Report on Hyperpolic Community detection

October 22, 2019

1 Introduction

The objective of this repport is to give tools and insight to execute and run the current project. Learning embedding within the poincare model is a difficult task much more than euclidean usual learning. Optimization in hyperbolic space often leads to machine precision issues and tweeking very precisely hyperparameters. Furthermore, coordinate of vectors is meaningfull contrary to euclidean, thus most centered element often means most general concept or in our work a most central nodes (e.g nodes with high degree relatively to others). This last point is especially important when develloping a cost function using negative sampling, indeed sampling negatives nodes randomly will accentuate the initial behaviour while sampling nodes based on the degree will soften it.

In this document we will go through issues encountered, by showing examples of errors that can occures and behaviour depending on model tunning.

2 KMeans and EM algorithm within Poincaré model

2.1 Barycenter hyper-parameters

2.2 EM: Machine precision

What to do : never clamp unormalised wik

2.3 EM: Updating parameters

Updating μ : Because of the gradient descent (or ascent) used for updating μ,this step is subject to failure or taking more or less time to process. Let be the two initalisation possible to start the gradient descent:

- $b_k = \frac{1}{n} \sum_{i=1}^{n} x_i$
- $b_k = \frac{1}{\sum_{j=1}^{n} w_{jk}} \sum_{i=1}^{n} x_i w_{ik}$

Obviously the second initalisation will lead to make less iteration to reach the convergence.

3 Learning embeddings

3.1 Batch, Mini-batch, sum or mean

3.2 Negative sampling

3.3 Compareason of optimization methods

3.4 Going out of the ball border

4 Parameters

:

5 Dataset

Parameter	Description
–init-lr	Initial learning rate for the first epoch, before applying O_3 loss
–lr	Learning rate applied on the gradient
–init-alpha	O_1 loss weight for the first epoch (if not given is set to alpha value)
–alpha	O_1 loss weitght
–init-beta	O_2 loss weight for the first epoch (if not given is set to beta value)
–beta	O_2 loss weight
–gamma	O_3 loss weight (this loss is firstly applied at the second iteration/epoch)
–n-gaussian	The number of gaussian in the gaussan mixture (mainly set to the number of community in the dataset)
–dataset	The dataset name in lowercase on which perform the experiment (see dataset section)
–walk-lenght	The size of the random walk usually set to 20
–context-size	The size of the context, the window around each nodes in a path
–precompute-rw	The number of path to consider from each nodes
–negative-sampling	The number of negative examples for the O_2 loss (between 5 and 20)
–epoch-embedding-init	The number of epoch (see all the nodes, couple on the graph) for the first embedding learning iteration, if not given is set to –epoch-embedding. We recommend to set more epoch for the first iteration than for others
–epoch-embedding	The number of epoch for learning embedding
–loss-aggregation	The type of aggregation of the loss must be "sum" or "mean". This paramaeters has a huge influence on the selection of learning rate and batch size. If choosing sum the learning rate must be low (between 1e-2 and 1e-4). If choosing mean the learning rate value must be high (from 1 to 10).
–batch-size	the size of the batch (number of element to require grad without updating parameters). A small batch-size is recommended in order to have a small number of iteration, however large batch size will drastically lower the time of epoch. The default batch size is set to 10000 (on small dataset one batch allow to visit every examples).

Dataset	#nodes	#edges	#community
karate	-	-	2
poolblogs	-	-	2
adjnoun	-	-	2
poolbooks	-	-	3
football	-	-	12
dblp	-	-	5