Introduction

The size of knowledge bases have grown significantly in recent years. Linked Open Data is more and more integrated in all large datasets and most search engines use graphs to analysis with. But doing inference, such as link prediction and entity prediction, over these large graphs can be difficult. As most prediction and clustering algorithms have a hard time understanding graphs and instead of using triples of [subject, predicate, object] these algorithms use feature vectors as inputs. As in recent years several of these algorithms have gained a lot of traction within the AI community, such as deep neural networks and convolutional neural networks. It would be great be able to use these algorithms for link prediction and node classification tasks. But this means that graphs and triples first must be transformed in feature vectors before the algorithm can be used.\\

In the last couple of years advances have been made to transform these triples into feature vectors. This technique, called graph embedding, tries to embed nodes to a point in space while preserving proximity to other nodes that are close, or are in other ways related to the node, by having for example similar connections.\\

The advantages of this transformation are that all nodes can now be embedded in a low dimensional Euclidean space. With the nodes now embedded applying any type of machine learning algorithm has become a lot easier.\\

There are different methods to embed graphs into low dimensional feature vectors. In this paper we focus on sequence-based graph embedding methods, such as \textit{Node2Vec}, \textit{Struc2Vec}, and \textit{Diff2Vec} are used. These methods are based on \textit{Word2Vec}; a method that learns word embeddings from raw text, \textit{Word2Vec} is computationally-efficient model and thus very useful for large datasets. The model walks over the graph and tries to build sequences by using the edges to walk from node to node.\\

The feature vectors must capture the underlying graph as good as possible. This can only be done if the sequences capture the graph correctly. This is a great challenge as graphs have a large variation of structures. Some graphs have many connections between nodes, and some graphs have sparse connections except for a few nodes. This is where the different embedding methods diverge. All three models try to capture different aspects of the graph. This can mean that some models work better on type of graph and other models on different types of graphs.\\

My Contribution. I propose a ensemble method that is more robust to different types of graphs, by combining \textit{Struc2Vec} and \textit{Diff2Vec}. I propose two techniques to make the models more robust, shown in figure: /ref{>><<}. The first technique uses both the \textit{Struc2Vec} and the \textit{Diff2Vec} model, but both vectors are concatenated and then inserted into the link prediction model. Which gives the link prediction model the possibility to decide which information to keep, which model is more useful to predict the link connection and which model is not as useful.  
The second method uses the vocabulary that \textit{Struc2Vec} and \textit{Diff2Vec} produce with their respective walks and combines the vocabulary to learn the feature vectors on this larger vocabulary. Then apply this to the link prediction model. In the remaining part of the paper the model is called the \textit{StrucDiff2Vec}. \\

The expectation is that the ensemble model will cover a broader spectrum of graphs while retaining the same accuracy. The expectation is that the concatenated method will score similar accuracy on the dataset as highest one of the two models. The second method will score accurately on a broad spectrum and will probably retain the same accuracy on many different graphs.

The paper will be structured as follows, In Section \ref{} we overview related work, on which this paper is based. The method used in this paper is discussed in Section \ref{}. Experimental results are analyzed in Section \ref{} and the paper concludes with Section \ref{}.\\

Related Work

As we propose an ensemble method it how the different parts of the ensemble method are built. The ensemble method uses two different models, namely \textit{Diff2Vec} and \textit{Struc2Vec}. Both these models originated from \textit{Node2Vec}. Which in its turn is an improvement on \textit{DeepWalk}

Node2Vec

Diff2Vec

Struc2Vec

Method