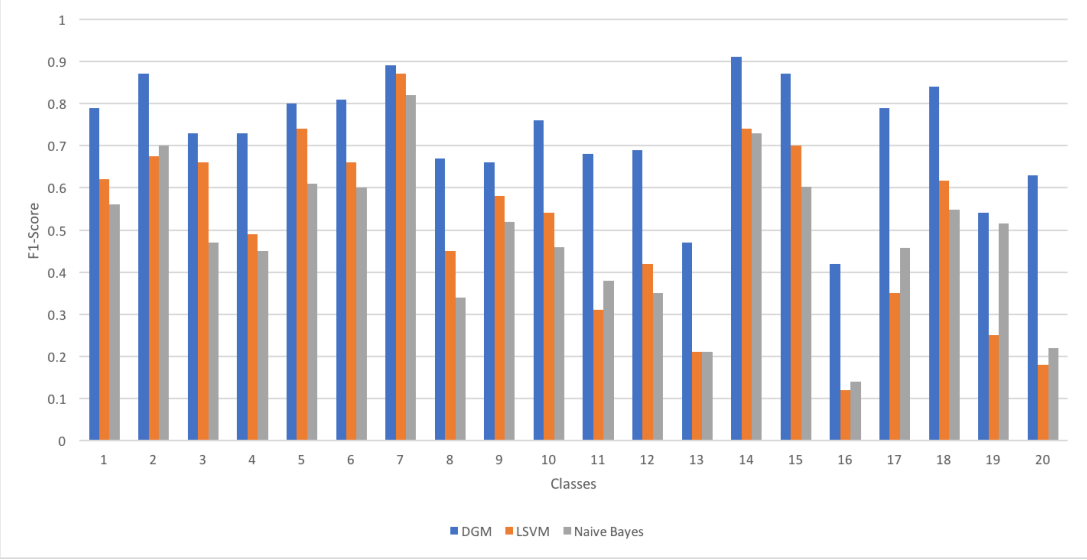


Figure 2: F1-score obtained for each class using the Naive-Bayes, Linear SVM and DGM methods respectively

scenario based and the use of a specific dataset for evaluation is not prevalent.

Another interesting observation from our result is the high presence of biological data in the *Neural Networks* and *Dimensionality Reduction* classes. We attribute this to the increased demand for more advanced models to handle the increasing data dimensionality, exploit and extract the inherent information and structure in the vast volumes of data which have prevailed in computational biology due to the burgeoning of modern technologies [17]. This need to handle larger data brought about the development and application of several dimensionality reduction techniques to better handle the data, and the introduction of neural networks and deep learning to bioinformatics because of their ability to process and learn from large and complex data [15], [22].

VI. Conclusion and Future works

We extended and investigated the use of deep generative models on multi-label graph-based semi-supervised document classification such that it can learn from both the text and graph information. The ability to learn from two inputs that could be of two different distribution means that it could be used not just for graph-based classification but also can be applied to data with additional information. We introduced the Delve citation dataset, a new document labeled multi-label citation dataset for graph-based document classification. We benchmark the Delve multi-labeled citation dataset lon the DGM framework, and we show that the semi-supervised DGM model can learn better classification models compared to several supervised learning algorithms. From the classification result on the Delve-ML, we extract and report the top ten dataset resources used by studies in

Table V: Top-10 dataset resources used in nine selected computer science fields in the delve database

	Natural Language Processing	**Bioinformatics**	**Recommendation Systems**
1	WordNet Data	Gene Ontology Consortium	Flickr Data
2	Linguistic Data Consortium	Gene Expression Omnibus	Facebook Data
3	TREC Data	UCSC Genome Browser	Youtube Data
4	Wikipedia Data	Ensembl Genome Browser	Yahoo data
5	Unified Medical Language System Data	European Bioinformatics Institute (EMBL-EBI)	IMDB Data
6	NIST DUC Data	Saccharomyces Genome Database (SGD)	Del.icio.us Data
7	Twitter Data	Basic Local Alignment Search Tool Data	Twitter Data
8	NTLK Data	The Arabidopsis Information Resource (TAIR)	Netflixprize Data
9	NATCORP (BNC) Data	UCL Statistical Parametric Mapping Data	Last.fm Data
10	NML Medical Subject Headings	UniProt Data	Grouplens Data
	Neural Networks	**Security & Privacy**	**Computer Vision**
1	UCL Statistical Parametric Mapping Data	Network Simulator (ns-2)	UCL Statistical Parametric Mapping
2	Gene Expression Omnibus	PlanetLab	Flickr Data
3	Gene Ontology Consortium	University of Oregon Route Views	TREC Video Retrieval Evaluation: TRECVID
4	UCSC Genome Browser	Gnutella (wego)	OpenStreetMap
5	Saccharomyces Genome Database (SGD)	Ebay (Auction) Data	The MNIST database
6	Ensembl Genome Browser	CAIDA Data	CMU Graphics Lab Motion Capture Database
7	UCI Machine Learning Repository	Common Vulnerabilities and Exposures (CVE) Data	Youtube Data
8	European Bioinformatics Institute (EMBL-EBI)	Skype Data	VICON Data
9	Arabidopsis Information Resource (TAIR) Data	National Vulnerability Database	BrainWeb: Simulated Brain Database
10	The DIP Database	UCI Learning Repository	CAVIAR project Data
	Agent Systems	**Information Retrieval**	**Dimensionality Reduction**
1	UCL Statistical Parametric Mapping	TREC Data	UCSC Genome Browser
2	JAVA Agent DEvelopment Framework	WordNet Data	Gene Ontology Consortium
3	Open Dynamics Engine (ODE)	Gene Ontology Consortium	Ensembl Genome Browser
4	NetLogo	GeoNames Data	UCI Machine Learning Repository
5	Jess Rule Engine	LinkedData Data	The Arabidopsis Information Resource (TAIR)
6	The Player Project	Network Simulator (ns-2)	Saccharomyces Genome Database (SGD)
7	Protégé	DBLP Data	Kyoto Encyclopedia of Genes and Genomes (KEGG)
8	Webots Robot Simulator	TREC Video Retrieval Evaluation: TRECVID	David Bioinformatics Resource
9	SWARM Agent-based Modeling Simulation Package,	Snowball Stemmers	ArrayExpress Data
10	Robot Operating System (ROS)	UCI KDD Archive	European Bioinformatics Institute (EMBL-EBI)

some selected subfields.

In future works, we plan to enlarge the labeled dataset and publish the full links of the top-k datasets from other fields. We also aim to improve the data quality further and provide it in a publicly usable format.

VII. Acknowledgement

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. 2639. This work was performed when Ke Sun was affiliated with KAUST.

References

[1] Uchenna Akujuobi and Xiangliang Zhang. Delve: a dataset-driven scholarly search and analysis system. *ACM SIGKDD Explorations Newsletter*, 19(2):36–46, 2017.

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[3] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of attributed graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

[4] Hong Cheng, Yang Zhou, and Jeffrey Xu Yu. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):12, 2011.

[5] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852. 2016.

[7] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[9] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer, 2010.

[10] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, 2008.

[11] Sang-Bum Kim, Hae-Chang Rim, Dongsuk Yook, and Heui-Seok Lim. Effective methods for improving naive bayes text classifiers. *PRICAI 2002: Trends in Artificial Intelligence*, pages 479–484, 2002.

[12] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NIPS*, pages 3581–3589, 2014.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] Vandana Korde and C Namrata Mahender. Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, 3(2):85, 2012.

[15] Lee J Lancashire, Christophe Lemetre, and Graham R Ball. An introduction to artificial neural networks in bioinformatics—application to complex microarray and mass spectrometry datasets in cancer studies. *Briefings in bioinformatics*, 10(3):315–329, 2009.

[16] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[17] Lexin Li. Dimension reduction for high-dimensional data. *Statistical methods in molecular biology*, pages 417–434, 2010.

[18] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[19] Chen Luo, Renchu Guan, Zhe Wang, and Chenghua Lin. Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks. In *European Conference on Information Retrieval*, pages 210–221. Springer, 2014.

[20] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *ICML*, pages 1445–1454, 2016.

[21] Tim Menzies and Charles Pecheur. Verification and validation and artificial intelligence. *Advances in computers*, 65:153–201, 2005.

[22] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, page bbw068, 2016.

[23] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2014.

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

[25] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[26] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[28] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 797–806. ACM, 2009.

[29] Hanghang Tong, Christos Faloutsos, Brian Gallagher, and Tina Eliassi-Rad. Fast best-effort pattern matching in large attributed graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746. ACM, 2007.

[30] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, pages 40–48, 2016.

[31] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[32] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2004.

[33] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 2010 IEEE 10th International Conference on Data Mining*, pages 689–698. IEEE, 2010.

[34] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep.*, 2002.

APPENDIX

A. Derivations of DGM

By assumption, the generative model is given by

$$
\begin{aligned}
p(z) &= \mathcal{N}(z \mid 0, I), \\
p(y) &= \begin{cases} \mathrm{Ber}\,(y \mid \gamma_t, \gamma_x) & \text{if } y \text{ is multi-label} \\ \mathrm{Cat}\,(y \mid \pi_t, \pi_x) & \text{if } y \text{ is multi-class} \end{cases}, \\
p(x \mid y, z) &= \begin{cases} \mathrm{Ber}\,(x \mid f_{\theta_1}(y, z)) \\ \qquad \text{if } x \text{ is binary} \\ \mathcal{N}\left(x \mid \mu_{\theta_1}(y, z),\, \mathrm{diag}(\sigma^2_{\theta_1}(y, z))\right) \\ \qquad \text{if } x \text{ is continuous} \end{cases}, \\
p(t \mid y, z) &= \begin{cases} \mathrm{Ber}\,(t \mid f_{\theta_2}(y, z)) \\ \qquad \text{if } t \text{ is binary} \\ \mathcal{N}\left(t \mid \mu_{\theta_2}(y, z),\, \mathrm{diag}(\sigma^2_{\theta_2}(y, z))\right) \\ \qquad \text{if } t \text{ is continuous} \end{cases},
\end{aligned}
$$

where y is either a one-hot vector (multi-class) or a binary vector (multi-label) of length D_y, z is a continuous latent

vector in $\Re^{D_z}$, x, t are continuous latent vectors in $\Re^{D_x}$, and $\Re^{D_t}$ respectively, $\mathrm{Cat}(\cdot \,|\, \pi_t, \pi_x)$ denotes a category distribution wrt the probability vector π_t and π_x denotes probability distributions vectors given t and x respectively; with $\pi_t, \pi_x = \frac{1}{D_y}\epsilon$, $\mathcal{N}(\cdot \,|\, \mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean μ and covariance matrix Σ, $\mathrm{Ber}(\cdot \,|\, \gamma_t, \gamma_x)$ denotes a multivariate Bernoulli distribution with independent random bits wrt the activation vectors γ_t, γ_x respectively. A corresponding inference model is assumed to be

$$q(y \,|\, t, x) = \begin{cases} \mathrm{Cat}\left(y \,|\, \pi_{\varphi_1}(x), \pi_{\varphi_1}(t)\right), \text{if } y \text{ is multi-class} \\ \mathrm{Ber}\left(y \,|\, \gamma_{\varphi_1}(x), \gamma_{\varphi_1}(t)\right), \text{if } y \text{ is multi-label} \end{cases},$$

$$q(z \,|\, t, x, y) = \mathcal{N}\left(z \,|\, \mu_{\varphi_2}(t, x, y),\, \mathrm{diag}(\sigma^2_{\varphi_2}(t, x, y))\right).$$

B. The Variational Bound

Given a set of labeled pairs $\{(t_l, x_l, y_l)\}$ and a set of unlabeled $\{t_u, x_u\}$, the task is to learn all the model parameters $\{\theta_1, \theta_2, \varphi_1, \varphi_2\}$ and to make prediction $t, x \to y$ based on the inference machine and $q(y \,|\, t, x)$. For a labeled pair (t, x, y), we have its negative log likelihood

$$\begin{aligned}
& -\log p(t, x, y) \\
=& -\log \int_z p(y)p(z)p(x \,|\, y, z)p(t \,|\, y, z)\mathrm{d}z \\
=& -\log \int_z q(z \,|\, t, x, y) \\
& \times \frac{p(y)p(z)p(x \,|\, y, z)p(t \,|\, y, z)}{q(z \,|\, t, x, y)}\mathrm{d}z \\
\leq & \int_z q(z \,|\, t, x, y) \\
& \times \log \frac{q(z \,|\, t, x, y)}{p(y)p(z)p(x \,|\, y, z)p(t \,|\, y, z)}\mathrm{d}z \\
=& \mathrm{KL}(q(z \,|\, t, x, y) \,:\, p(z)) \\
& -\log p(y) \\
& -\int_z q(z \,|\, t, x, y)\log p(x \,|\, y, z)\mathrm{d}z \\
& -\int_z q(z \,|\, t, x, y)\log p(t \,|\, y, z)\mathrm{d}z. \qquad (6)
\end{aligned}$$

The KL divergence between two Gaussian distribution has a closed form

$$\begin{aligned}
& \mathrm{KL}(\mathcal{N}(x \,|\, \mu_1, \Sigma_1) \,:\, \mathcal{N}(x \,|\, \mu_2, \Sigma_2)) \\
=& -\frac{1}{2}\log|\Sigma_1| - \frac{D_x}{2} + \frac{1}{2}\log|\Sigma_2| \\
& + \frac{1}{2}(\mu_1 - \mu_2)^\intercal \Sigma_2^{-1}(\mu_1 - \mu_2) + \frac{1}{2}tr(\Sigma_1\Sigma_2^{-1}). \qquad (7)
\end{aligned}$$

In the special case that $\Sigma_1 = \mathrm{diag}(\sigma_1^2)$, $\Sigma_2 = \mathrm{diag}(\sigma_2^2)$, we have

$$\begin{aligned}
& \mathrm{KL}(\mathcal{N}(x \,|\, \mu_1, \Sigma_1) \,:\, \mathcal{N}(x \,|\, \mu_2, \Sigma_2)) \\
=& \frac{1}{2}\sum_{i=1}^{D_x}\left[-\log\sigma_{1i}^2 - 1 + \log\sigma_{2i}^2 + \frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_{2i}^2} + \frac{\sigma_{1i}^2}{\sigma_{2i}^2}\right].
\end{aligned}$$

Therefore the first term on the RHS of Eq. (6) becomes

$$\begin{aligned}
& \mathrm{KL}(q(z \,|\, t, x, y) \,:\, p(z)) \\
\approx & \sum_{i=1}^{D_z}\Big[-\log\sigma^{(i)}_{\varphi_3}(t, x, y) - \frac{1}{2} \\
& + \frac{\mu^{(i)}_{\varphi_3}(t, x, y)^2 + \sigma^{(i)}_{\varphi_3}(t, x, y)^2}{2}\Big]. \qquad (8)
\end{aligned}$$

If x is binary, the third term on the RHS of eq. (6) is

$$\begin{aligned}
& -\int_z q(z \,|\, t, x, y)\log p(x \,|\, y, z)\mathrm{d}z \\
\approx & -\sum_{i=1}^{D_x}\log p(x^i \,|\, f^i_{\theta_1}(y, \hat{z},)). \qquad (9)
\end{aligned}$$

If t is binary, the last term on the RHS of eq. (6) is

$$\begin{aligned}
& -\int_z q(z \,|\, t, x, y)\log p(x \,|\, y, z)\mathrm{d}z \\
\approx & -\sum_{i=1}^{D_e}\log p(t^i \,|\, f^i_{\theta_2}(y, \hat{z},)). \qquad (10)
\end{aligned}$$

For continuous x, the third term is

$$\begin{aligned}
& -\int_z q(z \,|\, t, x, y)\log p(x \,|\, y, z)\mathrm{d}z \\
=& \sum_{i=1}^{D_x}\left[\frac{1}{2}\log 2\pi + \log\sigma_{\theta_2}(y, \hat{z}) + \frac{(x - \mu_{\theta_2}(y, \hat{z}))^2}{2\sigma^2_{\theta_2}(y, \hat{z})}\right] \qquad (11)
\end{aligned}$$

For continuous t, the last term is

$$\begin{aligned}
& -\int_z q(z \,|\, t, x, y)\log p(t \,|\, y, z)\mathrm{d}z \\
=& \sum_{i=1}^{D_e}\left[\frac{1}{2}\log 2\pi + \log\sigma_{\theta_2}(y, \hat{z}) + \frac{(e - \mu_{\theta_2}(y, \hat{z}))^2}{2\sigma^2_{\theta_2}(y, \hat{z})}\right]. \qquad (12)
\end{aligned}$$

Plugging the above eqs. (8-12) into eq. (6), we get a variational bound of the model evidence:

$$-\log p(x, y) \leq \mathcal{L}(x, y). \qquad (13)$$

If x and t have the same distribution, (e.g x and t are binary),

$$\begin{aligned}
\mathcal{L}(t, x, y) = & \sum_{i=1}^{D_z}\Big[-\log\sigma^{(i)}_{\varphi_2}(t, x, y) \\
& + \frac{\mu^{(i)}_{\varphi_2}(t, x, y)^2 + \sigma^{(i)}_{\varphi_2}(t, x, y)^2}{2}\Big] \\
& - \sum_{i=1}^{D_x}\log p(x^i \,|\, f^i_{\theta_1}(y, \hat{z})) \\
& - \sum_{i=1}^{D_e}\log p(t^i \,|\, f^i_{\theta_2}(y, \hat{z})) + constant, \qquad (14)
\end{aligned}$$

where $\hat{z} \sim q(z \mid x, y)$. If x and t have the different distributions, (e.g x is continuous and t is binary),

$$\begin{aligned}\mathcal{L}(t,x,y) = &\sum_{i=1}^{D_z}\Bigg[-\log\sigma^{(i)}_{\varphi_2}(x,t,y) \\ &+\frac{\mu^{(i)}_{\varphi_2}(t,x,y)^2+\sigma^{(i)}_{\varphi_2}(t,x,y)^2}{2}\Bigg] \\ &+\sum_{i=1}^{D_x}\left[\log\sigma_{\theta_1}(y,\hat{z})+\frac{(x-\mu_{\theta_1}(y,\hat{z}))^2}{2\sigma^2_{\theta_1}(y,\hat{z})}\right] \\ &-\sum_{i=1}^{D_e}\log p(t^i \mid f^i_{\theta_2}(y,\hat{z}))+constant. \end{aligned} \tag{15}$$

For an unlabeled (t, x), we have

$$\begin{aligned}&-\log p(t,x) \\ &=-\log\sum_y\int_z p(y)p(z)p(x \mid y,z)(x \mid y,z)\mathrm{dz} \\ &\leq \mathcal{U}(x). \end{aligned} \tag{16}$$

If x and t have the same distribution, (e.g x and t are binary),

$$\begin{aligned}\mathcal{U}(t,x) = &\sum_y\int_z q(y \mid t,x)q(z \mid t,x,y) \\ &\times\log\frac{q(y \mid t,x)q(z \mid t,x,y)}{p(y)p(z)p(x \mid y,z)p(t \mid y,z)}\mathrm{dz} \\ \approx &\sum_y\pi^y_{\varphi_1}(t,x)\Bigg\{\log\pi^y_{\varphi_1}(t,x) \\ &+\sum_{i=1}^{D_z}\Bigg[-\log\sigma^{(i)}_{\varphi_3}(t,x,y) \\ &+\frac{\mu^{(i)}_{\varphi_3}(t,x,y)^2+\sigma^{(i)}_{\varphi_3}(t,x,y)^2}{2}\Bigg] \\ &-\sum_{i=1}^{D_x}\log p(x^{(i)} \mid f^{(i)}_{\theta_1}(y,\hat{z})) \\ &-\sum_{i=1}^{D_e}\log p(t^{(i)} \mid f^{(i)}_{\theta_1}(y,\hat{z}))\Bigg\}+\text{constant}, \end{aligned} \tag{17}$$

where $\hat{y} \sim q(y \mid t, x)$ and $\hat{z} \sim q(z \mid t, x, \hat{y})$. As compared to Eq. (13), the only difference in Eq. (16) is the sum over y. If x and t have the different distributions, (e.g x is continuous and t is binary),

$$\begin{aligned}\mathcal{U}(t,x) \approx &\sum_y\pi^y_{\varphi_1}(t,x)\Bigg\{\log\pi^y_{\varphi_2}(t,x) \\ &+\sum_{i=1}^{D_z}\Bigg[-\log\sigma^{(i)}_{\varphi_2}(t,x,y) \\ &+\frac{\mu^{(i)}_{\varphi_2}(t,x,y)^2+\sigma^{(i)}_{\varphi_2}(t,x,y)^2}{2}\Bigg] \\ &+\sum_{i=1}^{D_x}\left[\log\sigma_{\theta_1}(y,\hat{z})+\frac{(x-\mu_{\theta_1}(y,\hat{z}))^2}{2\sigma^2_{\theta_1}(y,\hat{z})}\right] \\ &-\sum_{i=1}^{D_e}\log p(t^{(i)} \mid f^{(i)}_{\theta_1}(y,\hat{z}))\Bigg\}+\text{constant}. \end{aligned} \tag{18}$$

The summation over y in the unlabeled case increases exponentially with increasing number of classes in the multi-label case (i.e summation over all the possible configurations of the labels). To reduce the complexity, we implement a negative sampling version. We generate a negative label sample set C of size s (in our experiments we found $s = 10$ to be good enough) for each unlabeled data sample x^i_u, t^i_u. Each negative sample c_i is a multivariate Bernoulli distribution with independent random bits wrt the activation vector $1 - p$ such that labels with lower probabilities are selected. A positive sample wrt the activation vector p, assumed to be the positive label configuration is also added to the set C. Where p is the learned activation vector during each epoch.

We then calculate the loss for the unlabeled datasets $\mathcal{U}(t, x)$ using sampled label configuration in the set C. For instance equation 18 changes to :

$$\begin{aligned}\mathcal{U}(t,x) \approx &\sum_c\pi^c_{\varphi_1}(t,x)\Bigg\{\log\pi^c_{\varphi_2}(t,x) \\ &+\sum_{i=1}^{D_z}\Bigg[-\log\sigma^{(i)}_{\varphi_2}(t,x,c) \\ &+\frac{\mu^{(i)}_{\varphi_2}(t,x,c)^2+\sigma^{(i)}_{\varphi_2}(t,x,c)^2}{2}\Bigg] \\ &+\sum_{i=1}^{D_x}\left[\log\sigma_{\theta_1}(c,\hat{z})+\frac{(x-\mu_{\theta_1}(c,\hat{z}))^2}{2\sigma^2_{\theta_1}(c,\hat{z})}\right] \\ &-\sum_{i=1}^{D_e}\log p(t^{(i)} \mid f^{(i)}_{\theta_1}(c,\hat{z}))\Bigg\}+\text{constant}. \end{aligned} \tag{19}$$

The learning cost function is

$$\begin{aligned}E = &\sum_{x_l,t_l,y_l}\mathcal{L}(x_l,t_l,y_l)+\sum_{x_u,t_u}\mathcal{U}(t_u,x_u) \\ &+\beta\frac{N_l+N_u}{N_l}\sum_{x_l,t_l,y_l}\log q(y_l \mid t_l x_l), \end{aligned} \tag{20}$$

where $\beta > 0$ is a regularization strength parameter. N_l, N_u are the number of labeled and unlabeled samples respectively. The first two terms on the RHS of Eq. (20) is generative loss, the last term is discriminative loss. For the Multi-label case summing over all the y is expensive. To reduce the complexity of summing over the y in the multi-label case, we apply the pseudo labelling technique [16].