Predicting Citation Counts for Academic Literature using Graph Pattern Mining

Nataliia Pobiedina[1]* and Ryutaro Ichise[2]

[1] Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
pobiedina@ec.tuwien.ac.at
[2] Principles of Informatics Research Division
National Institute of Informatics, Japan
ichise@nii.ac.jp

Abstract. The citation count is an important factor to estimate the relevance and significance of academic publications. However, it is not possible to use this measure for papers which are too new. A solution to this problem is to estimate the future citation counts. There are existing works, which point out that graph mining techniques lead to the best results. We aim at improving the prediction of future citation counts by introducing a new feature. This feature is based on frequent graph pattern mining in the so-called citation network constructed on the basis of a dataset of scientific publications. Our new feature improves the accuracy of citation count prediction, and outperforms the state-of-the-art features in many cases which we show with experiments on two real datasets.

1 Introduction

Due to the drastic growth of the amount of scientific publications each year, it is a major challenge in academia to identify relevant literature among recent publications. The problem is not only how to navigate through a huge corpus of data, but also what search criteria to use. While the Impact Factor [1] and the h-index [2] measure the significance of publications coming from a particular venue or a particular author, the citation count aims at estimating the impact of a particular paper. Furthermore, Beel and Gipp find empirical evidence that the citation count is the highest weighted factor in Google Scholar's ranking of scientific publications [3]. The drawback about using the citation count as a search criteria is that it works only for the papers which are old enough. We will not be able to judge new papers this way. To solve this problem, we need to estimate the future citation count. An accurate estimation of the future citation count can be used to facilitate the search for relevant and promising publications.

A variety of research articles have already studied the problem of citation count prediction. In earlier work the researchers experimented on relatively small datasets and simple predictive models [4, 5]. Nowadays due to the opportunity to retrieve data from the online digital libraries the research on citation behavior is conducted on much larger datasets. The predictive models have also become more sophisticated due to the

* Supported by the Vienna PhD School of Informatics and NII International Internship Program.

advances in machine learning. The major challenge is the selection of features. Therefore, our goal is to discover features which are useful in the prediction of citation counts.

Previous work points out that graph mining techniques lead to good results [6]. This observation motivated us to formulate the citation count prediction task as a variation of the link prediction problem in the citation network. Here the citation count of a paper is equal to its in-degree in the network. Its out-degree corresponds to the number of references. Since out-degree remains the same over years, the appearance of a new link means that the citation count of the corresponding paper increases. In the link prediction problem we aim at predicting the appearance of links in the network. Our basic idea is to utilize frequent graph pattern mining in the citation network and to calculate a new feature based on the mined patterns – *GERscore* (Graph Evolution Rule score). Since we intend to predict the citation counts in the future, we want to capture the temporal evolution of the citation network with the graph patterns. That is why we mine frequent graph patterns of a special type - the so-called graph evolution rules [7].

The main contributions of this paper are the following:

- we study the citation count prediction problem as a link prediction problem;
- we adopt score calculation based on the graph evolution rules to introduce a new feature GERscore, we also propose a new score calculation;
- we design an extended evaluation framework which we apply not only to the new feature, but also to several state-of-the-art features.

The rest of the paper is structured as follows. In the next section we formulate the problem which we are solving. In the next section we formulate the problem at hand. Section 3 covers the state-of-the-art. In Section 4 we present our methodology to calculate the new feature. Section 5 describes our approach to evaluate the new feature. This section also includes the experimental results on two datasets followed by the discussion. Finally, we draw the conclusion and point out future directions for work.

2 Predicting Citation Counts

We want to predict citation counts for scientific papers. Formally, we are given a set of scientific publications $\mathcal{D}$, the *citation count* of a publication $d \in \mathcal{D}$ at time t is defined as: $Cit(d,t) = |\{d' \in \mathcal{D} : d \; is \; cited \; by \; d' \; at \; time \; t\}|$. To achieve our goal, we need to estimate $Cit(d, t + \Delta t)$ for some $\Delta t > 0$. We can solve this task by using either classification or regression.

Classification Task: Given a vector of features $\bar{X_d} = (x_1, x_2, \ldots, x_n)$ for each scientific publication $d \in \mathcal{D}$ at time t, the task is to learn a function for predicting $CitClass(d, t + \Delta t)$ whose value corresponds to a particular range of the citation count for the publication d at the time $t + \Delta t$.

Regression Task: Given a vector of features $\bar{X_d} = (x_1, x_2, \ldots, x_n)$ for a publication $d \in \mathcal{D}$ at time t, the task is to learn a function for predicting $Cit(d, t + \Delta t)$ whose value corresponds to the citation count of the publication d at the time $t + \Delta t$.

We suggest a new perspective on the citation count prediction problem. We construct a paper citation network from the set of scientific publications $\mathcal{D}$. An example

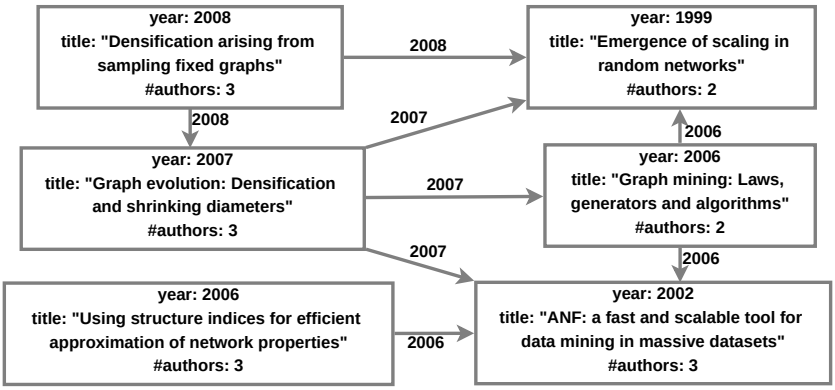


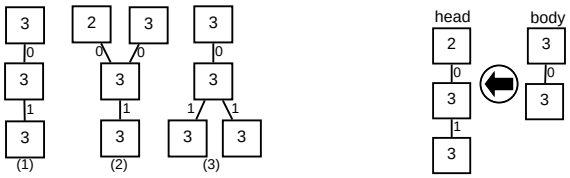
Fig. 1: Example of a citation network.

of a citation network is given in Figure 1. Nodes are papers. A link from one node to another means that the first paper cites the latter. We put the year of the citation as an attribute of the corresponding link. In this setting, citation count of a paper is equal to the in-degree of the corresponding node. Its out-degree is equal to the number of references. Since node's in-degree increases if a new link appears, we can regard the citation count problem as a variation of the link prediction problem in citation networks.

3 Related Work

Yan et al. find evidence that the citation counts of previous works of the authors are the best indicators of the citation counts for their future publications [8]. However, Livne et al. observe that the citation counts accumulated by the venue and by the references are more significant [6]. Furthermore, Shi et al. discover that highly cited papers share common features in referencing other papers [9]. They find structural properties in the referencing behavior which are more typical for papers with higher citation counts. These results indicate that graph mining techniques might be better suited to capture interests of research communities. That is why we formulate the problem of the citation count prediction as a link prediction problem in the citation network. Since feature-based link prediction methods, like [10, 11], can predict links only between nodes which already exist in the network, we use an approach which is based on graph pattern mining [7].

The estimation of future citations can be done with *classification* [12] or *regression* [8, 6, 13]. The classification task, where we predict intervals of citation counts, is in general easier, and in many applications it is enough. Furthermore, a dataset of publications from physics is used in [12], and from computer science in [8, 13]. There are also two different evaluation approaches. The first one is to test the performance for the freshly published papers [6, 12]. The second approach is to predict the citation counts for all available papers [8, 13]. To ensure a comprehensive study of performance of our new feature and several state-of-the-art features, our evaluation framework includes both classification and regression, two evaluation approaches and two datasets of scientific publications.



(a) Relative time patterns. (b) Graph evolution rule.

Fig. 2: Examples of relative time patterns and graph evolution rules. Node labels correspond to the number of authors.

4 GERscore

Our methodology to tackle the stated problem consists of several steps. First, we mine the so-called graph evolution rules in the citation network by using a special graph pattern mining procedure. Then we derive GERscore for each paper using several calculation techniques. We also calculate several state-of-the-art features. All features are obtained using data from previous years. To estimate the performance of these features, we use them in different predictive models on the testing datasets.

4.1 Mining Graph Evolution Rules

To calculate GERscore, we start with the discovery of rules which govern the temporal evolution of links and nodes. Formally, we are given a graph, in our case a citation network, $G = (V, E, \lambda, \tau)$ where λ is a function which assigns a label $l \in L_V$ to every node $n \in V$ and τ is a function which assigns a timestamp $t \in T$ to every edge $e \in E$. Though the citation network in our example is directed, we may infer the direction of links: they point from a new node towards the older one. That is why we ignore the direction and assume that the citation network is undirected.

Definition of relative time pattern [7]: A graph pattern $P = (V_P, E_P, \lambda_P, \tau_P)$ is said to be a *relative time pattern* in the citation network G iff there exist $\Delta \in \mathbb{R}$ and an embedding $\varphi : V_P \rightarrow V$ such that the following three conditions hold: (1) $\forall v \in V_P \Rightarrow \lambda_P(v) = \lambda(\varphi(v))$; (2) $\forall (u, v) \in E_P \Rightarrow (\varphi(u), \varphi(v)) \in E$; (3) $\forall (u, v) \in E_P \Rightarrow \tau(\varphi(u), \varphi(v)) = \tau_P(u, v) + \Delta$.

In Figure 2a we show examples of relative time patterns. For example, the pattern in Figure 2a(1) can be embedded with $\Delta = 2007$ or $\Delta = 2006$ into the citation network in Figure 1 while the pattern in Figure 2a(3) cannot be embedded at all.

Definition of evolution rule [7]: An *evolution rule* is a pair of two relative time patterns called $body$ and $head$ which is denoted as $head \Leftarrow body$. Given a pattern head $P_h = (V_h, E_h, \lambda, \tau)$, the body $P_b = (V_b, E_b, \lambda, \tau)$ is defined as: $E_b = \{e \in E_h : \tau(e) < max_{e* \in E_h}(\tau(e*))\}$ and $V_b = \{v \in V_h : deg(v, E_b) > 0\}$, where $deg(v, E_b)$ corresponds to the degree of node v with regard to the set of edges E_b.

An example of a graph evolution rule is given in Figure 2b. Do not get confused by the fact that body has less edges than head. The naming convention follows the one used for rules in logic. Considering the definition of the evolution rule, we can represent any evolution rule uniquely with its head. That is why relative time patterns in Figure 2a are also graph evolution rules.

To estimate frequency of the graph pattern P in the network G, we use *minimum image-based support* $sup(P) = \min_{v \in V_P} |\varphi_i(v) : \varphi_i \; is \; an \; embedding \; of \; P \; in \; G|$. The *support* of the evolution rule, $sup(r)$, is equal to the support of its head. The *confidence* of this rule, $conf(r)$, is $sup(P_h)/sup(P_b)$. Due to the anti-monotonous behavior of the support, confidence is between 0 and 1. The graph evolution rule from Figure 2b has a minimum image based support 2 in the citation network from Figure 1. The support of its body is also 2. Therefore, confidence of this rule is 1. We can interpret this rule the following way: if the body of this rule embeds into the citation network to a specific node at time t, then this node is likely to get a new citation at time $t + 1$.

Two additional constraints are used to speed up graph pattern mining. We mine only those rules which have support not less than *minSupport*, and which have number of links not more than *maxSize*. Moreover, we consider only those graph evolution rules where body and head differ in one edge. In Figure 2 all rules, except for (a3), correspond to this condition. Finally, we obtain a set $\mathcal{R}$ of graph evolution rules.

4.2 Calculating GERscore

To calculate GERscore, we modify the procedure from [7]. For each rule $r \in \mathcal{R}$ we identify nodes in the citation network to which this rule can be applied to. We obtain a set $\mathcal{R}_n \subset \mathcal{R}$ of rules applicable to the node n. Our assumption is that an evolution rule occurs in the future proportional to its confidence. That is why we put GERscore equal to $c * conf(r)$, where c measures the proportion of rule's applicability. We define three ways to calculate c. In the first case, we simply take $c = 1$. In the second case, we assume that evolution rules with higher support are more likely to happen, i.e., $c = sup(r)$. These two scores are also used for the link prediction problem in [7]. Lastly, if the evolution rule r contains more links, it provides more information relevant to the node n. We assume that such rule should be more likely to occur than the one with less edges. Since evolution rules are limited in their size by $maxSize$, we put $c = size(r)/maxSize$. Thus, we obtain three different scores: $score_1(n, r) = conf(r)$, $score_2(n, r) = sup(r) * conf(r)$, and $score_3(n, r) = conf(r) * (size(r)/maxSize)$.

Finally, we use two aggregation functions to calculate GERscore for node n:

- GERscore$_{1,i}(n) = \sum_{r \in \mathcal{R}_n} score_i(n, r)$,
- GERscore$_{2,i}(n) = \max_{r \in \mathcal{R}_n} score_i(n, r)$.

High values of GERscore can mean two things: either many rules or rules with very high confidence measures are applicable to the node. In either case, the assumption is that this node is very likely to get a high amount of citations. We may have here redundancies. For example, in Figure 2 rules (b) and (a1) are subgraphs of rule (a2). It might happen that these rules correspond to the creation of the same link. Still we consider all three rules, since we are interested to approximate the likelihood of increase in citation counts. Though the summation of individual scores is an obvious selection for the aggregation function, we also consider the maximum. It might turn out that graph evolution rules with the highest confidence are the determinants of future citations.

5 Experiment

5.1 Experimental Data

We use two real datasets to evaluate GERscore: *HepTh* and *ArnetMiner*. The first dataset covers arXiv papers from the years $1992-2003$ which are categorized as High Energy Physics Theory [12]. We mine graph evolution rules for the network up to year 1996 which has $9{,}151$ nodes and $52{,}846$ links. The second dataset contains papers from major Computer Science publication venues [8]. By taking papers up to year 2000, we obtain a sub-network with $90{,}794$ nodes and $272{,}305$ links.

We introduce two additional properties for papers: *grouped number of references* and *grouped number of authors*. For the first property the intervals are $0-1, 2-5, 6-14, 15\leq$. The references here do not correspond to all references of the paper, but only to those which are found within the dataset. We select the intervals $1, 2, 3, 4-6, 7\leq$ for the second property.

We construct several graphs from the described sub-networks which differ in node labels. Since we are not sure which label setting is better, we use either the grouped number of references, or the grouped number of authors, or no label. The choice of the first two label settings is motivated by [12]. With the help of the tool *GERM* [3], we obtain 230 evolution rules in the dataset HepTh, and $4{,}108$ in the dataset ArnetMiner for the unlabeled case. We have 886 rules in HepTh, and 968 in ArnetMiner for the grouped number of authors. For the grouped number of references the numbers are 426 and $1{,}004$ correspondingly.

In total, we obtain 18 different scores for each paper: $GERscore_{1,i}^{(j)}$ for summation and $GERscore_{2,i}^{(j)}$ for maximum, where i equals 1, 2, or 3 depending on the score calculation, and j corresponds to a specific label setting: $j=1$ corresponds to the grouped number of authors as node labels; $j=2$ stands for the unlabeled case; $j=3$ is for the grouped number of references. We report results only for one score for each label setting, because the scores exhibit similar behavior. Since our new score $score_3$ provides slightly better results, we choose $GERscore_{1,3}^{(j)}$ and $GERscore_{2,3}^{(j)}$. Additionally, feature *GERscore* is the combination of all scores.

5.2 Experimental Setting

To solve the classification task (Experiment 1), we consider three different models: multinomial Logistic Regression (mLR), Support Vector Machines (mSVM), and conditional inference trees (CIT). For the regression task (Experiment 2) we take Linear Regression (LR), Support Vector Regression (SVR), and Classification and Regression Tree (CART). We look at a variety of models because they make different assumptions about the original data. We do 1-year prediction in both tasks.

We consider two scenarios for evaluation which differ in the way we construct training and testing datasets. In Scenario 1 we predict the citation count or class label for the papers from the year t by using the data before year $t-1$, like in [6, 11, 12]. In Scenario 2 we take all papers from the year t and divide them into training and test datasets, e.g., as it is done in [8, 13]. We also perform five times hold-out cross-validation.

[3] `http://www-kdd.isti.cnr.it/GERM/`

Table 1: Distribution of instances according to classes (% Total).

Citation Class	**HepTh**			**ArnetMiner**		
	Scenario 1		*Scenario 2*	*Scenario 1*		*Scenario 2*
	Year 1996	Year 1997	Year 1997	Year 2000	Year 2001	Year 2001
Class 1	42.9%	40.33%	34.09%	97.27%	96.69%	88.86%
Class 2	29.81%	26.64%	30.32%	2.51%	3.13%	7.75%
Class 3	13.70%	18.77%	19.85%	0.19%	0.18%	2.40%
Class 4	13.58%	14.27%	15.74%	0.03%	0.01%	0.99%
Total Amount	2,459	2,579	12,113	30,000	25,919	399,647

To compare performance of our new feature, we calculate several state-of-the-art features: *Author Rank*, *Total Past Influence for Authors* (TPIA), *Maximum Past Influence for Authors* (MPIA), *Venue Rank*, *Total Past Influence for Venue* (TPIV), and *Maximum Past Influence for Venue* (MPIV) [8, 13]. To obtain Author Rank, for every author we calculate the average citation counts in the previous years and assign a rank among the other authors based on this number. We put maximum citation count for previous papers as MPIA. TPIA is equal to the sum of citation counts for previous papers. Venue Rank, TPIV and MPIV are calculated the same way using the venue of the paper.

5.3 Experiment 1

We assign class labels in the classification task with intervals $1, 2-5, 6-14, 15 \leq$ of citation counts. In Table 1 we summarize the distribution of instances according to these classes for the data which we use for the training and testing datasets.

We use *average accuracy* and *precision* to evaluate the performance of the classification. If class distribution is unbalanced, then precision is better suited for the evaluation [14]. We summarize the results of the classification task in Table 2. We mark in bold the features which lead to the highest performance measure in each column. The full model is indicated in the row *"All"*. All performance measures are average over the performance measures in 5 runs. The results indicate that the new feature is better than the baseline features and significantly improves the full model.

Due to a highly unbalanced distribution (Table 1), we observe only 1% improvement in accuracy for ArnetMiner in Scenario 2. In the case of HepTh, GERscore is at least 2% better in accuracy than the rest features. Furthermore, in Scenario 2 GERscore improves the accuracy of the full model by more than 9%. Statistical analysis shows that GERscore provides a significant improvement to the full model. If we compare precision rates, then we have that the full model with GERscore is more than 10% better than without it. Moreover, the best achieved accuracy for HepTh in Scenario 1 is 44% in previous work [12]. The accuracy of our full model mLR is 33% higher.

5.4 Experiment 2

To evaluate the performance of the regression models, we calculate the R^2 value as the *square of Pearson correlation coefficient* between the actual and predicted citation counts. In Table 3 we summarize the performance for the regression task. If a feature has "NA" as a value for R^2, it means we are not able to calculate it because the standard deviation of the predicted citation counts is zero. GERscore is significantly better

Table 2: Accuracy (%) and Precision (%) for the Classification Task.

		Scenario 1						Scenario 2					
	Feature	**HepTh**			**ArnetMiner**			**HepTh**			**ArnetMiner**		
		mLR	*mSVM*	*CIT*	*mLR*	*mSVM*	*CIT*	*mLR*	*mSVM*	*CIT*	*mLR*	*mSVM*	*CIT*
Accuracy	*GERscore*	**75.56**	**75.59**	**75.37**	**98.37**	**98.37**	98.36	**76.83**	**74.82**	**75.17**	**95.65**	**95.59**	**95.62**
	Author Rank	73.62	73.56	73.57	98.35	98.36	98.36	72.61	73.19	72.57	94.31	94.47	94.41
	MPIA	73.50	73.39	72.53	98.32	98.36	98.36	70.02	70.79	69.85	94.44	94.49	94.49
	TPIA	73.62	73.58	72.99	98.36	98.36	98.36	70.56	70.97	70.88	94.49	94.48	94.49
	Venue Rank	71.59	71.46	71.55	98.36	98.36	98.36	70	70.33	70.34	94.49	94.49	94.46
	MPIV	66.34	69.73	63.85	98.36	98.36	98.36	67.89	69.08	69.45	94.29	94.49	94.49
	TPIV	70.29	69.49	67.84	98.36	98.36	98.36	69.63	69.61	70.00	94.49	94.49	94.49
	All	76.85	75.91	76.54	98.37	98.36	98.36	81.37	81.11	79.35	96.11	96.11	96.05
	w/o GERscore	74.74	73.48	74.01	98.35	98.36	98.36	74.31	74.1	74.15	94.73	94.93	94.74
Precision	*GERscore*	**43.71**	**36.41**	**39.15**	**39.77**	**35.15**	**26.34**	**51.35**	**47.46**	44.02	**62.69**	**58.42**	**61.2**
	Author Rank	30.92	31.04	31.05	24.17	24.17	24.17	40.87	45.57	**45.15**	35.9	29.49	36.75
	MPIA	31.55	33.03	36.91	24.17	24.17	24.17	37.32	33.37	40.31	32.08	27.17	22.17
	TPIA	30.84	31.91	36.82	24.17	24.17	24.17	37.78	38.24	41.17	22.16	24.6	24.94
	Venue Rank	27.51	24	25.54	24.17	24.17	24.17	33.42	32.62	30.34	22.17	22.17	24.96
	MPIV	24.3	13.93	16.34	24.17	24.17	24.17	21.58	28.05	24.42	25.74	22.17	22.17
	TPIV	25.49	21.99	22.58	24.17	24.17	24.17	26.97	26.85	27.54	23.76	22.17	22.17
	All	48.9	47.75	41.83	39.63	34.04	29.78	61.47	61.2	57.56	62.19	63.82	62.19
	w/o GERscore	38.81	34.72	34.46	24.17	24.17	24.17	47.55	47.09	46.63	48.79	52.66	46.82

than the baseline features for ArnetMiner dataset. Though author related features lead to higher R^2 for HepTh, we see that GERscore still brings additional value to the best performing models (LR in Scenario 1 and CART in Scenario 2) . The analysis of variance (ANOVA) for two models, "All" and "All w/o GERscore", shows that GERscore improves significantly the full model. Our guess is that GERscore does not perform so well for HepTh due to the insufficient amount of mined evolution rules.

5.5 Discussion

Overall our new feature GERscore significantly improves citation count prediction. When classifying the future citations, GERscore is better than the baseline features in all cases. However, author-related features are still better in the regression task, but only for the dataset HepTh. HepTh provides better coverage of papers in the relevant domain, thus the citations are more complete. Another difference of HepTh from ArnetMiner is the domain: physics for the first and computer science for the latter. The last issue is the amount of mined graph evolution rules: we have only 230 unlabeled evolution rules for HepTh. We are not sure which of these differences leads to the disagreement in the best performing features. In [6] the authors argue that such disagreement arises due to the nature of the relevant scientific domains. However, additional investigation is required to draw a final conclusion.

We observe that CART performs the best for the regression task in Scenario 2 which agrees with the results in [8]. However, LR provides better results in Scenario 1. In general, the performance is poorer in Scenario 1. This means that it is much harder to predict citation counts for freshly published papers. It might be the reason why a simple linear regression with a better generalization ability performs well.

Out of all scores which constitute GERscore, the best results are gained for the scores calculated from the unlabeled graph evolution rules (see Table 3). When aggregating separate scores, summation is a better choice compared to maximum. This is

Table 3: Performance measures (R^2) for the Regression Task.

Feature	Scenario 1						Scenario 2					
	HepTh			**ArnetMiner**			**HepTh**			**ArnetMiner**		
	LR	*SVR*	*CART*	*LR*	*SVR*	*CART*	*LR*	*SVR*	*CART*	*LR*	*SVR*	*CART*
$GERscore^{(1)}_{1,3}$	0.011	0.028	0.07	0.02	0.009	0.021	0.063	0.069	0.137	0.138	0.13	0.154
$GERscore^{(2)}_{1,3}$	0.06	0.085	0.103	0.093	0.099	0.087	0.121	0.219	0.26	0.401	**0.431**	0.425
$GERscore^{(3)}_{1,3}$	0.009	0.053	0.091	0.157	0.14	0.169	0.009	0.011	0.065	0.188	0.211	0.209
$GERscore^{(1)}_{2,3}$	0.001	0.015	0.03	0.026	0.022	0.027	0.025	0.039	0.058	0.066	0.087	0.09
$GERscore^{(2)}_{2,3}$	0.005	0.005	NA	0.093	0.214	0.212	0.032	0.057	0.057	0.095	0.135	0.187
$GERscore^{(3)}_{2,3}$	0.069	0.094	0.088	0.097	0.125	0.162	0.001	0.009	0.002	0.094	0.102	0.108
GERscore	0.137	0.119	0.121	**0.233**	**0.219**	**0.213**	0.204	0.205	**0.271**	**0.483**	0.337	**0.429**
Author Rank	0.188	0.098	0.16	0.004	NA	0.004	0.204	**0.302**	0.266	0.133	0.15	0.174
MPIA	0.183	0.181	0.193	0.002	0.001	0.006	0.225	0.209	0.214	0.071	0.041	0.052
TPIA	**0.189**	**0.199**	**0.198**	0	0.001	0.005	**0.285**	0.232	0.21	0.004	0.072	0.063
Venue Rank	0.014	0.029	0.028	0.028	NA	0.014	0.051	0.061	0.05	0.037	0.058	0.054
MPIV	0.022	0.003	0.015	0.001	NA	0.014	0.039	0.048	0.035	0.024	0.023	0.037
TPIV	0.026	0.003	0.021	0	NA	0.004	0.039	0.048	0.035	0.024	0.023	0.037
All	0.245	0.192	0.161	0.235	0.184	0.175	0.371	0.357	0.395	0.513	0.317	0.544
w/o GERscore	0.203	0.120	0.164	0.01	0.004	0.013	0.312	0.289	0.274	0.157	0.149	0.19

an unfortunate outcome since aggregation with maximum would allow us to speed up the graph pattern mining by setting a high support threshold. The decrease in running time is also gained through mining labeled graph evolution rules. Though $GERscore^{(2)}_{1,i}$ provides better results compared to other label settings and aggregation technique, we still receive that the other scores contribute to the combined GERscore.

Our results are coherent with Yan et al. for ArnetMiner in Scenario 2 which is the only setting that corresponds to theirs: Author Rank is better than Venue Rank [8, 13]. However, we show that GERscore is even better in this case. Moreover, we arrive already at a better performance just by identifying graph evolution rules in the unlabeled citation network from the previous years.

6 Conclusion and Future Work

We have constructed a new feature - GERscore - for estimation of future citation counts for academic publications. Our experiments show that the new feature performs better than six state-of-the-art features in the classification task. Furthermore, the average accuracy of the classification is not affected much if we bring in other baseline features into the model. In the regression task the new feature outperforms the state-of-the-art features for the dataset of publications from computer science domain (ArnetMiner), though the latter still contribute to the performance of regression models. Thus, the application of graph pattern mining to the citation count prediction problem leads to better results. However, for the dataset of publications from physics (HepTh) GERscore is not as good as the author related features, i.e., author rank, MPIA and TPIA, though it does contribute to the increase of the performance. Additional investigation is required to identify the reason for the disagreement in the best performing features.

We have performed both classification and regression tasks for the prediction of citation counts in one year. It is interesting to investigate how well GERscore performs for the prediction over five and more years. Our results indicate that the performance

of the model does not always improve if we include more features. Thus, an important aspect to investigate is the optimal combination of features. Ultimately, we want to include our findings into a recommender system for academic publications.

Our future work includes thorough investigation how mined evolution rules influence the predictive power of GERscore. The first issue is to study the influence of input parameters, minimum support (minSup) and maximum size (maxSize), and what is the best combination for them. We need to take into consideration that by setting maxSize high and minSupport low we will obtain more evolution rules, however the computational time will grow exponentially. Another issue is that real-world networks change considerably over time. It may lead to the fact that the evolution rules which are frequent and have high confidence at time t may become rudimentary in ten years and will not be predictive of the citation counts. Thus, we plan to investigate for how long mined evolution rules on average stay predictive. This is an important question also because mining graph evolution rules is computationally hard, and reducing the amount of re-learning GERscores is extremely important.

References

1. Garfield, E.: Impact factors, and why they won't go away. Science **411**(6837) (2001) 522
2. Hirsch, J.: An index to quantify an individual's scientific research output. In: Proc. the National Academy of Sciences of the United States America. (2005) 102(46):16569
3. Beel, J., Gipp, B.: Google scholar's ranking algorithm: The impact of citation counts (an empirical study). In: Proc. RCIS. (2009) 439–446
4. Callaham, M., Wears, R., Weber, E.: Journal prestige, publication bias, and other characteristics associated with citation of published studies in peer-reviewed journals. Journal of the American Medical Association **287**(21) (2002) 2847–50
5. Kulkarni, A.V., Busse, J.W., Shams, I.: Characteristics associated with citation rate of the medical literature. PLOS one **2**(5) (2007)
6. Livne, A., Adar, E., Teevan, J., Dumais, S.: Predicting citation counts using text and graph mining. In: Proc. the iConference 2013 Workshop on Computational Scientometrics: Theory and Applications. (2013)
7. Bringmann, B., Berlingerio, M., Bonchi, F., Gionis, A.: Learning and predicting the evolution of social networks. IEEE Intelligent Systems **25** (2010) 26–35
8. Yan, R., Tang, J., Liu, X., Shan, D., Li, X.: Citation count prediction: learning to estimate future citations for literature. In: Proc. CIKM. (2011) 1247–1252
9. Shi, X., Leskovec, J., McFarland, D.A.: Citing for high impact. In: Proc. JCDL. (2010) 49–58
10. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science Magazine **286**(5439) (1999) 509–512
11. Munasinghe, L., Ichise, R.: Time score: A new feature for link prediction in social networks. IEICE Trans. **95-D**(3) (2012) 821–828
12. Mcgovern, A., Friedl, L., Hay, M., Gallagher, B., Fast, A., Neville, J., Jensen, D.: Exploiting relational structure to understand publication patterns in high-energy physics. SIGKDD Explorations **5** (2003) 2003
13. Yan, R., Huang, C., Tang, J., Zhang, Y., Li, X.: To better stand on the shoulder of giants. In: Proc. JCDL. (2012) 51–60
14. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management **45**(4) (2009) 427 – 437