Final Project Report (CSE 519): Ranking arXiv papers

Amol Damare
adamare@cs.stonybrook.edu
SBU ID: 107914028

Punit Mehta
pmmehta@cs.stonybrook.edu
SBU ID: 111461860

December 6, 2017

[Appointment time: 2nd slot on 11th December at 4 PM]

1 Introduction

In this project, we consider a problem of ranking an arXiv paper. Specifically, we try to rate/rank a paper in a network consisting of other relevant papers based on how popular the paper is. Ranking a scientific paper is a non-trivial task even for a field expert. We tried to solve some aspect of such paper ranking by transforming the underlying problem to predict a 'score' of a node in a graph. In particular, we create a citation graph from our dataset and apply deep-walk to generate the latent representations of each paper. Later, we consider different machine learning models to rank the papers in the citation graph based on such representations in our training data.

The organization of the report is as follows: In section 2, we start with the motivations on how we model our problem using latent representations. In section 3, we describe our general approach that we use across all the models to rank a paper. In section 4, we comment on our naive baseline model and some of its limitations. Section 5 proposes a better baseline model based on our leanings from the observations on the previous baseline. We outline an advanced model in Section 6 that we extensively evaluate in Section 7 and 8 comparing it with the two proposed baselines using various evaluation techniques. We show some results of our advanced model in Section 9 and conclude the report in Section 10.

2 Social behavior of scientific papers

A paper can be related to another paper through various ways such as - one paper cites other paper, both papers cite the same paper, they are accepted in the same journal/conference, they can have the same authors etc. We can use these relationships to form a network or graph of papers. When we think of scoring these papers, we can say that score of a paper will be dependent on the scores of the papers it is connected to in such graph. There is a correlation between scores of connected papers. So, when we have a new paper we can score it by looking at the scores of the papers it is connected to. In this way, network of papers behaves similar to any social network. This relationship allows us to apply recent developments in social relational learning [1] ,[11]to our problem. Also it allows us to model our problem as a relational learning problem [3], [2].

3 Our approach

The basic approach that will be used in scoring papers is given in Algorithm 1 Now, we will

Algorithm 1 High level algorithm to get ranks of papers

1: Create a graph of papers capturing relations such as citations, authors, conferences etc.
2: Get latent social embeddings for the vertices of this graph.
3: Create train and test data using some helper ranking function.
4: Train a model on this embedding to predict the score/rank of paper.

go in details of each step of this algorithm. Step 1 is trivial for our baseline model. We have created the graph using just citations as a criteria to get a citation graph of of papers. It's possible to extend this graph to include other relationships as well, as we have the necessary data to build such graph. Figure 1 shows the fields of our dataset we have for all the papers.

Field Name	Field Type	Description	Example
id	string	MAG or AMiner ID	53e9ab9eb7602d970354a97e
title	string	paper title	Data mining: concepts and techniques
authors.name	string	author name	Jiawei Han
author.org	string	author affiliation	department of computer science university of illinois at urbana champaign
venue	string	paper venue	Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial
year	int	published year	2000
keywords	list of strings	keywords	["data mining", "structured data", "world wide web", "social network", "relational data"]
fos	list of strings	fields of study	["relational database", "data model", "social network"]
n_citation	int	number of citation	29790
references	list of strings	citing papers' ID	["53e99ef4b7602d97027c2346", "53e9aa23b7602d970338fb5e", "53e99cf5b7602d97025aac75"]
page_stat	string	start of page	11
page_end	string	end of page	18
doc_type	string	paper type: journal, book title…	book
lang	string	detected language	en
publisher	string	publisher	Elsevier
volume	string	volume	10
issue	string	issue	29
issn	string	issn	0020-7136
isbn	string	isbn	1-55860-489-8
doi	string	doi	10.4114/ia.v10i29.873
pdf	string	pdf URL	//static.aminer.org/upload/pdf/1254/ 370/239/53e9ab9eb7602d970354a97e.pdf
url	list	external links	["http://dx.doi.org/10.4114/ia.v10i29.873", "http://polar.lsi.uned.es/revista/index.php/ia/ article/view/479"]
abstract	string	abstract	Our ability to generate…

Figure 1: Paper data model and its fields [9]

3.1 Latent social representation of graph

In step 2 of our algorithm 1, we need to get latent social representation of the graph we constructed in step 1. We have used 'DeepWalk' algorithm presented in [1] to get the social embedding of the citation graph. DeepWalk applies deep learning techniques to learn the latent social representations of each vertex in the network. To do this [1] has presented idea that random walks in graph is similar to short sentences in a special language whose vocabulary is the set of vertices of the network. Hence, we can apply same deep learning techniques that are used in neural language models to learn latent representation of each vertex. We will be using the same algorithm to get the representation of each vertex in our citation graph. There are other methods to get social representation of a vertex such as Spectral clustering[11] which uses spectral graph theory(eigen values of Laplcian of graph), [2] uses eigen values of Modularity matrix of graph etc. Deepwalk works well on all types graph and works very well even if the graph is sparsely labeled in relational classification task [1], which is exactly kind of representa-

tion we would need to rank the papers as we have very huge graph (166,192,182 vertices) often with very sparse labeling (as we will be using experts to rank few papers).

3.2 Intuitive Ranking Helper Function

We don't have any intuitive way to rank or score papers based on social representations we got from running DeepWalk on the graph. Also, it is very difficult to get a generalized scoring/ranking function that works universally well. But if we just consider one of the parameters to decide if a paper is good or not then we can have an intuitive ranking function that we can easily explain. But this ranking function does not generalize well. In this project, we propose the idea that we generalize this naive but intuitive ranking function by training some regression model on latent social representation of the graph.

4 Baseline (1) : Venue score is equivalent to a paper score

4.1 Idea

Initially, we considered a naive function that ranks a paper based on the conference or journal it was accepted in. (We can easily get the impact factors of conferences and journals and we follow the intuition that if a venue at which the paper is published is good then paper is as good as the venue itself.) Using this naive intuitive function to score a paper, we trained a linear regression model with input vectors as deep-walk's latent representations (using semi-supervised learning). In this baseline model, graph was constructed using citations and our ranking helper function was only considering conferences that these papers were published in. Algorithm 2 and 3 give detailed dynamics of the model. The detailed evaluation on this model is presented in Evaluation section.

Algorithm 2 Algorithm for Baseline model (1)

Create graph $G(V, E)$ using citations.
G_{train}, G_{test}=Split(G) ▷ Get random papers from graph to create training and testing datasets
IntuitiveRankingHelperFunction(V_{train})
IntuitiveRankingHelperFunction(V_{test})
Initialize F as array of feature vectors
F=DeepWalk(G) ▷ This runs deepwalk on graph to get vector representation of each vertex
model=LinearModel(G_{train}, F) ▷ Initialize a linear model
Evaluate *model* on G_{test}
ranks=model.predict(G)
return ranks

Algorithm 3 Algorithm for intuitively ranking papers

procedure IntuitiveRankingHelperFunction(V)
    Initialize VenueRank table ▷ this table has ranks of each conference and/or journal
    for $vinV$ **do**
        $rank[v] = rank[Venue[v]]$ ▷ Naive way to rank the paper, Paper is as good as conference/journal it is published in
    return $rank$

4.2 Observations

During the experiments with this baseline model, as the prediction of linear regression model highly depends on the output features in the training data (paper scores which in fact are venue scores for our model), we found that it's really important to give an appropriate score to each paper if we want to predict a reasonable score.

When we tested our baseline model (1), we found that some lower values in the rating-scale were missing due to the missing conferences in our training dataset as our intuitive ranking function completely depends on the venue-score. Hence, we need a better way to rank the papers in the first place in order to predict them better later.

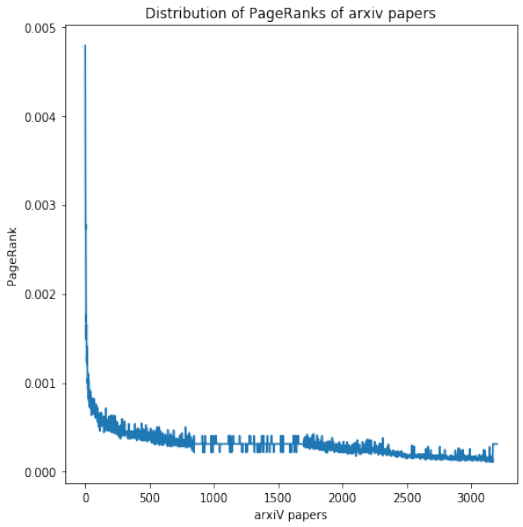

Figure 2: PageRank Ratings

5 Baseline (2) : PageRank as a scoring function

5.1 Idea

As we noted in the previous section, our baseline model (deep-walk on citation graph and linear regression on the deep-walk's parameters) highly depends on the score that we initially give to each paper (semi-supervised learning). If we can't give a reasonable rating to the papers in the first place, the linear regression will be trained on mostly erroneous data and the output prediction might not be helpful. Hence, to improve our base-line model, we clearly focus on the ways we can attach a score to a paper in order to create a trainable dataset. One of the most popular choices of giving a rank to a graph of nodes is PageRank [6].

Therefore, we again create a citation graph of all the papers and run PageRank algorithm on the ciration graph as part of our new baseline model. This way, we get a rank of each paper in the dataset and we can treat it as a score of our training dataset. The output of the PageRank algorithm is shown in Figure 2. Note the power law distribution generated by PageRank here. Here, x-axis corresponds to various arXiv papers and y-axis is rating generated by PageRank.

5.2 Linear regression using PageRank score

Now, we have a citation graph with scores from page rank algorithm and we generate the deep-walk representations for the same. The feature vector of deep-walk has 64 dimensions (which is a smaller representation of a node in such a huge citation graph) and is an input feature to our Linear Regression model. Here, we believe that since deep-walk representations capture the social relations between the nodes in the citation-graph, it's a good idea to train on them as academic papers connected to (referred by or being referred in) a paper with good score likely to be a good paper.

We train the regression model on latent representations of deep-walk (our input feature) and the scores (our output value) we got from PageRank algorithm. The trained model should predict the score of a paper given the latent representation of it in the citation graph. We evaluated this model on the previous baseline model and other competitive models. We noticed a significant improvement in the residual graph of this new baseline model. Detailed results are explained in the Evaluation section.

6 Advanced model : PageRank + DeepWalk + kNN

6.1 Idea

Intuitively, we believe that similar nodes (based on the out-degree of a node) in the citation-graph should be given similar scores. We note that we construct the graph in such a way that out-degree of a node corresponds to the number of citations of a paper mapped to that node in the network. To exploit such similarities, it's a good idea to apply k Nearest Neighbors algorithm to the DeepWalk feature vectors as DeepWalk also in some sense captures the same notion in a social graph.

Hence, we start with rating the papers using PageRank (similar to what we did for our improved baseline model) and generate DeepWalk representations. We take these feature-vectors (number of dimension = 64) along with the PageRank scores and apply k Nearest Neighbors algorithm in order to train our mode. We use a different subset of our dataset than the one used to train the model in order to test this advanced model so that we can see how well our advanced model generalizes. The evaluation on the test dataset is presented in the Evaluation section with more details. Here, the optimal value of k is 4 that we got by trying out various possibilities and observing the results. In the next section, we detail out the evaluation strategy and compare the results we got for all three models.

7 Evaluation

We consider five methods to evaluate all of our three models and compare them with each other: (1) Residual graphs of the error in prediction, (2) Citations of a paper as an evaluation metric, (3) Confusion Matrix of our ratings, (4) Error Histogram, (5) General error-scores (mean square, training/testing error etc.) We end the evaluation section by showing the permutation-test results of our advanced model.

7.1 Residual graphs of error in prediction

For Residual graphs, we plot the error in prediction (i.e prediction score - actual score) and see how the error points are distributed for each predicted value. We can see a trend of improvement

from 3 to 5 as the points are tending towards a random distribution from linear around the zero-error.

For our Baseline Model (1), we don't get the score which is less than 6. This is surprising as testing dataset is a random sample of our entire dataset and should ideally follow the same distribution that we have in our training dataset. We debugged down the issue and found that since we manually give scores to the conference, there are some conferences for which we don't have papers in our testing dataset and hence, linear regression couldn't predict the smaller values corresponding to the papers published in those less prestigious journals. Hence, we missed the lower values. That infact was one of our motivations to apply PageRank to all the papers and have an unbiased score that solely depends on the structure of the citation graph.

For our Baseline Model (2), we can see a linear trend around zero error and our kNN model has an almost random distribution in that region. Hence, we conclude that kNN performs better than our previous two baselines.

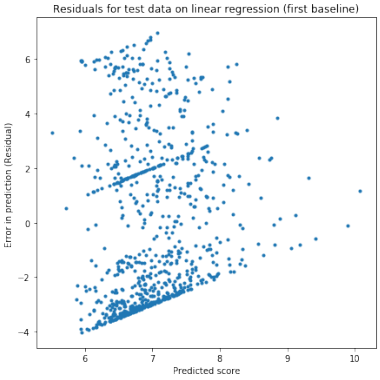

Figure 3: Residual plot: Baseline (1)

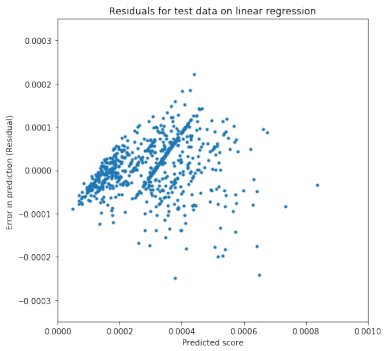

Figure 4: Residual plot: Baseline (2)

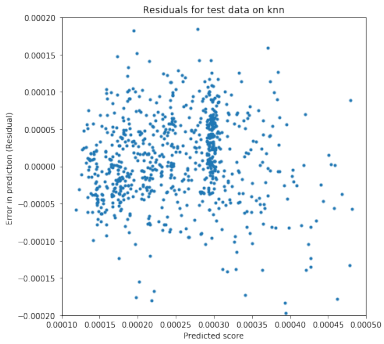

Figure 5: Residual plot: kNN

7.2 Citations as an evaluation metric

The second evaluation technique that we consider is about inferring the correctness of prediction from the citations of a paper. The fact that papers with high citations should relatively have higher score helps us in determining how well our models are scoring as against the citations of papers. For such an evaluation we plot the Figures 6, 7 and 8. We split the citations in three windows colored red, green and black (that is red points correspond to papers with citations from 0 to 3, green from 4 to 10 and black from 10 to higher values)

As we can see that, our baseline model (1) gives higher scores to papers with very low citations (red points on the right) and some papers with high citations are given relatively low score (green points on the left). This behavior is less observed in baseline model (2) and kNN tries to cluster the papers with similar number of citations. In that regard, kNN works better here. However, it does have some outliers (black points on the left with very low score) and kind of under-performs for such papers.

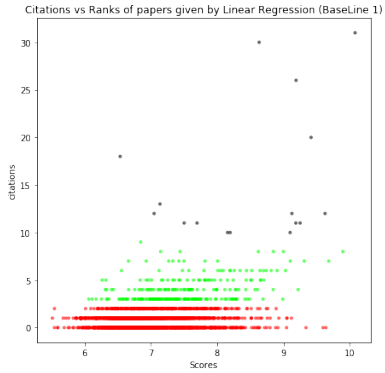

Figure 6: citations vs rating: Baseline (1)

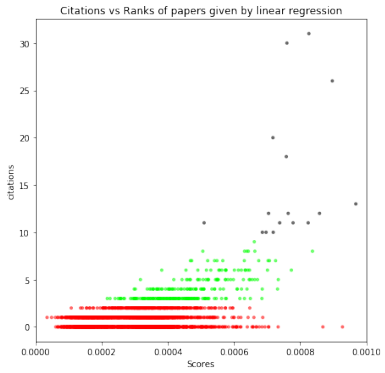

Figure 7: citations vs rating: Baseline (2)

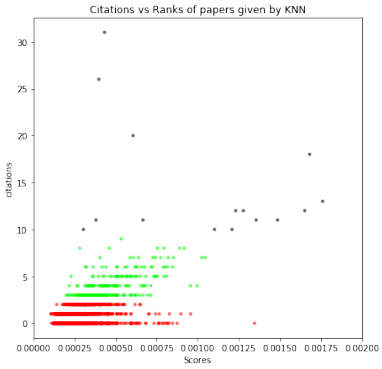

Figure 8: citations vs rating: kNN

7.3 Confusion matrix

We have confusion matrix for our three models that we considered in this project as shown in Figures 9, 10 and 11. We divided the scores we got from pagerank and scores we got from our model into 10 different bins to get the true and predicted label for each paper.

For baseline model (1), we can again observe that it doesn't predict any values which is less than 6 irrespective of what the actual score of the paper is. However, papers corresponding to lower actual values (true label) are given relatively low score (that is 6,7 in predicted label) and we can see papers with high predicted label when we approach high values in actual score. Having observed that, we can see a significant scope of improvement in this model. Baseline model (2) performs better than model (1) to distribute the papers across entire matrix in a meaningful way. However, we can still see many papers which are misclassified here as there are small values in the diagonal of its confusion matrix. The confusion matrix for kNN looks acceptable as there are high values in the diagonal and almost all the diagonal values are filled up and can conclude that kNN does a pretty good job here in rating the papers.

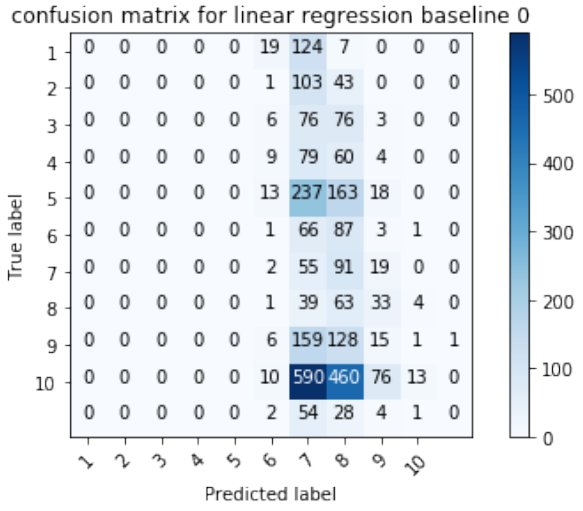

Figure 9: Confusion Matrix: Baseline (1)

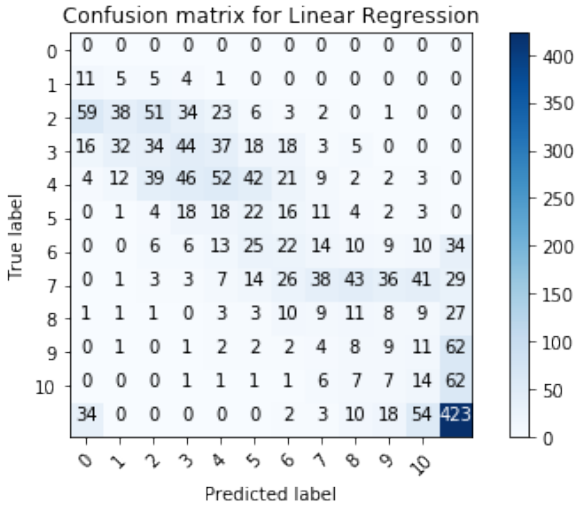

Figure 10: Confusion Matrix: Baseline (2)

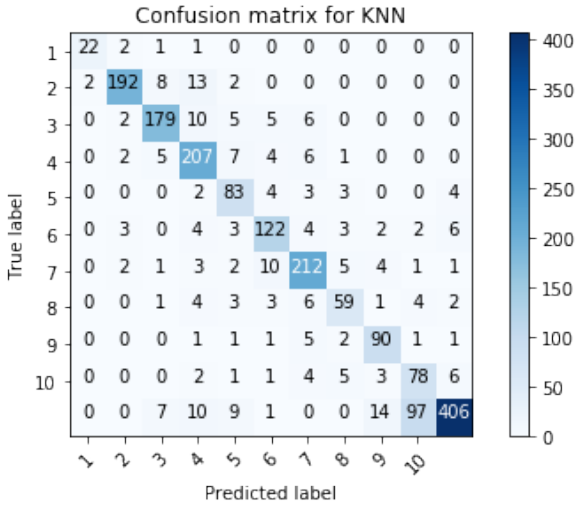

Figure 11: Confusion Matrix: kNN

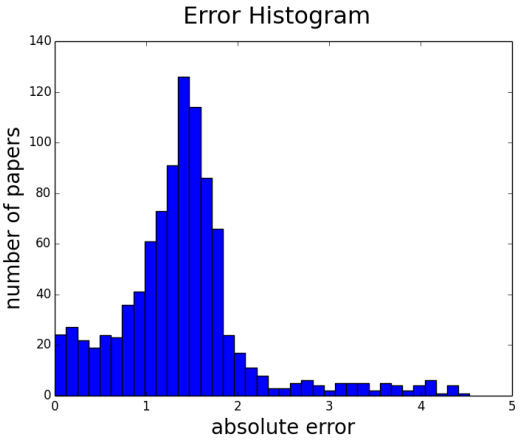

Figure 12: Error Histogram: Baseline (1)

7.4 Error histograms

We plot error histograms for both the baseline models and kNN as show in Figures 12, 13 and 14. We notice a reasonably good error-histogram for kNN and can see that most of the errors are completely centered around zero.

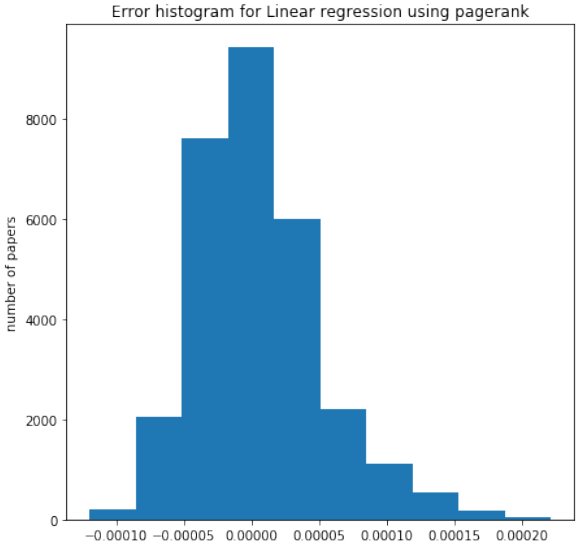

Figure 13: Error Histogram: Baseline (2)

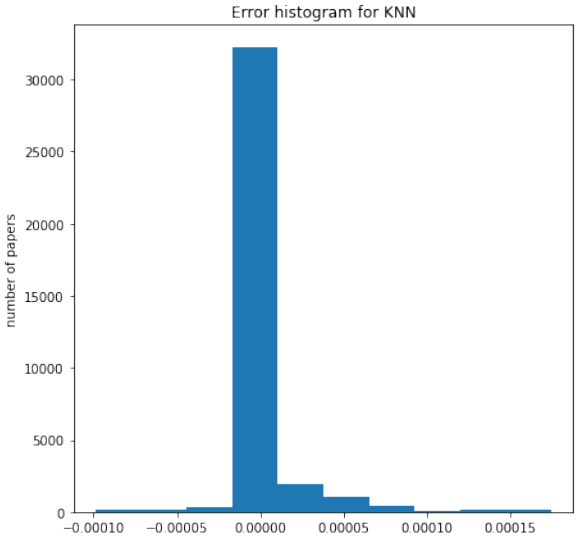

Figure 14: Error Histogram: kNN

7.5 General error-scores

We have tabulated the important error values for our new baseline and kNN in Figure 15. We can see a decline in the error values for our kNN model from our baseline model.

Errors \ Models	Linear Regression	KNN	Ridge Regression
Mean Squared Error	1.95E-08	1.41E-08	1.95E-08
Root Mean Squared Error	5.86E-05	1.90E-05	5.86E-05
Absolute Mean Error	5.86E-05	1.90E-05	5.86E-05
Mean P-value	0.0019	0.0059	0.0019

Figure 15: General error values

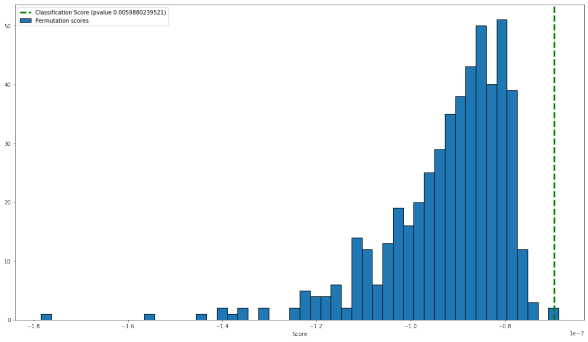

Figure 16: Permutation Test: kNN

7.6 P-test on advanced model

We performed a p-test on our advanced model that uses PageRank, DeepWalk and kNN and the results are shown in Figure 16. Our p-value is 0.005 for kNN. The p-test just proves that our models are better than just predicting on random noise. We have already shown in our evaluation using confusion matrix and residual graphs that kNN works much better.

8 Ridge Regression

During our analysis, we observed that the residual graph of linear regression (Figure 3) has some linear trend and were trying to understand it. That time, we found that if we have such linear trend, ridge regression is a good model to try for prediction. However, we got almost the same results (shown in Figures 17, 18, 19) as we got for linear regression and didn't notice any substantial difference.

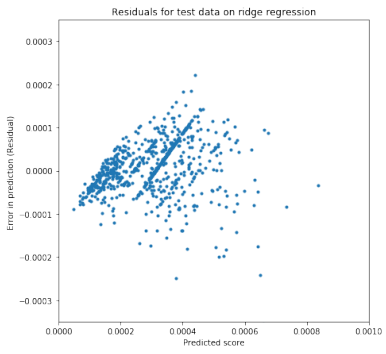

Figure 17: Residual Graph: Ridge Regression

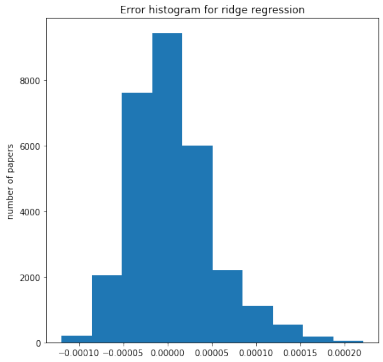

Figure 18: Error Histogram: Ridge regression

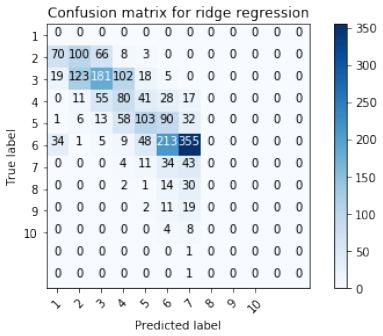

Figure 19: Confusion Matrix: Ridge Regression

9 Results of Advanced model

In this section, we show the results of our advanced model that we apply to our dataset. We basically listed out two types of papers: Top-5 and Bottom-5. Figure 20 and 21 show the respective category of papers from our dataset.

Authors	*Title*	*Venue*	*Year*
Jairo Rocha,Theo Pavlidis	A solution to the problem of touching and broken characters	ICDAR-1	1993
Kun Gao	A Uniform Parallel Optimization Method for Knowledge Discovery Grid	KES	2008
S. Navaladian, B. Viswanathan, T. K. Varadarajan, R. P. Viswanath	Fabrication of Worm-Like Nanorods and Ultrafine Nanospheres of Silver Via Solid-State Photochemical Decomposition.	Nanoscale research letters	2009
Shengqi Ye,Yingjia He,Zuo Zhang	Short-Term Traffic Flow Forecasting Based on MARS	FSKD	2008
Martin Mokrejs, Václav Vopálensky,Ondrej Kolenaty,Tomás Masek,Zuzana Feketová , Petra Sekyrová etc	IRESite: the database of experimentally verified IRES structures (www.iresite.org)	Nucleic Acids Research	2006

Figure 20: Top-5 papers

Authors	*Title*	*Venue*	*Year*
Bo-Yeong Kang,Qing Li	Fuzzy Ranking Model Based on User Preference	IEICE Transactions	2006
Th. Voigtmann, J. Horbach	The dynamics of silica melts under high pressure: mode-coupling theory results	Journal of Physics-condensed Matter	2008
Amit Kumar,Srinivas Peeta	Strategies to Enhance the Performance of Path-Based Static Traffic Assignment Algorithms.	Comp.-Aided Civil and Infrastruct. Engineering	2014
C. D. Mercier ,S. D. Hembree	What the Internet can do for you	IEEE Industry Applications Magazine	1998
Sabine Nick,Judith Andresen,Birgit Lübker,Luzie Thumm	Structure, Design, and Evaluation of an Online Chemistry Course		2003

Figure 21: Bottom-5 papers

10 Conclusion

As part of the course project for CSE 519 (Data Science Fundamentals), we tried to rate the academic papers. Specifically, we converted such rating problem to a graph problem (i.e. citation graph) and rank the nodes of the graph. We devised a naive baseline model to begin with and observed some of its limitations. We improved the baseline by proposing a modified version of it in which our intuitive scoring function uses PageRank than the scores of the venues. We applied DeepWalk in all of our three models and generate latent representations of the nodes in citation graph in order to exploit their social (structural) behavior in the graph. Later, we proposed an advanced model that applies kNN (k=4) using DeepWalk representations. We evaluated our models using standard evaluation techniques and compare their results with eachother. At the end of evaluation, we argue that kNN (our advanced model) performs better

than the baselines and does a fair job in rating the papers though it does have some false positives and true negatives.

11 Code Repository

We have made our code public. It can be found at https://github.com/amoldamare/CSE519-Project.

References

[1] Perozzi, B. ,Al-Rfou, R.,Skiena, S. DeepWalk: Online Learning of Social Representations Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '14

[2] Tang, Lei and Liu, Huan Relational learning via latent social dimensions Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,2009

[3] Tang, Lei and Liu, Huan Scalable learning of collective behavior based on sparse social dimensions Proceedings of the 18th ACM conference on Information and knowledge management,2009

[4] J. E. Hirsch. An index to quantify an individual's scientific research output, 2005, Proc.Nat.Acad.Sci.46:16569,2005; arXiv:physics/0508025. DOI: 10.1073/pnas.0507655102.

[5] Egghe, L. Theory and practise of the g-index Scientometrics (2006) 69: 131. https://doi.org/10.1007/s11192-006-0144-7

[6] Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank citation ranking: Bringing order to the web (Technical Report). Stanford InfoLab.

[7] Garfield, E. Citation analysis as a tool in journal evaluation Science,178,471-479.

[8] Pinski, G., Narin, F. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics Information Processing and Management, 12(5), 297-312

[9] Microsoft Academic Graph https://www.openacademic.ai/oag/

[10] arXiv.org https://arxiv.org/

[11] Tang, Lei and Liu, Huan Leveraging social media networks for classification Data Mining and Knowledge Discovery,2011,Springer

List of Figures