Gaussian Embedding of Linked Documents from a Pretrained Semantic Space

Antoine Gourru[1*], **Julien Velcin**[1] and **Julien Jacques**[1]
[1]Université de Lyon, Lyon 2, ERIC UR3083
{antoine.gourru, julien.velcin, julien.jacques}@univ-lyon2.fr

Abstract

Gaussian Embedding of Linked Documents (GELD) is a new method that embeds linked documents (e.g., citation networks) onto a pretrained semantic space (e.g., a set of word embeddings). We formulate the problem in such a way that we model each document as a Gaussian distribution in the word vector space. We design a generative model that combines both words and links in a consistent way. Leveraging the variance of a document allows us to model the uncertainty related to word and link generation. In most cases, our method outperforms state-of-the-art methods when using our document vectors as features for usual downstream tasks. In particular, GELD achieves better accuracy in classification and link prediction on Cora and Dblp. In addition, we demonstrate qualitatively the convenience of several properties of our method. We provide the implementation of GELD and the evaluation datasets to the community (https://github.com/AntoineGourru/DNEmbedding).

1 Introduction

Linked documents are everywhere, from web pages to bibliographic networks (e.g., scientific articles with citations) and social networks (e.g., tweets in a Follower-Followee network). The corpus structure provides rich additional semantic information. For example, in Scientific articles, page limitation often leads to short explanations that become clear if we read the linked papers (i.e., citations).

Many recent approaches propose to use low-dimensional document representation as a proxy for solving downstream tasks [Yang *et al.*, 2015]. These methods learn the representations using both textual and network information. They have many advantages: it accelerates the computation of similarities between documents and it drastically reduces the storage space needed. Moreover, they can significantly improve accuracy in information retrieval tasks, such as document classification [Yang *et al.*, 2015] and link prediction [Bojchevski and Günnemann, 2018]. TADW is the first approach that embeds linked documents [Yang *et al.*, 2015]. Subsequent methods are mainly based on matrix factorization [Brochier *et al.*, 2019; Huang *et al.*, 2017] and deep architectures [Liu *et al.*, 2018; Tu *et al.*, 2017; Kipf and Welling, 2016].

Most of those techniques learn documents as points in the embedding space. However, considering a measure of dispersion around those vectors brings useful information, as shown on corpus with no link between documents [Nikolentzos *et al.*, 2017]. In Graph2Gauss [Bojchevski and Günnemann, 2018], each document is associated with a measure of uncertainty along with its vector representation. However, the objective function optimizes the uncertainty using the network information and it does not model the dispersion at the word level. Additionnaly, variational methods such as [Kipf and Welling, 2016; Meng *et al.*, 2019] introduce gaussian posteriors, but the generative process uses the dot product between documents' mean only to model the adjacency and attribute matrix entries. Hence it is not clear what the uncertainty obtained from the variational variance captures.

Finally, none of earlier methods represents both documents and words in the same semantic space, as opposed to text-based methods such as [Le and Mikolov, 2014]. LDE [Wang *et al.*, 2016] and RLE [Gourru *et al.*, 2020] both build a joint space for embedding words and linked documents. However, these approaches do not take the uncertainty into account.

In this paper, we propose an original model that learns both a vector representation and a vector of uncertainty for each document, named GELD for Gaussian Embedding of Linked Documents. The uncertainty reveals both network and text variance. It will be higher if the document cites very different document sets and if it uses semantically distant words. In addition, documents and words lie in the same space: one can compute similarities between documents and words in this semantic space, enhance queries or describe document clusters by using close words.

After a review of related works in Section 2, we present our model in Section 3. We show that our representations outperform or match most of the recent methods in classification and link prediction on three datasets (two citation networks and a corpus of news articles) in Section 4. Additionally, we provide semantic insights on how to use the variance, and the shared latent space. We conclude and propose extensions of GELD in Section 5.

*Contact Author

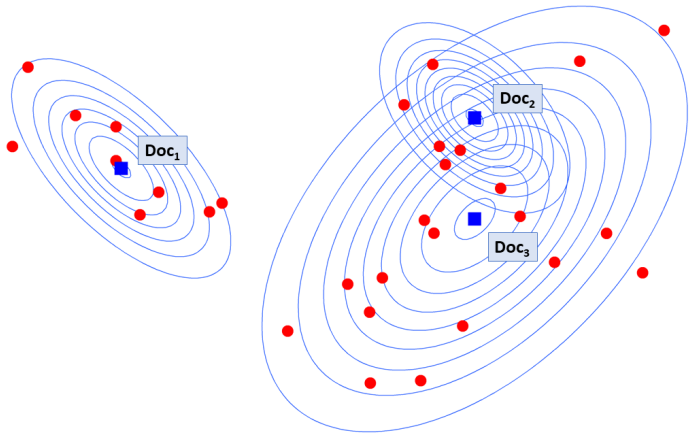

Figure 1: A document is represented as a Gaussian distribution (here, the blue circles). The blue squares represent the means. Words are in red. Doc_2 and Doc_3 are sharing words. In that case, Doc_3 cites Doc_2 but Doc_2 may not cite Doc_3. Doc_1 does not share any word with Doc_3 and Doc_2 and it does not cite them.

2 Related Works

In this section, we present recent approaches for embedding documents organized in a network.

2.1 Document Embedding

Since Word2vec models from [Mikolov *et al.*, 2013], representation learning for text has focused attention as it can improve many downstream tasks. Document embedding followed: many methods propose to represent documents as vectors. For example, [Le and Mikolov, 2014] extends the word2vec formulation. More precisely, the doc2vec models represent documents and words in the same space. One can therefore compute similarities between words and documents.

In many real-life problems, documents are organized as a network: the corpus forms an attributed network. Documents are nodes, citations are edges and the textual contents of the documents are the attributes. In the next section, we present several methods that take this network information into account when learning document representations.

2.2 Document Network Embedding

TADW is the first approach that embeds linked documents [Yang *et al.*, 2015]. It extends DeepWalk [Perozzi *et al.*, 2014], originally developed for network embedding, by formulating the problem as a matrix tri-factorization that includes the textual information. Subsequently, authors of GVNR-t [Brochier *et al.*, 2019] propose to extend Glove [Pennington *et al.*, 2014] in a similar way. AANE [Huang *et al.*, 2017] applies Laplacian Eigenmap to attributed network using Alternating Direction Method of Multipliers to accelerate the learning phase. Recent works mainly use deep architectures: STNE [Liu *et al.*, 2018] adapts the seq2seq architecture, CANE [Tu *et al.*, 2017] uses an attention mechanism.

These methods do not learn representations for both documents and words, as opposed to LDE [Wang *et al.*, 2016] and RLE [Gourru *et al.*, 2020]. LDE uses an extended formulation of [Mikolov *et al.*, 2013]. Nevertheless, it requires labels associated with nodes, which makes it a supervised approach. RLE [Gourru *et al.*, 2020] uses both text and network to build a vector that projects documents onto a pretrained word embedding space.

All these methods learn a single vector by document. This assumption is limited as documents, especially long ones, might be semantically rich. To this end, several methods propose to learn a vector of uncertainty associated to the vector representation.

2.3 Gaussian Document Embedding

Variational methods such as VGAE [Kipf and Welling, 2016] and CAN [Meng *et al.*, 2019] approximate the posterior of the latent variables (the document embedding) by gaussians. Nevertheless, the generative process uses the dot product between posterior means (the latent variables) to model the adjacency and attribute matrix entries. Hence, the variational variance is not an explicit measure of the document semantic and neighborhood uncertainty. The only method that explicitly model uncertainty is Graph2Gauss [Bojchevski and Günnemann, 2018]. A feed-forward neural network embeds the textual content and maps it to a mean and a variance, following an optimization process based on energy-based learning. The negative Kullback-Leibler divergence used as energy allows them to model the proximity between nodes. It should be higher between connected nodes than unconnected nodes. Therefore, Graph2Gauss does not explicitly model the semantic uncertainty of a document.

Another modeling assumption allows to learn documents as Gaussian Distributions. Documents are regarded as bag of word embeddings (see Figure 2), i.e. a multiset of vectors in the latent semantic space. Starting from this assumption, [Das *et al.*, 2015] proposes a new way to extract topics from document sets. Topics are Gaussian distributions that generate word vectors in a semantic space, and therefore a bag of word embeddings for each document. In [Nikolentzos

$$\mathcal{D}_\mathbf{i}^\mathbf{w} = \left\{ \begin{array}{ccccc} Everybody & loves & four & cheese & pizza \\ -0.36 & -0.38 & 0.34 & -0.05 & 0.62 \\ -0.01 & 0.68 & 0.45 & -0.31 & -0.49 \\ 0.21 & -0.54 & 0.55 & -1.32 & -0.69 \end{array} \right\}$$

Figure 2: The bag of word embeddings of a document.

et al., 2017], authors propose to model documents as Gaussians that generate bag of word embeddings. With pretrained word embeddings, they show that the optimal solutions for the means and variances are the empirical means and empirical variances of the vector representations of documents' words. Unfortunately, this solution does not hold with linked documents.

None of the above can, in the same time: i) represent documents and words in the same space, ii) learn a measure of uncertainty associated to the document embedding. We therefore propose a novel method, GELD, that has these properties.

3 GELD: Gaussian Embedding of Linked Documents

3.1 Data and Notations

We consider a corpus of n linked documents and a vocabulary of size v. We also consider a fixed representation in $\mathbb{R}^r$ for each word of the vocabulary. We write $u_k \in \mathbb{R}^r$ the vector representation of the k-th word w_k of the vocabulary, $u_{k,r}$ its r-th element. We note f the function that maps the word w_k to its pretrained representation $f(w_k) = u_k \in \mathbb{R}^r$. The user can either learn the word embeddings on the studied corpus with most recent methods [Devlin *et al.*, 2019; Mikolov *et al.*, 2013] or use a pretrained set of word embeddings built on a broader corpus[1] to reduce the computation time.

Each document d_i is then associated with the bag of word embeddings $\mathcal{D}_i^w = \{f(w^{i,1}), f(w^{i,2}), \ldots\}$. In our notations, $w^{i,1}$ is the first word used in the document i.

Our aim is to learn document representations as Gaussian distributions. Each document has two parameters: a mean μ_i in $\mathbb{R}^r$ and a diagonal variance $\sigma_i^2 I$, with $\sigma_i^2 \in \mathbb{R}^r$, that reveals the document *uncertainty*. We can use the document mean as vector representation when it is needed, e.g., in classification tasks. We note g the function that maps the document d_i to its mean $g(d_i) = \mu_i \in \mathbb{R}^r$. Using the network information, we therefore have, for each document, a bag of document embeddings cited by the document d_i, noted $\mathcal{D}_i^l = \{g(d^{i,1}), g(d^{i,2}), \ldots\}$, where $d^{i,1}$ is the first document cited by the document i.

We introduce two notations: $c_{i,k}$ the number of times the i-th document d_i uses the k-th word of the vocabulary and $a_{i,l}$ the number of times it cites the l-th document of the corpus.

By taking the union of these multisets, the corpus becomes n bags of vectors $\mathcal{D}_i = \mathcal{D}_i^w \cup \mathcal{D}_i^l$:

$$\mathcal{D}_i = \{f(w^{i,1}), f(w^{i,2}), \ldots, g(d^{i,1}), g(d^{i,2}), \ldots\} \quad (1)$$

whose j-th element is noted $\mathcal{D}_{i,j} \in \mathbb{R}^r$.

[1] e.g., https://fasttext.cc/

3.2 Model and Optimization

Similarly to [Das *et al.*, 2015; Nikolentzos *et al.*, 2017], we posit that the vectors in $\mathcal{D}_i$ are independently drawn from an isotropic Gaussian distribution, meaning that

$$\mathcal{D}_{i,j} \sim \mathcal{N}(\mu_i, \sigma_i^2 I) \quad (2)$$

.

The Gaussian is parametrized by a mean and a diagonal variance characterizing the document embedding and its uncertainty. The parameters to learn for each document are $\sigma_i^2 \in \mathbb{R}^r$ and $\mu_i \in \mathbb{R}^r$.

The log-likelihood of the proposed model is, with $|\mathcal{D}_i|$ the cardinality of $\mathcal{D}_i$:

$$\mathcal{L}(\mathcal{D}; \mu, \sigma^2) = \sum_{i=1}^{n} \sum_{j}^{|\mathcal{D}_i|} \log \mathcal{N}(\mathcal{D}_{i,j}; \mu_i, \sigma_i^2 I) \quad (3)$$

where $\mathcal{D} = \{D_i\}_{i=1}^n$, $\sigma^2 = \{\sigma_i^2\}_{i=1}^n$ and $\mu = \{\mu_i\}_{i=1}^n$.

We split this log-likelihood between the drawing regarding words ($\mathcal{L}_w$) and documents ($\mathcal{L}_d$).

$$\begin{aligned} \mathcal{L}(\mathcal{D}; \mu, \sigma^2) &= \sum_{i=1}^{n} \sum_{f(w) \in \mathcal{D}_i^w} \log \mathcal{N}(f(w); \mu_i, \sigma_i^2 I) \\ &+ \sum_{i=1}^{n} \sum_{g(d) \in \mathcal{D}_i^l} \log \mathcal{N}(g(d); \mu_i, \sigma_i^2 I) \\ &= \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{v} c_{i,k} \log \mathcal{N}(u_k; \mu_i, \sigma_i^2 I)}_{\mathcal{L}_w} \\ &+ \underbrace{\sum_{i=1}^{n} \sum_{l=1}^{n} a_{i,l} \log \mathcal{N}(\mu_l; \mu_i, \sigma_i^2 I)}_{\mathcal{L}_d} \end{aligned} \quad (4)$$

As citation and term frequencies are on different scales, we can observe imbalanced $\mathcal{L}_w$ and $\mathcal{L}_d$. This problem is frequent when modeling heterogeneous data using the same generative process [Wang, 2001]. We therefore optimize an alternative weighted likelihood. By defining $\eta \in [0, 1]$ denoting the importance given to the network information, we write the alternative function to optimize:

$$\widetilde{\mathcal{L}} = (1 - \eta)\mathcal{L}_w + \eta \mathcal{L}_d \quad (5)$$

Computing and annealing the gradient, we get the optimal solutions μ_i^* and $(\sigma_{i,r}^2)^*$:

$$\mu_i^* = \frac{\eta \sum_k \frac{c_{i,k} u_k}{\sigma_i^2} + (1-\eta) \sum_j \frac{a_{i,j} \mu_j}{\sigma_i^2} + (1-\eta) \sum_j \frac{a_{j,i} \mu_j}{\sigma_j^2}}{\eta \sum_k \frac{c_{i,k}}{\sigma_i^2} + (1-\eta) \sum_j \frac{a_{i,j}}{\sigma_i^2} + (1-\eta) \sum_j \frac{a_{j,i}}{\sigma_j^2}} \quad (6)$$

$$(\sigma_{i,r}^2)^* = \frac{\eta \sum_k c_{i,k} (\mu_{i,r} - u_{k,r})^2 + (1-\eta) \sum_j a_{i,j} (\mu_{i,r} - \mu_{j,r})^2}{\eta \sum_k c_{i,k} + (1-\eta) \sum_j a_{i,j}} \quad (7)$$

	Cora		Dblp		Nyt	
Train/Test ratio	10%	50%	10%	50%	10%	50%
DeepWalk	70.6 (2.0)	81.0 (0.7)	52.3 (0.4)	53.5 (0.2)	66.9 (0.7)	68.7 (0.9)
LSA	72.3 (1.9)	80.6 (0.7)	73.5 (0.2)	74.2 (0.2)	71.6 (1.0)	76.7 (0.7)
Concatenation	71.4 (2.1)	84.0 (1.1)	77.5 (0.2)	78.2(0.2)	77.9 (0.3)	81.1 (0.7)
TADW	81.9 (0.8)	87.4 (0.8)	74.8 (0.1)	75.5 (0.1)	75.8 (0.5)	79.4 (0.4)
AANE	79.8 (0.9)	84.4 (0.7)	73.3 (0.1)	74.2 (0.2)	71.7 (0.5)	76.9 (1.1)
GVNR-t	83.7 (1.2)	87.0 (0.8)	69.6 (0.1)	70.2 (0.2)	74.3 (0.4)	76.7 (0.6)
RLE	84.0 (1.3)	87.7 (0.6)	79.8(0.2)	81.2 (0.1)	77.7 (0.7)	80.0 (0.6)
VGAE	72.3 (1.7)	81.1 (0.7)	Memory Overflow		68.1 (0.8)	70.1 (0.6)
G2G	79.0 (1.5)	84.8 (0.7)	70.8 (0.1)	71.5 (0.2)	69.0 (0.5)	71.5 (0.8)
STNE	79.4 (1.0)	86.7 (0.8)	73.8 (0.2)	74.5 (0.1)	75.1 (0.7)	78.1 (0.6)
GELD	**84.3** (1.1)	**88.3** (0.4)	**81.63** (0.1)	**82.3** (0.1)	**78.5** (0.8)	**81.2** (0.3)

Table 1: Comparison of Micro-F1 results on a classification task for different train/test ratios. We provide the standard deviation in parentheses. GELD outperforms most recent methods on every dataset train/test ratio. On Dblp, it outperforms TADW by 7 points.

We maximize $\widetilde{\mathcal{L}}$ in each parameter and repeat the process until convergence. The optimization is iterative, and each parameter update depends on current values of the other parameters. Because of this dependency, we propose to adopt the Robbins-Monro method [Robbins and Monro, 1951] to prevent going too fast to a mediocre local solution. It yields good results in our experiments. The updated value of parameter μ_i at epoch k, given the optimal values at epoch k computed using Equation 6 we note $\mu_i^{*(k)}$, is:

$$\mu_i^{(k)} = \lambda^{(k)}\mu_i^{*(k)} + (1-\lambda^{(k)})\mu_i^{(k-1)} \tag{8}$$

We update σ_i^2 similarly. We propose to use $\lambda^{(k)} = (\delta k)^{-\gamma}$, $\gamma \leq 1$ as done in [Barkan, 2017] to ensure convergence conditions. $\delta \in [0,1]$ is the importance given to the optimal solution at the beginning of the optimization. In our experiments, we obtain higher results with low values of δ. In other words, we do not *trust* the first optimal solutions as the optimization is iterative.

The initial means and variances, noted by μ_i^0 and $(\sigma_i^2)^0$, are the empirical means and variances computed on the bag of word embedding only:

$$\begin{aligned} \mu_i^0 &= \frac{1}{|\mathcal{D}_i^w|}\sum_{f(w)\in\mathcal{D}_i^w} f(w) \\ (\sigma_{i,r}^2)^0 &= \frac{1}{|\mathcal{D}_i^w|}\sum_{f(w)\in\mathcal{D}_i^w} (f(w)_r - \mu_{i,r}^0)^2 \end{aligned} \tag{9}$$

Algorithm 1 GELD Algorithm

Input: $\mathcal{D}, U$
Parameters: η, λ, k
Output: μ, σ^2

for each: document i **do**
  initialize μ_i^0 and $(\sigma_i^2)^0$ according to Equation 9
end for
k=1
repeat
  for each: document i **do**
    compute $\mu_i^{*(k)}$ and $(\sigma_i^2)^{*(k)}$ using Eq. 6 and 7
    update $\mu_i^{(k)}$ and $(\sigma_i^2)^{(k)}$ with Equation 8
  end for
  $k = k+1$
until convergence
return μ, σ^2

4 Experiments

4.1 Datasets and Evaluation Tasks

We experiment on three datasets. Cora [Tu *et al.*, 2017] and Dblp [Tang *et al.*, 2008; Pan *et al.*, 2016] are two citation networks. Cora contains 2,211 abstracts of tagged scientific documents (7 classes) with 5,001 edges. Dblp has 60,744 documents titles (4 classes) and 52,914 edges between them. Additionally, we use the Nyt dataset from [Gourru *et al.*, 2020] containing press articles from January 2007. It has 4 classes, 5,135 documents and 3,050,513 edges. For each dataset, we filter the vocabulary by withdrawing stop words with the scikit-learn package[2] and we remove common and rare words (i.e., words appearing less than 4 times and in more than 25% of the documents). We obtain vocabulary size of 4,390 on Cora, 3,763 on Dblp and 6,407 for the Nyt dataset. Our method learns a mean and a variance for each document. In many real-world scenarios, downstream tasks require a single vector representation for a document. We therefore evaluate the relevance of using documents' mean in standard evaluation tasks: classification in Section 4.3 and link prediction in Section 4.4. We also provide qualitative insights on the variance possible use in Section 4.5. Besides, we also demonstrate the benefit of embedding documents and words in the same space.

4.2 Parameters Tuning

We compare our approach to recent baselines. We use four matrix factorization-based approaches: TADW, AANE, GVNR-t and RLE, and three deep neural network models: VGAE, Graph2Gauss and STNE. We also compare our method to DeepWalk, that considers the network information

[2]https://scikit-learn.org/

	Cora		Dblp		Nyt	
% edges hidden	50%	25%	50%	25%	50%	25%
DeepWalk	73.2 (0.6)	80.9 (1.0)	89.7 (0.0)	93.2 (0.2)	88.1 (0.0)	88.1 (0.0)
LSA	87.4 (0.6)	87.2 (0.8)	54.2 (0.1)	54.8 (0.0)	54.9 (0.0)	54.9 (0.1)
Combination	77.9 (0.3)	83.7 (0.8)	88.8 (0.0)	92.6 (0.3)	88.2 (0.0)	88.3 (0.0)
TADW	90.1 (0.4)	93.3 (0.4)	61.2 (0.1)	65.0 (0.5)	**88.5 (0.0)**	**88.5 (0.0)**
AANE	83.1 (0.8)	86.6 (0.8)	67.4 (0.1)	66.5 (0.1)	58.9 (0.2)	61.2 (0.2)
GVNR-t	83.9 (0.9)	91.5 (1.1)	88.1 (0.3)	91.4 (0.1)	61.2 (0.2)	61.3 (0.3)
RLE	94.3 (0.2)	94.8 (0.2)	89.3 (0.1)	91.2 (0.2)	77.5 (0.3)	77.8 (0.2)
VGAE	87.1 (0.4)	88.2 (0.7)	Memory Overflow		88.4 (0.0)	88.4 (0.0)
Graph2Gauss	92.0 (0.3)	93.8 (1.0)	88.0 (0.1)	92.1 (0.5)	88.3 (0.0)	88.2 (0.0)
STNE	83.1 (0.5)	90.0 (1.0)	45.6 (0.0)	53.4 (0.1)	88.4 (0.0)	88.4 (0.0)
GELD	**95.3 (0.1)**	**95.8 (0.1)**	**92.6 (0.2)**	**94.7 (0.3)**	88.3 (0.0)	88.3 (0.0)

Table 2: Comparison of mean AUC on a link prediction task for different percentages of edges hidden. We randomly remove the edges and repeat this procedure 3 times. We provide the standard deviation in parentheses. GELD outperforms most recent methods, up to 40 points for STNE on Dblp. On Nyt, it is comparable to TADW that achieves the best performance.

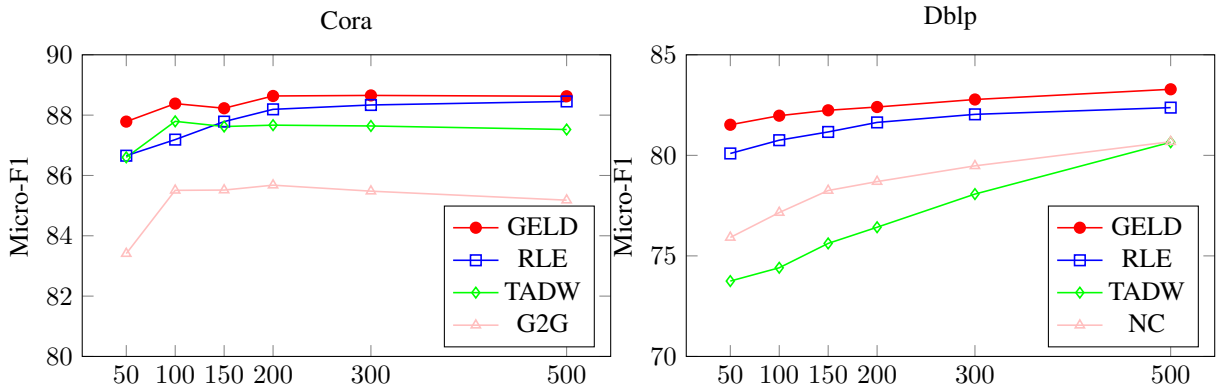

Figure 3: Accuracy with 50% test/train ratio by embedding dimension for the four best methods on the classification task on Dblp and Cora. NC stands for Naive Combination. GELD performs constantly better than competitors, even in low dimension.

Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
network	reinforcement	posterior	pac	genetic	casebased	ilp
networks	rl	bayesian	schapire	ga	knowledge	clause
neural	barto	gibbs	error	mutation	reasoning	kira
feedforward	qlearning	models	queries	gp	experiences	literals
multilayer	multiagent	model	set	search	design	relational

Table 3: Class descriptions on Cora. We show the top 5 words closest to the class centroids.

Title	Variance	Class
Collective Latent Dirichlet Allocation	545	3
Spatial Latent Dirichlet Allocation	605	1
Distributed Inference for Latent Dirichlet Allocation	590	1
Fast collapsed gibbs sampling for latent dirichlet allocation	513	3
Latent Dirichlet Co-Clustering.	398	3
A perceptual hashing algorithm using latent dirichlet allocation	604	2

Table 4: Six Nearest Neighbors in the embedding for the article "Latent Dirichlet Allocation" by Blei et al., obtained on Dblp. We provide the total variance summed by axes.

only, LSA [Deerwester *et al.*, 1990], that embeds the document with regard to the textual information, and a concatenation of embeddings obtained with those two methods we call "Concatenation" as done by [Yang *et al.*, 2015]. We use all the original implementations provided by the authors.

We use hyper parameters recommended by the authors, for those who use similar datasets, and grid-search hyper-parameters using the document classification task otherwise. For TADW, we use $\lambda = 0.2$ [Yang *et al.*, 2015] and dimension 200 for the reduced representation of documents' content. For AANE, we use optimal λ and ρ obtained via grid-search, as x_{min} for GVNR-t. For RLE we use $\lambda = 0.7$ and build the word vectors as specified by the authors. For VGAE, we use the author architecture, and $K = 1$ for Graph2Gauss. VGAE could not handle DBLP on our machine in reasonable time. For STNE, we determine depth using grid-search. For DeepWalk, we perform 40 walks of length 40 by nodes, and we set the window size to 10. We run all the experiments in parallel with 20 physical cores (Intel® Xeon® CPU E5-2640 v4 @ 2.40GHz) and 96GB of RAM. We use $r = 160$ as embedding dimension for every method following [Yang *et al.*, 2015].

Similarly, we report the optimal parameters for GELD obtained via grid-search on the classification task: $\delta = 0.1$, $\gamma = 0.2$, $\eta = 0.99$ for Cora, $\eta = 0.8$ for Dblp and $\eta = 0.95$ for Nyt. To learn word vectors, we adopt Skip-gram with negative sampling [Mikolov *et al.*, 2013] implemented in gensim[3]. We use window size of 15 for Cora, 10 for Nyt, 5 for DBLP (depending on documents size), and 5 negative examples for both. It only takes 46 seconds on Cora, 84 on DBLP and 42 on Nyt.

4.3 Classification Results

We adopt standard evaluation tasks following similar works [Yang *et al.*, 2015; Bojchevski and Günnemann, 2018]. We perform classification with a SVM classifier with L2 regularization. The optimal regularization is fixed, for each method and dataset, using grid search. We run the algorithms 10 times and report the mean Micro-F1 and standard deviation in Table 1.

Our method outperforms every competitor on each dataset (Table 1). To the exception of VGAE, linked document methods demonstrate higher accuracy than Deepwalk and LSA on Cora but they fail to outperform the Combination on Dblp and Nyt. GELD performs consistently better, possibly due to the impact of the variance during the learning phase: by inspecting Equation 6, we can see that the optimal value gives less weight to documents with high variance (i.e. uncertain or too general documents). Interestingly, with optimal L2 regularization, TADW yields better results than more recent baselines on every dataset with 50% train/test ratio. Figure 3 presents results with 50% train/test ratio with different dimensions for the four best models. GELD performs better in each dimension. TADW outperforms RLE in dimension 100, but adding dimension seems to deteriorate the results until convergence.

[3]https://radimrehurek.com/gensim/

4.4 Link Prediction Results

For the link prediction task, we hide a random set of edges to learn the representations. Then, a random set of unconnected pairs of documents is drawn as negative examples. We compute cosine similarity between pairs of documents in the hidden edge set and the negative example set. We then report Area Under the Curve (AUC) in Table 2, obtained with 3 runs and different percentages of hidden edges.

GELD outperforms baselines on Cora and Dblp, but is beaten by TADW on Nyt (Table 2). Nevertheless, every method except LSA, AANE and GVNR-t obtains AUC between 88.2 and 88.5, for $\%25$ and $\%50$ of edges hidden. This is due to the nature of the network: mean degree is around 500, i.e. 11% of the network. Even with 50% of edge hidden, the network information is well represented. Furthermore, TADW fails to produce good representations for link prediction on Dblp which is less dense while GELD performs constantly for different network topologies.

4.5 Qualitative Insights

As stated earlier, GELD represents words and documents in the same space. To demonstrate the interest of this property, we compute, for each annotated class of Cora, the average vector of document means μ_i inside this class. We present the five closest words to these class centroids in Table 3. It is easy to grasp the class content by looking at these descriptors. For example, Class 3 contains documents on Bayesian models and Class 1 on neural networks.

In Table 4, we present the six closest documents to "Latent Dirichlet Allocation", along with their variance (the sum of each axis variance). The papers "A perceptual hashing algorithm using latent Dirichlet allocation" and "Spatial Latent Dirichlet Allocation" both apply LDA to images. Therefore, they have a greater variance as they must cite papers from different areas. Meanwhile, "Latent Dirichlet Co-clustering" is in the same class than LDA, and it is more likely to cite documents from this class. Interestingly, papers with lower variance are all in the same class than "Latent Dirichlet Allocation" (Class 3).

5 Conclusion

We presented Gaussian Embedding of Linked Documents (GELD), a generative model that represents a document as a Gaussian Distribution in a word vector space. Learned through maximum likelihood estimation, it outperforms existing methods on Cora and Dblp, and it matches baselines on a New York Times dataset. Adding the variance during the learning phase seems to provide better vector representations since it gives less importance to documents with high variance when updating parameters. In further studies, we will focus on: 1) integrating the variance in downstream tasks, 2) developing a fully Bayesian version of our model to add priors on mean and variance.

Acknowledgments

We thank Adrien Guille for his support, and for helping to write the code used in our experiments.

References

[Barkan, 2017] Oren Barkan. Bayesian neural word embedding. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 3135–3143, 2017.

[Bojchevski and Günnemann, 2018] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *Proceeding of the International Conference on Learning Representations*, ICLR, 2018.

[Brochier *et al.*, 2019] Robin Brochier, Adrien Guille, and Julien Velcin. Global vectors for node representations. In *Proceedings of the World Wide Web Conference*, WWW, pages 2587–2593, 2019.

[Das *et al.*, 2015] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, 2015.

[Deerwester *et al.*, 1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[Gourru *et al.*, 2020] Antoine Gourru, Adrien Guille, Julien Velcin, and Julien Jacques. Document network projection in pretrained word embedding space. In *European Conference on Information Retrieval*, pages 150–157. Springer, 2020.

[Huang *et al.*, 2017] Xiao Huang, Jundong Li, and Xia Hu. Accelerated attributed network embedding. In *Proceedings of the SIAM International Conference on Data Mining*, SDM, pages 633–641, 2017.

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

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[Liu *et al.*, 2018] Jie Liu, Zhicheng He, Lai Wei, and Yalou Huang. Content to node: Self-translation network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1794–1802. ACM, 2018.

[Meng *et al.*, 2019] Zaiqiao Meng, Shangsong Liang, Hongyan Bao, and Xiangliang Zhang. Co-embedding attributed networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 393–401, 2019.

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

[Nikolentzos *et al.*, 2017] Giannis Nikolentzos, Polykarpos Meladianos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Multivariate gaussian document representation from word embeddings for text categorization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 450–455, 2017.

[Pan *et al.*, 2016] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI, pages 1895–1901, 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[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.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 990–998, 2008.

[Tu *et al.*, 2017] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1722–1731, 2017.

[Wang *et al.*, 2016] Suhang Wang, Jiliang Tang, Charu Aggarwal, and Huan Liu. Linked document embedding for classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 115–124. ACM, 2016.

[Wang, 2001] Steven Xiaogang Wang. *Maximum weighted likelihood estimation*. PhD thesis, University of British Columbia, 2001.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *International Joint Conference on Artificial Intelligence*, 2015.