12-2014

Probabilistic Latent Document Network Embedding

Tuan M. V. LE
Singapore Management University, vmtle.2012@smu.edu.sg

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Probabilistic Latent Document Network Embedding

Tuan M. V. Le
School of Information Systems
Singapore Management University
Singapore
email: vmtle.2012@phdis.smu.edu.sg

Hady W. Lauw
School of Information Systems
Singapore Management University
Singapore
email: hadywlauw@smu.edu.sg

***Abstract*—A document network refers to a data type that can be represented as a graph of vertices, where each vertex is associated with a text document. Examples of such a data type include hyperlinked Web pages, academic publications with citations, and user profiles in social networks. Such data have very high-dimensional representations, in terms of text as well as network connectivity. In this paper, we study the problem of embedding, or finding a low-dimensional representation of a document network that "preserves" the data as much as possible. These embedded representations are useful for various applications driven by dimensionality reduction, such as visualization or feature selection. While previous works in embedding have mostly focused on either the textual aspect or the network aspect, we advocate a holistic approach by finding a unified low-rank representation for both aspects. Moreover, to lend semantic interpretability to the low-rank representation, we further propose to integrate topic modeling and embedding within a joint model. The gist is to join the various representations of a document (words, links, topics, and coordinates) within a generative model, and to estimate the hidden representations through MAP estimation. We validate our model on real-life document networks, showing that it outperforms comparable baselines comprehensively on objective evaluation metrics.**

***Keywords*—document network; embedding; visualization; topic modeling; generative model; dimensionality reduction;**

I. Introduction

We are increasingly leaving a greater amount of digital footprints. Most of this content is unstructured, primarily in the form of text, but also with a high degree of connectivity among different pieces of content. We refer to this data type that can be described as a network of entities, where each entity is associated with text content, as a *document network*. There are examples abound of such data type. For one, we are familiar with academic publications and the citations linking them. For another, we frequently encounter Web pages and the hyperlinks among them. In social networks, such as LinkedIn, Facebook, or Twitter, we have user profiles and connections.

Due to their importance and wide applicability, document networks have been an intensive subject of research, particularly in information retrieval and link analysis. Relatively less attention has been paid to much-needed methods for conducting *exploratory analysis* on document networks. Analyzing a document network is very challenging because of the high-dimensional nature of the data. In one sense, a document can be expressed in terms of the occurrences of words (i.e., the dimensionality of text). In another sense, a document can also be expressed in terms of its connectivity to the other documents (i.e., the dimensionality of network).

Problem. In this work, we focus on the *embedding* problem. Given a document network, our objective is to "embed" (or reduce) the documents' high-dimensional representations (both in terms of text as well as network connnectivity) in a low-dimensional space that would still preserve as much of the "properties" of the original data as possible.

The resulting low-dimensional representations have several important applications. One major application towards exploratory analysis that we focus on is *visualization*. By interpreting the documents' reduced representations as coordinates on a two or three-dimensional space, we can produce a scatterplot visualization that is spatially informative in terms of the relative similarities or differences among entities. This form of visualization is grounded on the principle of dimensionality reduction [1], [2], rather than on aesthetics ground [3]. Such a visualization may serve as a component within a larger document organization system to assist users in categorizing documents, or as a part within a larger retrieval system. Other than visualization, the low-dimensional representations can also be used as a form of lossy compression, or for feature selection in learning tasks such as clustering or classification.

Embedding is a well-recognized problem in machine learning (see Section II). However, existing methods have not been designed with a document network in mind. We identify two issues that affect the fittingness of these methods for embedding a document network. The first issue is the *lack of connection between text and network*. Most methods have been designed either for embedding text documents, or for embedding a network. Obtaining either one embedding alone may offer a potentially distorted or incomplete view of the data. Obtaining both embeddings separately may produce two different representations that are not easily reconciled.

The second issue is the relative *lack of semantic interpretability*. Previous embedding methods produce low-dimensional representations that are not easily interpretable (other than as axes of the scatterplot visualization). In this respect, we are inspired by topic modeling [4], which obtains low-rank representations (i.e., topics) that are semantically interpretable (through high-probability words of each topic). However, topic modeling is not a solution to the embedding problem. For instance, to produce two-dimensional (2D) visualization, we can represent documents' topic distributions on a 2D simplex space, but this is only possible for three topics, which would be severely limiting as most applications of topic modeling require tens, if not hundreds, of topics [4].

Proposed Approach. To address the above issues, we propose a holistic and integrated approach based on two key princi-

ples. The first principle is to *embed both text and network representations of a document into a single unified low-rank representation*. This is grounded in the intuition that text content and network connectivity can inform each other. On one hand, text content can help to resolve ambiguities in the network. For instance, unobserved edges in a network may indicate either a genuine absence or a missing presence. If two documents are different in text content, the former is more likely than the latter. On the other hand, network connectivity can help to resolve the ambiguities in text through observed edges among documents that use different words for the same concept (synonymy), or missing edges among documents that use common words to refer to different concepts (polysemy).

The second principle is to *incorporate both a topic model and an embedding model within a single joint model*. To make our discussion more concrete, without loss of generality, we assume that the low-rank embedding takes the form of 2D visualization coordinates. This joint modeling is mutually beneficial to both topic modeling and visualization. By incorporating a topic model, we can infuse the visualization with semantic interpretability. Each point on a 2D scatterplot can be associated with the most likely topics or words [5]. By incorporating an embedding model, the mapping between topics and visualization may eventually offer a natural interface for user interaction to tune the underlying topic model [6].

We are thus motivated to tie together the four representations of each document in a document network, namely: the two high-dimensional representations in terms of word occurrences and network connectivity respectively, the intermediate representation in terms of a topic distribution as in topic modeling, as well as the low-rank representation in terms of visualization coordinates as in embedding. One framework to join these disparate representations is *generative modeling*, a probabilistic model for the generation of observable data through modeling random variables (that encode the representations mentioned above). Generative modeling has been the bedrock for much of the topic modeling works that build on [4], though it has not been as widely applied to embedding.

Contributions. *First*, our novelty arises from the holistic approach to topic-based embedding of document networks. In comparison, previous works, reviewed in Section II, have attempted this as separate segments, namely: embedding of documents, embedding of networks, or topic modeling, but have not recognized the embedding of a document network as a distinct problem to be addressed in its own entirety.

Second, to address this problem, we develop a generative modeling approach, and propose a model called PLANE, which stands for *P*robabilistic *LA*tent Document *N*etwork *E*mbedding. In Section III, we describe the process of generation of observable data (text and network) from latent representations (topics and visualization coordinates). In Section IV, we outline an MAP inference algorithm to estimate the hidden parameters of this model through EM.

Third, to validate this model, we conduct comprehensive experiments (Section V) on four real-life document networks derived from a benchmark collection of academic publications. We compare our model, quantitatively as well as qualitatively, against comparable baselines on both aspects (embedding and topic modeling) on a number of objective evaluation metrics.

II. Related Work

In terms of embedding. While we focus on embedding a document network, there are previous efforts on embedding documents, or embedding a network, which we review below.

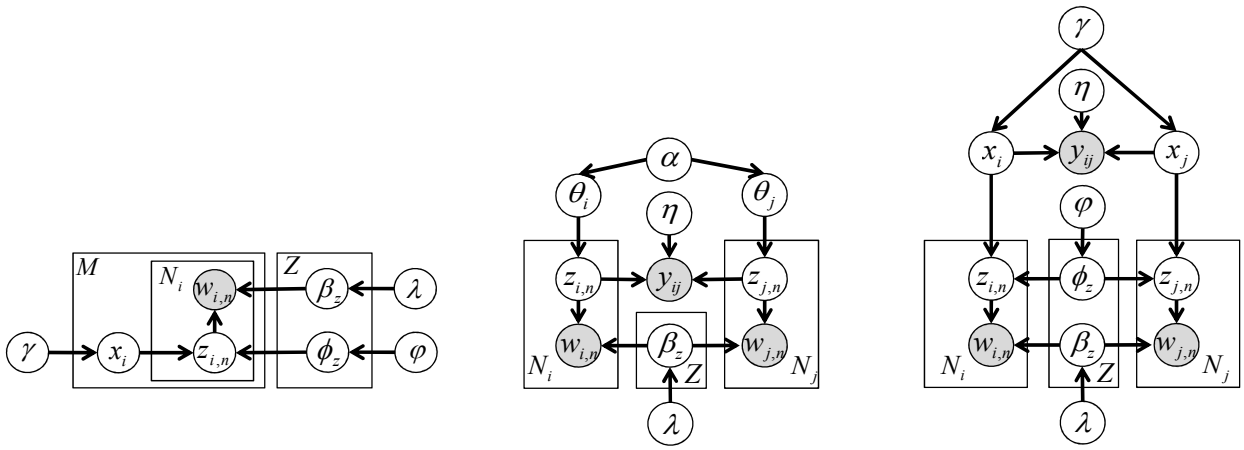
To embed documents, we can employ embedding techniques, which take as input M high-dimensional vectors $\{v_i\}_{i=1}^M$ and generate as output M low-rank vectors $\{x_i\}_{i=1}^M$. For instance, the v_i's may be the bag-of-words representations of documents, and the x_i's may be visualization coordinates. Good embedding produces x_i's that represent the v_i's "faithfully". In traditional embedding [7], [8], [9], this criterion is frequently formulated as preserving the distances among v_i's in the distances among x_i's. More recent approaches [2], [10] formulate this in terms of probabilities.

Recent works advocate having an intermediate representation, which is the topic space. The closest one to ours is PLSV [5], which pioneers the integration of topic modeling and visualization in a joint model. Figure 1(a) shows the graphical model of PLSV. Its generative process is as follows. For each topic z, we draw its word distribution β_z from a Dirichlet with parameter λ, as well as its coordinate ϕ_z from a Normal distribution with mean 0 and variance φ^{-1}. For each document v_i, we draw its coordinate x_i from Normal with mean 0 and variance γ^{-1}. To generate each of the N_i words in v_i, we draw a topic $z_{i,n}$ based on the relative distance between x_i and topic coordinates, and draw a word from the selected topic's word distribution $\beta_{z_{i,n}}$. Since PLSV models only documents, our model builds on it by integrating a network model.

There also exist other joint embedding models that focus on orthogonal features that complement, rather than compete with this work. Their innovations center around manifold regularization [11] or spherical representation [12], which could potentially be incorporated into our problem independently. Importantly, these works do not seek to model network links.

To embed a network, we can employ graph embedding techniques, of which there are broadly two main categories of approaches. The first category is *spectral embedding*, where the focus is on dimensionality reduction. For instance, the adjacency matrix representing the graph can be used as input to SVD [13] or PCA [1], whose objective is compressibility (preserving the variance in the data). To produce a low-dimensional embedding, the first few principal eigenvectors (with the largest eigenvalues) can be used as the coordinates $\{x_i\}_{i=1}^M$. This approach has been widely used for various large-scale graphs [14]. Building on this, SPE [15] attempts to preserve the neighboring structure as well, but since it is formulated as semidefinite programming, it is computationally very expensive for large-scale graphs [14].

The second category is *spring embedding*, also known as force-directed graph drawing. One example is the Fruchterman and Reingold layout [16] (FR-layout), which simulates a force system where spring-like attractive forces on links pull connected nodes together. The simulation is repeated iteratively till a mechanical equilibrium state is reached (energy minimization). Another approach Kamada and Kawai layout [17] (KK-layout) is also based on the idea of a balanced spring system and energy minimization, but achieves faster convergence due to the use of derivatives. These layouts are commonly found in graph visualization programs [18], [19].



(a) PLSV (b) RTM (two-document segment) (c) PLANE (two-vertex segment)

Fig. 1. Graphical Models of PLSV (a), RTM (b) and PLANE (c)

In terms of topic modeling. Topic modeling is originally designed for documents [4], where each document is associated with a topic distribution, and each topic is associated with a word distribution. There also exists similar statistical modeling of networks as surveyed in [20]. For instance, in mixed membership stochastic blockmodel [21], each user is associated with a distribution over "communities", which explain the generation of links among users.

Recognizing the wide availability and applicability of document networks as a distinct data type, subsequent works seek to combine text and networks. One example is through a regularization framework [22], which however is not a joint model, and therefore does not model the generation of links. Yet others [23], [24], [25], [26] focus on modeling the generation of both text documents and network links jointly.

Our work builds on the Relational Topic Model (RTM) [23], which we review briefly below. Its graphical model is shown in Figure 1(b). Each document v_i is associated with a topic distribution θ_i. To generate the n^{th} word in v_i, we first pick a topic $z_{i,n}$ from θ_i, then pick a word $w_{i,n}$ from $z_{i,n}$'s topic multinomial $\beta_{z_{i,n}}$. θ_i and β_z have Dirichlet priors of α and λ respectively. In turn, each link y_{ij} between a pair of documents v_i and v_j is generated from a link probability function based on the topics that occur in v_i and v_j. The more they share common topics, the more likely there to be a link between them. There are a number of key differences between RTM and PLANE. Most importantly, we need to consider the low-rank embedding objective. We also model link generation based on coordinates instead of topic distributions. In our model likelihood, we also incorporate "virtual" negative links, not just observed positive links (see Section III).

There are also some works on visualizing topic models [27], [28], [29], [30], where the focus is on visualizing which topics are important in a corpus, or which words are important in a topic. While they convey some information visually, they are orthogonal to our objective. They are not low-rank embedding techniques, and do not produce a low-rank representation for each document, which can also be used in non-visualization applications such as dimensionality reduction or compression.

III. Generative Model

Here, we describe the framework and the generative process of our proposed model PLANE, whose graphical representation in terms of a plate diagram is shown in Figure 1(c).

Framework. We consider as input a document network, represented as a graph $G = (V, E)$. V is a set of M vertices. Each vertex $v_i \in V$ refers to a document, and is associated with a bag of words. We denote $w_{i,n}$ to be the n^{th} word token in v_i, and N_i to be the total number of word tokens in v_i. Each token has a symbol drawn from the vocabulary of words W. In turn, E is a set of edges in G, where each edge $e_{ij} \in E$ connects two vertices v_i and v_j. In this work, we would model an undirected graph, i.e., $e_{ij} = e_{ji}$, as our emphasis is on connectivity, rather than on directionality. The model could still apply to directed graphs by dropping the edge directions. In this paper, we use the term "edge" and "link" interchangeably.

As output, we aim for dual objectives as follows.

- *Embedding*: For each vertex v_i, we seek to learn its low-rank representation x_i, expressed as coordinates on a D-dimensional space. In this paper, which is framed in terms of embedding in a visualization space, we assume $D = 2$, without loss of generality.
- *Topic Modeling*: For each vertex v_i, we also seek to learn its representation in the topic space, expressed as a probability distribution $\{\mathrm{P}(z|v_i)\}_{z=1}^{Z}$ over a specified number of Z topics, where $D \ll Z \ll |W|$ is expected in most cases. Correspondingly, each topic z is associated with β_z, a probability distribution over words $\{\mathrm{P}(w|\beta_z)\}_{w \in W}$, where words with high probabilities provide semantic meaning to the topic.

To unify the dual objectives above, we need to concretely define how the two objectives are correlated with each other. This can be achieved by a mapping function from the visualization space to the topic space. Towards realizing this mapping, we associate each topic z with a visualization coordinate ϕ_z in the same D-dimensional space. If we model each ϕ_z to be the mean of a unit-variance Gaussian, and x_i to have been drawn from a mixture of Gaussians centered at ϕ_z's (with uniform

mixture weights), we can express $\mathrm{P}(z|v_i)$ as the *responsibility* of the z's component of the Gaussian mixtures [31], as shown in Equation 1, which has also been used in [10], [5]. Here, $||\cdot||$ is the Euclidean norm defined on the visualization space, and $\Phi = \{\phi_z\}_{z=1}^{Z}$ refers to the collection of all topic coordinates.

$$\mathrm{P}(z|v_i) = \mathrm{P}(z|x_i, \Phi) = \frac{\exp(-\frac{1}{2}||x_i - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_i - \phi_{z'}||^2)} \quad (1)$$

This mapping has an intuitive meaning. The closer is x_i to ϕ_z in the visualization space, the greater is the probability of topic z in vertex v_i. It follows that if two vertices are close in the visualization space, they will also share similar topic distributions, thus encoding the above-mentioned embedding objective of finding similar low-rank representations for documents with similar high-dimensional representations.

Generative Process. We now describe the full generative process of our proposed model PLANE below.

1) For each topic $z = 1, \ldots, Z$:
   a) Draw z's word distribution:
   $$\beta_z \sim \mathrm{Dirichlet}(\lambda)$$
   b) Draw z's coordinate:
   $$\phi_z \sim \mathrm{Normal}(0, \varphi^{-1} I)$$
2) For each vertex v_i, where $i = 1, \ldots, M$:
   a) Draw v_i's coordinate:
   $$x_i \sim \mathrm{Normal}(0, \gamma^{-1} I)$$
   b) For each word $w_{i,n}$, where $n = 1, \ldots, N_i$:
      i) Draw a topic:
      $$z_{i,n} \sim \mathrm{Categorical}(\{\mathrm{P}(z|x_i, \Phi)\}_{z=1}^{Z})$$
      ii) Draw a word:
      $$w_{i,n} \sim \mathrm{Categorical}(\beta_{z_{i,n}})$$
3) For each pair of vertices v_i and v_j:
   a) Draw e_{ij}'s binary indicator:
   $$y_{ij} \sim \mathrm{Bernoulli}\left(\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta)\right)$$

Step 1 shows the generation of the parameters for each topic z. Like classical topic models [4], its word distribution β_z has a Dirichlet prior (with hyper parameter λ). Its visualization coordinate ϕ_z has a Normal prior (centered at 0 with precision φ). The mean at 0 determines the locality of the visualization.

Step 2a shows the generation of parameter for each vertex v_i, which is its visualization coordinate x_i, from a Normal distribution with mean 0 and precision γ. Following Equation 1, this coordinate is mapped to v_i's representation in the topic space, which is a probability distribution over the Z topics, i.e., $\{\mathrm{P}(z|x_i, \Phi)\}_{z=1}^{Z}$.

Step 2b encodes the *document embedding* step, where the "embedded" low-dimensional representation x_i generates the high-dimensional text representation (bag of words). Based on x_i's topic space representation, we repeatedly draw a topic $z_{i,n}$ from $\{\mathrm{P}(z|x_i, \Phi)\}_{z=1}^{Z}$, followed by drawing a word $w_{i,n}$ from the topic's word distribution $\beta_{z_{i,n}}$.

Step 3 encodes the *network embedding* step, where the "embedded" low-dimensional representation x_i generates the high-dimensional network representation (i.e., which other vertices v_i is connected to). We associate each edge e_{ij} with a binary random variable denoted by y_{ij}, with a value of 1 if the edge is present ($e_{ij} \in E$), and 0 otherwise ($e_{ij} \notin E$). This random variable is drawn from a Bernoulli distribution. The Bernoulli parameter is denoted by $\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta) \in [0, 1]$, which determines the probability that an edge exists between two vertices based on the vertices' latent coordinates x_i and x_j, and a parameter η (to be defined shortly).

Naturally, for network embedding, we desire that connected vertices would share similar embedded parameters. In that sense, the more similar are x_i and x_j, the higher is the $\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta)$. Since x_i and x_j are coordinates, their "similarity" can be measured in terms of Euclidean distance $||x_i - x_j||$. To transform this distance into a probability value, we adopt the exponential probability function [31], as shown in Equation 2, where η is a parameter to be learned. In this work, we seek to study the connectivity hypothesis itself. While there could be other ways to realize the edge probability function, we keep the exploration in that direction to future work.

$$\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta) = \exp(-\eta \cdot ||x_i - x_j||^2) \quad (2)$$

Our modeling of edge probability function based on distance ties together all the representations (document, networks, visualization coordinates, topics). This sets us apart from others that model only subsets of these representations (e.g., documents and networks but not visualization [23], documents and visualization but not networks [5]).

Model Likelihood. PLANE's graphical model in Figure 1(c) shows how the various representations are related to one another. Importantly, the observed (shaded) variables are only the words $\{w_{i,n}\}$ in vertex v_i, as well as the edges' indicators $\{y_{ij}\}$. Equation 3 shows the log-likelihood function for generating these observed variables in the input graph $G = (V, E)$ based on the hidden parameters, such as embedding coordinates $\{x_i\}$ and topic multinomials $\{\beta_z\}$. The first component corresponds to the text associated with vertices in V. The second component corresponds to the edges.

$$\mathcal{L}(G) = \sum_{i=1}^{M} \sum_{n=1}^{N_i} \log \sum_{z=1}^{Z} \mathrm{P}(w_{i,n}|\beta_z)\mathrm{P}(z|x_i, \Phi) + \sum_{ij} \log \mathrm{P}(y_{ij}|x_i, x_j, \eta) \quad (3)$$

We need to decide how to model observed and unobserved edges. One way is to set $y_{ij} = 1$ when an edge is observed between vertices v_i and v_j, and $y_{ij} = 0$ otherwise. As stated in [23], this approach may be inappropriate when the absence of an edge cannot be used as evidence for $y_{ij} = 0$. To resolve this, they decided to model only observed edges (i.e, y_{ij} is either 1 or unobserved) [23]. While doing so can speed up computation, it falls short of the full discriminating power because the hidden structure of the corpora cannot be described fully only based on the positive observations ($y_{ij} = 1$). The negative observations ($y_{ij} = 0$) should also be considered.

Due to the reason above, we decide to model both observed and unobserved edges. We treat observed edges as positive observations ($y_{ij} = 1$). For unobserved edges, we assume that only a subset of them would be negative ($y_{ij} = 0$). It is not necessary to specify which particular edges are negative. Let ρ be the expected number of these "virtual" negative observations (to be learned from the data), and $U = \frac{M \times (M-1)}{2} - |E|$ be the total number of unobserved edges. The expected log likelihood of these negative observations is as follows.

$$\frac{\rho}{U} \sum_{e_{ij} \notin E} \log \mathrm{P}(y_{ij} = 0 | x_i, x_j, \eta) \tag{4}$$

Therefore, the final log-likelihood of our model will be computed as follows.

$$\begin{aligned}\mathcal{L}(G) = & \sum_{i=1}^{M} \sum_{n=1}^{N_i} \log \sum_{z=1}^{Z} \mathrm{P}(w_{i,n}|\beta_z)\mathrm{P}(z|x_i, \Phi) + \\ & \sum_{e_{ij} \in E} \log \mathrm{P}(y_{ij} = 1 | x_i, x_j, \eta) + \\ & \frac{\rho}{U} \sum_{e_{ij} \notin E} \log \mathrm{P}(y_{ij} = 0 | x_i, x_j, \eta)\end{aligned} \tag{5}$$

IV. Parameter Estimation

We estimate the parameters based on maximum a posteriori (MAP) estimation using EM algorithm [32]. The parameters that need to be estimated are the word probabilities $\{\beta_z\}_{z=1}^{Z}$, the topic coordinates Φ, the vertex coordinates $\{x_i\}_{i=1}^{M}$. η and ρ will also be learned from data. Since η and ρ are positive, let $\eta = \eta_{sqr}^2$ and $\rho = \rho_{sqr}^2$. Instead of directly learning η and ρ, we will learn η_{sqr} and ρ_{sqr} to avoid imposing the positivity constraints when optimizing the likelihood. We denote the collection of the unknown parameters as Ψ.

The conditional expectation of the complete-data log likelihood in MAP estimation with priors is:

$$\begin{aligned}\mathcal{Q}(\Psi|\hat{\Psi}) = & \sum_{i=1}^{M} \sum_{n=1}^{N_i} \sum_{z=1}^{Z} \mathrm{P}(z|i, n, \hat{\Psi}) \log \left[\mathrm{P}(z|x_i, \Phi)\mathrm{P}(w_{i,n}|\beta_z)\right] \\ & + \sum_{i=1}^{M} \log(\mathrm{P}(x_i)) + \sum_{z=1}^{Z} \log(\mathrm{P}(\phi_z)) + \sum_{z=1}^{Z} \log(\mathrm{P}(\beta_z)) \\ & + \sum_{e_{ij} \in E} \log \mathrm{P}(y_{ij} = 1 | x_i, x_j, \eta) + \\ & + \frac{\rho}{U} \sum_{e_{ij} \notin E} \log \mathrm{P}(y_{ij} = 0 | x_i, x_j, \eta)\end{aligned}$$

$\hat{\Psi}$ is the current estimate. $\mathrm{P}(z|i, n, \hat{\Psi})$ is the class posterior probability of the i^{th} document and the n^{th} word in the current estimate. $\mathrm{P}(\beta_z)$ is a symmetric Dirichlet prior with parameter λ for word probability β_z. $\mathrm{P}(x_i)$ and $\mathrm{P}(\phi_z)$ are Gaussian priors with a zero mean and a spherical covariance for the document coordinates x_i and topic coordinates ϕ_z. We set the hyper-parameters to $\lambda = 0.01$, $\varphi = 0.1M$ and $\gamma = 0.1Z$ as in [5].

In the E-step, $\mathrm{P}(z|i, n, \hat{\Psi})$ is updated as follows.

$$\mathrm{P}(z|i, n, \hat{\Psi}) = \frac{\mathrm{P}(z|\hat{x}_i, \hat{\Phi})\mathrm{P}(w_{i,n}|\hat{\beta}_z)}{\sum_{z'=1}^{Z} \mathrm{P}(z'|\hat{x}_i, \hat{\Phi})\mathrm{P}(w_{i,n}|\hat{\beta}_{z'})}$$

In the M-step, by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t β_{zw}, the next estimate of word probability β_{zw} is as follows.

$$\beta_{zw} = \frac{\sum_{i=1}^{M} \sum_{n=1}^{N_i} \mathrm{I}(w_{i,n} = w)\mathrm{P}(z|i, n, \hat{\Psi}) + \lambda}{\sum_{w'=1}^{W} \sum_{i=1}^{M} \sum_{n=1}^{N_i} \mathrm{I}(w_{i,n} = w')\mathrm{P}(z|i, n, \hat{\Psi}) + \lambda W}$$

.

$\mathrm{I}(.)$ is the indicator function. ϕ_z and x_i cannot be solved in a closed form, and are estimated by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ using quasi-Newton [33].

We compute the gradients of $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t ϕ_z, x_i, ρ_{sqr}, η_{sqr} respectively as follows.

$$\begin{aligned}\frac{\partial Q}{\partial \phi_z} = & \sum_{i=1}^{M} \sum_{n=1}^{N_i} \big(\mathrm{P}(z|x_i, \Phi) - \mathrm{P}(z|i, n, \hat{\Psi})\big)(\phi_z - x_i) - \varphi\phi_z \\ \frac{\partial Q}{\partial x_i} = & \sum_{n=1}^{N_i} \sum_{z=1}^{Z} \big(\mathrm{P}(z|x_i, \Phi) - \mathrm{P}(z|i, n, \hat{\Psi})\big)(x_i - \phi_z) - \gamma x_i \\ & - \sum_{e_{ij} \in E} 4\eta_{sqr}^2 (x_i - x_j) \\ & + \frac{4\rho_{sqr}^2 \eta_{sqr}^2}{U} \sum_{e_{ij} \notin E} (x_i - x_j) \frac{\exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)}{(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2))} \\ \frac{\partial Q}{\partial \rho_{sqr}} = & \frac{2\rho_{sqr}}{U} \sum_{e_{ij} \notin E} \log(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)) \\ \frac{\partial Q}{\partial \eta_{sqr}} = & -2\eta_{sqr} \sum_{e_{ij} \in E} ||x_i - x_j||^2 \\ & + \frac{2\rho_{sqr}^2 \eta_{sqr}}{U} \sum_{e_{ij} \notin E} ||x_i - x_j||^2 \frac{\exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)}{(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2))}\end{aligned}$$

V. Experiments

The objective of experiments is to validate the effectiveness of our topic-based embedding method PLANE. First, we describe the setup, in terms of the datasets (V-A) as well as the comparable baselines (V-B). Thereafter, we conduct the primary comparison in terms of the goodness of embedding coordinates (V-C). This is done both quantitatively by using the coordinates as features in a classification task, as well as qualitatively by inspecting some example visualizations. Finally, we compare the effectiveness of PLANE as a topic model for document network (V-D).

A. Datasets

For repeatability, we rely on a publicly-available benchmark data source, which is a representative example of document networks. Cora[1] is a collection of academic publications and their citation networks from various categories [34]. Each document is an abstract. Two documents are connected by an undirected edge if one document cites the other. Documents in Cora are divided into general categories. Following [35], we use the following categories as four separate datasets: *Data Structure* (DS), *Hardware and Architecture* (HA), *Machine Learning* (ML), and *Programming Language* (PL).

[1] http://people.cs.umass.edu/~mccallum/data/cora-classify.tar.gz

TABLE I. DATASETS OF CORA

	#classes	#documents	#edges	vocabulary
Data Structure (DS)	9	570	1336	3085
Hardware and Architecture (HA)	6	223	515	2073
Machine Learning (MA)	7	1980	5638	4431
Programming Language (PL)	9	1553	4851	4105

TABLE II. COMPARATIVE METHODS

	Document embedding	Network embedding	Topic model	Joint model
PLANE	✓	✓	✓	✓
RTM+PE	✓	✓	✓	
PLSV	✓		✓	✓
KK		✓		
SVD		✓		

For each dataset, each document is further classified into one of several sub-fields. For DS, the nine sub-fields are: *Computational Complexity*, *Computational Geometry*, *Formal Languages*, *Hashing*, *Logic*, *Parallel*, *Quantum Computing*, *Randomized*, and *Sorting*. The other three datasets each have their own respective sub-fields as well. We treat these sub-fields as class labels, which are not used as input, but rather for evaluation in Section V-C. We also remove documents that are not connected to any document within the same dataset.

Table I lists the sizes of these datasets in terms of the number of classes, documents, edges, and the vocabulary sizes.

B. Comparative Methods

In Table II, we list the methods that we will be comparing, and highlight the properties of each method.

Proposed approach. As a topic-based embedding model, **PLANE** is our method that models both document and network embeddings, as well as topic model in a joint manner.

Pipelined approach. Since there is no other existing model with all the properties, the most direct baseline is a composite approach that pipelines two methods. First, a document network is reduced into a set of topic distributions (one for each document) by the relational topic model RTM [23]. As recommended in [23], α is set such that the total mass of the Dirichlet hyperparameter is 5. λ is set to 0.01 (same as PLANE and PLSV) following [5]. Then, these topic distributions are embedded in a 2D visualization space using PE [10], an embedding approach designed for probability distributions. This composite, called **RTM+PE**, is our primary baseline that allows us to validate the utility of modeling both topics and embedding jointly, as opposed to modeling them separately.

Document embedding. While document embedding is not a direct baseline, because it does not model the network aspect, a comparison to it allows us to evaluate the contribution of network embedding to our model. As a representative of document embedding, the closest one to ours is **PLSV** [5], which models topic-based document embedding.

Network embedding. Network embedding is not a direct baseline either, because it models neither documents nor topics. For completeness, we include a comparison to two categories of network embedding. As a representative of spring embedding, we use **KK** layout [17]. As a representative of spectral embedding, we use **SVD** [13]. These are among the most popular methods in their respective categories.

For the probabilistic methods (i.e., PLANE, RTM+PE, PLSV), we average the performance numbers across ten independent runs. For each run, the parameter estimation is based on 100 learning iterations. We set the number of iterations for each Gibbs sampling E-step of RTM to 1000. As much as possible, we have used public implementations. For RTM, we use its original authors' implementation[2]. For KK, we use the implementation in the JUNG library[3]. For SVD, we use the implementation in R software[4]. We implement our own method PLANE, as well as the baselines PE and PLSV[5].

C. Embedding

As our primary objective is to embed a document network in a low-dimensional space, we first evaluate the quality of the resulting embedding coordinates against all the baselines.

Metric. Since embedding seeks to "preserve" the original data as much as possible in the reduced dimensions, one well-accepted means for embedding evaluation is to use the low-dimensional coordinates as features in a learning task [5], [15]. Since class labels are available (but not used as input), we conduct evaluation based on classification. The more the features help to predict the classes, the more the low-dimensional coordinates (features) have preserved the properties of the data (embedding objective). Because what is evaluated are the features, we use a simple k-nearest neighbor classification. For each document, we hide its true label, and predict its label as the majority label among its k-nearest neighbors (based on Euclidean distance in the embedding coordinates). The metric $accuracy(k)$ is the fraction of documents for which the predicted label matches the hidden true label.

Vary number of topics. First, we investigate the effect of the number of topics Z on accuracy. Figure 2 shows the $accuracy(10)$ values for the four datasets. Similar observations regarding the relative standings of various methods can be made for other k values as well.

The accuracy values are relatively stable across different numbers of topics. Figure 2(b) for HA shows a small increase from $Z = 10$ to $Z = 20$, after which accuracies remain flat. For subsequent experiments, we will use $Z = 20$ by default.

In absolute terms, PLANE achieves high accuracies of around 0.8 for HA, and 0.7 for DS, ML, and PL. This is notable as PLANE only uses 2-dimensional features for the k-NN classification. This helps to validate the quality of the embedding in preserving the high-dimensional representations.

In relative terms, PLANE has higher accuracies than all the baselines. This outperformance is statistically significant in all cases. It outperforms RTM+PE, which helps to validate the utility of having a joint modeling of embedding and topics. It also outperforms document embedding (PLSV) and network embedding (KK, SVD), which justifies embedding documents and network with a unified low-rank representation.

Among the baselines themselves, there is no consistent ordering across datasets in terms of which is better. For the

[2]http://cran.r-project.org/web/packages/lda/

[3]http://jung.sourceforge.net/

[4]http://stat.ethz.ch/R-manual/R-devel/library/base/html/svd.html

[5]We could not find a public or an original implementation by their authors.

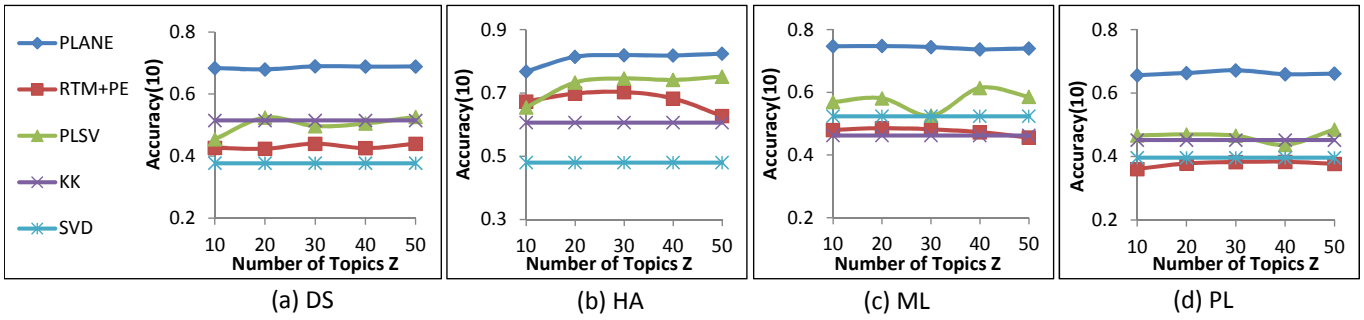


Fig. 2. Accuracy at $k = 10$ nearest neighbors for varying number of topics Z

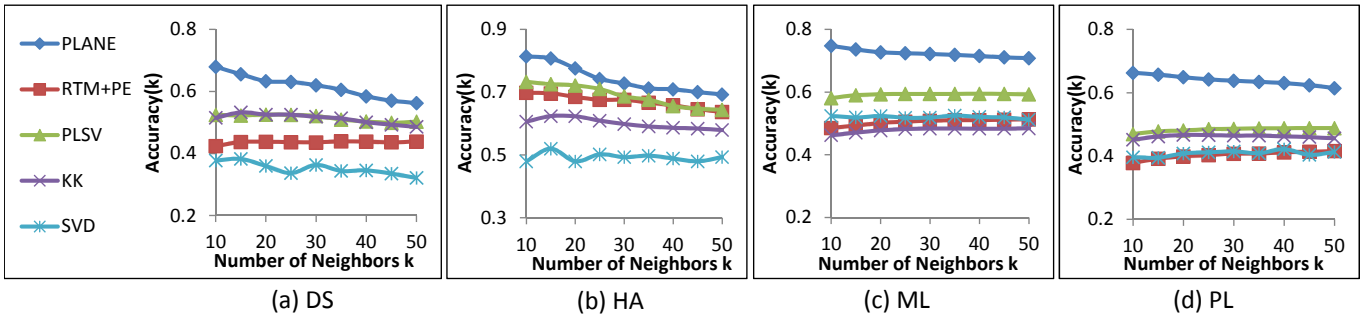


Fig. 3. Accuracy at varying k nearest neighbors for $Z = 20$ topics

network embedding KK and SVD, accuracies are flat across different Z's because they are not topic-based approaches.

Vary neighborhood size. We now investigate how the accuracy is affected by different neighborhood sizes for the k-NN classification. Figure 3 shows the $accuracy(k)$ values for the four datasets when $Z = 20$. As shown by the earlier consistency among different Z's, similar observations can be made for other number of topics as well. For all the methods, there is a general tendency that accuracy decreases at larger k's. This is reasonable, because as k increases, we use a greater number of neighbors to arrive at the classification, which dilutes the quality of classification. Importantly, in relative terms, the outperformance by PLANE still stands across different k's, for the reasons explained above.

Visualization. To gain a sense of the visualization quality obtained by embedding the documents in a two-dimensional scatterplot, we show several examples for the various datasets.

We begin with the Data Structure (DS) dataset. Figure 4(a) shows the visualization generated by PLANE. Each document is a dot placed in the scatterplot according to their 2D embedding coordinates. Each dot is painted with a color that represents its sub-field or class. The legend specifies the colors assigned to each class. Edges are lines between two connected documents. There are two key observations. First, note how the different classes are quite well-separated from one another (the class information itself was never used for learning). The red *Parallel* documents are at the lower right, while the grey *Sorting* documents are at the center. Second, note how the edges are hardly visible, which is a good sign because it means connected documents are placed as close neighbors in the visualization space. Otherwise, we would have witnessed criss-crossing lines all over. These observations support the hypothesis that having a joint model for embedding documents and network results in better embedding overall.

Still for the DS dataset, Figure 4(b) for RTM+PE does not show a good separation between classes, and has many criss-crossing edge lines. This is because while the network is used to influence the topic distributions, because of the disjoint embedding through PE, the network effect does not get enforced in the embedding process. PLSV in Figure 4(c) looks more coherent than RTM+PE, but not as clean as PLANE. For one thing, the grey *sorting* documents are spread apart, while in PLANE they are clustered together. For another thing, there are still criss-crossing edges due to separation of connected documents as PLSV models text content only. In contrast, KK in Figure 4(d) models only network embedding. Thus connected edges are tightly clustered together. However, because it does not model content, documents of the same class without connection to each other are spread far apart (e.g., the red *parallel*). Due to space constraint, here we do not show SVD (which has the lowest accuracy for DS in Figure 3(a)).

To show that the observations for PLANE apply to other datasets as well, in Figure 5, we show PLANE's visualization for HA, ML, and PL datasets. Evidently, PLANE can group together documents of the same class well, and place connected documents as neighbors in the visualization space.

D. Topic Modeling

While our main objective is to improve the embedding of document networks, it is important to ensure that the gains in embedding and visualization quality have not come at the expense of the topic model. Since ours is a topic model for a document network, the appropriate comparison is to a baseline that also models the generation of both words and links,

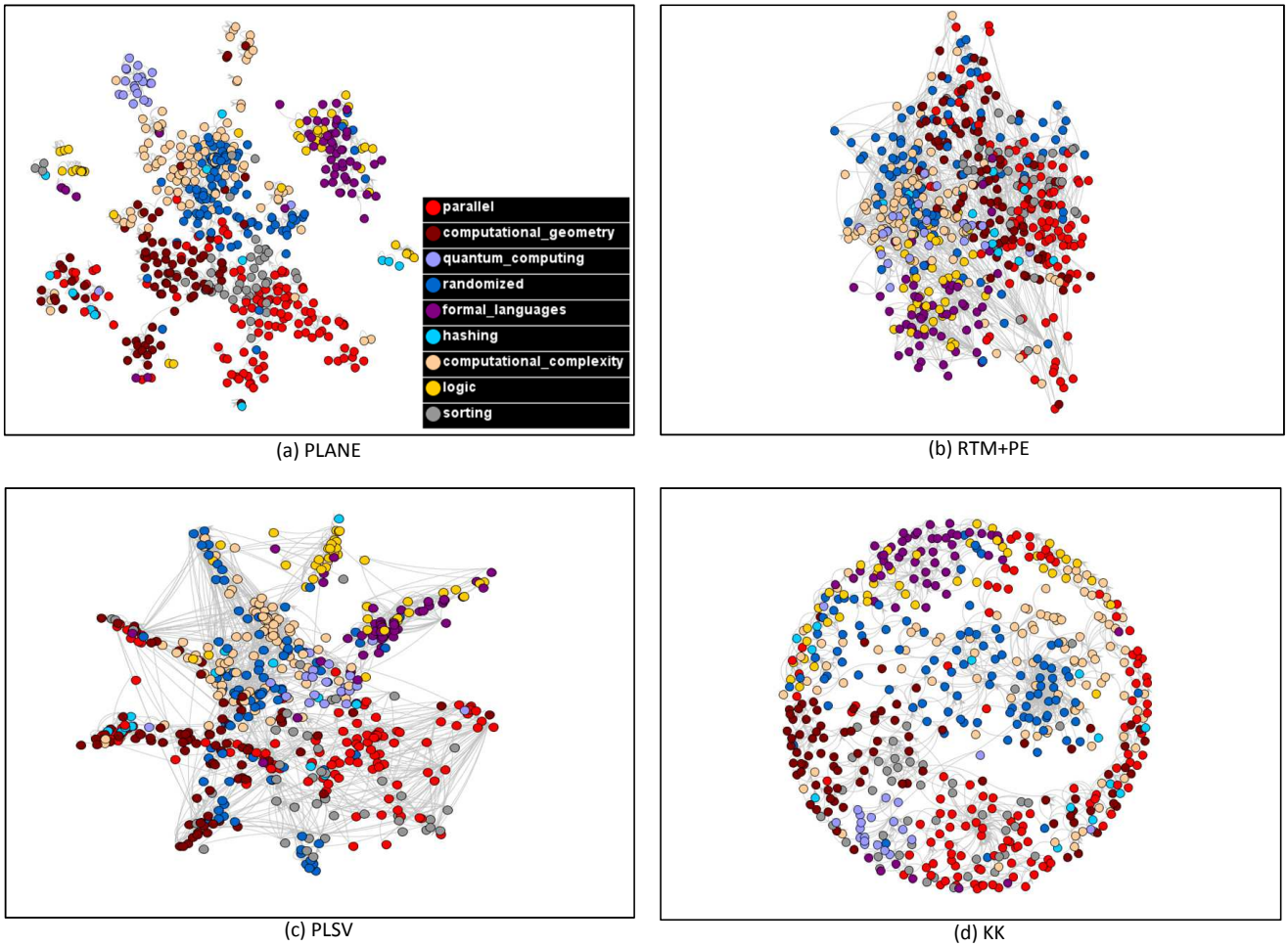


Fig. 4. Visualizations of Data Structure (DS) dataset for $Z = 20$ (best seen in color)

namely RTM [23]. In the following, we compare PLANE and RTM, in terms of the topic words, as well as the links.

1) **Topic Interpretability:** As modeling topics with embedding is to improve the interpretability of embedding, we evaluate the topics on how interpretable the topic words are.

Metric. Pointwise Mutual Information (PMI) is an established measure for how coherent the top words in a topic are [36]. PMI for two words w_i and w_j is defined in Equation 6.

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{6}$$

PMI uses an external corpus to estimate $p(w_i, w_j)$ and $p(w_i)$. As in [36], we use *Google Web 1T 5-gram Version 1* [37], a corpus of n-grams generated from 1 trillion word tokens. $p(w_i)$ is estimated from the frequencies of 1-grams. $p(w_i, w_j)$ is estimated from the frequencies of 5-grams. For each topic, we average the pairwise PMI's among the topic's top 10 words. For each model, we average the topic-level PMI's. Higher PMI indicates that the words in a topic are correlated, and the topic is more coherent and interpretable.

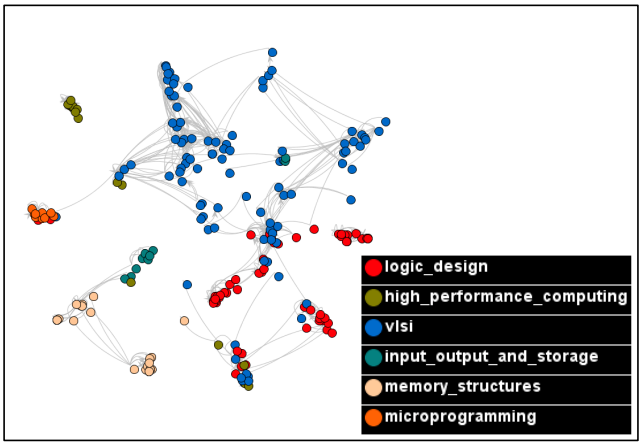
PMI Scores. Table III shows the PMI scores for the four datasets for $Z = 20$ topics. The figures for other numbers of topics are consistent as well. Averaging across the four datasets, PLANE and RTM have very similar PMI's of around 0.5. This suggests that PLANE is at least not inferior to RTM, even with the constraint of modeling embedding coordinates. This shows a great promise by PLANE in enriching the visualization with coherent semantic interpretability.

TABLE III. PMI SCORES FOR TOPIC INTERPRETABILITY ($Z = 20$)

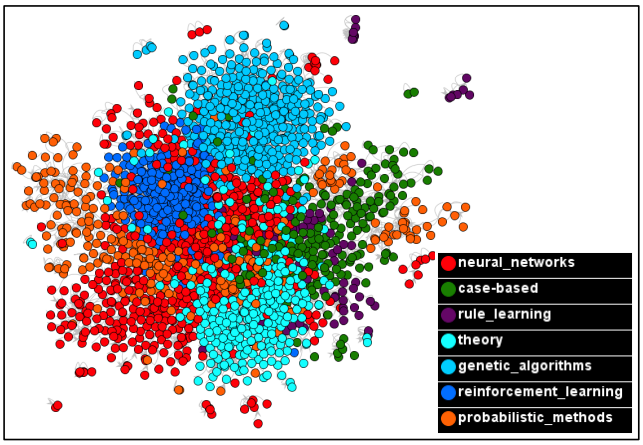
	DS	HA	ML	PL	*Average*
PLANE	0.59	0.53	0.43	0.51	*0.51*
RTM	0.54	0.48	0.51	0.50	*0.50*

To get a sense of the topic interpretability, we show in Table IV the top ten words of each of the 20 topics learned by PLANE for the DS dataset. These keywords are strongly suggestive of the various sub-fields/classes in DS. For instance, topic 0 is probably about *Computational Complexity*, topic 1 is about *Randomized*, while topic 4 is about *Parallel*. Topic 13 are strongly suggestive of *Quantum Computing*, whereas topic 19 seems to capture *Computational Geometry*. In general, the top words in each topic are indeed coherent and meaningful.

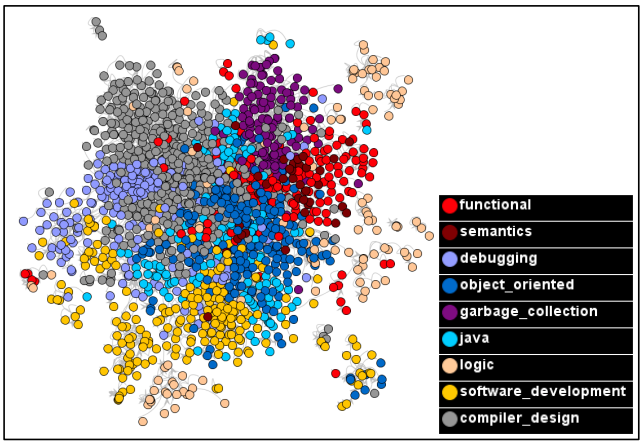
2) **Link Generation Probability:** In addition to words, both PLANE and RTM also model edges or links. In order to evaluate their effectiveness in modeling link generation, we



(a) Hardware and Architecture (HA)



(b) Machine Learning (ML)



(c) Programming Language (PL)

Fig. 5. PLANE's Visualizations for $Z = 20$ (best seen in color)

compare the two methods in terms of link prediction. Note that this is confined to an evaluation task, and our goal is not to propose or compare to state-of-the-art link prediction methods.

For each document (with at least three links), we randomly hide one link. In total, we have around 13%-14% of all links hidden. The task is thus to predict these hidden links based on the observations on the texts and the remaining links. To estimate these hidden links, for PLANE, we use

TABLE IV. TOP 10 WORDS FOR TOPICS IN DS BY PLANE ($Z = 20$)

ID	Top 10 words
0	class, complexity, set, result, measure, prove, study, language, theorem, hierarchy
1	time, log, bound, construction, random, application, work, small, size, construct
2	problem, space, algorithm, time, class, set, competitive, prove, analysis, structure
3	module, approach, paper, program, time, computational, composition, performance, logic, type
4	parallel, processor, machine, communication, algorithm, model, performance, implementation, memory, message
5	result, proof, number, approximate, random, paper, program, property, bit, graph
6	code, algorithm, performance, implementation, parallel, matrix, system, computation, routine, communication
7	algorithm, graph, log, bind, lower, tree, problem, deterministic, network, edge
8	time, structure, query, datum, log, geometric, point, tree, problem, decomposition
9	model, algorithm, parallel, problem, pram, sort, optimal, memory, number, design
10	paper, algorithm, language, matrix, grant, work, number, support, computer, science
11	method, domain, finite, equation, solution, problem, element, mixed, mesh, system
12	function, polynomial, test, proof, program, code, property, interactive, system, testing
13	quantum, problem, computer, machine, computation, model, time, turing, number, polynomial
14	algorithm, complexity, number, problem, set, polynomial, bind, case, real, space
15	system, hybrid, state, property, control, method, verification, transition, game, linear
16	digraph, vertex, result, algorithm, polynomial, path, problem, number, find, bind
17	type, set, constraint, algebra, result, system, space, term, kleene, program
18	time, automaton, model, clock, logic, simulation, real-time, language, checking, symbolic
19	algorithm, mesh, problem, equation, base, ratio, string, generation, quality, triangulation

the document coordinates to compute the probability of a hidden link according to Equation 2. For RTM, we compute the probability of a hidden link according to [23], which shares a comparable exponential link probability function but based on topic distributions (instead of latent coordinates).

Metric. One possibility is to compute the likelihood of generating these hidden links. However, this may not be an appropriate measure, because we will be computing only the likelihood of some links being present (but not of links being absent), thus favoring a model that simply produces higher probability values across the board for all possible links. For instance, consider how in Equation 2, one can produce a higher likelihood simply with a lower η, even while keeping all x_i and x_j's the same, which is inappropriate because the model complexity is in deriving the coordinates to determine which documents should (or should not) be neighbors.

Therefore, a more appropriate metric is to evaluate whether the model assigns a higher probability to the hidden link (which is factually present, though not used for learning) than to other unobserved links. For each document with a hidden link, we rank all the unobserved links of this document in terms of their generation probabilities. The highest rank is 1. Intuitively, the hidden link is expected to have a rank as close to 1 as possible, because it is indeed a factual link that was simply hidden from the model. We borrow a metric from information retrieval, called mean reciprocal rank [38] or MRR, which is defined in Equation 7, where E' is the set of hidden links, and $\mathrm{rank}(\mathrm{e_{ij}})$ is the ranking of the hidden link e_{ij} among the unobserved links of the document from which it is hidden. The higher is the MRR of a method, the better is the method at placing the hidden links in the high ranks.

$$\mathrm{MRR} = \frac{1}{|E'|} \sum_{e_{ij} \in E'} \frac{1}{\mathrm{rank}(\mathrm{e_{ij}})} \tag{7}$$

MRR Scores. Table V shows the MRR scores for the four datasets for $Z = 20$ topics. The figures for other numbers of topics are consistent as well. We see that PLANE produces significantly higher MRR scores than RTM across all the

TABLE V. MRR SCORES FOR LINK PREDICTION ($Z = 20$)

	DS	HA	ML	PL	*Average*
PLANE	0.328	0.207	0.194	0.219	*0.237*
RTM	0.005	0.009	0.0001	0.002	*0.004*

datasets. Averaging across the datasets, PLANE has a score of 0.237, which implies that it generally places the hidden links in the top 5 in terms of link generation probability. In contrast, RTM's score of 0.005 implies that the hidden links tend to be placed around the two hundredths' rank positions.

We attribute PLANE's higher performance in this task to the way we infer the parameters of the model. As discussed in Section III, by modeling some amount of "virtual" negative links we force the model to discriminate between close neighbors (more likely to be positive links) and distant documents (more likely to be negative links). In contrast, by modeling only positive links, RTM is not as able to sharply discriminate genuine neighbors from unrelated documents. The trade-off is that PLANE requires more run time than RTM, because the former models both positive as well as "virtual" negative links, whereas RTM models positive links only (of which there are relatively few in a sparse network).

VI. CONCLUSION

We address the problem of embedding a document network's high-dimensional representations in terms of text and network connectivity in a low-dimensional space. We formulate this as a generative model tying together the various representations of a document (words, links, topics, and coordinates), which we call PLANE. Through comprehensive experiments on four real-life datasets extracted from the Cora collection, we show that it outperforms existing baselines in topic modeling, document embedding, and network embedding, especially in terms of the quality of embedding coordinates (as features in classification and scatterplot visualization). For future work, we plan to consider extensions such as generalizing to directed graph, and pursuing computational optimizations such as hyper-threading or parallel processing.

ACKNOWLEDGMENT

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

REFERENCES

[1] I. Jolliffe, *Principal Component Analysis*. Wiley Online Library, 2005.

[2] L. V. der Maaten and G. Hinton, "Visualizing data using t-SNE," *JMLR*, vol. 9, 2008.

[3] H. C. Purchase, "Metrics for graph drawing aesthetics," *JVLC*, 2002.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, 2003.

[5] T. Iwata, T. Yamada, and N. Ueda, "Probabilistic latent semantic visualization: topic model for visualizing documents," in *KDD*, 2008.

[6] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *TVCG*, 2013.

[7] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, 1964.

[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, 2000.

[9] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, 2000.

[10] T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, and J. B. Tenenbaum, "Parametric embedding for class visualization," *Neural Computation*, 2007.

[11] T. M. V. Le and H. W. Lauw, "Manifold learning for jointly modeling topic and visualization," in *AAAI*, 2014.

[12] ——, "Semantic visualization for spherical representation," in *KDD*, 2014.

[13] G. H. Golub and C. F. V. Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.

[14] A. Talwalkar, S. Kumar, M. Mohri, and H. Rowley, "Large-scale SVD and manifold learning," *JMLR*, 2013.

[15] B. Shaw and T. Jebara, "Structure preserving embedding," in *ICML*, 2009.

[16] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and Experience*, 1991.

[17] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, 1989.

[18] M. Bastian, S. Heymann, M. Jacomy *et al.*, "Gephi: an open source software for exploring and manipulating networks." *ICWSM*, 2009.

[19] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz - open source graph drawing tools," in *Graph Drawing*, 2002.

[20] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," *FTML*, 2010.

[21] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," in *NIPS*, 2009.

[22] Q. Mei, D. Cai, D. Zhang, and C. Zhai, "Topic modeling with network regularization," in *WWW*. ACM, 2008, pp. 101–110.

[23] J. Chang and D. M. Blei, "Relational topic models for document networks," in *AISTATS*, 2009.

[24] R. Nallapati and W. W. Cohen, "Link-PLSA-LDA: A new unsupervised model for topics and influence of blogs." in *ICWSM*, 2008.

[25] Y. Sun, J. Han, J. Gao, and Y. Yu, "iTopicModel: Information network-integrated topic modeling," in *ICDM*, 2009.

[26] Y. Liu, A. Niculescu-Mizil, and W. Gryc, "Topic-Link LDA: Joint models of topic and author community," in *ICML*, 2009.

[27] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang, "Tiara: a visual exploratory text analytic system," in *KDD*, 2010.

[28] B. Gretarsson, J. O'donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth, "TopicNets: Visual analysis of large text corpora with topic modeling," *TIST*, 2012.

[29] A. J.-B. Chaney and D. M. Blei, "Visualizing topic models." in *ICWSM*, 2012.

[30] J. Chuang, C. D. Manning, and J. Heer, "Termite: visualization techniques for assessing textual topic models," in *AVI*, 2012.

[31] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, 1977.

[33] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, 1989.

[34] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, 2000.

[35] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining content and link for classification using matrix factorization," in *SIGIR*, 2007.

[36] D. Newman, S. Karimi, and L. Cavedon, "External evaluation of topic models," in *ADCS*, 2009.

[37] T. Brants and A. Franz, "Web 1T 5-gram Version 1," 2006.

[38] N. Craswell, "Mean reciprocal rank," in *Encyclopedia of Database Systems*, 2009.