Extraction of Defect features from Thermographic welding videos using Machine Learning

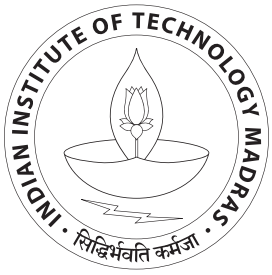
A Project Report

submitted by

VISHRUIT KULSHRESHTHA

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
and
MASTER OF TECHNOLOGY



DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Extraction of Defect features from Thermographic welding videos using Machine Learning**, submitted by **Vishruit Kulshreshtha (ME12B159)**, to the Indian Institute of Technology Madras, for the award of the degree of **Bachelor of Technology in Mechanical Engineering** and **Master of Technology in Thermal Engineering**, is a bonafide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Balaraman Ravindran
Research Guide
Associate Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Prof. Prabhu Rajagopal
Research Guide
Associate Professor
Dept. of Mechanical Engineering
IIT-Madras, 600 036

Prof. Srikrishna Sahu
Research Guide
Assistant Professor
Dept. of Mechanical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: May 2017

ACKNOWLEDGEMENTS

TODO

I wish to express my sincere thanks to **Dr. Balaraman Ravindran**, my research adviser, for his tremendously invaluable and patient guidance throughout the course of this project. The discussions I have had with Prof. Ravindran have been a constant source of encouragement and hope that helped me develop an obsession with the problem and helped me keep exploring for ideas.

I would also like to offer my special thanks to **Karthik Thyagrajan**, (MS Student, Department of Computer Science IIT Madras) for his constant support and motivation as a friend.

I would like to to express my sincere thanks and deep sense of indebtedness to my guide Dr. Balaraman Ravindran for his guidance and motivation throughout my work. His inspiring suggestions motivated me to solve problems efficiently. I am also grateful to my guide also for providing access to the RISE LAB GPUs without which none of my experiments could have been performed.

I would also thank Priyesh Vijayan, Sivasankar, Beethika Tripathy and all my colleagues in Rise LAB who always helped me whenever needed in my research.

Finally, I wish to thank my parents, my friends, and the faculty of Department of Computer Science, IIT Madras for their kind support and assistance.

ABSTRACT

KEYWORDS: Thermography, CMT Welding, Non-Destructive Evaluation (NDE), COMSOL simulations, deep learning, CNN, Video Segmentation

Deep learning techniques have proven successful in Computer Vision for its adaptability to capture the minute information in any data. But it is less studied in relational networks. Recent papers tried to incorporate deep learning techniques on single attribute single relational data where they get the embeddings of the nodes and based on those embeddings, one can do collective classification, link prediction, anomaly detection and various others methods which requires feature vector representations. I focus on extending deep learning techniques to learn the embedding of single-attribute multi-relational (SAMR) data for which till recent time, very few papers has been published. I propose two novel architectures which learns embedding in SAMR data and using which I focus on problems of link prediction and collective classification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
HMM	Hidden Markov Model
RF	Random Forest
RLCS	Rough Longest Common Subsequence
TF-IDF	Term Frequency-Inverse Document Frequency
MFCC	Mel Frequency Cepstral Coefficients
SVM	Support Vector Machine
IOI	Inter Onset Interval
LCS	Longest Common Subsequence

CHAPTER 1

INTRODUCTION

1.1 Overview

Rather than addressing problems like collective classification and link prediction directly, in recent years, many papers focused on creating a latent representation of the network so that those feature representations can be fed to off-the-shelf machine learning algorithms to solve our problems. At the same time, Deep learning has seen wide range of growth from image and text representations to graph network. It is more intuitive that deep learning performs extremely well in achieving state-of-the-art performance almost in all fields due to its computational power.

Many deep learning based embedding papers such as Mikolov *et al.* [2013*a*]; Perozzi *et al.* [2014]; J *et al.* [2016]; Pan *et al.* [2016]; Chang *et al.* [2015] have been proposed in recent two-three years to address embeddings in various network models. All those papers doesn't learn embeddings in multi-relational data whereas Heterogeneous Network Embedding (HNE) Chang *et al.* [2015] is too complex in its architecture as it uses convolutional neural network. Many real world applications consist of multi-relational data and not many papers have been published in this type of network.

1.2 Motivation

1.3 Major Contribution

- System is based on content based recommendation rather than widely used collaborative filtering based recommendation.
- System eliminates the cold start problem where a new song is hard to recommend.
- System takes into consideration various aspects of music like genre, melody and lyrics to recommend which makes it highly versatile.

1.4 Organization of Thesis

I propose two simple methods to address the multi-relational network problem. In Chapter **??**, I will be talking about my literature survey in various fields followed by Section III in which I will show some of my results achieved in implementing two papers. Section IV depicts the proposed architecture and Section V is the future work.

Chapter **??** discusses in detail the collective classification task performed for seven classes of songs. Next chapter in line is chapter **??** where initial attempts for classification of songs based on likes and dislikes of user is given. This chapter discusses direct song to song matching techniques that were tried for recommending songs. After attempting this task, model based techniques were tried which is discussed in chapter **??**. Experiments performed on all the above discussed techniques are discussed in the respective chapters only. Chapter 7 concludes the work with critical analysis of work done.

CHAPTER 2

LITERATURE SURVEY

In this section, I have categorized my literature survey into three parts namely link prediction, collective classification and network embedding with deep learning techniques. I started my survey by reading papers on different link prediction methods and implemented a paper which is explained in Section III. Then I read papers on collective classification and papers based on network embedding.

2.1 Non Destructive Evaluation (NDE)

Link prediction attempts to estimate the likelihood of the existence of a link between two nodes based on observed links and attributes of nodes. Given a graph,

$$G = (V, E)$$

where V is set of vertices and E is set of edges between V, U denotes set of all possible edges, $N = U - E$ denotes set of non-existent edges and $M \subset U$ represents missing or future edges, one has to find the missing or the future edges M. General problem in link prediction is the class imbalance between missing links with that of possible links. To solve using machine learning algorithms, one has to perform either sampling, active learning or cost-sensitive learning. Link prediction has many applications such as in biological networks where it is expensive to experimentally establish each link, spurious link detection where

we need to remove those links from the network, link discovery problems like recommendation of products or friends to users, etc.

The Link Prediction Problem for Social Networks Liben-Nowell and Kleinberg [2003] :

This is the basic paper to address the link prediction problems and its solutions are categorized into proximity based link predictions, random walk based link predictions and matrix factorization based link predictions.

Proximity based Link Predictions : Proximity measure can be divided into local neighborhood and global path based approach. Local neighborhood approach is based on the neighborhood information available in the network. They capture similarities between nodes and they predict links such that more similar nodes typically form a link. Global path based approach calculates promixity of two nodes by capturing the paths between them in various ways in the network. Global path based approach is computationally expensive when compared to local neighborhood based approach as they may need to scan across the network if the two nodes are apart from each other.

Random Walk based Link Predictions : Performing random walk from a vertex captures the neighborhood information of that vertex. Two nodes can be similar if they are present in the random walk generated by those vertices. Random walks generated by the two vertices may not be same.

Matrix Factorization based Link Predictions : The adjacency matrix can be factorized based on low rank approximation to represent the latent feature vectors in low dimensional space where the factorized matrix captures the information of users.

Method	Formula
Common Neighbors	$\|\Gamma(x) \cap \Gamma(y)\|$
Adamic/Adar	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log\|\Gamma(z)\|}$
Resource Allocation	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\|\Gamma(z)\|}$
Jaccard's Coefficient	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x) \cup \Gamma(y)\|}$
Preferential Attachment	$\|\Gamma(x)\| \cdot \|\Gamma(y)\|$
Salton Index	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{\sqrt{\|\Gamma(x)\| \cdot \|\Gamma(y)\|}}$
Sorenson Index	$\frac{2\|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x)\| + \|\Gamma(y)\|}$
Hub Promoted Index	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{min(\|\Gamma(x), \Gamma(y)\|)}$
Hub Depressed Index	$\frac{\|\Gamma(x) \cap \Gamma(y)\|}{max(\|\Gamma(x), \Gamma(y)\|)}$
Shortest Path	$min\|p_{u->v}\|$
Katz	$\sum_{l=1}^{\infty} \beta_l \|p^l_{u->v}\|$

Table 2.1: Proximity based Link Prediction methods

Chance-Constrained Programs for Link Prediction Janardhan Rao Doppa :

This paper aims to handle imbalance between classes explicitly. Semi-definite programs such as inter point methods can be efficiently used to solve the convex optimization problem of Chance constraint and Second Order Cone Programs (SOCP). These SOCP methods are employed to handle for feature selection, missing features and unbalanced data. They proposes clustering based SOCP (CBSOCP) where they assume the class-conditional densities of positive (link formation) and negative classes as mixture models and they assume those components have spherical co-variances. Then they learn (μ, σ^2) for clusters by clustering the mixture models separately. Finally they find an hyperplane to separate positive and negative clusters. Fig 2.1 depicts the geometric interpretation of CBSOCP.

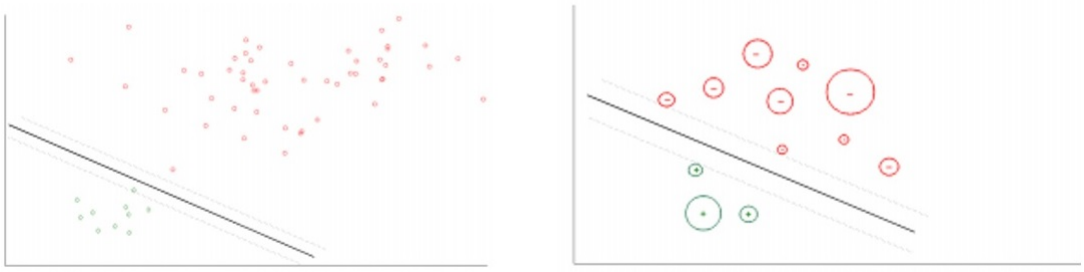


Figure 2.1: Geometric interpretation for SVM and CBSOCP.Janardhan Rao Doppa

Centrality-Aware Link Recommendations Parotsidis *et al.* [2016] :

This paper aims to reduce the closeness centrality of all nodes. They try to bring small world effect in the network where any node in the network can be reached from any node within few hops. They use existing link recommendations algorithms to suggest possible link formation to users where they aim on selecting those links such that closeness centrality of those nodes is reduced.

Collective Prediction of Multiple Types of Links in Heterogeneous Information Networks Cao *et al.* [2014] :

As conventional promixity measures fail to work on heterogeneous information network, they design a relatedness measure (RM) based on linkage homophily principle where two nodes can form a link if most of their similar nodes are linked together. They extend co-training method which handles label correlation between homogeneous network to heterogeneous network such that most confident predictions of unlabelled links are shared between the network.

Handling Class Imbalance in Link Prediction Using Learning to Rank Techniques Bopeng Li :

Due to the presence of large number of absent links, the misclassification error is not suitable for performance measure. Instead of misclassification error, many suggests to use ranking performance measures such as area under curve (AUC), average precision and normalized discounted cumulative gain (NDCG). This paper talks about link prediction as learning to rank problem and they develop a link prediction method based on cross entropy surrogate used in ListNet ranking algorithm (Cao et al. 2007) **?**.

2.2 Thermography

Collective Classification is the problem of label predictions in the network. This problem finds many application in our day-to-day life like who influence other objects in the network, epidemiological network, predicting political affiliations and so on. Many real world data do not provide adequate label information, so this problem tends to be semi-supervised classification problem. This problems get more difficult when only a few labeled data is available. The semi-supervised learning method should be capable to handle this deficiency of labeled data. Label correlation is the key aspect in collective classification where the label of object o can have correlation with that object's attribute or they can have correlation with the neighborhood attributes or they can have correlation with the unobserved labels in the neighborhood.

Approximate Inference Algorithms for Approaches based on Local Conditional Classifiers :

Iterative Classification Algorithm (ICA) : This is an iterative algorithm where in each iteration, all unobserved nodes are taken and for those nodes in that iteration,

we calculate the likelihood of the labels of a node based on its neighborhood labels predicted by the local classifier. We only assign a label to a node which has the maximum likelihood for that label. This iteration process can be continued till all class labels have been classified or till a threshold is reached.

Gibbs Sampling (GS) : The original algorithm of Gibbs Sampling is computationally expensive as they need to determine the convergence of the procedure. Thus an approximate version of gibbs sampling is proposed where they assume the conditional probability distribution given by the local classifier is correct. Gibbs sampling is very similar to ICA except that the number of iterations is initially fixed a to period called "burn-in". Once the burn-in period is reached, the algorithm start having a statistical count for each unobserved label, a count of labels assigned to the nodes. After certain sampling of the nodes, each node is assigned a label which has the maximum count from the statistical count.

Multi-label Collective Classification in Multi-attribute Multi-relational Network DataVijayan *et al.* [2014] :

This paper is the one of the very few paper which handles multi-label classification in multi-attribute and multi-relational (MAMR) data. They propose a co-training style framework which learns a classifier (f^l) for each label (l ϵ L). The classifier f^l can be break down into two classifiers f_a^l and f_g^l where the classifier f_a^l is learned on each attribute view (a ϵ A) and the classifier f_g^l is learned on each relational view (g ϵ G) separately. This is the minimization problem of the three loss functions L_1, L_2 and L_3 where L_1 loss function is the loss on labeled data on both relational and attribute views. L_2 and L_3 loss functions updated in two steps

in the co-training style proposed framework.

$$min(\underbrace{L_1(A,G,L,T)}_{\text{Loss on labeled data}} + \underbrace{L_2(A,G,L,U) + L_3(A,G,L,U)}_{\text{Loss on unlabeled data}})$$

$$L_1(A,G,L,T) = \sum_{l \in L}(\underbrace{\sum_{a \in A} \mathcal{L}(f_a^l(A^a_{T_l}), Y^l_{T_l})}_{\text{Attribute views' disagreement}} + \underbrace{\sum_{g \in G} \mathcal{L}(f_g^l(G^g_{T_l}), Y^l_{T_l})}_{\text{Relational views's disagreement}})$$

$$L_2(A,G,L,U) = \sum_{l \in L}(\underbrace{\sum_{a \in A}(f_a^l(A^a_{U_l}) - \prod_{a \in A} f_a^l(A^a_{U_l}))^2}_{\text{Disagreemnet among attribute views}} + \underbrace{\sum_{g \in G}(f_g^l(G^g_{U_l}) - \prod_{g \in G} f_g^l(G^g_{U_l}))^2}_{\text{Disagreemnet among relational views}})$$

$$L_3(A,G,L,U) = \sum_{l \in L}(\underbrace{\sum_{a \in A}(f_a^l(A^a_{U_l}) - \prod_{g \in G} f_g^l(G^g_{U_l}))^2}_{\text{Attribute views' disagreemnet with relational views}} + \underbrace{\sum_{g \in G}(f_g^l(G^g_{U_l}) - \prod_{a \in A} f_a^l(A^a_{U_l}))^2}_{\text{Relational views' disagreemnet with attribute views}})$$

2.2.1 Network Embedding with Deep Learning Techniques

Efficient estimation of word representations in vector space Mikolov *et al.* [2013*a*] :

This paper generally called as word2vec proposes two new log-linear models for learning distributed representations of words. In order to effectively minimize computational complexity caused by the non-linear hidden layer in the existing neural network language models (NNLM), they propose two simpler models which can be trained on large data more efficiently.

Continuous Bag-of-Words Model (CBOW) : This is similar to feedforward NNLM where non-linear hidden layer is removed and the projection layer is shared across all words. A log-linear model is built using few words from history

and few words from future as input and the output is the current word. In general, the model will find the likelihood of the current word given those words.

Continuous Skip-gram Model (Skipgram) : This model is exactly opposite to CBOW model described above. Given the current word as input, it tries to maximize the likelihood of those words left and right to the current word.

Hierarchical Softmax is used in the output layer as present in the typical NNLM but instead of balanced binary tree, huffman tree based hierarchical softmax is employed to give better computationally better speed up. As the embedding of the words are generated using anyone of the above two methods, one can do some simple algebraic operations between vectors and the resultant vector thus produced would be closer to the target word embedding. A simple architecture of these two models is shown in Fig 2.2. Example : Vector(”King”) - Vector(”Man”) + Vector(”Women”) is close to the word representation of the Vector(”Queen”).

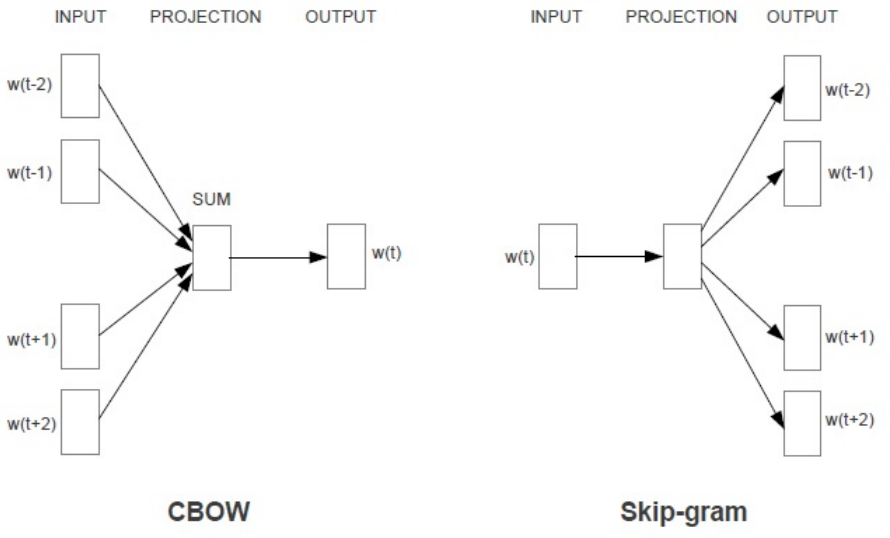


Figure 2.2: The CBOW architecture predicts the current word based on the context, and the skipgram model predicts the surrounding words given the current word Mikolov *et al.* [2013*a*].

Distributed Representations of Words and Phrases and their Compositionality Mikolov *et al.* [2013*b*] :

This paper proposes several variants to skipgram model presented in Mikolov et al. Mikolov *et al.* [2013*a*]. They replace the complex hierarchical softmax used in the output layer with a simplified variant of noise contrastive estimation (NCE) called Negative Sampling (NEG). The Skipgram and the CBOW models do not need a full probabilistic model as achieved by hierarchical softmax, so they are trained using logistic regression where they try to discriminate the noise targets from real targets. Thus NEG samples k words from the noise distribution and maximize the log likelihood of the target words. In addition to NEG, they proposed subsampling of frequent words. The main idea is that the frequent words like ”the”, ”and” and ”a” occurs millions of times and when they are added to words like ”France” or ”Paris”, it doesn't makes the phrase meaningful. To avoid those words, each word is discarded with the probability,

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

where t is the threshold and f(w) is the frequency of the word.

Heterogeneous Network Embedding via Deep Architectures Chang *et al.* [2015] :

This paper mainly talks about data embedding where it is used to create low-dimensional feature representations by providing deep embedding algorithm for networked data. There are very few papers which talked about linking structural and content information using deep learning techniques. This paper does the same in completely supervised scheme. A non-linear multi-layered embedding

function can capture linkage information and rich content and produce multi dimensional representation for an object in low dimensional space. After that, one generally need to feed it to any link prediction algorithm to solve it. This paper proposes heterogeneous network embedding (HNE) which maps the multi-attribute data to a unified latent representation. They learn embedding in three ways : i) They learn embedding between two images (node - node), ii) They learn embedding between texts of the same image (attribute - attribute) and iii) They learn embedding between image and texts of the same image (node - attribute). They use convolutional neural network to learn all these embedding where weights are shared between all these learning. Different embeddings are generated at these three learnings and then the way to combine the embeddings to form a single latent representation is unclear in the paper. They also proposed an extension of this architecture for MAMR data by stacking the relational matrices which they claim that it is not scalable for large graphs.

DeepWalk: Online Learning of Social RepresentationsPerozzi *et al.* [2014] :

Deepwalk learns latent representation of nodes in a graph by embedding the structural property of the nodes. On obtaining those embedding, one can perform off-the-shelf machine learning algorithms to perform link prediction, collective classification, anomaly detection, cluster analysis, etc. It gets the local information of the nodes by a series of short random walks performed at each node and then it is fed to the skipgram model proposed by Mikolov et al. Mikolov *et al.* [2013*a*,*b*]. The reason of choosing the skipgram model is that the distribution of words appearing in natural language is similar to the distribution of vertices appearing in random walk which follows a power-law distribution.

This vertex representation modeling yields the following optimization problem :

$$min(\phi) \qquad -logPr(v_{i-w}, ..., v_{i-1}, v_{i+1}, ..., v_{i+w})|\phi(v_i))$$

where given a random walk sequence of a vertex, we need to find the probability of the other vertices occurring in that random walk. This increases the probability of the vertices in the neighborhood of the current vertex thus capturing the structural information of the graph. Once this model is trained with hierarchical softmax at output layer, the embedding are retrieved from the weight matrix in the projection layer.

Author2Vec: Learning Author Representations by Combining Content and Link Information J *et al.* [2016] :

This is a short paper where they extend deepwalk Perozzi *et al.* [2014] approach to learn both content and link information thus addressing single attribute single relational network data. They focus on co-authorship network where they learn embeddings for authors. They first learn content information of an author and then enrich that author's representation by capturing link information to get the final embeddings.

Content-Info Model : The textual content here is the abstract of the author and its feature representation is retrieved by paragraph2vec Le and Mikolov [2014] which converts paragraph to vector representation. This model gets the author's vector representation closer to the vector representation of the author's published papers whereas it deviates from other irrelevant papers' vector representation away from the author's vector representation.

Link-Info Model : This model is similar to the content-info model and it aims

to bring the similarity between authors who published papers together closer in vector representation and the other authors away from it.

Tri-Party Deep Network RepresentationPan *et al.* [2016] :

This paper published in IJCAI-2016 is very similar to that of author2vec (WWW-16) J *et al.* [2016] described above has an additional model designed to learn the labels of the vertices. In architecture wise, author2vec learns vertex representation in content-info model and then fed to link-info model, where as in this paper two embeddings are jointly learned in a coupled neural network model. The architecture of this model is given in Fig. 2.3.

Inter-node Relational Modeling : It learns the embedding of the nodes given the

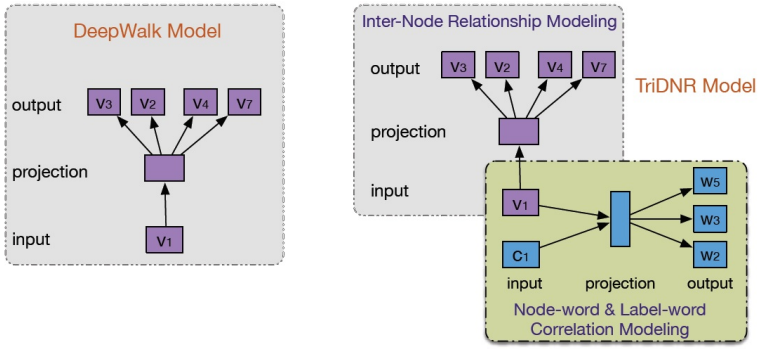


Figure 2.3: Architecture of Deepwalk model and TriParty model.Pan *et al.* [2016]

series of random walk sequences of all nodes.

Node-content Correlations Assessing and Label-content Correspondence Modeling : Given a node and labels of its documents, it learns both the representation of its contents as well as the input label vectors.

Connections : The two layers mentioned above are coupled together and the representation of nodes are learned together.

Scaling up Link Prediction with Ensembles Duan *et al.* [2016] :

This paper applies Non-Negative Matrix Factorization (NMF) in the ensemble decomposed components of the original graph such that it approximates the search space of possible link predictions to $O(n^2)$ comparisons. NMF for collaborative filtering algorithms are widely used in recommender systems for its effective prediction of score of any user with any item given the matrix is sparse. The authors have chosen Symmetric NMF of form,

$$W = FF^T$$

where the forbenious norm of $(W - FF^T)$ is minimized.

But the challenge in link prediction is that the adjacency matrix is very large when the number of nodes goes over million. To counter this problem, ensemble decomposition of the original graph is proposed such that applying Symmetric NMF over those decomposed components is feasible. Ensemble decomposition is now achieved by following sampling methods :

1. Random Node Bagging : Randomly select all nodes with probability f, where f is the fraction of the graph to be broken and select all its neighbors.

2. Edge Based Bagging : It is similar to snowball sampling where they randomly select a node and its subsequent neighbors with probability 0.5 and this process repeats until the total number of nodes reaches the fraction f.

2.3 Machine Learning

2.3.1 Support Vector Machines

2.3.2 Convolutional Neural Networks

2.3.3 Autoencoders

2.3.4 Segmentation / Object Detection / Object localization

Optical Flows

CHAPTER 3

DATASET

CHAPTER 4

CLASSIFICATION ON ATTRIBUTE

4.1 Introduction

Attribute information in a network is generally an information about the nodes itself. While we do any classification on a network, it is better to know about the nodes apart from the structural information of the network. These information can be in the form of text as in the case of social network where the information can be about the users, bio data, educational qualifications, current workplace, etc., and in the case of movies, it can be plot, genre, summary or cast of the movie. In various cases, one can find these attribute information serves better clarity in classification of the network when compared to the structural information. For example, summary information of the movies gives as more clarity to classify its genres than given the structural information of the movies where two movies have links if they same actor or director. So in the context of embedding on attribute information, I tend to classify the network only considering the attribute information and later use it to learn collective classification along with structural information in Chapter 6.

4.2 Related Work

Attribute information in the form of text are studied as the problem of text summarization and classification based on attributes in the network can be related

to the problem of document classification. Various natural language processing based text summarization techniques can be incorporated to get attribute classification. We focus on creating an embedding from term frequencyinverse document frequency (tf-idf) technique to achieve our task since our dataset contains tf-idf matrices. Simple ways of getting embeddings from tf-idf matrix are Latent Semantic Analysis (LSA), Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). I tend to get the embeddings from tf-idf matrix through denoising autoencoder.

4.3 Dataset

Datasets for classification on attribute information are the tweets of the users in twitter dataset and the tags of the movies in movielens dataset. Both of them are available to us in tf-idf matrices.

- Twitter Dataset : Tf-idf matrix of the tweet contents is of size 854 X 35352, where there are 854 users and 35352 words in the tweets.
- Movielens Dataset : Tf-idf matrix of the tags of the movie is of size 3911 X 1854, where there are 3911 movies and 1854 tags.

4.4 Denoising Autoencoder

We use denoising autoencoder to get the embeddings of the tf-idf matrix. There is only one hidden layer of size equivalent to the size of our embeddings (128). The stacked autoencoder of several levels turned out to be a poor performer in our case as the fed in features are very sparse and the model gave an embedding more or less same values for all the 128 features.

- Model : $\hat{x}_i = f(x_i) = O(W^{1T}(W^1 x + b_1) + b_2)$
- Parameters : $\theta = W^1, b_1, b_2$
- Loss Function : $min \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2$
- Optimizer : RMSProp

4.5 Experimental Results

CHAPTER 5

CLASSIFICATION ON STRUCTURAL INFORMATION

5.1 Introduction

Any task done on the network typically considers the structure of the network rather than the attribute information of the nodes in the network. The structural information generally considers various features related to the neighbors or other nodes in the network like degree, centrality measures, random walk, neighbors, similarity measures, etc. Taking other nodes of the network into account, one can solve numerous tasks related to the network like clustering, classification, community detection, link prediction, etc. In this chapter, I deal with only structure of the network and I tend to learn the representation of the network with it. The relational data which is basically the network of nodes provides key information in solving in most cases as opposed to the attribute information, for example in the case of social network, one can find the community of the user based on his/her connections in the social network.

5.2 Related Work

Embedding on the network considering structural information has wide range of studies and in the past three years, there are numerous papers have been published

in the field of deep learning itself. Starting from deepwalk, Perozzi *et al.* [2014]

5.3 Dataset

The dataset for this task are the relational information of twitter dataset and movielens dataset.

- Twitter Rugby Dataset : It has 6 relational views created based on the twitter users' followers, followed by, retweets, retweeted by, mentions and mentioned. There are 854 users and not every view has all 854 users as nodes when they form the network.
- Movielens Dataset : It has 2 relational views such as actor-actor graph and director-director graph created out the movielens dataset, where two movies are linked if they have same actor/director.

5.4 Model : Deepwalk (tensorflow)

The simple and the effective way of getting an embedding for relational data is Perozzi *et al.* [2014]. More faster and efficient way of the author's version has been implemented by gensim. Their version is only cpu based code and still one can get good performance in terms of speed. But to incorporate the attribute information into this structural embedding in 6, gensim's version of code didn't result in good performance in terms of classification. So I modelled the working of deepwalk in a simpler way in tensorflow to certain extent such that it solves the basic ideas of Mikolov *et al.* [2013*a*] considering not only the random walks as proposed in Perozzi *et al.* [2014] but also including the neighbors of the node while training the model.

- Data : $\{x_i, y_i\}_{i=1}^{N}$, where x_i is the one-hot vector of the node and y_i is the multi-hot vector in which nodes appearing in the context window of random of the

node as well as the neighbors of the node will have ones in the vector and all others zero.

- Model : $\hat{x}_i = f(x_i) = O(W^{1T}(W^1 x + b_1) + b_2)$
- Parameters : $\theta = W^1, b_1, b_2$
- Loss Function : $min \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2$
- Optimizer : Adam

5.5 Experimental Results

CHAPTER 6

CLASSIFICATION ON ATTRIBUTE AND STRUCTURAL

6.1 Introduction

In chapter 4 and chapter 5, we saw the performance of attribute and relational data when considered individually. In this chapter 6, we focus on our main goal of embedding on single attribute multi-relational data. I

6.2 Related Work

6.3 Dataset

6.4 Feature Selection

6.5 Song Summarization

6.6 Classifiers Used

6.7 Experimental Results

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

7.2 Criticism and Possible Future Work

Thus I have proposed two simple frameworks to address multi-relational network problem. They are capable to use the existing deepwalk model Perozzi *et al.* [2014] with an expense of additional computation required to get the information across the relations. The normal deepwalk code which is available in online is CPU based. So I'll try to use converted tensorflow based deepwalk in GPU for scalability.

APPENDIX A

CLASSIFICATION AND FEATURE EXTRACTION TECHNIQUES USED

A.1 Collective Classification

Lorem ipsumLorem ipsumLorem ipsumLorem ipsumLorem ipsumLorem ipsum-Lorem ipsumLorem ipsumLorem ipsumLorem ipsumLorem ipsumLorem ipsum-Lorem ipsumLorem ipsumLorem ipsumLorem ipsumLorem ipsum

Approximate Inference Algorithms for Approaches based on Local Conditional Classifiers :

Iterative Classification Algorithm (ICA) : This is an iterative algorithm where in each iteration, all unobserved nodes are taken and for those nodes in that iteration, we calculate the likelihood of the labels of a node based on its neighborhood labels predicted by the local classifier. We only assign a label to a node which has the maximum likelihood for that label. This iteration process can be continued till all class labels have been classified or till a threshold is reached.

Gibbs Sampling (GS) : The original algorithm of Gibbs Sampling is computationally expensive as they need to determine the convergence of the procedure. Thus an approximate version of gibbs sampling is proposed where they assume the conditional probability distribution given by the local classifier is correct. Gibbs sampling is very similar to ICA except that the number of iterations is initially fixed

a to period called ”burn-in”. Once the burn-in period is reached, the algorithm start having a statistical count for each unobserved label, a count of labels assigned to the nodes. After certain sampling of the nodes, each node is assigned a label which has the maximum count from the statistical count.

A.2 Classification Techniques

A.2.1 Logistic Regression

REFERENCES

Adamic, L. and **E. Adar** (2001). Friends and neighbors on the web. *Social Networks*, **25**, 211–230.

Backstrom, L. and **J. Leskovec**, Supervised random walks: Predicting and recommending links in social networks. *In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0493-1.

Bopeng Li, A. T., Sougata Chaudhuri, Handling class imbalance in link prediction using learning to rank techniques.

Cao, B., **X. Kong**, and **P. S. Yu**, Collective prediction of multiple types of links in heterogeneous information networks. *In Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14. IEEE Computer Society, Washington, DC, USA, 2014. ISBN 978-1-4799-4302-9.

Chang, S., **W. Han**, **J. Tang**, **G.-J. Qi**, **C. C. Aggarwal**, and **T. S. Huang**, Heterogeneous network embedding via deep architectures. *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15. ACM, New York, NY, USA, 2015. ISBN 978-1-4503-3664-2.

Duan, L., **C. Aggarwal**, **S. Ma**, **R. Hu**, and **J. Huai**, Scaling up link prediction with ensembles. *In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-3716-8.

J, G., **S. Ganguly**, **M. Gupta**, **V. Varma**, and **V. Pudi**, Author2vec: Learning author representations by combining content and link information. *In Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016. ISBN 978-1-4503-4144-8.

Janardhan Rao Doppa, P. T. L. G., Jun Yu, Chance-constrained programs for link prediction.

Le, Q. V. and **T. Mikolov** (2014). Distributed representations of sentences and documents. *CoRR*, **abs/1405.4053**.

Liben-Nowell, D. and **J. Kleinberg**, The link prediction problem for social networks. *In Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03. ACM, New York, NY, USA, 2003. ISBN 1-58113-723-0.

Menon, A. K. and **C. Elkan**, Link prediction via matrix factorization. *In Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'11. Springer-Verlag, Berlin, Heidelberg, 2011. ISBN 978-3-642-23782-9.

Mikolov, T., **K. Chen**, **G. Corrado**, and **J. Dean** (2013*a*). Efficient estimation of word representations in vector space. *CoRR*, **abs/1301.3781**.

Mikolov, T., **I. Sutskever**, **K. Chen**, **G. Corrado**, and **J. Dean** (2013*b*). Distributed representations of words and phrases and their compositionality. *CoRR*, **abs/1310.4546**.

Pan, S., **J. Wu**, **X. Zhu**, **C. Zhang**, and **Y. Wang**, Tri-party deep network representation. *In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 2016.

Parotsidis, N., **E. Pitoura**, and **P. Tsaparas**, Centrality-aware link recommendations. *In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-3716-8.

Perozzi, B., **R. Al-Rfou**, and **S. Skiena** (2014). Deepwalk: Online learning of social representations. *CoRR*, **abs/1403.6652**.

Prithviraj Sen, M. B. L. G. B. G. T. E.-R., Galileo Namata, Collective classification in network data.

Satchidanand, S. N., **H. Ananthapadmanaban**, and **B. Ravindran**, Extended discriminative random walk: a hypergraph approach to multi-view multi-relational transductive learning. *In Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015.

Tang, J., **M. Qu**, **M. Wang**, **M. Zhang**, **J. Yan**, and **Q. Mei** (2015). LINE: large-scale information network embedding. *CoRR*, **abs/1503.03578**.

Vijayan, P., **S. Subramanian**, and **B. Ravindran**, Multi-label collective classification in multi-attribute multi-relational network data. *In* **X. Wu**, **M. Ester**, and **G. Xu** (eds.), *ASONAM*. IEEE Computer Society, 2014. ISBN 978-1-4799-5877-1.

Wang, C., **V. Satuluri**, and **S. Parthasarathy**, Local probabilistic models for link prediction. *In Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07. IEEE Computer Society, Washington, DC, USA, 2007. ISBN 0-7695-3018-4.