Diffusion Maps for Textual Network Embedding

Xinyuan Zhang, Yitong Li, Dinghan Shen, Lawrence Carin

Duke University

December 4, 2018

Textual Information Network Embedding

- Networks are ubiquitous, such as social networks (*e.g.*, Twitter) or citation networks of research papers (*e.g.*, arXiv).
- A ***textual information network*** is $G = (V, E, T)$, where $V = \{v_i\}_{i=1}^N$ is the set of vertices, $E = \{e_{i,j}\}_{i,j=1}^N$ is the set of edges, and $T = \{t_i\}_{i=1}^N$ is the set of texts associated with vertices.
- ***Network embedding*** aims to learn a low-dimensional representation $\boldsymbol{v}_i \in \mathbb{R}^d$ for vertex $v_i \in V$.

Motivation

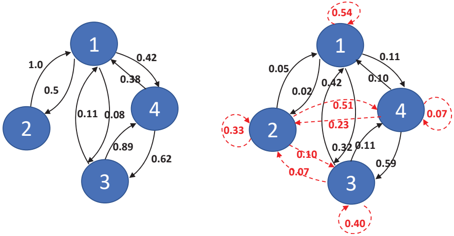
Problem:

- How to measure the complete level of connectivity between any two texts in the graph?

Solutions:

- We propose DMTE which captures the semantic relatedness between texts by applying a diffusion-convolution operation on the text inputs.
- We design a new objective that preserves high-order proximity, by including a diffusion map in the conditional probability.

- $\mathbf{P} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is the ***transition matrix***, with $p_{i,j}$ representing the transition probability from vertex v_i to vertex v_j within one step.
- We introduce the power series of $\mathbf{P}$ for the diffusion process.



(a) Original graph (b) Forth order diffusion graph.

- The ***diffusion map*** of vertex v_i is $\boldsymbol{u}_i$, which maps from vertices and their embeddings to the results of a diffusion process that begins at vertex v_i.

To incorporate both the structure and textual information of the network, we adopt two types of embeddings $\boldsymbol{v}_i^s$ and $\boldsymbol{v}_i^t$ for each vertex v_i.

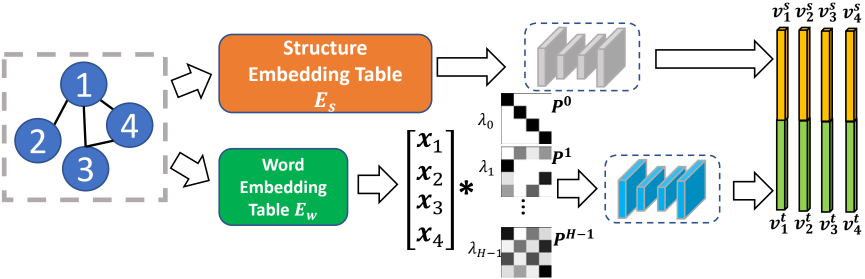


Figure: An illustration of our framework for textual network embedding.

Given the set of edges E, the goal of DMTE is to maximize the following overall objective function:

$$\mathcal{L} = \sum_{e \in E} L(e) = \sum_{e \in E} \alpha_{tt} L_{tt}(e) + \alpha_{ss} L_{ss}(e) + \alpha_{st} L_{st}(e) + \alpha_{ts} L_{ts}(e). \quad (1)$$

The objective function consists of four parts which measure both the structure and text embeddings.

$$L_{tt}(e) = s_{i,j} \log p(\mathbf{v}_i^t | \mathbf{v}_j^t), \quad L_{ss}(e) = s_{i,j} \log p(\mathbf{v}_i^s | \mathbf{u}_j^s) \quad (2)$$

$$L_{st}(e) = s_{i,j} \log p(\mathbf{v}_i^s | \mathbf{v}_j^t), \quad L_{ts}(e) = s_{i,j} \log p(\mathbf{v}_i^t | \mathbf{u}_j^s) \quad (3)$$

Experiments

- We achieve state-of-the-art results on two textual information network embedding tasks: (*i*) link prediction, where we predict the existence of an edge given a pair of vertices; and (*ii*) multi-label classification, where we predict the labels of each text.
- **Case study**:

Query: The K-D-B-Tree: A Search Structure For Large Multidimensional Dynamic Indexes.
1. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects.
2. The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries.
3. Segment Indexes: Dynamic Indexing Techniques for Multi-Dimensional Interval Data.
4. Generalized Search Trees for Database Systems.
5. High Performance Clustering Based on the Similarity Join.

Table: Top-5 similar vertex search based on embeddings learned by DMTE.

Poster: 10:45 AM – 12:45 PM @ Room 210 & 230 AB #152

Diffusion Maps for Textual Network Embedding

Xinyuan Zhang, Yitong Li, Dinghan Shen, Lawrence Carin
Duke University, Durham, NC 27707
Email: xy.zhang@duke.edu

Abstract

Textual network embedding leverages rich text information associated with the network to learn low-dimensional vectorial representations of vertices. Rather than using typical natural language processing (NLP) approaches, recent research exploits the relationship of texts on the same edge to graphically embed text. However, these models neglect to measure the *complete* level of connectivity between any two texts in the graph. We present diffusion maps for textual network embedding (DMTE), integrating global structural information of the graph to capture the semantic relatedness between texts, with a diffusion-convolution operation applied on the text inputs. In addition, a new objective function is designed to efficiently preserve the high-order proximity using the graph diffusion. Experimental results show that the proposed approach outperforms state-of-the-art methods on the vertex-classification and link-prediction tasks.

Problem Defination

Definition 1. A ***textual information network*** is $G = (V, E, T)$, where $V = \{v_i\}_{i=1,\cdots,N}$ is the set of vertices, $E = \{e_{i,j}\}_{i,j=1}^N$ is the set of edges, and $T = \{t_i\}_{i=1,\cdots,N}$ is the set of texts associated with vertices. Each edge $e_{i,j}$ has a weight $s_{i,j}$ representing the relationship between vertices v_i and v_j. If v_i and v_j are not linked, $s_{i,j} = 0$. If there exists an edge between v_i and v_j, $s_{i,j} = 1$ for an unweighted graph, and $s_{i,j} > 0$ for a weighted graph. A *path* is a sequence of edges that connect two vertices. The text of vertex v_i, t_i, is comprised of a word sequence $< w_1, \cdots, w_{|t_i|} >$.

Definition 2. Let $\mathbf{S} \in \mathbb{R}^{N\times N}$ be the adjacency matrix of a graph whose entry $s_{i,j} \geq 0$ is the weight of edge $e_{i,j}$. The ***transition matrix*** $\mathbf{P} \in \mathbb{R}^{N\times N}$ is obtained by normalizing rows of $\mathbf{S}$ to sum to one, with $p_{i,j}$ representing the transition probability from vertex v_i to vertex v_j within one step. Then an h-step transition matrix can be computed with $\mathbf{P}$ to the h-th power, *i.e.*, $\mathbf{P}^h$. The entry $p_{i,j}^h$ refers to the transition probability from vertex v_i to vertex v_j within exactly h steps.

Definition 3. A ***network embedding*** aims to learn a low-dimensional vector $\boldsymbol{v}_i \in \mathbb{R}^d$ for vertex $v_i \in V$, where $d \ll |V|$ is the dimension of the embedding. The embedding matrix $\mathbf{V}$ for the complete graph is the concatenation of $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N\}$. The distance between vertices on the graph and context similarity should be preserved in the representation space.

Definition 4. The ***diffusion map*** of vertex v_i is $\boldsymbol{u}_i$, the i-th row of the diffusion embedding matrix $\mathbf{U}$, which maps from vertices and their embeddings to the results of a diffusion process that begins at vertex v_i. $\mathbf{U}$ is computed by $\mathbf{U} = \sum_{h=0}^{H-1} \lambda_h \mathbf{P}^h \mathbf{V}$, where λ_h is the importance coefficient that typically decreases as the value of h increases. The high-order proximity in the network is preserved in diffusion maps.

Method

We employ a diffusion process to build long-distance semantic relatedness in text embeddings, and capture global structural information in the objective function. To incorporate both the structure and textual information of the network, we adopt two types of embeddings $\boldsymbol{v}_i^s$ and $\boldsymbol{v}_i^t$ for each v_i vertex. In this work, $\boldsymbol{v}_i$ is learned by an unsupervised approach, and it can be used directly as a feature vector of vertex v_i for various tasks.

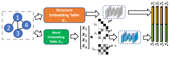
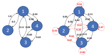


Figure 1: An illustration of our framework for textual network embedding.

Diffusion Process

Initially the network only has a few active vertices, due to sparsity. Through the diffusion process, information is delivered from active vertices to inactive ones by filling information gaps between vertices; vertices may be connected by indirect, multi-step paths. We introduce the transition matrix $\mathbf{P}$ and its power series for the diffusion process.



(a) Original graph (b) Forth order diffusion graph.

Figure 2: A simple example of diffusion process in a directed graph.

Text Embedding

A word sequence $t = < w_1, \cdots, w_{|t|} >$ is mapped into a set of d_t-dimensional real-valued vectors $< \boldsymbol{w}_1, \cdots, \boldsymbol{w}_{|t|} >$ by looking up the word embedding matrix $\mathbf{E}_w$. We can obtain a simple text representation $\boldsymbol{x}_i \in \mathbb{R}^{d_t}$ of vertex v_i by taking the average of word vectors. The input texts can be represented by matrix $\mathbf{X} \in \mathbb{R}^{N\times d_t}$.

$$\boldsymbol{x} = \frac{1}{|t|}\sum_{i=1}^{|t|} \boldsymbol{w}_i, \qquad \mathbf{X} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \cdots \oplus \boldsymbol{x}_N. \tag{1}$$

Alternatively, we can use the bi-directional LSTM. Text inputs are represented by the mean of all hidden states.

$$\overrightarrow{\boldsymbol{h}}_i = LSTM(w_i, \boldsymbol{h}_{i-1}), \qquad \overleftarrow{\boldsymbol{h}}_i = LSTM(w_i, \boldsymbol{h}_{i+1}) \tag{2}$$

$$\boldsymbol{x} = \frac{1}{|t|}\sum_{i=1}^{|t|} (\overrightarrow{\boldsymbol{h}}_i \oplus \overleftarrow{\boldsymbol{h}}_i), \qquad \mathbf{X} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \cdots \oplus \boldsymbol{x}_N. \tag{3}$$

However, the above embeddings do not leverage the semantic relatedness indicated from the graph. To address this issue, we employ the diffusion convolutional operator to measure the level of connectivity between any of two texts in the network.

Let $\mathbf{P}^* \in \mathbb{R}^{N\times H\times N}$ be a tensor containing H hops of power series of $\mathbf{P}$, *i.e.*, the concatenation of $\{\mathbf{P}^0, \mathbf{P}^1, \cdots, \mathbf{P}^{H-1}\}$. $\mathbf{V}_t^* \in \mathbb{R}^{N\times H\times d}$ is the tensor version of the text embedding representation, after the diffusion convolutional operation.

$$\mathbf{V}_t^* = f(\mathbf{W} \odot \mathbf{P}^*\mathbf{X}), \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{H\times d}$ is the weight matrix, f is a nonlinear differentiable function, and $\odot$ represents element-wise multiplication. With longer paths discounted more than shorter paths, the text embedding matrix $\mathbf{V}_t$ is given by

$$\mathbf{V}_t = \sum_{h=0}^{H-1} \lambda_h \mathbf{V}_t^{*(:,h,:)}. \tag{5}$$

Through the diffusion process, text representations, *i.e.*, rows of $\mathbf{V}_t$ are not embedded independently. With the whole graph being smoothed, indirect relationships between texts that are not on the same edge can be considered to learn embeddings.

Objective Function

Given the set of edges E, the goal of DMTE is to maximize the following overall objective function:

$$\mathcal{L} = \sum_{e\in E} L(e) = \sum_{e\in E} \alpha_{tt}L_{tt}(e) + \alpha_{ss}L_{ss}(e) + \alpha_{st}L_{st}(e) + \alpha_{ts}L_{ts}(e). \tag{6}$$

The objective function consists of four parts, which measure both the structure and text embeddings. Each part is to measure the log-likelihood of generating v_i conditioned on v_j, where v_i and v_j are on the same directed edge.

$$L_{tt}(e) = s_{i,j}\log p(\boldsymbol{v}_i^t|\boldsymbol{v}_j^t) = s_{i,j}\log\frac{\exp(\boldsymbol{v}_i^t\cdot\boldsymbol{v}_j^t)}{\sum_{\boldsymbol{v}_k^t\in\mathbf{V}_t}\exp(\boldsymbol{v}_k^t\cdot\boldsymbol{v}_j^t)}, \tag{7}$$

$$L_{ss}(e) = s_{i,j}\log p(\boldsymbol{v}_i^s|\boldsymbol{u}_j^s) = s_{i,j}\log\frac{\exp(\boldsymbol{v}_i^s\cdot\boldsymbol{u}_j^s)}{\sum_{\boldsymbol{v}_k^s\in\mathbf{V}_s}\exp(\boldsymbol{v}_k^s\cdot\boldsymbol{u}_j^s)}, \tag{8}$$

$$L_{st}(e) = s_{i,j}\log p(\boldsymbol{v}_i^s|\boldsymbol{v}_j^t) = s_{i,j}\log\frac{\exp(\boldsymbol{v}_i^s\cdot\boldsymbol{v}_j^t)}{\sum_{\boldsymbol{v}_k^s\in\mathbf{V}_s}\exp(\boldsymbol{v}_k^s\cdot\boldsymbol{v}_j^t)}, \tag{9}$$

$$L_{ts}(e) = s_{i,j}\log p(\boldsymbol{v}_i^t|\boldsymbol{u}_j^s) = s_{i,j}\log\frac{\exp(\boldsymbol{v}_i^t\cdot\boldsymbol{u}_j^s)}{\sum_{\boldsymbol{v}_k^t\in\mathbf{V}_t}\exp(\boldsymbol{v}_k^t\cdot\boldsymbol{u}_j^s)} \tag{10}$$

Note that $p(\cdot|\boldsymbol{u}_j^s)$ computes the probability conditioned on the diffusion map of vertex v_j, and $p(\cdot|\boldsymbol{v}_j^t)$ computes the probability conditioned on the text embedding of vertex v_j.

Experiments

We evaluate the proposed method for the multi-label classification and link prediction tasks.

- Given a pair of vertices, ***link prediction*** seeks to predict the existence of an unobserved edge using the trained representations.
- ***Multi-label classification*** seeks to classify each vertex into a set of labels using the learned vertex representation as features.

Dataset

- DBLP is a citation network that consists of 60744 papers in 4 research areas: database, data mining, artificial intelligence, and computer vision. The network has 52890 edges indicating the citation relationship between papers.
- Cora is a citation network that consists of 2277 machine learning papers in 7 classes and 5214 edges indicating the citation relationship between papers.
- Zhihu is a Q&A based community social network in China. In our experiments, 10000 active users are collected as vertices and 43894 edges. The description of their interested topics are used as text information.

Results

Table 1: AUC scores for link prediction on Cora(top) and Zhihu(bottom).

% of edges	15%	25%	35%	45%	55%	65%	75%	85%	95%
Deep Walk	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
LINE	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
node2vec	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
TADW	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
TriDNR	85.9	88.6	90.5	91.2	91.3	92.4	93.0	93.6	93.7
CENE	72.1	86.5	84.6	88.1	89.4	89.2	93.9	95.0	95.9
CANE	86.8	91.5	92.2	93.9	94.6	94.9	95.6	96.6	97.7
DMTE (w/o diffusion)	87.4	91.2	92.0	93.2	93.9	94.6	95.5	95.9	96.7
DMTE (text only)	82.6	84.0	85.7	87.3	89.1	91.1	92.0	92.9	94.2
DMTE (Bi-LSTM)	86.3	88.2	90.7	92.7	94.1	94.8	96.0	97.3	98.1
DMTE (WAvg)	**91.3**	**93.1**	**93.7**	**95.0**	**96.0**	**97.1**	**97.4**	**98.2**	**98.8**
Deep Walk	56.6	58.1	60.1	60.0	61.8	61.9	63.3	63.7	67.8
LINE	52.3	55.9	59.9	60.9	64.3	66.0	67.7	69.3	71.1
node2vec	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
TADW	52.3	54.2	55.6	57.3	60.8	62.4	65.2	63.8	69.0
TriDNR	53.8	55.7	57.9	59.5	63.0	64.6	66.0	67.5	70.3
CENE	56.2	57.4	60.3	63.0	66.3	66.0	70.2	69.8	73.8
CANE	56.8	59.3	62.9	64.5	68.9	70.4	71.4	73.6	75.4
DMTE (w/o diffusion)	56.2	58.4	61.3	64.0	68.5	69.7	71.5	73.3	75.1
DMTE (text only)	55.9	57.2	58.8	61.6	65.3	67.6	69.5	71.0	74.1
DMTE (Bi-LSTM)	56.3	60.3	64.9	69.8	73.2	76.4	**78.7**	**80.3**	**82.2**
DMTE (WAvg)	**58.4**	**63.2**	**67.5**	**71.6**	**74.0**	**76.7**	78.5	79.8	81.5

Figure 3: *Left*: The link prediction results w.r.t. the number of hops H at different training ratios. *Right*: F1-Macro scores for multi-label classification on *DBLP*.

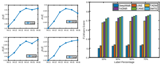


Table 2: Top-5 similar vertex search based on embeddings learned by DMTE.

Query: The K-D-B-Tree: A Search Structure For Large Multidimensional Dynamic Indexes.
1. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects.
2. The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries.
3. Segment Indexes: Dynamic Indexing Techniques for Multi-Dimensional Interval Data.
4. Generalized Search Trees for Database Systems.
5. High Performance Clustering Based on the Similarity Join.

Conclusions

- We propose DMTE which integrates global structural information of the graph to capture the level of connectivity between any two texts, by applying a diffusion convolutional operation on the text inputs.
- We design a new objective that preserves high-order proximity, by including a diffusion map in the conditional probability.
- Experimental results on the vertex-classification and link-prediction tasks show the superiority of the proposed approach.