Survey on Feature Engineering of Author-Paper Pair Matching in Bibliography Data

Samani Ankit , Sanjay Bhanderi

Department of Computer Engineering,
MEF Group of Institutions, Rajkot,
Gujarat, India

ankit.samani@gmail.com,sdbhanderi@gmail.com

***Abstract*— Everyday thousands of research work is published in variety of journals, conference proceedings by academic and industrial researcher across hundreds of various disciplines. The tools and techniques which provide efficient search capability and provide the information for measures about publication like who has been published, which publication and when modern research is challenge. KDDCUP'13 addressed this challenge where given paper has been written by particular author (i.e. author-paper identification) for biographic dataset provided by Microsoft Academic Search Database. Many researchers has published their solutions for this problem based on incorporating selected proper features with different model like, Random Forest, GBDT (Gradient Boosting Decision Tree), FICO scorecard model, RGF. This paper focused on complete review for the solution published for KDDCUP'13 challenges.**

Keywords—Authorship, Feature engineering, Mean Average Precision

I. INTRODUCTION

This work disseminates the first 5 winning papers in KDDCUP-2013 Challenge which positively associates the authorship of a paper by using the concept of Mean Average Precision. The working dataset has been provided by Microsoft Academic Search which is an open platform that provides a variety of metrics and experiences for the research community, in addition to literature search. It covers more than 50 million publications and over 19 million authors across a variety of domains, with updates added each week.

Incorporating Feature Engineering is the first and foremost approach to be followed in which preprocessing and classification based on author & paper combination is performed. A variety of Classification Techniques can be employed to target the above mentioned problem set. Some of the widely used techniques worthwhile to highlight are tabulated in Table I.

II. SURVEY

From the experimental study it is observed that the efficiency of identification of author-paper relation varies with employment of different techniques and features extracted. The evaluation parameter enforced by KDDCUP-2013 Challenge was MAP(Mean Average Precision) which is the average of areas under the precision-recall curve.

Definition: 1 Average Precision

$$Ave_p = \frac{\sum_{k=1}^{N}(P(k) * rel(k))}{N_{pos}}$$

where N is the number of samples (author-paper pairs), Npos is the number of confirmed samples, P(k) is the precision at cut-off k, rel(k) is an indicator function equal to 1 if the sample at rank k is confirmed, 0 otherwise.

A. *Survey Based on Feature Engineering*

Generating Features can build a good model which is critical step and important task which is used to normalize and minimize data for predicting the authorship of given paper. For this, certain features have to be extracted through which the authorship of paper can be decided. Some of the Features Based on Cluster are clusterCoauthor (a, p): the number of coauthors of paper p that matches with any coauthor in the cluster of correct paper coauthors for author a, Features Based on Authors describes Node in bipartite graph are Count-Features in which total count of journals in which author has published his paper and some more like NLP(Natural Language Processing) Feature in which the use of 'Term Frequency(tf)' and 'Inverse Document Frequency(idf)' is calculated for measuring the keywords of all authors with respect to given paper, Features Based on Paper are Count-Feature in which total count of authors are calculated in specific same journal and the other feature used is NLP in which the calculation of tf-idf for measuring keywords of paper with respect to all journals present in dataset has performed, calculation of tf and idf can be done by using

Definition: 2 Term Frequency

$$\mathrm{tf} = \frac{1}{N_p}$$,[2]

where N_p is a number of words in the paper.

Definition: 3 Inverse Document Frequency

$$\mathrm{idf} = \log\left(\frac{N}{N_w}\right)$$,[2]

where N is a number of papers, N_w is a number of papers where the word is occurred[2].

TABLE I : DIFFERENT CLASSIFICATION TECHNIQUES

Sr No.	Technique Name	Central Idea
1.	Random Forest	The technique built multiple decision trees using randomly sub-sampled features which outputs the result by averaging the prediction of individual trees.
2.	Gradient Boosting Decision Tree (GBDT)	The technique built multiple decision trees having the goal to optimize deviance (logistic regression)
3.	LambdaMart	LambdaMART is the combination of GBDT and LambdaRank.
4.	Gradient Boosting Machine	GBM is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble.
5.	Support Vector Machine	The basic idea is to find a hyper plane which separates the d-dimensional data perfectly into its two classes.
6.	K-means	In clustering K-Means algorithm, K initial pointers are chosen to represent initial cluster centers, all data points are assigned to the nearest one, the mean value of the points in each cluster is computed to form its new cluster centre and iteration continues until there are no changes in the clusters. The K-means algorithms iterates over the whole dataset until convergence is reached. [7]
7.	K-Nearest Neighbour Classification	*k*-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.
8.	AdaBoost	ADABOOST is an ensemble (or meta-learning) method that constructs a classifier in an iterative fashion. In each iteration, it calls a simple learning algorithm (called the base learner) that returns a classifier, and assigns a weight coefficient to it.
9.	Page Rank	PageRank produces a static ranking of the given paper list according to
10.	Classification and Regression Tree(CART)	The CART decision tree is a binary recursive partitioning procedure capable of processing continuous and nominal attributes both as targets and predictors with a sequence of nested pruned trees.
11.	Scorecard Technique	Scorecard model has a binning step to divide each predictor space into bins, and then assign score weight to each bin.

Now the Features based on Author-Paper describes the edge in bipartite graph are Multiple Source in which there are many records having same paper id and author id and the examples for this number of the author-paper pair is to be identified from dataset. . Other feature which are based on name and affiliation in which the author affiliation is the key characteristics by which prediction can be done for identifying the author-paper pair and example of extracted values which can be used are name Match(a,p) matches the name of coauthor of papered p with author id a and affiliation Match(a,p) count the number of coauthor affiliation matches with author affiliation from coauthor paper list.

Feature based on Target Leaking has been implemented due to replication of paper id of the author in the dataset and to remove replication *dupPaperId (a, p)* is used in which it is checked whether paper *p* has duplicate ids in the paper id list for author *a* in the train dataset.Similarly there are certain features based on Coauthor are CountCoaNotInAutCsv(a, p) used to find the number of the coauthor which does not appear in the Author dataset, SumPaperOfCoauthor(a,p) used to calculate the total number of paper written together by author and coauthor together. Not only this there are certain similarity features also exist through which name similarity between author can be verified and ambuigity can be reduced/removed by using the formula as highlighted:

NameSimi(name1,name2)

= MWBP(graph(words1,words2))

Min(size(words1),size(words2))

Where MWBP means the maximum weighted bipartite matching on the graph builded by the words1 and words2. There are also many features used like year(p) which extracts the year of paper published ,NumCoauthor(p) which is used to measure the number of different coauthor ids exists in paper id p of given author in dataset.

B. Survey Based on Classification Model

A classification task begins with build data (also known as *training data*) for which the target values (or class assignments) are known. Different classification algorithms use different techniques for finding relations between the predictor attributes' values and the target attribute's values in the build data. These relations are summarized in a model, which can then be applied to new cases with unknown target values to predict target values. A classification model can also be used on build data with known target values, to compare the predictions to the known answers [6]. Some of models used in the papers are Random Forest, Gradient Boosting Decision Tree (GBDT), LambdaMart, Gradient Boosting Machine (GBM) with Bernoulli distribution and Scorecard technique.

1) Random Forest

Random Forests is a tree based learning method introduced by Leo Breiman. The algorithm constructs multiple decision trees using randomly sub-sampled features and outputs the result by averaging the prediction of individual trees. The use of multiple trees reduces the variance of prediction, so Random Forests are robust.[1].Random forests rely on simple averaging of models in the ensemble[11].

2) Gradient Boosting Decision Tree(GBDT)

Gradient boosting Decision Tree is tree based learning algorithm in which tree boosting creates a series of decision trees which together form a single predictive model. A GBDT model is built sequentially by using weak decision tree learners on reweighted data[1].one the most effective advantage of this model is that it is not necessary to initialize predictor variable from the starting of technique, it is allowed to introduce predictor variable in between the process and one of the major drawback of this technique is it cannot build trees in parallel which creates less number of tree during construction of final ensemble model.

3) LambdaMart

LambdaMART is the boosted tree version of Lambda-Rank, which is based on RankNet[9].LambdaMART is a learning to rank algorithm based on Multiple Additive Regression Tree and additively It is nonlinear and computationally efficient.

4) Gradient Boosting Machine(GBM) with Bernoulli distribution

Gradient boosting machines are a family of powerful machine-learning techniques [10]. In gradient boosting machines, or simply, GBMs, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable. The principle idea behind this algorithm is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble. The loss functions applied can be arbitrary, but to give a better intuition, if the error function is the classic squared-error loss, the learning procedure would result in consecutive error-fitting. In general, the choice of the loss function is up to the researcher, with both a rich variety of loss functions derived so far and with the possibility of implementing one's own task-specific loss [10].

5) Scorecard technique

Scorecard technique is based on supervised learning model in which it has a binning step to divide each predictor space into bins, and then assign score weight to each bin.

$$f \ = w_0 + \sum_{i=1}^{I} f_i(x_i),$$

Where $f_i(x_i)$ is the predictor score:

$$f_i(x_i) = \sum_{j=1}^{J_i} w_{ij} b_{ij}(x_i)$$

$$b_{ij}(x_i) = \begin{cases} 1 \text{ if value of } x_i \text{ belongs to the jth bin} \\ 0 \text{ else} \end{cases}$$

w_{ij} : the score weights associated with bin j for predictor x_i.

b_{ij} : the dummy indicator variables for the bins of predictor x_i.

Missing values can be handled easily by using a missing value bin. Given enough bins for a predictor *xi*, the above predictor score function $f_i(x_i)$ is flexible to approximate any general function based on a single predictor, and the scorecard model is the sum of such functions. The complete, compact representation of a model by its bin definitions and weights makes the scorecard a popular, transparent, and easily understood model formulation [3]. By looking into the bin weights for the predictors, lots of insight can be gained for the data which is core advantage for the critical variable generation step for this author-paper pair matching challenge.

III. ANALYSIS

Analysis phase has been carried out through which we obtained the best method which gives good Mean Average Precision and paper[1] has good MAP which gives precise author-paper pair upto 0.98259 MAP and the analysis has been highlighted in Table II. Many new techniques has been used to find the author-paper pair but method used by [1] is they conduct a simple weighted average ensemble and a post-processing procedure by utilizing some strong features. During each stage, they cautiously use the internal validation or the official Valid set to potentially avoid the over-fitting issue. This step is crucial for them to get the best performance on the private leaderboard for predicting data in the Test set.

IV. CONCLUSION

Incorporating all techniques from different papers, we encounters many such Features which are used to classify the dataset and then implied to appropriate classification model which impairs the author-paper relationship and this result can be increased by using certain other available classifiers and ensemble it using proper feature, we can achieve still good result.

TABLE II : ANALYSIS OF DIFFERENT PAPERS BASED ON THEIR MODELS, TECHNIQUES AND FEATURES

Reference Paper	Used Features Based on	Algorithm use	Map	Final model
[1]	• Coauthor Name Matching • Author Consistency • Publication Time • Heterogeneous Bibliographic Networks	Random Forests	0.98259	• Random Forests • Gradient Boosting Decision Tree • LambdaMart
[2]	• Author • Paper • Author-paper	Combination of deep feature engineering and GBM	0.98144	• Gradient Boosting Machine (GBM) with Bernoulli distribution
[3]	• Name and Affiliation • Target Leaking Feature	Scorecard		• Scorecard • Variable Interaction
[4]	• Author Paper Correlation • Author • Paper • Coauthor	GBDT and RGF	0.98120	
[5]	• Contextual Rule-based Feature • Author-Paper Pairs • Author • Paper	Model Averaging	0.9800	• Divide-Conquer

ACKNOWLEDGEMENT

We are thankful to Nitul Dutta for his consistent support, attention and motivation for completing this manuscript.

REFERENCES

[1] Chun-Liang Li, Yu-Chuan Su, Ting-Wei Lin, Cheng-Hao Tsai, Wei-Cheng Chang, Kuan-Hao Huang,Tzu-Ming Kuo and team, “Combination of Feature Engineering and Ranking Models for Paper-Author Identification in KDD Cup 2013,” in KDDCup-2013 Author-Paper Identification Challenge.

[2] Dmitry Efimov, Lucas Silva, Benjamin Solecki,”KDD Cup 2013 - Author-Paper Identification Challenge:Second Place Team” in KDD Cup-2013 Author-Paper Identification Challenge.

[3] Xing Zhao, “The Scorecard Solution to the Author-Paper Identification Challenge” in KDD Cup-2013 Author-Paper Identification Challenge.

[4] Jiefei Li, Xiaocong Liang, Weijie Ding, Weidong Yang, Rong Pan “Feature Engineering and Tree Modeling for Author-Paper Identification Challenge,” in KDDCup-2013 Author-Paper Identification Challenge.

[5] rheng Zhong, Lianghao Li, Naiyan Wang, Ben Tan, Yin Zhu, Lili Zhao, Qiang Yang, “Contextual Rule-based Feature Engineering for Author-Paper Identification,” in KDDCup-2013 Author-Paper Identification Challenge

[6] L. Breiman. Random forests. Machine Learning, 2001.

[7] Julie M. David and Kannan Balakrishnan, “Significance of Classification Techniques In Prediction Of Learning Disabilities”.

[8] XindongWu,Vipin Kumar,J. Ross Quinlan,Joydeep Ghosh,Qiang Yang and Team,”Top 10 algorithms in data mining” ,Springer-Verlag London Limited 2007.

[9] Christopher J.C. Burges,”From RankNet to LambdaRank to LambdaMART: An Overview”, Microsoft Research Technical Report MSR-TR-2010-82.

[10] Alexey Natekin ,Alois Knoll,”Gradient boosting machines, a tutorial”, Front Neurorobot. 2013

[11] Random Forest(http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm)