Appl Intell (2016) 44:252–268
DOI 10.1007/s10489-015-0657-y

# Citation count prediction as a link prediction problem

**Nataliia Pobiedina · Ryutaro Ichise**

Published online: 3 April 2015

**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.

**Keywords** Citation count · Graph pattern mining · Feature selection

## 1 Introduction

Due to the drastic growth of the amount of scientific publications each year, it is a major challenge in academia to identify important 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 [2] and the $h$-index [3] measure the significance of 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 [4]. In other bibliography search systems the citation count is also considered as one of the major search criteria [5]. 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 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 [6–8]. 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 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 [9]. 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

This is an extended and enhanced version of the results published in [1].

N. Pobiedina (✉)
Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
e-mail: pobiedina@ec.tuwien.ac.at

R. Ichise
Principles of Informatics Research Division,
National Institute of Informatics, Tokyo, Japan
e-mail: ichise@nii.ac.jp

the corresponding paper increases. In the link prediction problem we aim at predicting the appearance of links in the network. However, we do not solve the link prediction problem since we need to estimate only the amount of new links for a specific node, but not with other nodes in the network in which it gets connected. Our 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 [10].

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 for solving the citation count prediction problem, 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. 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. For this purpose we take the definition of the citation count problem introduced by Yan et al. [11]. 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 $\bar{X}_{d,t} = (x_1, x_2, \ldots, x_n)$ 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 $\bar{X}_{d,t} = (x_1, x_2, \ldots, x_n)$ 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 propose 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 of a citation network is given in Fig. 1. Nodes represent scientific papers. A link from one node to another means that the first paper cites the latter. As we see, nodes and links have attributes which we will discuss later on. In this setting, the citation count of a paper is equal to the in-degree of the corresponding node. Its out-degree corresponds to the number of references present in the network and does not change over time. Since a 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. Generally, the link prediction problem answers the question whether there will be a link between two disconnected nodes in the network. In our case there are two major differences from the general link prediction problem. Firstly, new links are formed with the nodes which do not yet exist in the network (since the corresponding papers are not yet published). Therefore, we cannot use classical link prediction methods. Secondly, for a specific node we are not interested to

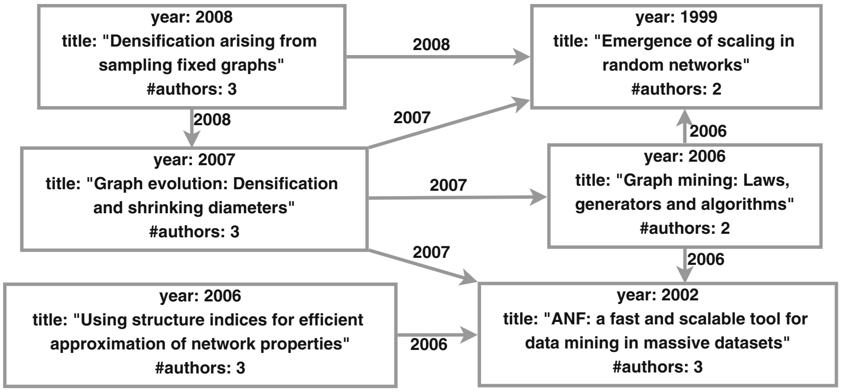

year: 2008
title: "Densification arising from sampling fixed graphs"
#authors: 3
2008
year: 1999
title: "Emergence of scaling in random networks"
#authors: 2
2008
2007
2006
year: 2007
title: "Graph evolution: Densification and shrinking diameters"
#authors: 3
2007
year: 2006
title: "Graph mining: Laws, generators and algorithms"
#authors: 2
2007
2006
year: 2006
title: "Using structure indices for efficient approximation of network properties"
#authors: 3
2006
year: 2002
title: "ANF: a fast and scalable tool for data mining in massive datasets"
#authors: 3

**Fig. 1** Example of a citation network

identify the nodes with which it will form links in the future, rather we want to estimate the amount of such nodes. Thus, we need to construct a suitable link prediction method to estimate future citation counts for scientific publications.

## 3 Related work

To solve the problem at hand, we build upon the works studying the citation count prediction and link prediction problems. The former works provide the baseline to compare our new feature, while the latter are used to construct it.

### 3.1 Citation count prediction

The task of predicting the citation counts for scientific publications as well as the general study of citing behavior have long attracted attention in the academic world. For example, Callaham et al. predicted citation counts for 204 publications from the 1991 emergency medicine specialty meeting [6]. They used decision trees and showed that the journal's impact factor is the most significant factor. Kulkarni et al. studied 328 medical articles published in 1999 and 2000 [7]. By using linear regression they achieved $R^2$ of 0.2 for predicting citation counts five years ahead.

Nowadays with the majority of digital libraries, such as ACM, IEEE, arXiv, etc., providing access online, it is possible to retrieve data about scientific publications automatically and to conduct studies of citation behavior on a large scale. Recent studies on citation count problem are performed on much larger datasets using more sophisticated predictive models and features of papers. Using a dataset of 30,199 papers from the arXiv, McGovern et al. suggested to predict non-self citations for a set of papers by performing a classification task of papers into quartiles $\{0-1, 2-5, 6-14, > 14\}$ according to their citations [12]. When constructing a training dataset, they considered characteristics of papers, the referenced papers, authors, number of pages and previous papers written by the authors. On several data samples the authors achieved an average classification accuracy of 44 % using relational probability trees. As an outcome of their study, several patterns are outlined according to which paper has an 85 % probability of obtaining more than 14 non-self citations. For example, one of the patterns is that the paper has more than 8 references. However, the authors do not provide detailed description of the features in their prediction model as well as their performance. The main focus of the paper is to uncover interesting patterns of citing and publishing behavior in the corresponding physics community.

Yan et al. introduce the citation counts prediction task [11]. They propose several factors which correlate with citation counts. These factors are based on content, author, venue and publication year of scientific publications, e.g., they use such features as author and venue ranks. To obtain author rank, the average citation counts in the previous years for every author is determined and a rank is assigned based on this number among the other authors. Venue rank is calculated the same way using the venue of the paper instead of the authors. In the succeeding work Yan et al. extend the list of factors, but they still show that the author rank is the most influential factor among those considered [13]. In these works the authors have also compared the performance of different predictive models with Classification and Regression Tree (CART) and Gaussian Process Regression (GPR) providing higher $R^2$ values compared to k-nearest neighbor (kNN), support vector regression (SVR) and linear regression (LR) models. The dataset which is used in their experiments is publicly available. Yan et al. do not use any features constructed from the citation network [11, 13].

Livne et al. extract a large and diverse dataset from Microsoft Academic Search [9]. This dataset contains 38 million papers which they group into seven major academic domains. For the citation count problem they construct features based on the authors, author institutions, venue, references and content of the papers. By using SVR they show that the most significant group of features is the one based on the citation network. However, the venue factor is more significant in two out of seven domains. The authors suggest that graph mining techniques might be better suited to capture the interest of research community. Similar results are obtained by Didegah and Thelwall when analyzing a set of papers published in nanoscience and nanotechnology journals from 2007 to 2009 [8]. They observe that the impact factor of the publication venue and of the references are the most significant determinants of the citation count. Summarizing the results of the recent work [8, 9, 13], we come to the conclusion that properties of papers which are related either to the paper co-authorship or citation networks are among the most significant factors for the paper citation prediction task. This observation indicates that formulating this problem as a link prediction problem in the citation network might be a promising approach. None of the above mentioned works considered a link prediction method to predict future citation counts. We will show that it is possible to solve the citation count prediction problem with a link prediction method and that by doing so we improve the performance.

### 3.2 Link prediction

A known model for the link prediction task in social networks is the preferential attachment model [14]. This model assumes that new nodes are more likely to form relationships with those nodes in the network which have already high degree. This behavior creates the so called "rich-get-richer" effect. Among the other methods for link prediction in social networks, there are Adamic-Adar [15], Liben-Nowell and Kleinberg [16]. Munasinghe and Ichise introduce a time-aware feature which considerably improves the performance of classical models for link prediction [17]. However, these methods can predict links only between nodes which already exist in the network. The citation count for a given paper increases if the corresponding node in the network gets an incoming relationship from a new node. By introducing graph evolution rules Bringmann et al. illustrate a way to predict links between an existing node and a new node in the network [10]. Their approach is based upon mining graph evolution rules in a network where links are stamped with the creation time and nodes may have up to one integer label. In our example network (see Fig. 1) we introduce link attributes which correspond to the years when the corresponding references appear. As a possible node label, Bringmann et al. take the degree of the node. In our work we consider number of authors and number of references as possible node labels. The obtained rules provide an opportunity to capture the temporal evolution of the network.

Another evidence that graph mining in the citation network might lead to good results for the citation count prediction can be found in [18]. Shi et al. investigate the patterns of citations by constructing citation projection graphs. The citation projection graph of a specific publication is a subgraph of the citation network which includes the references and citations among the papers which are referenced by this publication and also cite it. The authors observe that certain properties of the projection graphs are more common for papers with high impact. The impact of a publication is measured by its citation count normalized by the average citation count for all other papers published in the same year. The publications are classified into three classes according to their impact – high, medium and low. Though the authors apply a graph mining technique to study the citing behavior in three domains (natural sciences, social sciences and computer sciences), they do not use any link prediction method. The patterns which they uncover are not graph patterns in the classical understanding (their structure is not fixed and is more or less unique for each node), but rather these patterns refer to the structural properties of the citation projection graphs which differ for papers with high, medium and low impact. We, on the other hand, mine the local graph patterns which have specific properties in the whole network. These patterns have fixed structure, capture the temporal aspect of the citation counts and can be used for link prediction unlike the work of Shi et al. [18].

Thus, we suggest a new feature GERscore which is based upon frequent patterns of a specific form, i.e., graph evolution rules, mined from the citation network.

### 3.3 Evaluation

The estimation of future citations can be done with *classification* [12] or *regression* [9, 11, 13]. The classification task, where we predict intervals of citation counts, is in general easier [19, Ch. 6.7], and in many applications it is enough. For example, papers with more than 100 citations are referred to as influential in [13]. Shi et al. also study the properties of papers with regard to three classes of normalized citation counts [18]. Though both classification and regression tasks provide estimations for future citation counts in our case, there is a fundamental difference: the former estimates the probability of a paper to belong to a specific interval of citation counts, whereas the latter estimates the real citation count for this paper. Yan et al. apply the regression task to predict future citation counts and then use the results to construct a recommender system for scientific literature [13]. We also perform the regression task to predict the exact future citation counts. Furthermore, a dataset of publications from physics is used in [12], and from computer science in [11, 13]. There are also two different evaluation approaches. The first one is to test the performance for the freshly published papers [9, 12]. The second approach is to predict the citation counts for all available papers [11, 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. Furthermore, we include two performance measures for each of the learning tasks: average accuracy and precision for classification, $R^2$ and $RMSE$ for regression. The previous works report their results in terms of one performance measure. To sum it up, we extend the evaluation frameworks from the previous works, and we use the works of Mcgovern et al. and Yan et al. as our baseline [11, 12].

## 4 GERscore

Our methodology to tackle the stated problem consists of several steps which are depicted in Fig. 2. First, we construct a citation network from a publication database (block (1)), and by using additional constraints we mine the so-called graph evolution rules in this network (block (2)). Then we derive the GERscore for each paper using several calculation techniques (block (3)). We also calculate several state-of-the-art features (block (4)). All features are obtained using data from previous years. To estimate the performance of these features, we prepare training and testing datasets following two different scenarios (blocks (5)-(6)) and construct several predictive models for the classification and regression tasks (block (8)). In the rest of this

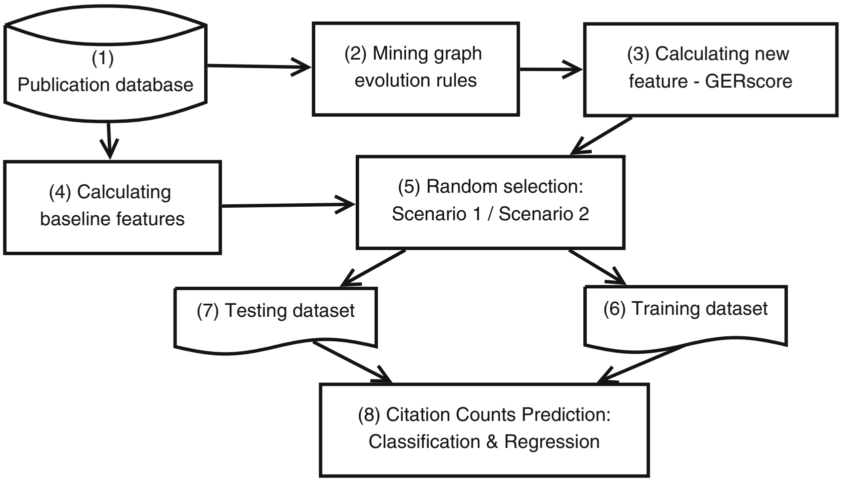
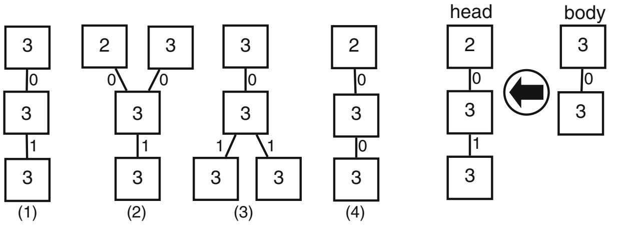

(1) Publication database
(2) Mining graph evolution rules
(3) Calculating new feature - GERscore
(4) Calculating baseline features
(5) Random selection: Scenario 1 / Scenario 2
(7) Testing dataset
(6) Training dataset
(8) Citation Counts Prediction: Classification & Regression

**Fig. 2** Process flow for predicting citation counts using a modified link prediction method

section, we explain the process of obtaining graph evolution rules and GERscores (blocks (2) and (3)). Blocks (1) and (4)-(8) are described in Section 5.

### 4.1 Mining graph evolution rules

To calculate the GERscore, we start with the discovery of rules which govern the temporal evolution of links and nodes. These rules are based on the frequent patterns of a special form, the so-called *relative time patterns*, and are introduced in [10]. Informally, a relative time pattern is a connected graph with one type of label over nodes (exactly one integer label or no label at all) and one integer label over links which represents relative time. Examples of relative time patterns are given in Fig. 3a. We can embed this pattern into a network if we can match each node of the pattern to some node in the network by preserving node labels and the structure of links between these nodes. Additionally, link labels in the pattern should correspond with a fixed gap to the labels of the matched links in the network. In Fig. 1 the network is directed, but to apply the notion of relative time patterns we ignore the direction of links. Besides, we may infer the direction of links: they point from a new node towards the older one. As link attributes, we have the year of link appearance in the network which corresponds to the year of publication of the citing paper. Nodes can have various attributes in the citation network, but we focus on three possible options: no label, the number of authors and the number of references of the corresponding paper. If we consider only the number of authors as a node label in the example citation network, then the pattern in Fig. 3a(1) can be embedded with the time gap 2007 or 2006 into the citation network in Fig. 1 while the pattern in Fig. 3a(3) cannot be embedded at all.



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

**Fig. 3** Examples of relative time patterns and graph evolution rules. Node labels correspond to the number of authors

A *graph evolution rule* is a pair of relative time patterns called *body* and *head* which is denoted as *head* $\Leftarrow$ *body* [10]. Informally, the *body* can be represented as the *head* without links which have the highest label. An example of a graph evolution rule is given in Fig. 3b. Do not get confused by the fact that body has less links 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 Fig. 3a(1)-(3) are also graph evolution rules. However, the relative time pattern in Fig. 3a(4) cannot be regarded as an evolution rule since both link labels equal zero.

To estimate frequency of the relative time pattern in the given network, we use *minimum image-based support*. It roughly equals to the minimum number of different nodes in this network to which one of the nodes in the pattern can be matched. The *support* of the evolution rule, $sup(r)$, is equal to the support of its head. The *confidence* of this rule, $conf(r)$, is equal to the ratio of the supports of the head and the body. The graph evolution rule from Fig. 3b has a minimum image based support 2 in the citation network from Fig. 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 $t + 1$. We assume that the likelihood of such event is proportional to the confidence of the rule. To determine all graph evolution rules in a network, we need to employ a graph pattern mining procedure.

Since graph pattern mining is computationally hard, two additional constraints are used to speed up the process of mining graph evolution rules in a network. We mine only those rules which have support not less than *minSupport*, and which have number of links not more than *maxSize*. The higher *minSupport* or the lower *maxSize*, the faster the graph pattern mining process will finish and the less patterns we will obtain. In case we have node labels in the network, we will also often arrive at better running times compared to the case when no labels are used over the nodes. Among the uncovered patterns, we identify graph evolution rules. In other words, we look for the patterns which have at least two different values on the links. Furthermore, we consider only those graph evolution rules where body and head differ in one link. In Fig. 3 all rules, except for a(3), correspond to this condition. Finally, we obtain a set $\mathcal{R}$ of graph evolution rules.

### 4.2 Calculating GERscore

To calculate the GERscore, we modify the procedure from [10]. We need to do this modification since the suggested approach in the previous work is a link prediction method, whereas we need to adapt it to our problem. The main task is to aggregate the information about the obtained graph evolution rules for a scientific publication. For each publication $n$ in the citation network we identify rules from the set $\mathcal{R}$ which can be applied to it. We say that a graph evolution rule can be applied to the node $n$ if its body can be embedded into the network so that one of the matched nodes is $n$. 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 the 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 [10]. 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:

1. $score_1(n, r) = conf(r)$,
2. $score_2(n, r) = sup(r) * conf(r)$,
3. and $score_3(n, r) = conf(r) * (size(r)/maxSize)$.

In the previous work the authors also experiment with different score calculation techniques, and they show that the best results for the link prediction problem are obtained by using $score_2(n, r)$ [10]. However, we will still run experiments with all three scores since we solve a different problem.

Finally, we use two functions to accumulate the final GERscore for the node $n$:

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

Here $score_i(n, r)$ corresponds to one of three possible score calculations. Therefore, we obtain six possible scores for our new feature. Throughout the paper, whenever we use the word "score", we always refer to one of the possible calculations for the GERscore.

Both aggregation techniques, maximum and summation, are used in [10]. The authors show that summation leads to better results. Though it might be intuitive to select the rule with the maximum score (which corresponds to the usage of the maximum as an aggregation function), but taking into consideration all rules, which can be applied to the node, might provide better estimation about the evolution. However, if it turns out that graph evolution rules with the highest support are the determinants of future citations, it has good implications in the sense that we can set the support threshold for the graph pattern mining procedure very high, thus reducing the running time.

High values of the 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 the situation when different rules correspond to the appearance of the same link. For example, in Fig. 3 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. With this regard, the constructed GERscore is similar to the network measures discussed in the work of Shi et al. [18]. However, our feature is based on a link prediction method which makes it distinct from the measures in this work.

## 5 Experiment

### 5.1 Experimental data

We use two real datasets to evaluate the 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 [11]. 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.

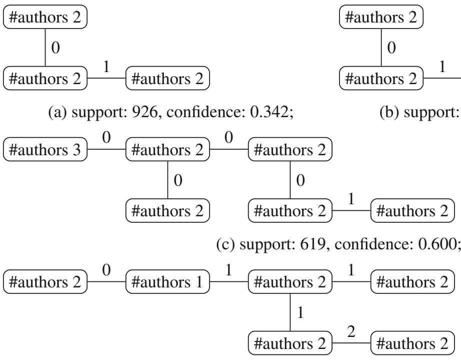
We construct several graphs from the described sub-networks which differ in node labels. It is good to have a label over nodes because this speeds up graph pattern mining. Since we are not sure which label setting is better, we use either the grouped number of references, or the grouped number of authors, or no label. The choice of the first two label settings is motivated by the uncovered citing patterns in [12]. Bringmann et al. show that graph evolution rules with no node labels lead to good results when solving the link prediction problem [10]. Also, they use the node degree as a label to obtain labeled graph evolution rules. In our case, it makes more sense to use the out-degree (or the number of references) since it does not change over time. Since it does not make sense to have continuous values as node labels (possible rules will be too rare and their interpretation will get harder), we group the values into categories. Bringmann et al. use also the grouped values of node's degree as labels [10].

In Table 1 we show the amount of graph evolution rules obtained with the help of the tool *GERM*[1] for different label settings for our two datasets. It is clear that the amount of evolution rules is considerably smaller than the amount of mined frequent patterns: not every relative time pattern is a graph evolution rule and we consider only graph evolution rules where body and head differ in one link. We obtain 230 evolution rules in the dataset HepTh, and 4, 108 in the dataset ArnetMiner for the unlabeled case (when no label over the nodes is used). 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. For both datasets we mine rules of the size up to five. However, support thresholds are set different since the datasets have considerably different amount of nodes. In our experiments we identified that this combination of input parameters is feasible enough to obtain results within one month for both datasets. The most crucial parameter is the size of the evolution rules, and it drastically affects the running times.

**Table 1** Results of graph pattern mining

| | | min | max | | # evolution |
|---|---|---|---|---|---|
| Network | Setting | support | size | # patterns | rules |
| Hep-Th | no label | 1,000 | 5 | 1,412 | 230 |
| upto | # authors | 500 | 5 | 7,441 | 886 |
| 1996 | # references | 500 | 5 | 6,565 | 426 |
| Arnet | no label | 5,000 | 5 | 6,742 | 4,108 |
| upto | # authors | 2,000 | 5 | 4,838 | 968 |
| 2000 | # references | 2,000 | 5 | 4,366 | 1,004 |

Figure 4 contains examples of graph evolution rules which we obtain for the citation network with grouped # authors as node labels. As we mention earlier, there are two main measures to estimate the frequency for each evolution rule: support and confidence. Thus, Figs. 4a and 4b contain the rules which have the highest support in HepTh and ArnetMiner correspondingly. In both cases the rules have the same structure and same node labels, but they have different supports and confidence measures: a lower support in HepTh than in ArnetMiner, but a higher confidence at the same time. However, the rules with the highest confidences are different for our datasets (see Figs. 4c, 4d). Though they both have five links, they differ in structure, node and link labels. Furthermore, the rule for ArnetMiner (Fig. 4d) has higher support as well as higher confidence. Such information already indicates that there are differences in the temporal evolution of the considered citation networks. Firstly, the amount of mined rules is considerably less for HepTh. Secondly, the differences in confidence measures will affect the probabilities of link formation.



(a) support: 926, confidence: 0.342;

(b) support: 4336, confidence: 0.192;

(c) support: 619, confidence: 0.600;

(d) support: 2026, confidence: 0.767;

**Fig. 4** Examples of graph evolution rules mined from HepTh and ArnetMiner datasets using number of authors as node label: (**a**) rule with highest support for HepTh, (**c**) rule with highest confidence for HepTh, (**b**) rule with highest support for ArnetMiner, (**d**) rule with highest confidence for ArnetMiner

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

### 5.2 Experimental setting

For a comprehensive study we perform two experiments. In the first experiment we aim at classifying papers into quartiles according to the future citation counts. We consider the following models for the classification task:

1. Multinomial Logistic Regression (mLR) which is a generalization of logistic regression for the case of more than two discrete outcomes,
2. Multi-class Support Vector Machines (mSVM) which construct a hyperplane or a set of hyperplanes to separate the training instance with the largest distance to the nearest data point of any class [20],
3. Conditional Inference Trees (CIT) which recursively perform univariate splits of the dependent variable and use a significance test to select variables [21].

We predict the real future citation counts for papers in the second experiment. Here, we consider such models for the regression task:

1. Linear Regression (LR) which approximates the dependent variable linearly based on the independent variables and intercept,
2. Support Vector Regression (SVR) which is an adaptation of SVM to perform the regression task [20],
3. Classification and Regression Tree (CART) which recursively perform univariate splits of the dependent variable and use the Gini coefficient to select variables [22].

We use the implementation of these models in R [23]. We look at a variety of models because they make different assumptions about the original data. Therefore, it is not guaranteed that features perform equally well in different models.

For each of the learning tasks (experiments) we consider two scenarios for evaluation which differ in the way we construct training and testing datasets. In Scenario 1 we train the models on the papers published one year before. A similar evaluation approach is undertaken in [9, 12, 17]. In Scenario 2 the training and testing datasets are constructed on the same pool of papers, e.g., as it is done in [11, 13].

In both scenarios we use a slightly modified $5 \times 2$ cross-validation [24], where each fold contains a stratified selection of scientific publications, i.e., $1,000$ instances for HepTh and $10,000$ for ArnetMiner. We chose such approach to remain in line with the previous works where 10,000 papers are chosen both into the training and testing datasets for ArnetMiner [11, 13]. We do not perform complete cross-validation procedure in Scenario 1: the training dataset contains papers from the year $t$ and the testing dataset has papers from the year $t+1$, so changing the training and testing datasets does not make sense here. We use $\Delta t = 1$ both for classification and regression tasks which corresponds to the prediction of citation counts for the next year. Additionally, we perform 5 year prediction in the case of the regression task to provide a more comprehensive comparison to the previous works.

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), *Maximum Past Influence for Venue* (MPIV) and *Recency* [11, 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 identify the author with the maximum citation counts in the previous years and put this total citation count as MPIA for the paper. TPIA is equal to the sum of citation counts for the previous papers of the authors. Venue Rank, TPIV and MPIV are calculated the same way using the venue of the paper. Recency is the absolute difference in years between the publication and current years, and it is used only in Scenario 2. Though recency is not a good feature, we want to verify its performance in the classification task. Livne et al. introduce several features based on the references of the paper in their work [9]. We do not use all the features since we do not aim at constructing a comprehensive model for citation count prediction, but we rather show the viability of our new feature for this task. Since we are not sure which feature performed the best in their work, we select three features. Following their work and preserving the naming convention of the earlier features, we calculate *References Rank*, *Total Past Influence for References* (TPIR) and *Maximum Past Influence for References* (MPIR). All these features are used as the baseline to compare our new feature.

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 the $GERscore_{1,3}^{(j)}$ and $GERscore_{2,3}^{(j)}$. Additionally, feature *GERscore* is the combination of our new 18 scores.

In Fig. 5 we illustrate the dependence between the features author rank, venue rank, recency and GERscore on one hand and average citation counts on the other hand for our two datasets. There is a similar dependence between author

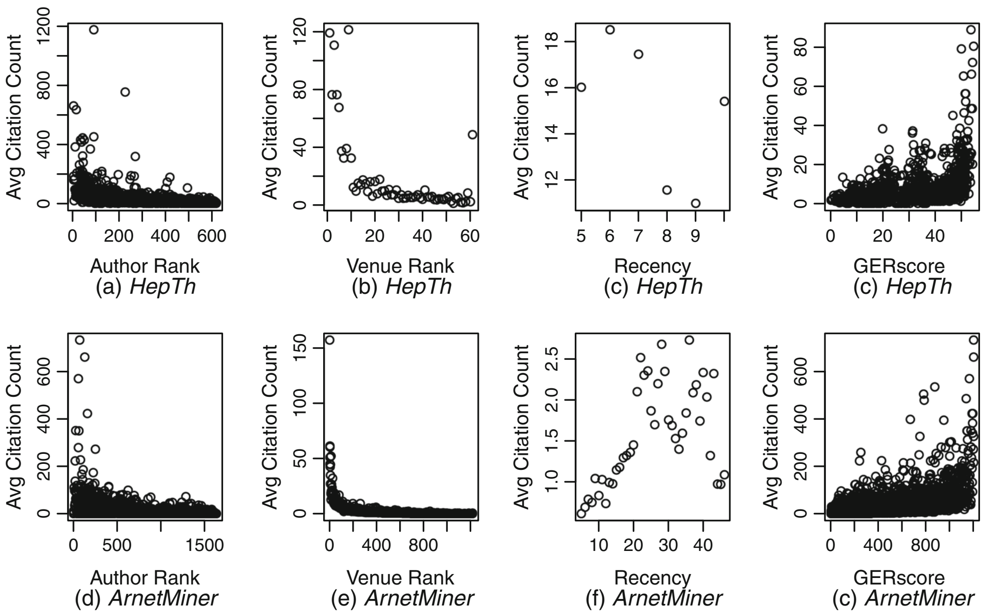


**Fig. 5** Correlation between average citation count and features: author rank, venue rank, recency and $GERscore^2_{1,3}$

ranks and average citation counts in HepTh and ArnetMiner datasets (see Fig. 5a and d). The dependence is clearer for the dataset ArnetMiner: papers with higher rank (which corresponds to the lower value of the variable author rank) have on average higher citation counts. However, there is even more obvious dependence between venue rank and average citation count for both datasets: papers with higher venue rank (which corresponds to the lower value of the variable venue rank) have considerably higher citation counts. The dependence between recency and average citation counts does not seem to be of a specific character (see Fig. 5c and f) which can cause its poor performance in the learning tasks. The last feature, $GERscore^2_{1,3}$, shows an inverse trend compared to Author Rank and Venue Rank: papers with a higher score have on average higher citation count. This observation supports the intuition of the score construction.

### 5.3 Classification task

In this experiment we compare how the calculated features perform with regard to classifying academic publications according to future citation counts. We assign class labels with intervals $1, 2-5, 6-14, > 14$ of citation counts. Such intervals are chosen in correspondence to the previous work [12]. There the classes were specified for HepTh dataset. It might occur that such distribution is not optimal for ArnetMiner. Shi et al. define classes dynamically for each dataset [18]. Such approach ensures that class distribution is the same across different datasets, but the disadvantage is that class boundaries are not fixed and may vary even over time. Therefore, we take the approach from the work of McGovern et al. [12]. In Table 2 we summarize the distribution of instances according to these classes for the training and testing datasets. As we see, it is the case that the class distribution is extremely skewed for ArnetMiner, especially in Scenario 1. We do not change the intervals because we want to have the same setting for both datasets. To construct training and testing datasets, we randomly select $1,000$ papers from Year 1996 into the training dataset, and from Year 1997 into the testing dataset in Scenario 1 for HepTh. $1,000$ instances are selected from Year 1997 into the training data in Scenario 2 for HepTh, and another $1,000$ instances are selected from the rest into the testing data. For ArnetMiner the procedure is the same, except that we select $10,000$ papers from years 2000 and 2001 correspondingly. In all cases we construct stratified folds and repeat the procedure 5 times.

We use *average accuracy* and *precision* to evaluate the performance in the classification task. If we put $tp_i$ true positives, $tn_i$ true negatives, $fp_i$ false positives, and $fn_i$ false negatives for class $i$, then average accuracy of the classifier is:

$$Accuracy = \frac{1}{l} * \sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fp_i + fn_i + tn_i}.$$

If class distribution is unbalanced, then precision is better suited for the evaluation [25]:

$$Precision = \frac{1}{l} * \sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}.$$

The performance of the features for the classification task is presented in Tables 3 and 4. We report average accuracy and precision for the new feature GERscore and the baseline features: Author Rank, MPIA, TPIA, Venue Rank, MPIV, TPIV, References Rank, MPIR, TPIR and Recency.

**Table 2** Distribution of instances according to classes (% Total)

| | HepTh | | | ArnetMiner | | |
|---|---|---|---|---|---|---|
| | Scenario 1 | | Scenario 2 | Scenario 1 | | Scenario 2 |
| Citation Class | 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 |

We mark in bold the features which lead to the highest performance measure in each column. In both scenarios we obtain that the highest accuracy and precision are for the GERscore.

However, due to a highly unbalanced distribution (Table 2), we observe only 1 % advantage in accuracy for ArnetMiner in Scenario 2 and almost none in Scenario 1. Moreover, we obtain that the average accuracy for ArnetMiner is very high and there is almost no difference in the performance measures for different features. If the classifier puts all observations into the first class in Scenario 1, we arrive at an average accuracy of around 97 % and a precision rate of about 24 %. Therefore, it is rather difficult to draw conclusions about the performance of the studied features on ArnetMiner dataset.

In the case of HepTh, the GERscore is at least 2 % better in accuracy than the other features in both scenarios. The increase is more obvious in terms of precision. Recency, as expected, is not good for predicting future citation counts.

The full model with all considered features as independent variables is indicated in the row *"All"* in Tables 5 and 6. To be more precise, this model contains all features listed in Table 3 in Scenario 1 and all features listed in Table 4 in Scenario 2. To verify how much the GERscore improves the performance, we construct a full model without the new feature denoted as *"-GERscore"* in the tables.

Statistical analysis shows that the GERscore provides a significant improvement to the full model. The average accuracy does not improve considerably if we add the GERscore: for HepTh around 2 % increase in Scenario 1 and less than 6% in Scenario 2; for ArnetMiner the increase is less than 2 %. But if we compare precision rates, then we have that the full model with the GERscore ("All") is more than 10 % better than without it ("-GERscore"). The best achieved accuracy for HepTh in Scenario 1 is 44 % in previous work [12]. The accuracy of our full model mLR is 81 %, but we cannot directly compare to the previous work since we take into account self-citations and the sampling technique is different. Still we see that constructed classification models provide accurate and rather precise results for both datasets. Suppose we are to identify relevant literature, then we could use classification as the first step to select potential

**Table 3** Accuracy (%) and Precision (%) for the different features for the Classification Task in Scenario 1

| | | Accuracy | | | | | | Precision | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| | Feature | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT |
| New | GERscore | **75.93** | **75.35** | **75.3** | **98.35** | **98.35** | **98.34** | **44.67** | **36.54** | **39.65** | **38.91** | **36.29** | **31.33** |
| | Author Rank | 73.86 | 73.95 | 73.75 | 98.31 | 98.34 | 98.34 | 32.2 | 33.64 | 28.49 | 24.17 | 24.17 | 24.17 |
| | MPIA | 73.76 | 73.39 | 72.99 | 98.31 | 98.34 | 98.34 | 33.46 | 27.9 | 31.59 | 24.17 | 24.17 | 24.17 |
| | TPIA | 73.91 | 73.86 | 73.09 | 98.31 | 98.34 | 98.34 | 34.84 | 32.66 | 31.94 | 24.17 | 25.84 | 24.17 |
| | Venue Rank | 71.37 | 71.74 | 71.9 | 98.31 | 98.34 | 98.34 | 29.8 | 26.98 | 22.97 | 24.17 | 24.17 | 24.17 |
| Baseline | MPIV | 66.44 | 69.83 | 63.24 | 98.31 | 98.34 | 98.34 | 25.23 | 12.52 | 20.72 | 24.17 | 24.17 | 24.17 |
| | TPIV | 70.87 | 69.82 | 67.93 | 98.31 | 98.34 | 98.34 | 27.89 | 22.39 | 22.48 | 24.17 | 24.17 | 24.17 |
| | References Rank | 70.8 | 69.14 | 71.72 | 98.31 | 98.34 | 98.34 | 22.03 | 23.66 | 24.68 | 24.17 | 24.17 | 24.17 |
| | MPIR | 72.9 | 71.79 | 71.76 | 98.31 | 98.34 | 98.34 | 29.3 | 26.3 | 25.8 | 28.19 | 24.17 | 24.17 |
| | TPIR | 73.69 | 73.31 | 72.25 | 98.31 | 98.34 | 98.34 | 28.69 | 28.38 | 32.51 | 28.35 | 29.17 | 24.17 |

**Table 4** Accuracy (%) and Precision (%) for the different features for the Classification Task in Scenario 2

| | | Accuracy | | | | | | Precision | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| | Feature | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT |
| New | GERscore | **76.95** | **75.07** | **76.35** | **95.81** | **95.51** | **95.54** | **51.32** | **47.16** | **52.27** | **67.91** | **66.58** | **66.13** |
| | Author Rank | 72.88 | 73.3 | 72.97 | 94.18 | 94.44 | 94.39 | 41.85 | 43.32 | 42.03 | 38.77 | 29.51 | 30.38 |
| | MPIA | 70.1 | 70.88 | 70.79 | 94.4 | 94.43 | 94.43 | 35.84 | 34.13 | 30.81 | 33.56 | 24.72 | 22.22 |
| | TPIA | 70.95 | 71.36 | 71.47 | 94.43 | 94.43 | 94.42 | 38.49 | 35.89 | 36.9 | 22.22 | 22.22 | 23.26 |
| | Venue Rank | 69.98 | 70.09 | 69.98 | 94.43 | 94.43 | 94.44 | 28.47 | 29.23 | 27.93 | 22.22 | 22.22 | 27.23 |
| Baseline | MPIV | 68.79 | 69.08 | 69.02 | 94.31 | 94.41 | 94.43 | 22.88 | 27.97 | 21.28 | 24.29 | 22.22 | 23.16 |
| | TPIV | 69.74 | 69.74 | 70.01 | 94.43 | 94.43 | 94.43 | 27.21 | 27.21 | 27 | 22.22 | 22.22 | 22.22 |
| | Recency | 67.77 | 67.36 | 67.37 | 94.39 | 94.3 | 94.4 | 16.78 | 16.79 | 13.27 | 22.17 | 22.17 | 22.16 |
| | References Rank | 69.01 | 68.91 | 69.04 | 94.36 | 94.43 | 94.43 | 26.94 | 24.91 | 31.28 | 27.47 | 22.22 | 22.22 |
| | MPIR | 69.06 | 69.08 | 69.23 | 94.38 | 94.41 | 94.43 | 25.21 | 27.97 | 26.5 | 28.74 | 28.51 | 22.22 |
| | TPIR | 69.4 | 69.83 | 69.73 | 94.4 | 94.43 | 94.43 | 28.55 | 30.51 | 30.74 | 34.15 | 31.48 | 23.17 |

papers and then perform regression on the selected papers to obtain a better ranking.

Overall, the results of the classification task indicate that the new feature is better than the baseline features and significantly improves the full model. Statistical significance of the improvement has been identified with analysis of variance (ANOVA) conducted for two models, "All" and "-GERscore", since these models are built on the same datasets. Surprisingly, mLR turns out to be the best performing method.

## 5.4 Regression task

In the second experiment we compare how the constructed features perform with regard to predicting real values of future citation counts for academic publications. To evaluate the performance in this task, we calculate the $R^2$ value as the *square of Pearson correlation coefficient* between the actual and predicted citation counts:

$$R^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}}{s_X}\right)\left(\frac{Y_i - \bar{Y}}{s_Y}\right),$$

where $n$ is the sample size, $X = (X_1, ..., X_n)$ correspond to the real citation counts, $\bar{X}$ is the mean of $X$, $s_X$ is the standard deviation of $X$, $Y = (Y_1, ..., Y_n)$ correspond to the predicted citation counts, $\bar{Y}$ is the mean of $Y$, and $s_Y$ is the standard deviation of $Y$. $R^2$ measures how good the constructed model is in relative terms. To measure the model's fitness also in absolute terms, we calculate the root of mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2}.$$

$R^2$ is a value between 0 and 1 while the value of $RMSE$ depends on the units of the response variable (in our case on the real citation counts). That is why we can compare $R^2$ for ArnetMiner and HepTh, but not $RMSE$: the ranges of the citation counts are quite different for them. $R^2$ measures how good the model is fit, hence, bigger values correspond to a better model. On the other hand, $RMSE$ estimates the non-fit of the model, therefore, it increases if the model becomes worse. Hence, we expect that the model with a higher $R^2$ would have a lower $RMSE$.

**Table 5** Accuracy (%) and Precision (%) for the full model with and without the GERscore for the Classification Task in Scenario 1

| | Accuracy | | | | | | Precision | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Model | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT |
| All | 77.08 | 74.85 | 75.8 | 98.37 | 98.36 | 98.36 | 50.05 | 47.62 | 40.83 | 41.84 | 35.31 | 31.7 |
| -GERscore | 74.69 | 70.44 | 73.53 | 98.31 | 98.35 | 98.33 | 43.74 | 36.24 | 37.19 | 24.17 | 30.01 | 26.79 |

**Table 6** Accuracy (%) and Precision (%) for the full model with and without the GERscore for the Classification Task in Scenario 2

| | Accuracy | | | | | | Precision | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Model | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT | mLR | mSVM | CIT |
| All | 80.92 | 80.62 | 79.28 | 96.22 | 95.91 | 96.01 | 60.42 | 59.76 | 56.68 | 69.17 | 67.56 | 67.52 |
| -GERscore | 74.71 | 74.89 | 74.41 | 94.6 | 94.71 | 94.71 | 48.13 | 48.05 | 46.25 | 49.93 | 58.48 | 50.61 |

We summarize the performance of various features for the regression task in Tables 7, 8 in terms of $R^2$ and in Tables 9, 10 in terms of $RMSE$. The results are indicated for 1-year ($\Delta t = 1$) and 5-year prediction ($\Delta t = 5$). Again, we mark with bold font those features which give the highest performance measure in each column. 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.

The GERscore leads to better $R^2$ values than the baseline features for ArnetMiner dataset. We showed that the GERscore is also the best performing feature for HepTh dataset in the classification task. However, it is no longer true in the regression task. An author-related feature, TPIA, yields the best $R^2$ results in Scenario 1 for HepTh in all cases (see Table 7). Now if we examine Scenario 2, the best models (LR and SVR) are still constructed with the author-related features (see Table 8). But if we use CART as a learning method, we arrive at a better $R^2$ with our new feature than with the others. Interestingly, the results of $RMSE$ are not always coherent with the results of $R^2$. Depending on a learning method, we arrive at a lower $RMSE$ using our new feature (e.g.,see Table 9). But it is still true that the lowest $RMSE$ is obtained for the author-related features using SVR.

Though author related features result in higher $R^2$ for HepTh, we obtain that the GERscore still brings additional value to the full models (Tables 11 and 12). The improvement is less obvious in Scenario 1, especially for the 5-year prediction for HepTh. To verify the statistical significance of the improvement, we conduct ANOVA for two models, "All" and "-GERscore". The analysis shows that the GERscore improves significantly the full model in all cases.

**Table 7** Performance measure $R^2$ for the different features for the Regression Task in Scenario 1

| | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Feature | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| GERscore$^{(1)}_{1,3}$ | 0.01 | 0.033 | 0.072 | 0.02 | 0.014 | 0.015 | 0.01 | 0.026 | 0.044 | 0.028 | 0.026 | 0.019 |
| GERscore$^{(2)}_{1,3}$ | 0.075 | 0.11 | 0.122 | 0.111 | 0.095 | 0.073 | 0.06 | 0.083 | 0.084 | 0.123 | 0.13 | 0.101 |
| GERscore$^{(3)}_{1,3}$ | 0.01 | 0.033 | 0.106 | 0.147 | 0.129 | 0.158 | 0.004 | 0.038 | 0.07 | 0.121 | 0.122 | 0.103 |
| GERscore$^{(1)}_{2,3}$ | 0.003 | 0.017 | 0.023 | 0.026 | 0.029 | 0.04 | 0.002 | 0.015 | 0.015 | 0.022 | 0.028 | 0.022 |
| GERscore$^{(2)}_{2,3}$ | 0.006 | 0.006 | NA | 0.108 | **0.231** | **0.232** | 0.004 | 0.004 | NA | 0.108 | 0.153 | **0.166** |
| GERscore$^{(3)}_{2,3}$ | 0.067 | 0.111 | 0.104 | 0.109 | 0.131 | 0.182 | 0.04 | 0.073 | 0.069 | 0.117 | 0.125 | 0.123 |
| GERscore | 0.161 | 0.152 | 0.142 | **0.221** | 0.215 | 0.19 | 0.111 | 0.107 | 0.093 | **0.165** | **0.155** | 0.124 |
| Author Rank | 0.224 | 0.089 | 0.155 | 0.005 | NA | 0.003 | 0.183 | 0.076 | 0.136 | 0.008 | 0.004 | 0.009 |
| MPIA | 0.258 | 0.188 | 0.176 | 0.003 | 0.003 | 0.005 | 0.236 | 0.166 | 0.176 | 0.005 | 0.007 | 0.012 |
| TPIA | **0.275** | **0.219** | **0.193** | 0.001 | 0.001 | 0.01 | **0.249** | **0.189** | **0.195** | 0.002 | 0.006 | 0.013 |
| Venue Rank | 0.041 | 0.052 | 0.055 | 0.018 | NA | 0.029 | 0.03 | 0.054 | 0.035 | 0.063 | 0.062 | 0.063 |
| MPIV | 0.04 | 0.011 | 0.031 | 0.002 | NA | 0.034 | 0.037 | 0.001 | 0.027 | 0.004 | 0.011 | 0.035 |
| TPIV | 0.048 | 0.01 | 0.032 | 0.002 | NA | 0.021 | 0.037 | 0.002 | 0.022 | 0.001 | 0.026 | 0.027 |
| References Rank | 0.092 | 0.012 | 0.035 | 0.004 | NA | 0.029 | 0.073 | 0.004 | 0.029 | 0.02 | 0.023 | 0.026 |
| MPIR | 0.092 | 0.083 | 0.074 | 0.005 | 0.001 | 0.037 | 0.071 | 0.044 | 0.053 | 0.004 | 0.031 | 0.035 |
| TPIR | 0.076 | 0.1 | 0.095 | 0.008 | 0.001 | 0.037 | 0.061 | 0.066 | 0.058 | 0.006 | 0.028 | 0.033 |

**Table 8** Performance measure $R^2$ for the different features for the Regression Task in Scenario 2

| | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Feature | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| $\text{GERscore}^{(1)}_{1,3}$ | 0.062 | 0.072 | 0.121 | 0.154 | 0.172 | 0.173 | 0.055 | 0.065 | 0.099 | 0.236 | 0.224 | 0.275 |
| $\text{GERscore}^{(2)}_{1,3}$ | 0.126 | 0.226 | **0.247** | 0.382 | **0.374** | 0.428 | 0.106 | 0.17 | **0.176** | 0.447 | **0.448** | 0.472 |
| $\text{GERscore}^{(3)}_{1,3}$ | 0.019 | 0.017 | 0.009 | 0.186 | 0.193 | 0.228 | 0.013 | 0.012 | 0.011 | 0.226 | 0.243 | 0.258 |
| $\text{GERscore}^{(1)}_{2,3}$ | 0.027 | 0.036 | 0.049 | 0.066 | 0.092 | 0.101 | 0.024 | 0.033 | 0.045 | 0.095 | 0.128 | 0.132 |
| $\text{GERscore}^{(2)}_{2,3}$ | 0.033 | 0.059 | 0.058 | 0.086 | 0.139 | 0.186 | 0.031 | 0.058 | 0.058 | 0.114 | 0.171 | 0.215 |
| $\text{GERscore}^{(3)}_{2,3}$ | 0.006 | 0.015 | NA | 0.093 | 0.102 | 0.114 | 0.005 | 0.011 | 0.013 | 0.128 | 0.138 | 0.154 |
| GERscore | 0.188 | 0.201 | 0.217 | **0.445** | 0.298 | **0.444** | 0.14 | 0.161 | 0.157 | **0.543** | 0.355 | **0.483** |
| Author Rank | 0.201 | **0.264** | 0.179 | 0.118 | 0.121 | 0.17 | 0.164 | 0.21 | 0.17 | 0.133 | 0.181 | 0.159 |
| MPIA | 0.235 | 0.235 | 0.21 | 0.061 | 0.055 | 0.067 | 0.19 | 0.207 | 0.147 | 0.07 | 0.025 | 0.054 |
| TPIA | **0.303** | 0.239 | 0.236 | 0.002 | 0.067 | 0.056 | **0.25** | **0.216** | 0.167 | 0.004 | 0.062 | 0.071 |
| Venue Rank | 0.056 | 0.065 | 0.051 | 0.035 | 0.057 | 0.053 | 0.043 | 0.063 | 0.049 | 0.04 | 0.048 | 0.05 |
| MPIV | 0.046 | 0.056 | 0.044 | 0.031 | 0.027 | 0.035 | 0.032 | 0.043 | 0.038 | 0.025 | 0.009 | 0.039 |
| TPIV | 0.047 | 0.053 | 0.038 | 0.02 | 0.023 | 0.035 | 0.035 | 0.04 | 0.036 | 0.017 | 0.006 | 0.035 |
| Recency | 0.002 | 0.002 | NA | 0.006 | NA | NA | 0.002 | 0.003 | 0.004 | 0.002 | NA | NA |
| References Rank | 0.096 | 0.083 | 0.059 | 0.026 | 0.016 | 0.023 | 0.094 | 0.086 | 0.034 | 0.025 | 0.009 | 0.017 |
| MPIR | 0.094 | 0.088 | 0.086 | 0.018 | 0.022 | 0.02 | 0.098 | 0.092 | 0.065 | 0.018 | 0.014 | 0.019 |
| TPIR | 0.117 | 0.096 | 0.065 | 0.026 | 0.022 | 0.026 | 0.109 | 0.096 | 0.057 | 0.026 | 0.012 | 0.018 |

**Table 9** Performance measure $RMSE$ for the different features for the Regression Task in Scenario 1

| | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Feature | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| $\text{GERscore}^{(1)}_{1,3}$ | 16.068 | 16.294 | 15.723 | 0.668 | 0.668 | 0.672 | 39.722 | 39.891 | 39.814 | 1.959 | 1.966 | 1.966 |
| $\text{GERscore}^{(2)}_{1,3}$ | 15.495 | 15.864 | **15.192** | 0.633 | 0.639 | 0.702 | **38.85** | 39.301 | **38.826** | 1.848 | 1.834 | 2.034 |
| $\text{GERscore}^{(3)}_{1,3}$ | 16.002 | 16.566 | 15.552 | **0.624** | 0.628 | 0.669 | 39.646 | 40.197 | 39.481 | 1.876 | 1.866 | 1.982 |
| $\text{GERscore}^{(1)}_{2,3}$ | 16.062 | 16.624 | 15.915 | 0.677 | 0.662 | 0.663 | 39.702 | 40.383 | 39.554 | 1.97 | 1.967 | 1.964 |
| $\text{GERscore}^{(2)}_{2,3}$ | 16.015 | 16.668 | 16.057 | 0.645 | 0.597 | **0.583** | 39.533 | 40.515 | 39.589 | **1.834** | 1.824 | **1.811** |
| $\text{GERscore}^{(3)}_{2,3}$ | 15.729 | 15.739 | 15.566 | 0.625 | 0.634 | 0.631 | 39.58 | 39.063 | 39.504 | 1.844 | 1.869 | 1.881 |
| GERscore | **15.317** | 15.004 | 15.689 | 0.625 | **0.593** | 0.651 | 39.516 | 37.97 | 40.22 | 1.88 | **1.811** | 1.995 |
| Author Rank | 16.742 | 15.522 | 16.261 | 0.674 | 0.672 | 0.689 | 41.791 | 38.494 | 44.437 | 1.973 | 2.004 | 1.986 |
| MPIA | 17.892 | 14.683 | 16.301 | 0.677 | 0.671 | 0.68 | 48.113 | **36.809** | 43.368 | 1.979 | 1.999 | 1.994 |
| TPIA | 20.059 | **14.636** | 18.913 | 0.673 | 0.672 | 0.687 | 53.046 | 37.625 | 49.093 | 1.966 | 1.999 | 2.003 |
| Venue Rank | 15.941 | 16.297 | 15.753 | 0.667 | 0.672 | 0.672 | 39.768 | 39.711 | 39.478 | 1.875 | 1.949 | 1.951 |
| MPIV | 18.887 | 16.662 | 16.189 | 0.673 | 0.672 | 0.672 | 47.044 | 40.418 | 40.378 | 1.988 | 1.991 | 1.946 |
| TPIV | 18.21 | 16.369 | 16.355 | 0.673 | 0.672 | 0.674 | 45.499 | 40.167 | 41.119 | 1.983 | 1.984 | 1.981 |
| References Rank | 17.847 | 16.344 | 16.593 | 0.674 | 0.671 | 0.666 | 42.742 | 40.046 | 41.651 | 1.969 | 2.053 | 1.964 |
| MPIR | 19.653 | 15.653 | 16.108 | 0.672 | 0.672 | 0.667 | 46.93 | 39.345 | 40.496 | 1.977 | 1.955 | 1.952 |
| TPIR | 22.359 | 15.462 | 17.775 | 0.67 | 0.67 | 0.667 | 53.624 | 38.715 | 45.761 | 1.98 | 1.958 | 1.977 |

**Table 10** Performance measure $RMSE$ for the different features for the Regression Task in Scenario 2

| | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| Feature | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| GERscore$^{(1)}_{1,3}$ | 25.644 | 26.194 | 25.006 | 5.202 | 5.318 | 5.168 | 47.473 | 48.119 | 46.625 | 5.542 | 5.823 | 5.41 |
| GERscore$^{(2)}_{1,3}$ | 24.962 | 24.631 | **23.439** | 4.43 | **4.669** | 4.248 | 46.371 | 46.453 | **44.973** | 4.725 | **4.997** | 4.605 |
| GERscore$^{(3)}_{1,3}$ | 26.286 | 26.806 | 26.678 | 5.089 | 5.281 | 4.931 | 48.421 | 49.107 | 49.104 | 5.542 | 5.799 | 5.439 |
| GERscore$^{(1)}_{2,3}$ | 26.113 | 26.641 | 25.84 | 5.431 | 5.485 | 5.334 | 48.012 | 48.804 | 47.636 | 5.998 | 6.075 | 5.883 |
| GERscore$^{(2)}_{2,3}$ | 26.055 | 26.491 | 25.751 | 5.353 | 5.388 | 5.067 | 47.841 | 48.495 | 47.328 | 5.927 | 6.018 | 5.602 |
| GERscore$^{(3)}_{2,3}$ | 26.383 | 26.854 | 26.422 | 5.345 | 5.444 | 5.283 | 48.417 | 49.152 | 48.519 | 5.9 | 6.048 | 5.812 |
| GERscore | 24.263 | 24.651 | 24.346 | **4.198** | 4.935 | **4.201** | 45.867 | 46.296 | 46.237 | **4.287** | 5.386 | **4.571** |
| Author Rank | 24.032 | **23.994** | 24.783 | 5.26 | 5.403 | 5.119 | 45.015 | 45.383 | 45.999 | 5.863 | 5.924 | 5.789 |
| MPIA | 23.462 | 24.344 | 24.085 | 5.43 | 5.589 | 5.42 | 44.636 | 45.686 | 46.001 | 6.066 | 6.245 | 6.133 |
| TPIA | **22.512** | 24.086 | 23.785 | 5.585 | 5.561 | 5.472 | **44.45** | **44.874** | 46.847 | 6.261 | 6.193 | 6.049 |
| Venue Rank | 25.818 | 26.378 | 25.892 | 5.494 | 5.557 | 5.466 | 47.725 | 48.342 | 47.454 | 6.154 | 6.22 | 6.131 |
| MPIV | 25.923 | 26.504 | 25.974 | 5.522 | 5.608 | 5.514 | 47.873 | 48.623 | 47.703 | 6.2 | 6.263 | 6.167 |
| TPIV | 25.922 | 26.495 | 26.051 | 5.544 | 5.613 | 5.513 | 47.824 | 48.642 | 47.72 | 6.224 | 6.267 | 6.175 |
| Recency | 26.441 | 27.024 | 26.422 | 5.574 | 5.648 | 5.589 | 48.428 | 49.332 | 48.438 | 6.266 | 6.274 | 6.274 |
| References Rank | 25.33 | 26.16 | 26.269 | 5.533 | 5.619 | 5.549 | 46.593 | 48.006 | 49.467 | 6.202 | 6.259 | 6.245 |
| MPIR | 25.332 | 26.07 | 25.467 | 5.547 | 5.606 | 5.553 | 46.568 | 47.813 | 47.469 | 6.223 | 6.25 | 6.228 |
| TPIR | 25.055 | 25.792 | 27.29 | 5.528 | 5.606 | 5.542 | 46.162 | 47.371 | 47.961 | 6.203 | 6.251 | 6.223 |

We also observe that it becomes harder to predict citation counts over longer periods. The only exception is ArnetMiner dataset in Scenario 2. In the previous work the authors also showed better performance over longer time periods in this setting [11]. Our explanation is that it happens due to the specifics of the considered datasets and evaluation approaches. The average citation count for ArnetMiner dataset remains around 1 throughout years, while it gradually grows till 12 for HepTh. Thus, the performance of 5-year prediction for ArnetMiner does not drop so much as for HepTh. The same observation holds for the dataset in [11]. If we study the results in terms of $RMSE$, we observe that 5-year prediction is harder: the values more than double for both datasets in Scenario 1 (see Table 11), but the relative drop in Scenario 2 is not so high, especially for ArnetMiner. Again, the reason is the difference between the average citation counts for these datasets (HepTh has it higher). This also explains why the values of $RMSE$ are higher for HepTh.

**Table 11** Performance measures ($R^2$ and $RMSE$) for the full model with and without the GERscore for the Regression Task in Scenario 1

| | | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| | Model | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| $R^2$ | All | 0.3 | 0.19 | 0.218 | 0.224 | 0.197 | 0.193 | 0.273 | 0.147 | 0.154 | 0.173 | 0.162 | 0.136 |
| | -GERscore | 0.288 | 0.115 | 0.215 | 0.022 | 0.011 | 0.026 | 0.269 | 0.085 | 0.141 | 0.063 | 0.037 | 0.057 |
| RMSE | All | 21.984 | 14.586 | 21.584 | 0.626 | 0.599 | 0.644 | 57.694 | 37.471 | 58.282 | 1.872 | 1.796 | 1.954 |
| | -GERscore | 24.156 | 16.246 | 20.899 | 0.667 | 0.668 | 0.673 | 60.632 | 41.296 | 59.817 | 1.892 | 1.956 | 1.941 |

Another interesting observation is that $R^2$ is higher in Scenario 2 compared to Scenario 1. We have expected that predicting citation counts for freshly published papers is more difficult since not much is known about them. However, if we compare $RMSE$ across scenarios, we notice that the values are lower in Scenario 1. The explanation is quite simple: average citation counts in Scenario 2 are higher than in Scenario 1, and this leads to a higher error. So, comparing our two scenarios in terms of $RMSE$ does not make sense.

### 5.5 Discussion

Overall our new feature GERscore significantly improves citation count prediction. The statistical significance of the improvement has been verified for the full models using ANOVA test. When classifying the future citations, the 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 the previous work the authors argue that such disagreement arises due to the nature of the relevant scientific domains [9]. 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 previous work [11]. However, LR provides better results in Scenario 1. In general, the performance in Scenario 1 is not as good as in Scenario 2. 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. Surprisingly, CART does not yield the best results for the 5-year prediction which contradicts to the previous work [11]. However, if we leave out the GERscore from the full model ("-GERscore" in Table 12), firstly, we have that non-linear models, namely SVR and CART, perform better. Secondly, CART yields the best result for ArnetMiner for predicting citation counts over 5 years. For the full model it is the case that the performance drops for HepTh and increases for ArnetMiner over the longer time period. We face again the challenge that more datasets are required to determine whether the nature of the scientific domain influences these results.

Out of all scores which constitute the GERscore, the best results are gained for the scores calculated from the unlabeled graph evolution rules (see Tables 8 and 10). When aggregating separate scores, summation is a better choice compared to maximum. This is 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 $GERscore_{1,i}^{(2)}$ 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 [11, 13]. However, we show that the 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 ten 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) the 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. Additionally, we predict the actual citation counts in 5 years. We observe that it becomes harder to predict citation counts over longer periods. Our results also indicate that the performance of the model does not always improve if we include more features. We have not included all features from the previous works in our evaluation framework, e.g., content related features [9, 11, 13] or network related features [9, 18]. Thus, an important aspect to investigate is the performance of various features on different datasets and their optimal combination where dimension reduction methods

**Table 12** Performance measures ($R^2$ and $RMSE$) for the full model with and without the GERscore for the Regression Task in Scenario 2

| | | $\Delta t = 1$ | | | | | | $\Delta t = 5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HepTh | | | ArnetMiner | | | HepTh | | | ArnetMiner | | |
| | Model | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART | LR | SVR | CART |
| $R^2$ | All | 0.346 | 0.28 | 0.366 | 0.27 | 0.452 | 0.474 | 0.269 | 0.285 | 0.226 | 0.562 | 0.312 | 0.527 |
| | -GERscore | 0.296 | 0.221 | 0.324 | 0.129 | 0.213 | 0.149 | 0.243 | 0.272 | 0.154 | 0.162 | 0.141 | 0.175 |
| RMSE | All | 22.483 | 23.438 | 21.674 | 4.977 | 4.166 | 4.089 | 44.485 | 43.258 | 45.939 | 4.198 | 5.478 | 4.415 |
| | -GERscore | 23.112 | 24.689 | 22.275 | 5.325 | 4.998 | 5.172 | 44.686 | 43.755 | 47.81 | 5.767 | 5.933 | 5.778 |

might be of help. Ultimately, we want to include our findings into a recommender system for academic publications.

Our future work includes thorough investigation how the mined evolution rules influence the predictive power of the GERscore. Here we want to investigate in several directions. 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. Pobiedina N, Ichise R (2014) Predicting citation counts for academic literature using graph pattern mining. In: Proceeding IEA/AIE, pp 109–119
2. Garfield E (2001) Impact factors, and why they won't go away. Science 411(6837):522
3. Hirsch J (2005) An index to quantify an individual's scientific research output. Proc the National Academy of Sciences of the United States America 102(46):16569
4. Beel J, Gipp B (2009) Google scholar's ranking algorithm: The impact of citation counts (an empirical study). In: Proceeding RCIS, pp 439–446
5. Bethard S, Jurafsky D (2010) Who should I cite: learning literature search models from citation behavior. In: Proceeding CIKM, pp 609–618
6. Callaham M, Wears R, Weber E (2002) Journal prestige, publication bias, and other characteristics associated with citation of published studies in peer-reviewed journals. J. Am. Med. Assoc. 287(21):2847–50
7. Kulkarni AV, Busse JW, Shams I (2007) Characteristics associated with citation rate of the medical literature. PLOS One 2(5)
8. Didegah F, Thelwall M (2013) Determinants of research citation impact in nanoscience and nanotechnology. JASIST (JASIS) 64(5):1055–1064
9. Livne A, Adar E, Teevan J, Dumais S (2013) Predicting citation counts using text and graph mining. In: Proceeding the iConference 2013 Workshop on Computational Scientometrics: Theory and Applications
10. Bringmann B, Berlingerio M, Bonchi F, Gionis A (2010) Learning and predicting the evolution of social networks. IEEE Intell Syst 25:26–35
11. Yan R, Tang J, Liu X, Shan D, Li X (2011) Citation count prediction: learning to estimate future citations for literature. In: Proceeding CIKM, pp 1247–1252
12. Mcgovern A, Friedl L, Hay M, Gallagher B, Fast A, Neville J, Jensen D (2003) Exploiting relational structure to understand publication patterns in high-energy physics. SIGKDD Explorations 5:2003
13. Yan R, Huang C, Tang J, Zhang Y, Li X (2012) To better stand on the shoulder of giants. In: Proceeding JCDL, pp 51–60
14. Barabasi AL, Albert R (1999) Emergence of scaling in random networks. Sci Mag 286(5439):509–512
15. Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Networks 25(3):211–230
16. Liben-Nowell D (2007) The link-prediction problem for social networks. JASIST 58(7):1019–1031
17. Munasinghe L, Ichise R (2012) Time score: A new feature for link prediction in social networks. IEICE Trans 95-D(3):821–828
18. Shi X, Leskovec J, McFarland DA (2010) Citing for high impact. In: Proceeding JCDL, pp 49–58
19. Devroye L, Gyrfi L, Lugosi G (1996) A Probabilistic Theory of Pattern Recognition. Springer
20. Chang CC, Lin CJ (2011) Libsvm: A library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
21. Hothorn T, Hornik K, Zeileis A (2006) Unbiased recursive partitioning: A conditional inference framework. J Comp Graph Stat 15(3):651–674
22. Breiman L, Friedman J, Stone CJ, Olshen R (1984) Classification and Regression Trees. Chapman and Hall/CRC
23. The R project for statistical computing http://www.r-project.org/ (January 2013)
24. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10(7):1895–1923
25. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437

**Nataliia Pobiedina** is a PhD student at the Vienna PhD School of Informatics, a special program at Vienna University of Technology, Austria. In 2013 she won a scholarship for a three month internship at the National Institute of Informatics in the group of Ryutaro Ichise, Tokyo, Japan. Her research interests include machine learning, data mining and network analysis.

**Ryutaro Ichise** received his Ph.D. degree in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2000. From 2001 to 2002, he was a visiting scholar at Stanford University. He is currently an associate professor in the Principles of Informatics Research Division at the National Institute of Informatics in Japan. His research interests include machine learning, semantic web, and data mining.