Property Preserving Network Embedding

Instruction

Blind

Blind
Blind University

13 September 2016

Contents

Introduction

1 Files

Here we submit three folds: modeltrain ,citeseer and classification. In *modeltrain* folder, the training code of $\text{PPNE}_{\text{ineq}}$ and PPNE_{num} are presented. In *citeseer* folder, we show the input of the above two models of citeseer datasets. In *classification* folder, we show the classification tool used in this paper and the representation vectors learned by various embedding results of citeseer datasets. This ensures the experimental results in the paper can be re-implemented. Here is the table of the submitted files and their descriptions:

Table 1: File Description

File	Directory	Description
PPNEinqTrain.c	modeltrain	Training code of $\text{PPNE}_{\text{ineq}}$ model
PPNEnumTrain.c	modeltrain	Training code of PPNE_{num} model
citeseer.walks.txt	citeseer	Node sequences generated by random walk
train.citeseer.inequationconstraints.txt	citeseer	inequation constraints for $\text{PPNE}_{\text{ineq}}$ model
train.citeseer.numeric.constraints.txt	citeseer	numeric constraints for PPNE_{num} model
valid.citeseer.inequation.constraints.txt	citeseer	validation constraints for $\text{PPNE}_{\text{ineq}}$ model
classification.py	classification	SVM model implemented using sklearn
embedding.citeseer.deepwalk.txt	classification	embedding results of DeepWalk
embedding.citeseer.inequation.constraints.txt	classification	embedding results of $\text{PPNE}_{\text{ineq}}$ model
embedding.citeseer.LINE.txt	classification	embedding results of LINE
embedding.citeseer.naivecombination.txt	classification	embedding results of Naive Combination Method
embedding.citeseer.numeric.constraints.txt	classification	embedding results of PPNE_{num} Method
embedding.citeseer.propertyfeatures.txt	classification	embedding results of Property Features Method
embedding.citeseer.TADW.txt	classification	embedding results of TADW Method
group.txt	classification	the categories of citeseer data

2 Model Training

In the corresponding paper we propose two types of TPNE model. For each model we submit a training file using C. There is little difference in the parameter update process

between these two training files. The input parameters of these two models are mostly same. Here we introduce the input parameters used in this work:

Table 2: Parameters Description

Parameter	Description
layer1_size	Train Setting embedding size
window	Train Setting window size
sample	Train Setting sample value
negative	Train Setting negative sampling number
num_threads	Running Threads
iter	Iteration Times
PPNE_inequation_file	constraints training file. For inequation model, it contains lines as: $\{A, B, A, D\}$, refers to $sim(A, B) > sim(A, D)$. For numeric model, it contains lines as:$\{A, B, Score\}$, refers to $sim(A, B) = Score$
PPNE_inequation_fileCV	constraints valid file, only used in inequation model
PPNE_add_time	PPNE Add Time
PPNE_weight_decay	PPNE Weight Decay
PPNE_inter_coeff	PPNE Inter Coeff
PPNE_hinge_margin	PPNE Norm Hinge Margin
train_file	node sequences file, each line is a node sequence
output_file	Final embedding saved file, similar to word2vec format

3 Key Notes

- This code can be easily compiled and running in the Windows 7 and Visual Studio 2013.
- Math Kernel Library can significantly reduce the training time, for example, the training process in citeseer dataset will be reduce to three minutes.
- We are rewriting the code with python and we want to make it a public python package can be downloaded with pip.
- Other datasets and experimental results will be published later in the website of the first author (blind) due to the file uploaded limitation of github (25M).