Tag2Gauss: Learning Tag Representations via Gaussian Distribution in Tagged Networks*

Yun Wang[1†], **Lun Du**[1†], **Guojie Song**[1 ‡], **Xiaojun Ma**[1], **Lichen Jin**[1], **Wei Lin**[2], **Fei Sun**[2]

[1] Key Laboratory of Machine Perception (MOE), Peking University, Beijing, China

[2] Alibaba Group, Beijing, China

{wang.yun, dulun, gjsong, mxj, jlcbluecat}@pku.edu.cn, {yangkun.lw, ofey.sf}@alibaba-inc.com

Abstract

Keyword-based tags (referred to as *tags*) are used to represent additional attributes of nodes in addition to what is explicitly stated in their contents, like the hashtags in YouTube. Aside of being auxiliary information for node representation, tags can also be used for retrieval, recommendation, content organization, and event analysis. Therefore, tag representation learning is of great importance. However, to learn satisfactory tag representations is challenging because 1) traditional representation methods generally fail when it comes to representing tags, 2) bidirectional interactions between nodes and tags should be modeled, which are generally not dealt within existing research works. In this paper, we propose a tag representation learning model which takes tag-related node interaction into consideration, named **Tag2Gauss**. Specifically, since tags represent node communities with intricate overlapping relationships, we propose that Gaussian distributions would be appropriate in modeling tags. Considering the bidirectional interactions between nodes and tags, we propose a tag representation learning model mapping tags to distributions consisting of two embedding tasks, namely Tag-view embedding and Node-view embedding. Extensive evidence demonstrates the effectiveness of representing tag as a distribution, and the advantages of the proposed architecture in many applications, such as the node classification and the network visualization.

1 Introduction

Tagged networks, namely networks including not only relationships between nodes but also the subordinate relationship between nodes and tags, are ubiquitous. Keyword-based tags (referred to as *tags*) are used to represent additional attributes of nodes in addition to what is explicitly stated in their contents. Taking YouTube as an example, a million-video sample shows that 6 tags are applied to each video on average, the majority of which do not exist in the title [Geisler and Burns, 2007; Aggarwal, 2011; Huang *et al.*, 2010; Godin *et al.*, 2013].

In addition to mere literal meanings expressed in the words of tags, tags also convey more complex semantics illustrated in the way they interact with tagged nodes, thereby providing nodes with information highly useful to users. Consequently, tags can be used for different tasks, such as information retrieval, item recommendation, and event analysis [Chang, 2010; Tsur and Rappoport, 2012].

However, it is challenging to learn satisfactory tag representations.

- Traditional representation methods generally fail when it comes to representing tags. On one hand, naive one-hot representations of tags ignore the semantic relationships between tags. On the other hand, representing tags with mere word embedding focus more on the literal meaning of keyword-based tags, which ignores their interaction with tagged nodes and is not suitable for tag representation. In addition, conventional representation learning models, which map objects to vectors also fail to generate satisfactory representations, since, a single point in the vector space is unable to represent complex relationships between tags, such as inclusion, entailment and hierarchy. Consequently, more flexible representation methods should be involved.
- Bidirectional interactions between nodes and tags should be modeled, which are generally not dealt within existing research works. For example, in *Bilibili*, the tag "*MAD*" is similar to the tag "*ANIMATION*" according to tagged videos, however it is difficult to find the correlation in general literal corpus. More extremely, some tags with mosaic keywords cannot convey any useful literal information. As discussed above, the meanings tags express are largely defined by the nodes adhered with them, and vise versa. However, explicit similarity measures, such as links between tag pairs, are not available in tagged networks, which underscores the need for us to design models capturing complex interactions between nodes and tags.

To cope with these problems, we study the tag embed-

*This work was supported by the National Natural Science Foundation of China (Grant No. 61876006 and No. 61572041).

†These authors contributed equally to the work.

‡Corresponding Author

ding problem in node-tag hybrid networks derived from a tagged network (see Figure 1) and propose our model, named **Tag2Gauss**, which deals with the challenges correspondingly. On one hand, since tags represent node communities with intricate overlapping relationships, we propose that distributions, namely a region associated with different intensity, would be more appropriate in modeling tags containing nodes with scattering positions in vector space. On the other hand, considering the bidirectional interactions between nodes and tags, we propose a tag representation learning model mapping tags to distributions consisting of two embedding tasks, namely **tag-view embedding** and **node-view embedding**. Specifically, in tag-view embedding module, tags are represented as Gaussian distributions. To capture similarity between tags, a walking strategy, named *Hybrid Walker*, is proposed based on the node-tag hybrid network, followed by a max-margin ranking objective to optimize tag representations. In node-view embedding, we present a generative model for node representations inspired by the Gaussian Mixture Model. In this way, tag distributions are directly used for node embedding generation, enabling the joint learning of tag representations and node representations in the multi-task learning framework we propose.

To summarize, we make the following contributions:

- We represent tags as Gaussian distributions, which is able to characterize the complex semantic relationships between tags, such as inclusion, entailment and hierarchy.
- We design a multi-task learning framework, **Tag2Gauss**, involving both tag representation learning and node representation learning that can be optimized jointly.
- We conduct extensive experiments on three real-world tagged networks, and the results demonstrate that our model can capture the rich information carried in both nodes and tags and achieve superior performance from its counterparts.

2 Related Work

Network Embedding. Network embedding aims to map the vertices or edges of a network into low-dimensional vector space for the sake of better performance of learning tasks such as node classification and link prediction [Goyal and Ferrara, 2018]. Some methods proposed are based on context of the vertex to preserve its structural properties [Perozzi *et al.*, 2014; Tang *et al.*, 2015; Du *et al.*, 2018a; Du *et al.*, 2018b], others are based on graph factorization or deep learning [Goyal and Ferrara, 2018; Wang *et al.*, 2016]. Embedding of attributed network whose nodes are each associated with their rich features [Huang *et al.*, 2017], has received attention in recent years. TADW leverages rich text features in graph factorization [Yang *et al.*, 2015], while GCN and GraphSAGE turn to scalable deep learning architecture with neighborhood aggregation [Kipf and Welling, 2017; Hamilton *et al.*, 2017]. However, stable discrete node features as tags are often treated as one-hot vector in those works and interactions between them are overlooked.

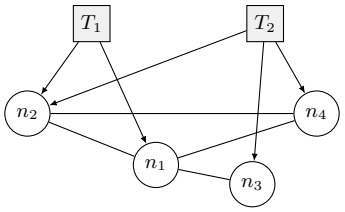


Figure 1: The node-tag hybrid network, which has two types of nodes, plain nodes n_i and tag nodes T_k; and two types of edges, interaction edges (undirected) and affiliation edges (directed).

Variational Representation with Gaussian Distribution. Gaussian distribution is used to describe the variation bound for posterior probabilities in Variational Autoencoders (VAE) [Kingma and Welling, 2014]. Uncertainty of Gaussian distribution has also been utilized in word embedding models to make the parameters more expressive both in prior [Vilnis and McCallum, 2015] and posterior [Brazinskas *et al.*, 2017] ways. When it comes to network/graph, KG2E train Gaussian embeddings for entities and relations in knowledge graph [He *et al.*, 2015]; VGAE, based on VAE, embeds the network with Gaussian using GCN encoder and inner dot decoder [Kipf and Welling, 2016]. However, to the best of our knowledge, expressive Gaussian methods has not been generalized to tagged networks yet.

3 Problem Definition

Definition 1 (***Tagged Network***)**.** A **tagged network** is defined as $G = <V, E, T>$, where $V = \{v_i\}$, $i = 1, 2, \ldots, n$ represents the set of nodes, $E = \{e_{ij}\}$, $i, j = 1, 2, \ldots, n, i \neq j$ represents the set of edges, and $T = \{t_k\}$, $k = 1, 2, \ldots, |T|$ represents the set of tags that each node belongs to. Each node v_i has k tags, *i.e.*, T_{v_i}=$\{t_{i_1}, t_{i_2}, \ldots, t_{i_k}\}$, $t_{i_j} \in T$. If node v_i is marked with tag t, node v_i has a tag t, or tag t belongs to node v_i.

Tags are literal symbols to label characteristics of a node. The semantics of tag is essentially determined by the characteristics of the set of nodes marked with it. In order to explore the tag representation based on the interaction of nodes, we introduce the definition of node-tag hybrid network as shown in Figure 1 as follows:

Definition 2 (***Node-Tag Hybrid Network***)**.** A **node-tag hybrid network** H=$<V_H, E_H>$ is derived from a tagged network G=$<V, E, T>$, where $V_H = V \cup T$, $E_H = E \cup E_T$. For $v_H \in V_H$, v_H is a plain node if $v_H \in V$. Otherwise, it is a tag node if $v_H \in T$. For $e_H \in E_H$, e_H is an interaction edge if $e_H = <v_1, v_2> \in E$. Otherwise, it is an affiliation edge $e_H = <v, t>$ and node v is marked with tag t.

Due to the bidirectional interactions between nodes and tags, tag representation learning task and node representation learning task can reinforce each other and be jointly optimized. Next, we introduce the definition of node-view embedding for node representation and tag-view embedding for tag representation, respectively.

Definition 3 (***Node-view Embedding***)**.** Given a tagged network $G = <V, E, T>$, the problem of **node-view embed-**

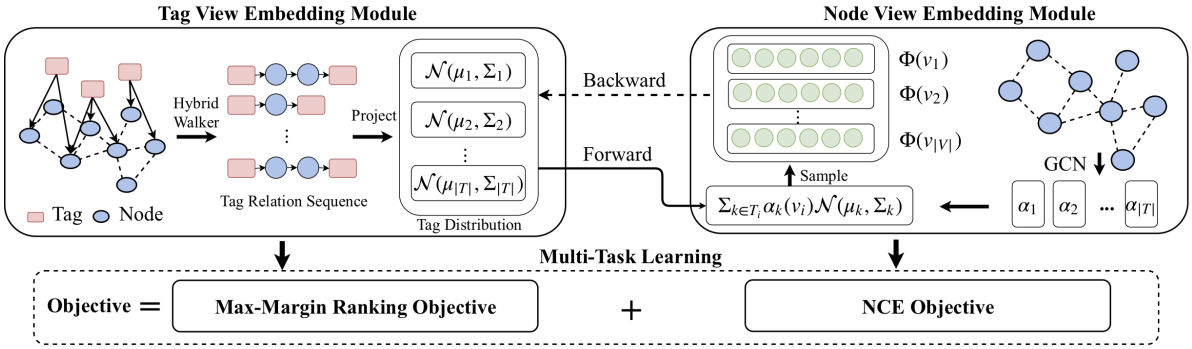


Figure 2: An overview of Tag2Gauss. Tag2Gauss consists of two major components: (a) tag view embedding module projects tags to Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ and preserve the proximity exploring from the nodes interaction with the help of the hybrid walker; (b) node view embedding module maps nodes to a single point in the vector space which preserves the neighbor-aware proximity and the tag-aware proximity with the help of GCN and GMM, respectively. Besides, the multi-task learning part is proposed to jointly optimize the two tasks, *i.e.,* tag-view embedding and node-view embedding.

ding aims to represent each node $v \in V$ attached with multiple tags T_v into a low-dimensional Euclidean space conditional on tags adhered with it, *i.e.*, learning a mapping function $\Phi(V|T) : V \to \mathbb{R}^d$, where $d \ll |V|$.

According to the above analysis, a single point in the vector space is unable to represent complex relationships between tags. Inspired by the Word2Gauss [Vilnis and McCallum, 2015], we represent each tag as a Gaussian distribution. Next, we introduce the definition of tag-view embedding for tag representation learning.

Definition 4 (*Tag-view Embedding*). Given a tagged network $G = <V, E, T>$, the problem of **tag-view embedding** aims to represent each tag $t \in T$ and preserve tag proximity into a low-dimensional Euclidean space based on the interaction of nodes V. The tag $t_k \in T$ is mapped to a Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$ in a d-dimensional space, where $\mu_k \in \mathbb{R}^d$ is a mean vector and $\Sigma_k \in \mathbb{R}^{d \times d}$ is a covariance matrix.

4 Tag2Gauss

In this section, we introduce the overview of our model **Tag2Gauss** and more specifically introduce two main modules, *i.e.*, tag-view embedding module and node-view embedding module. Finally, we propose a multi-task learning framework to jointly learn tag representations as well as node representations.

4.1 Framework Overview

In the framework of Tag2Gauss (see Figure 2), the parameters set $\{\mathcal{N}(\mu_k, \Sigma_k))\}$, where $k = 1, 2, \ldots, |T|$ are the tag representations to be learned, and $\mathcal{N}(\mu_k, \Sigma_k)$ is the tag t_k corresponding Gaussian distribution. In tag-view embedding module, we design a hybrid walker to generate tag pair corpus and train them with the max-margin ranking objective. In node-view embedding module, we present a generative model based on Gaussian Mixture Model for the node representation generation [Rasmussen, 2000]. During the joint learning process, tag representation can be directly used to generate the node embedding. Meanwhile, the optimization process of the node embedding can also tune the tag representation by the back-propagation.

4.2 Tag-view Embedding

In this section, we design the tag-view embedding module to map each tag to a Gaussian distribution based on the interaction of nodes. Because there is no explicit link between tag pairs in the tagged network, we design a walking strategy based on the node-tag hybrid network, namely hybrid walker, to capture the tag proximity which is preserved by optimizing the max-margin ranking objective.

Hybrid Walker

We first define "tag relation sequence" as such a walking sequence which starts and ends with tag nodes, and the rest are plain nodes (see Figure 3).

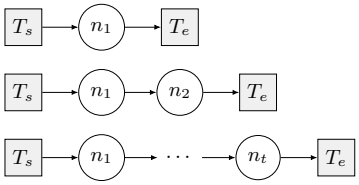


Figure 3: The tag relation sequences.

The hybrid walker strategy is as follows: v refers to the current position, Γ is the type mapping function,

$$\Gamma(v) = \begin{cases} 1 & \text{type}(v) = node \\ 0 & \text{type}(v) = tag \end{cases}. \quad (1)$$

According to the definition of "tag relation sequence", $\Gamma(v^0) = 0$, and the walking process ends up with v^L, where $\Gamma(v^L) = 0$. The transition probability $p(v^{i+1}|v^i)$ at step i is defined as follows:

$$p(v^{i+1}|v^i) = p(t|v^i)p(v^{i+1}|t, v^i), \quad (2)$$

where,

$$t = \Gamma(v^i), \quad (3)$$

$$p(t|v^i) = (\frac{1}{c^i})^t + (1 - \frac{1}{c^i})^{1-t}, \quad (4)$$

$$p(v^{i+1}|v^i, t=1) = \begin{cases} \frac{w_{i,i+1}}{\sum w_{i,*}} & (v^{i+1}, v^i) \in E \\ 0 & (v^{i+1}, v^i) \notin E \end{cases}, \quad (5)$$

$$p(v^{i+1}|v^i, t=0) = \begin{cases} \frac{\hat{w}_{i,i+1}}{\sum \hat{w}_{i,*}} & (v^{i+1}, v^i) \in E_T \\ 0 & (v^{i+1}, v^i) \notin E_T \end{cases}, \quad (6)$$

where, $w_{i,i+1}$ refers to the edge weight of plain nodes, $\hat{w}_{i,i+1}$ refers to the edge weight between plain nodes and tag nodes, and c is a damping factor, which punishes the length of the walking sequence, such as $c = 2$. $p(v^{i+1}|v^i, t = 0)$ is the transition probability from a node v^i to a tag node v^{i+1}. $p(v^{i+1}|v^i, t = 1)$ is the transition probability from a node v^i to another plain node v^{i+1}. Such a walk strategy essentially learns the number of different-order co-neighbors of tags corresponding node groups. Therefore, the closer the semantic distance between the node groups, the higher the probability of the corresponding tags appearing in the same "tag relation sequence".

Max-margin Ranking Objective

From each tag relation sequence, we can get the tag and its context. Considering mapping tags to Gaussian distributions, we train them with max-margin ranking objective O_{tag_view}, which pushes scores of positive pairs above negatives by a margin [Vilnis and McCallum, 2015]:

$$\mathrm{O}_{tag\_view} = \sum_{\substack{u_{tag}\sim\mathcal{Z} \\ v_{tag}\sim\mathcal{P}_{sim_G}(u,\cdot)}} \sum_{\tilde{v}_{tag}\sim\mathcal{Q}} \mathcal{L}(u_{tag}, v_{tag}, \tilde{v}_{tag}), \quad (7)$$

$$\mathcal{L}(u_{tag}, v_{tag}, \tilde{v}_{tag}) = \max\big(0, m - \mathcal{S}(u_{tag}, v_{tag}) + \mathcal{S}(u_{tag}, \tilde{v}_{tag})\big), \quad (8)$$

$$\begin{aligned} \mathcal{S}(tag_i, tag_j) &= -\mathcal{D}_{\mathrm{KL}}(\mathcal{N}_{tag_j} \| \mathcal{N}_{tag_i}) \\ &= -\frac{1}{2}\Bigg(\mathrm{tr}(\Sigma_i^{-1}\Sigma_j) + (\mu_i - \mu_j)^\top \Sigma_i^{-1}(\mu_i - \mu_j) \\ &\quad - d - \log\frac{\det(\Sigma_j)}{\det(\Sigma_i)}\Bigg). \end{aligned} \quad (9)$$

where u_{tag} is drawn from some distribution $\mathcal{Z}$, v_{tag} is drawn from the similarity distribution of $\mathcal{P}_{sim_G}(u_{tag}, \cdot)$ in network G. In our model, $\mathcal{P}_{sim_G}(u_{tag}, \cdot)$ is determined by the frequency of tag occurrence in tag pair corpus from the hybrid walker. $\tilde{v}_{tag}$ is drawn from noise distribution $\mathcal{Q}$. KL divergence is used to measure the similarity $\mathcal{S}(tag_i, tag_j)$ to help learn the complex semantic relationship between tags, such as inclusion, entailment and hierarchy.

4.3 Node-view Embedding

In this section, we design the node-view embedding module to generate node embeddings with the distribution of tags. This feedback from the node embeddings tunes the tag representations during training. When it comes to the node embedding, compared with neighbor-aware proximity (*e.g.* first-order and second-order proximity in [Tang *et al.*, 2015]), tag-aware proximity does not require two nodes to be directly linked or share many "contexts" for being close, even with high order. Therefore, not only neighbor-aware proximity, tag-aware proximity is taken into consideration in the node embedding.

Algorithm 1 Node Embedding Generation Algorithm

Input: Tagged Network $G = <V, E, T>$; input features $\{\mathbf{x}_v\}$; tag sets $\{T_i\}$; neighborhood function $\mathrm{N} : v \to 2^V$
Output: Node representations $\boldsymbol{U}$ for all $v \in V$; Tag representations $\boldsymbol{U}_{tag}$ for all $t \in T$

```
Initialize the set of parameters N(μ_k, Σ_k) for all tag t_k ∈ T
Initialize weight matrix W
for v_i in V do
    n⃗_i = GCN(v_i)
    α = softmax(W · n⃗_i)
    sample t⃗_i from Σ_{t_k ∈ |T_i|} α_k N(μ_k, Σ_k)
    Φ(v_i) = MLP([n⃗_i, t⃗_i])
```

Because tags are represented as distributions, we elaborate a generative node embedding model, where the node representation $\Phi(v_i)$ is drawn from $\Pr(v_i|N_{v_i}, T_{v_i})$ which is the conditional distribution given the neighbor structure N_{v_i} and the tag set T_{v_i} of the node, *i.e.,*

$$\Phi(v_i) \sim \sum_{k \in T_{v_i}} \alpha_k(N_{v_i})\mathcal{N}(\mu_k, \Sigma_k) \quad (10)$$

where T_{v_i} is the tag set of v_i, and N_{v_i} is the neighbor information of v_i. Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$, where $k = 1, 2, \ldots, |T|$ is the representation of k-th tag in T_{v_i}. In order to model the node representation distribution $\Pr$ with the neighbor information, we design the α as the function of N_{v_i}, *i.e.*, $\alpha_k(N_{v_i})$.

Node Embedding Generation

According to Equation 10, we describe the node embedding generation (Algorithm 1). To capture the neighbor-aware proximity, Graph Convolutional Networks [Kipf and Welling, 2017] are introduced to aggregate information. The parameters to be tuned in the training process are the Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$ for any tag $t_k \in T$, and the weight matrix $\boldsymbol{W}$ for modelling mixing coefficients $\boldsymbol{\alpha}$. The tag-aware information $\vec{t}_i$ is drawn from the mixture distribution and next bonded with the neighbor-aware information $\vec{n}_i$ to generate the final node representation $\Phi(v)$.

NCE Objective Function

In order to learn useful and predictive representations in a fully unsupervised setting, we apply the following NCE loss function to distinguish between node samples u and v from the empirical similarity distribution $\mathcal{P}_{sim_G}$ and those negative samples $\{\tilde{v}\}$ generated by a noise distribution Q over the nodes, which encourages nearby nodes to have similar representations:

$$\begin{aligned} \mathrm{O}_{node\_view} = \sum_{\substack{u\sim\mathcal{Z} \\ v\sim\mathcal{P}_{sim_G}(u,\cdot)}} \Big[&-\log\big(\sigma\big(\mathbb{E}_{\Pr}(\Phi(u))^\top \mathbb{E}_{\Pr}(\Phi(v))\big)\big) \\ &-k \cdot \mathbb{E}_{\tilde{v}\sim Q} \log\big(\sigma\big(-\mathbb{E}_{\Pr}(\Phi(u))^\top \mathbb{E}_{\Pr}(\Phi(\tilde{v}))\big)\big)\Big] \end{aligned} \quad (11)$$

where $\sigma(\cdot) = 1/(1 + \exp(-x))$ is the sigmoid function, u is drawn from some distribution $\mathcal{Z}$, v is drawn from the similar-

ity distribution $\mathcal{P}_{sim_G}(u,\cdot)$ which is determined by the plain edge weights linked from u. $\tilde{v}$ is drawn from noise distribution $\mathcal{Q}$. $\mathbb{E}_{\Pr}(\Phi(u))$ is the expectation of $\Phi(u)$, which can be calculated by drawing multiple times from the distribution $\Pr$ defined in Equation 10.

4.4 Multi-task Learning

Based on previous analysis, O_{tag_view} and O_{node_view} can be jointly optimized for better learning tag representations, as well as node representations. In the multi-task learning framework, the whole objective is designed as follows:

$$\min \mathrm{O}_{tag\_view} + \lambda \mathrm{O}_{node\_view} \tag{12}$$

Obviously, we can optimize O_{tag_view} with Stochastic Gradient Descendent (SGD). However, generating $\mathbb{E}_{\Pr}(\Phi(u))$ with multiple times sampling is a non-continuous operation and has no gradient, which can not be back-propagated the error with SGD. Next, we introduce the solution for this problem, called "reparameterization trick" which extends from [Kingma and Welling, 2014]

Reparameterization Trick

For any Gaussian Mixture Model $P(x)$, *i.e.*,

$$P(x) = \sum_i \alpha_i \mathcal{N}_i(x) \tag{13}$$

we can get the expectation $\mathbb{E}_{P(x)}$ by first sampling from each Gaussian distribution $\mathcal{N}_i(x)$ to get the expectation $\mathbb{E}_{\mathcal{N}_i(x)}$, then mixing these expectations with α to get $\mathbb{E}_{P(x)}$, which is proved as follows:

$$\begin{aligned}\mathbb{E}_{P(x)} &= \int_x \sum_i \alpha_i \mathcal{N}_i(x) \cdot x \mathrm{d}\, x = \sum_i \alpha_i \int_x \mathcal{N}_i(x) \cdot x \mathrm{d}\, x \\ &= \sum_i \alpha_i \mathbb{E}_{\mathcal{N}_i(x)}\end{aligned} \tag{14}$$

Also, given μ and Σ of a Gaussian distribution, we can sample x from $\mathcal{N}(\mu,\Sigma)$ by first sampling $\varepsilon \sim \mathcal{N}(0,I)$, then computing $x = \mu + \Sigma^{\frac{1}{2}} * \varepsilon$, *i.e.*,

$$\mathbb{E}_{\mathcal{N}}(x) \Leftrightarrow \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,I)}(x = \mu + \Sigma^{\frac{1}{2}} * \varepsilon) \tag{15}$$

Therefore, the sampling operation in our model can be moved to an input layer. With the help of reparameterization trick, none of the expectations are related to distributions that depend on our model parameters. Therefore we can optimize O_{node_view} with SGD.

5 Experiments

5.1 Experiment Setup

Data Sets

Leetcode[1] is an online programming website. Nodes stand for open problems, edges mean direct hyperlink between problems, and tags are certain categories associated with the problem; we treat difficulty as the label (562 nodes, 1095 edges, 34 tags, 3 classes).

Bilibili[2] is a start-up Chinese video website. Nodes resemble videos, edges are the links of "related video", and tags are certain short words added by uploaders. We collect a sample of videos and their links using BFS and regard the genre (namely type name) of each video as the label (11727 nodes, 187149 edges, 151 tags, 10 classes).

Cora. We also convert the open data set Cora [Sen *et al.*, 2008] to the tagged network where each of the 1433 words resembles a tag (2707 nodes, 5429 edges, 1433 tags, 7 classes).

Baselines

We choose the following methods as baselines for classification: DeepWalk [Perozzi *et al.*, 2014], Node2Vec [Grover and Leskovec, 2016], LINE [Tang *et al.*, 2015], and GraphSAGE [Hamilton *et al.*, 2017].

Parameters Settings

In Tag2Gauss, diagonal covariance is set as the covariance of Gaussian distribution and λ in Equation 12 is selected according to the grid search. The learning rate is 0.001. The experimental results are reported in our experiment with the 10-fold cross-validation.

5.2 Node Classification

For classification, the embedding dimension across different models is 64. We classify the pre-trained embeddings using Logistic Regression over different training size. We take the mean Macro-F_1 as the result. Table 1 shows the Macro-F_1 on node classification. As Tag2Gauss takes the polysemy and complex semantic relations of tags, it outperforms all other methods. Especially, the necessity of discriminating the tag from node feature can be verified from the outperformance of Tag2Gauss to GraphSage.

5.3 Tag Representation Capacity

We prove it is more advantageous for network mining tasks when tags are mapped to distribution through node classification experiment. A more detailed analysis is as follows. Figure 5 shows the node classification accuracy on data set Leetcode on different models, *i.e.*, DeepWalk, Hybrid DeepWalk, and Tag2Gauss. As a typical algorithm that preserves the structure, DeepWalk is introduced into our experiments to verify the validity of regarding tags as auxiliary information. Especially, Hybrid DeepWalk, applies DeepWalk algorithm to node-tag hybrid network and learn the representations of nodes and tags simultaneously and finally joint node embedding with corresponding tag embedding. Tag2Gauss takes the representations of nodes from node-view embedding module.

The figure shows that tags do play a role in network representation learning as DeepWalk is worse than Hybrid DeepWalk and Tag2Gauss; however, Hybrid DeepWalk is inadvisable as it is not significantly improved and lacks robustness. This naive method cannot exploit the implicit information between tags and nodes well. In such situations, Tag2Gauss shows its incomparable advantages on account of exploiting the implicit information of tags and keeping robust by mapping tags to distributions.

5.4 Tag Representation Visualization

In this section, we verify the advantages of mapping tags to Gaussian distributions. Different from a single point, the representation of each tag can learn the semantics and polysemy

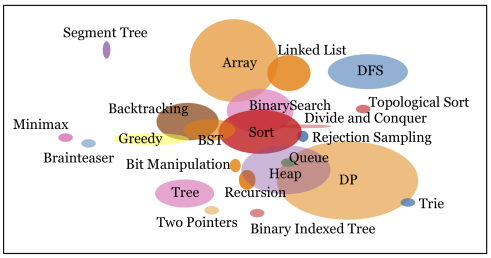
[1] https://leetcode.com/

[2] https://www.bilibili.com/

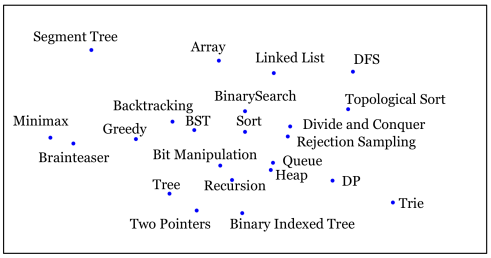
Model	Leetcode					Bilibili					Cora				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
Node2Vec	36.37%	36.37%	38.68%	37.63%	39.68%	48.19%	48.19%	45.36%	45.36%	42.88%	57.12%	57.40%	57.40%	50.84%	48.84%
LINE	34.41%	38.59%	35.89%	33.66%	40.46 %	6.55%	7.21%	7.65%	8.30%	9.28%	49.00%	49.96%	46.23%	45.48%	39.13%
GraphSage	34.00%	37.37%	36.65%	39.77%	44.37%	61.48%	60.81%	60.52%	59.02%	54.26%	50.95%	51.63%	49.10%	45.70%	34.15%
Tag2Gauss	**42.27%**	**42.68%**	**43.70 %**	**44.04%**	**45.03%**	**61.65%**	**61.23%**	**60.83%**	**60.58%**	**56.85%**	**68.45%**	**67.21%**	**66.56%**	**64.87%**	**63.26%**

Table 1: The comparison of node classification measured by Macro-F_1 on different models and different training size.

with the help of the mean vector and variance matrix in Gaussian distribution. We train 2-dimension embeddings with diagonal covariance to visualize the tags in a plane. Figure 4 depicts the visualization of a subset of tag representations in Leetcode. In Figure 4(a), we draw every tag representation as an ellipse with a mean vector and three times of standard derivation. For comparison, Figure 4(b) shows the tags drawn with the mean vector.



(a) Shown as ellipses



(b) Shown as scatters

Figure 4: A subset of tags visualization in Leetcode dataset.

Compared with point representations, ellipses are of great benefit because the overlaps illustrate the inclusion, entailment and hierarchy of tags, hence they reveal the correlation between tags more precisely. The ellipse region of tags in Leetcode shows the applicability of related algorithm among these programming problems, and the intersection of ellipses means that the corresponding tag has the same set of solvable problems. For example, tag *DP* has a larger set of solvable problems than tag *DFS*. As for the complicated interaction among tags, point representation can only show the relationship as distance, thus limited in the expression ability. For example, figure 4(b) shows that *Heap* is close to *Queue*, but *Heap* actually contains *Queue* which can be seen in figure 4(a).

5.5 Cold Start

In cold start, a model generates embedding for new nodes unseen during training. For Tag2Gauss, tag representations

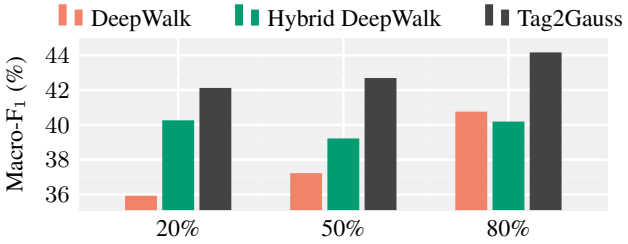


Figure 5: Node classification on data set leetcode. Each bar represents an algorithm in a certain training ratio, plotted by training ratios on the horizontal axis and Macro-F_1 on the vertical.

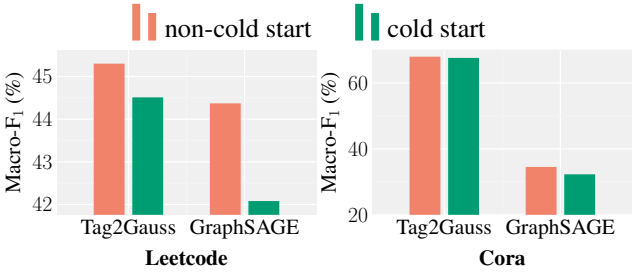


Figure 6: Cold start on Leetcode and Cora dataset

provide the auxiliary information for new nodes representing. To validate the effectiveness of Tag2Gauss in cold start, we divide the network into two parts, *i.e.*, the initial network $\mathcal{G}_{init}$ and the new node set $\mathcal{G}_{new}$. $\mathcal{G}_{init}$ is used to train the embedding model, and then the learned node embedding is used to learn a logistic regression classifier(same as above). In the testing process, embedding of nodes in $\mathcal{G}_{new}$ are inferred with the trained embedding model and classifier. Figure 6 shows the node classification accuracy on $\mathcal{G}_{new}$ with 10-fold cross-validation. It can be seen from Figure 6 that Tag2Gauss prevail over GraphSage on Leetcode and Cora.

6 Conclusion

In this paper, we propose Tag2Gauss to learn tag representation via Gaussian distribution in the tagged network. Specifically, we represent the tag as a Gaussian distribution to characterize the complex semantic relationships between tags, such as inclusion, entailment and hierarchy. Moreover, we propose a tag representation learning model mapping tags to distributions consisting of two embedding tasks, namely tag-view embedding and node-view embedding. Empirically, the extensive experimental results on node classification and network visualization, as well as cold start of new nodes, demonstrate the advantages of Tag2Gauss.

References

[Aggarwal, 2011] Charu C Aggarwal. An introduction to social network data analytics. In *Social network data analytics*, pages 1–15. Springer, 2011.

[Brazinskas *et al.*, 2017] Arthur Brazinskas, Serhii Havrylov, and Ivan Titov. Embedding words as distributions with a bayesian skip-gram model. *CoRR*, abs/1711.11027, 2017.

[Chang, 2010] Hsia-Ching Chang. A new perspective on twitter hashtag use: Diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010.

[Du *et al.*, 2018a] Lun Du, Zhicong Lu, Yun Wang, Guojie Song, Yiming Wang, and Wei Chen. Galaxy network embedding: A hierarchical community structure preserving approach. In *IJCAI*, pages 2079–2085, 2018.

[Du *et al.*, 2018b] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. Dynamic network embedding: An extended approach for skip-gram based network embedding. In *IJCAI*, pages 2086–2092, 2018.

[Geisler and Burns, 2007] Gary Geisler and Sam Burns. Tagging video: Conventions and strategies of the youtube community. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 480–480. ACM, 2007.

[Godin *et al.*, 2013] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596. ACM, 2013.

[Goyal and Ferrara, 2018] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

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

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[He *et al.*, 2015] Yonghao He, Jian Wang, Cuicui Kang, Shiming Xiang, and Chunhong Pan. Large scale image annotation via deep representation learning and tag embedding learning. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 523–526. ACM, 2015.

[Huang *et al.*, 2010] Jeff Huang, Katherine M Thornton, and Efthimis N Efthimiadis. Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 173–178. ACM, 2010.

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

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations. [Online]. Available:https://arxiv.org/abs/1312.6114*, 2014.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *Proceedings of NIPS Bayesian Deep Learning Workshop*, 2016.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Proceedings of the 5th International Conference on Learning Representations. [Online]. Available:https://arXiv:1609.02907*, 2017.

[Perozzi *et al.*, 2014] 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.

[Rasmussen, 2000] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560, 2000.

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

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

[Tsur and Rappoport, 2012] Oren Tsur and Ari Rappoport. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 643–652. ACM, 2012.

[Vilnis and McCallum, 2015] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. In *Proceedings of the 3rd International Conference on Learning Representations. [Online]. Available:https://arxiv.org/abs/1412.6623*, 2015.

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

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2111–2117. AAAI Press, 2015.