**E-LSTM-D: A Deep Learning Framework for Dynamic Network Link Prediction**

Like Common Neighbor (CN) and Resource Allocation Index (RA), are widely used in link prediction of static networks, but they can hardly deal with the changes of the network structure directly.

* There are some method can improve that, but depend on simple statistics of networks and thus cannot effectively deal with high non-linearity.
* In order to deal that, network embedding techniques were proposed to learn the representations of networks that can preserve high-order proximity (e.g. DeepWalk, node2vec, SDNE, GCN, TS-RW, RBM, ctRBM, GRU)

**What is this paper address to?**

-Focus on predicting network global structure in the future and check the links are disappearing or not.

-Provide E-LSTM-D model to predicting links in dynamic network. Has encoder-decoder architecture help the model learn network representation and reconstruct a graph on the grounds of extracted information. And low dimensional representations for the sequences of graphs can be well learned from the stacked LTSM module placed right behind the encoder.

-Conduct comprehensive experiments on five real-world datasets. The results show that our model significantly outperforms the current state-of-the-art methods.

**What contributions they have done?**

-Propose a general end-to-end deep learning framework, namely E-LSTM-D, for link prediction in dynamic networks.

-Model competent to make long term prediction tasks with only slight drop of performances, predicting links disappearing or not.

-Define a new metric, Error Rate, to measure the performance of dynamic network link prediction.

-Conduct extensive experiments, comparing our ELSTM-D model with five baseline methods on various metrics.

**Some basic definitions need to know**

1. **Dynamic networks:** Link prediction makes full use of info extracted from previous graphs to reveal the underlying network evolving patterns, then predict the future network status.
2. **Dynamic Network Link Prediction:** Aims to learn function that maps the input sequence. Find the networks are going to appear or disappear at the next timespan.

**What is E-LSTM-D Framework?**-A combination of encoder-decoder architecture and stacked LSTM.

-Encoder as highly non-linear network structure (dealing with spatial non-linearity and sparsity) and decoder as extracted features back to the original space (learn temporal dependencies).

**What is Encoder-decoder architecture?**

-Inspired by Autoencoder, which can efficiently learn representations of data in a supervised way.

-Encoder is composed of multiple non-linear perceptions, projects the high dimensional graph data into a relatively lower dimensional vector space.

-Encoder layer processes every term separately and then concatenates all the activations in the order of time

-Decoder layer uses sigmoid as the activation function rather than ReLU .

-And the number of units of the output layer always equals to the number of nodes.

**What is Stacked LSTM?**

-As a special kind of recurrent neural network (RNN)

-Can learn long-term dependencies

-Use of LSTM cells to store long-term memory and filter out the useless information.

- The encoder at the entrance could reduce the dimension for each graph and thus keep the computation of the stacked LSTM at a reasonable cost. And the stacked LSTM which is advanced at dealing with temporal and sequential data is supplementary to the encoder in turn.

**Major experiments for the reading**

1. Select Datasets: Perform the experiments on five real-world dynamic networks

* CONTACT (human contact dynamic network of face-to-face proximity)
* ENRON (email networks)
* RADOSLAW (email networks)
* FB-FORUM (Facebook-like online forum, online social network)
* LKML (linux kernel mailing list)

1. Baseline Methods

* Node2vec (maps the nodes of a network from a high dimensional space to a lower dimensional vector space)
* TNE (models network evolution as a Markov process and then use the matrix factorization to get the embedding vector for each node.)
* CtRBM (first generates a vector for each node based on temporal connections and predict future linkages by integrating neighbor information.)
* GTRBM (takes the advantages of both tRBM and GBDT to effectively learn the hidden dynamic patterns)
* DDNE (uses a GRU as an encoder to read historical information and decodes the concatenated embeddings of previous snapshot into future network structure.)

1. Evaluation Metrics

* Area Under the ROC Curve (AUC) (measure the performance of a dynamic link predictor)
* GMAUC (a metric specifically designed for measuring the performance of DNLP. It combines PRAUC and AUC by taking their geometric mean)
* Error Rate (defined as the ratio of the number of mis-predicted links)

**Experiments Result**

-E-LSTM-D model outperforms all the baseline methods in almost all the cases, no matter the network is large or small, dense or sparse, for both short-term and long-term prediction

-E-LSTM-D and other DNLP baselines can get much better performances, due to their dynamic nature.

-node2vec can easily predict much more links than the truly existing ones, leading to relatively large Error Rates.

-TNE performances poorly on Error Rate

-The dramatic difference of the Error Rate between E-LSTM-D and TNE indicates that this metric is a good addition to AUC to comprehensively measure the performance of DNLP.

-ctRBM and DDNE, have similar performances while they could not compete with E-LSTM-D in most cases.

-long-term prediction on structure is indeed relatively difficult for most dynamic networks

-We can make better long-term prediction on FB-FORUM and LKML.

-Although some methods have excellent performances on AUC, they might mis-predict many links

**Parameter Sensitivity**

1. Influence of the model’s structure: the general structure of E-LSTM-D can achieve state-of-art performance in most cases.
2. Influence of historical snapshot length N: phenomenon suggests us to choose N = 10 in the previous experiments.
3. Influence of the penalty coefficient β: suggest us to choose a relatively small β.