Netflix Movie Recommendation with DeepWalk and Node2Vec

Xizhi Wu
xiw183@pitt.edu

Motivation

Model the netflix movie dataset as a graph
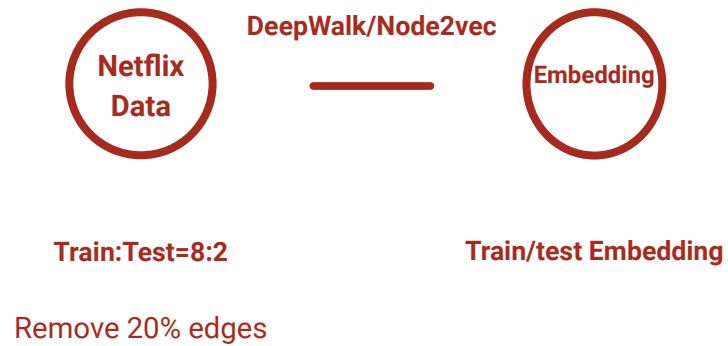
movies and users as nodes

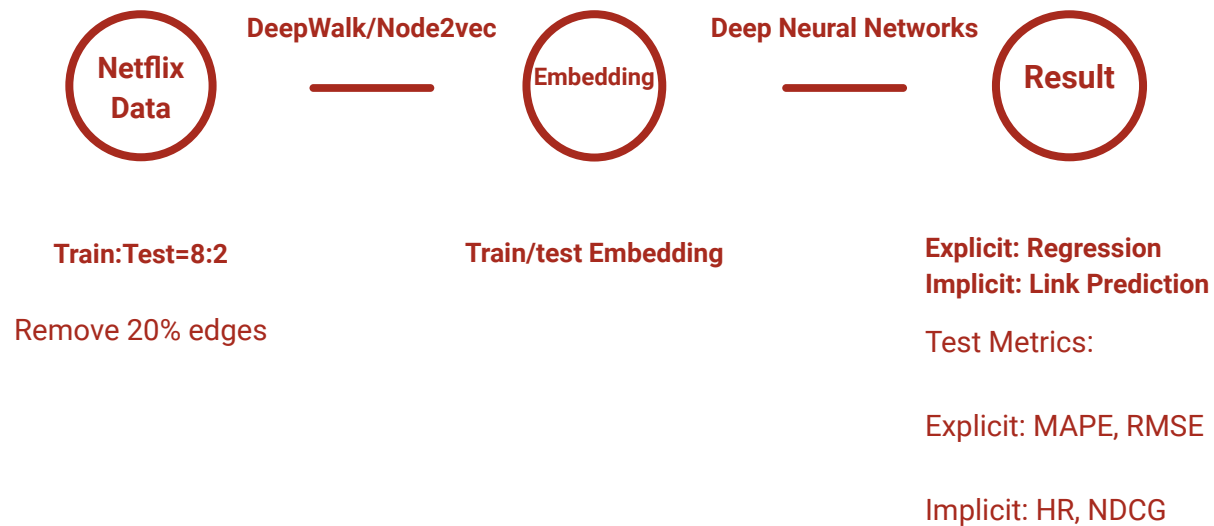Users watch a certain movie, there would be a edge.

	movie	user	rating
0	m11405	u28966	4
1	m15532	u28966	1
2	m8903	u28966	1
3	m10809	u1086	4
4	m829	u1086	3
...	...	...	...

Netflix dataset

Goals

Netflix Data

DeepWalk/Node2vec

Embedding

Train:Test=8:2

Train/test Embedding

Remove 20% edges

Goals

Netflix Data

DeepWalk/Node2vec

Embedding

Deep Neural Networks

Result

Train:Test=8:2

Remove 20% edges

Train/test Embedding

Explicit: Regression
Implicit: Link Prediction

Test Metrics:

Explicit: MAPE, RMSE

Implicit: HR, NDCG

Methods: Generate Embedding

DeepWalk:

dimensions = 32

num_walks = 10

walk_length = 40

window_size = 5

Node2Vec:

p=1.0

q=0.5

dimensions = 32

num_walks = 10

walk_length = 40

window_size = 5

Make sure all the nodes are reachable when we remove edges

Methods: Generate test Dataset

Graph test: original graph   —-----------> test embedding

Graph train: remove 20% edges   —-----------> train embedding

The removed edges are used for prediction in testing our embedding

Methods: Generate Train Dataset

In graph train: remove another 20% of the train graph

The removed edges are used to test our choice of processing embedding

Such as L1, concatenate.. Next slide

Methods: Processing Embedding

Movie : 1x32 vector

User : 1x32 vector

Operator L1: (Movie vector - User Vector)

Operator Concat: (Movie vector concatenate with User Vector)

They will be the input to our regression model and link prediction model

Methods: regression models

L1 regression model:

Input dim:32

Loss function: MSE

Optimizer: Adam

Activation: relu and tanh

Output: ratings

Concat regression model:

Input dim:64

Loss function: MSE

Optimizer: Adam

Activation: relu and tanh

Output: ratings

Methods: Link prediction models

L1 link prediction model:

Input dim:32

Loss function: MSE

Optimizer: Adam

Activation: relu and tanh

Output: 1/0

Concat link prediction model:

Input dim:64

Loss function: MSE

Optimizer: Adam

Activation: relu and tanh

Output: 1/0

Results: DeepWalk Implicit

L1 model:

Accuracy:0.91

Concat model:

Accuracy:0.96

Testing Result:

HR: 0.4936

NDCG: 0.4724

The results here are my first run, the result in the paper is my second run after the presentation, which is a bit different from the first. I increased the number of training set samples in the second run

Results: DeepWalk Explicit

I only used concat model because it
seems to be better

Testing Result:

RMSE: 1.09

MAPE: 0.34

Results: Node2Vec Implicit

L1 model:

Accuracy:0.919

Concat model:

Accuracy:0.49

Testing Result:

HR: 0.54

NDCG: 0.310

Testing Result:

HR: 0.45

NDCG: 0.314

Results: Node2Vec Explicit

L1 model:

Accuracy:0.919

Concat model:

Accuracy:0.96

Testing Result:

RMSE: 1.08

MAPE: 0.352

Testing Result:

RMSE: 1.07

MAPE: 0.328

Results: Comparison between L1 and Concat

Concatenation on the User-Movie embeddings are easier to overfit on implicit data, because the input has more dimensions and output is 1 or 0.

For explicit data, Concatenation is better than L1 on the User-Movie embedding

MAPE and RMSE are slightly better.

Results: Comparison between DeepWalk and Node2Vec

DeepWalk is performing better on NDCG, Node2Vec is performing better on HR

It could be the p and q value for Node2Vec are not the best for my models. Because I only tried p=1.0 and q=0.5. Other combinations might be better.

Results: Comparison to my best model in HW1 and HW2

Best model in HW1:

Base SVD:

RMSE: 0.99

MAPE: 30.42

Best model in HW2:

GMF:

HR: 0.4

NDCG: 0.32

Best Explicit Model:

Node2Vec Concat:

RMSE: 1.07

MAPE: 32.8

Best Implicit Model:

DeepWalk Implicit:

HR: 0.49

NDCG: 0.4724

Reference

Keras Library

StellarGraph Library

Thank You!