Global Vectors for Node Representations

Robin Brochier [1,2], Adrien Guille [1], Julien Velcin [1]

[1] Université de Lyon, Lyon 2, ERIC EA3083 [2] Peerus, DSRT F-69003

Presented at the 2019 World Wide Web Conference

Introduction

We present *GVNR* (Global Vectors for Node Representation), a model to learn representations of nodes in a network. We formulate a factorization problem on a thresholded co-occurrence matrix obtained by random walks. We specify this problem so that the reconstruction error is measured on all the positive entries and some randomly sampled zero entries. Furthermore, we show how to extend this model under the name *GVNR-t*, to deal with networks where nodes are text documents.

Matrix to factorize

From a network, sequences of nodes are generated by random walks. A matrix of weighted co-occurrence X is constructed by counting the co-occurrences in these sequences. For each node j visited within q steps, with $q \leq l$, from a node i, we increase X_{ij} by $\frac{1}{q}$.

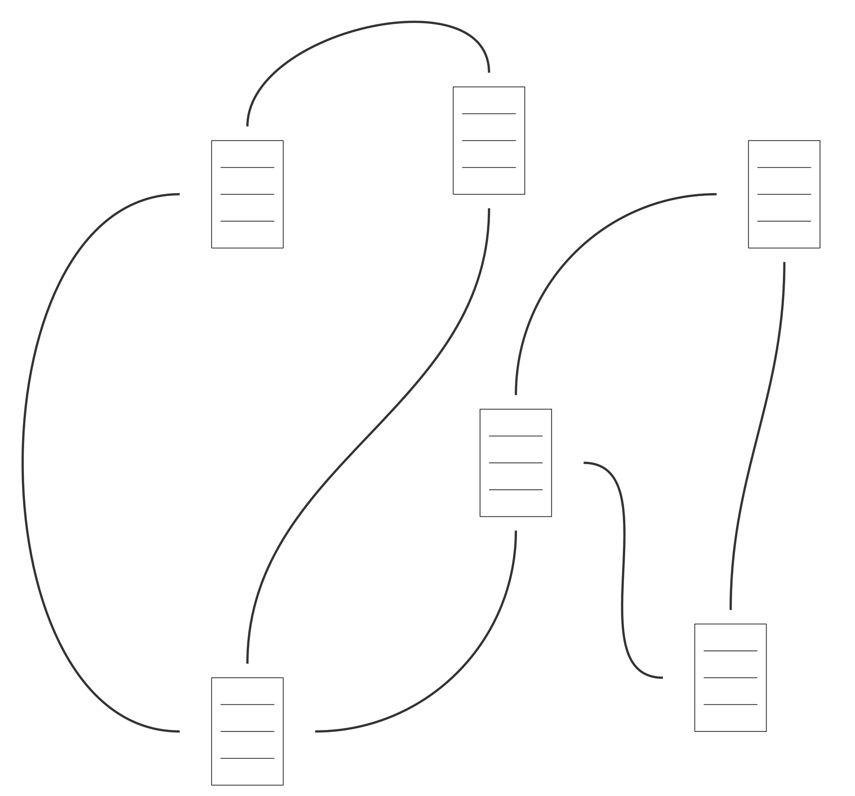

Figure 1: An hypothetical network of documents on which we perform random walks.

X is sparse but many of its coefficients are close to zero since many pairs of nodes rarely co-occur. We introduce a threshold parameter x_{min} to zero out all entries below this threshold, considering them as noise.

Subsampling

In order to factorize this matrix, we propose to deal with all positive entries and to subsample k zero entries for each positive entry.

$$X = \begin{pmatrix} 0 & 4 & 8 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 6 & 1 & 0 \end{pmatrix}$$

x: strictly positive entries

x: sampled zero entries

Figure 2: A co-occurrence matrix from which strictly positive and zero entries are sampled. Here $k = 1$ i.e. the same number of positive entries as the number of zero entries are sampled.

General model

GVNR trains node vectors U and V and their biases b^U and b^V to approximate $\log(X)$:

$$\underset{U,V,b^U,b^V}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} s(x_{ij}) \big(u_i \cdot v_j + b_i^U + b_j^V - \log(c + x_{ij})\big)^2$$

- $c \in]0;1]$ allows for smoothing X while making the logarithm negative when $x_{ij} = 0$.
- $s(x_{ij})$ selects the coefficients of X for measuring the reconstruction error:

$$s(x_{ij}) = \begin{cases} 1 & \text{if } x_{ij} > 0, \\ m_i & \text{else, with } m_i \sim \text{Bernoulli}(\alpha_i). \end{cases}$$

- α_i depends on the odds that a randomly chosen coefficient on row i of X is a positive coefficient, p_i being the proportion of positive coefficients on row i.

$$\alpha_i = \begin{cases} k \times \frac{p_i}{1-p_i} & \text{if } p_i \leq (k+1)^{-1}, \\ 1 & \text{otherwise} \end{cases}$$

Extension: accounting for text

We extend the model to handle networks of documents. Assuming word order is negligible, a document is represented by a vector $\delta \in \mathbb{N}^{+m}$ (bag of words), with m the size of the vocabulary. We replace the contextual embedding v_j by the weighted average of the word embeddings of a document. We thus learn a word embedding matrix $W \in \mathbb{R}^{m \times d}$:

$$\underset{U,W,b^U,b^V}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} s(x_{ij}) \times \Big(u_i \cdot \frac{\delta_j \, W}{|\delta_j|_1} + b_i^U + b_j^V - \log(c + x_{ij})\Big)^2$$

Experiments: node classification

Table 1: Accuracy on the citation (1) network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	57.7	62.4	69.5	72.8	73.8
NetMF	65.7	**72.9**	**76.4**	**78.6**	79.4
DeepWalk	67.8	71.6	74.5	75.8	79.2
GVNR ($x_{\text{min}} = 0$)	58.5	62.5	70.7	73.4	75.0
GVNR ($x_{\text{min}} = 1$)	**69.5**	72.6	75.9	78.1	**80.2**

Table 2: Accuracy on the citation (2) network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	42.8	53.5	55.3	56.2	56.8
NetMF	**51.2**	54.8	55.1	55.0	54.8
DeepWalk	41.3	52.5	54.5	55.5	56.0
GVNR ($x_{\text{min}} = 0$)	38.7	46.8	49.1	50.4	50.9
GVNR ($x_{\text{min}} = 1$)	45.6	**55.6**	**57.3**	**58.7**	**59.0**

Table 3: Accuracy on the citation (1) network, considering the text features.

	% of training data				
	10%	20%	30%	40%	50%
LSA	54.7	61.0	62.4	63.0	62.8
DeepWalk+LSA	73.8	77.9	78.4	78.1	78.1
TADW	77.1	78.8	78.2	78.8	78.6
GVNR-t	**79.3**	**80.7**	**80.8**	**81.4**	**81.1**

Table 4: Accuracy on the citation (2) network, considering the text features.

	% of training data				
	10%	20%	30%	40%	50%
LSA	52.0	54.7	54.7	58.4	65.7
DeepWalk+LSA	58.3	60.7	61.1	60.0	61.2
TADW	60.6	60.1	60.1	66.2	69.3
GVNR-t	**63.3**	**62.5**	**64.9**	**68.6**	**70.4**

Parameters sensitivity

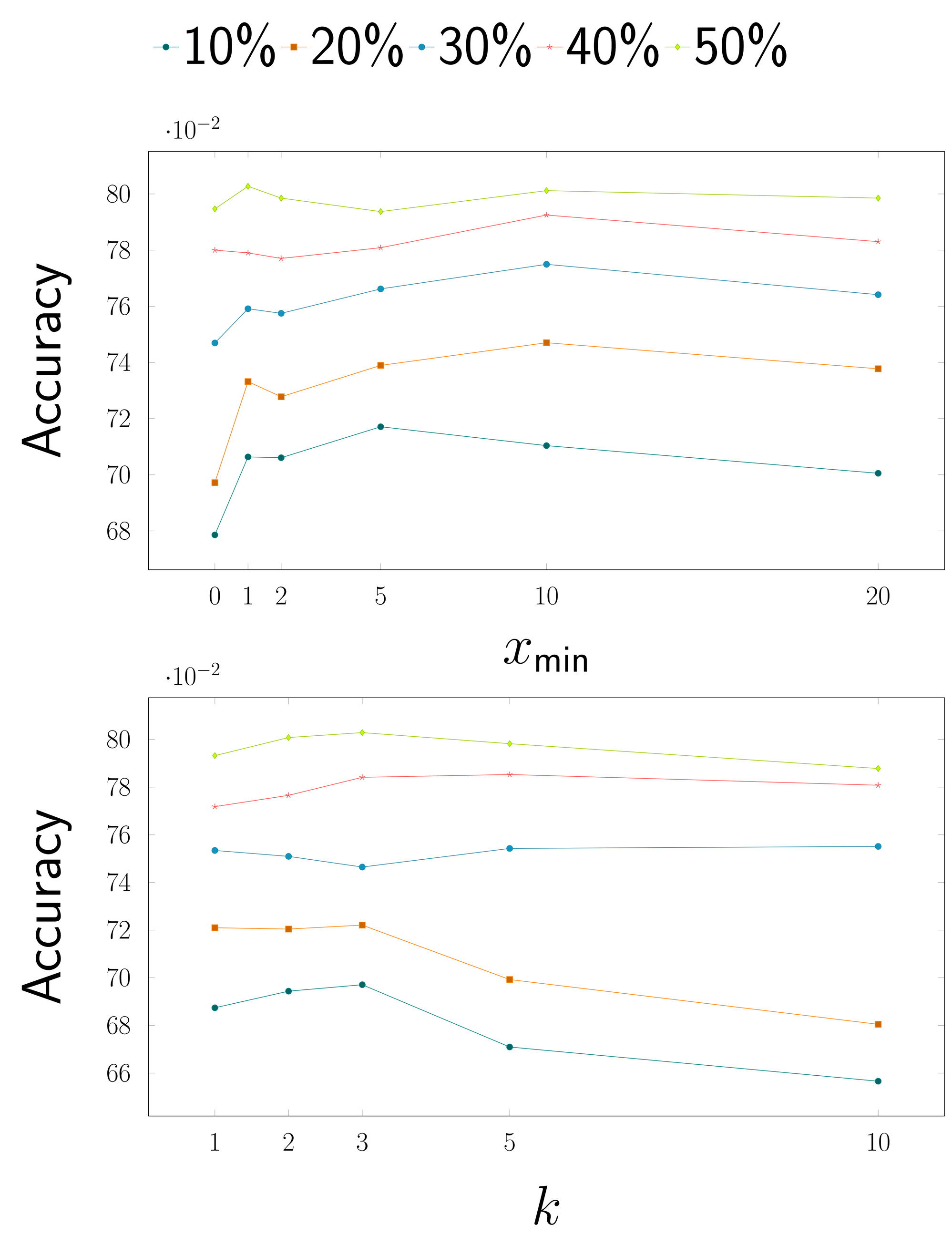

Figure 3: Sensitivity of *GVNR* to the hyper-parameters x_{min} and k on the citation (1) network.

Application to keyword recommendation

Table 5: Keyword recommendation by selecting the closest word embeddings w_k to both embeddings u (node) and $\frac{\delta_j \, W}{|\delta_j|_1}$ (content) of an input document.

Document	***A brief survey of computational approaches in social computing*** *Web 2.0 technologies have brought new ways of connecting people in social networks for collaboration in various on-line communities. Social Computing is a novel and emerging computing paradigm...*
Closest words to u	*cold start problem, storylines, document titles, movielens data, computational humor*
Closest words to $\frac{\delta_j \, W}{|\delta_j|_1}$	*social, social network, enron email corpus, social networks, extremely large datasets, sites blogs*

Contact

- GitHub: https://github.com/brochier/gvnr
- Web: http://www.robinbrochier.com
- Emails: robin.brochier@univ-lyon2.fr