SCHOLAR TREND LEARNER:

PREDICTING SCHOLAR POPULARITY AS EARLY AND ACCURATE AS POSSIBLE

MASOUMEH NEZHADBIGLARI

SCHOLAR TREND LEARNER:

PREDICTING SCHOLAR POPULARITY AS EARLY AND ACCURATE AS POSSIBLE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: MARCOS ANDRÉ GONÇALVES
COORIENTADORA: JUSSARA MARQUES DE ALMEIDA GONÇALVES

Belo Horizonte

Outubro de 2016

MASOUMEH NEZHADBIGLARI

SCHOLAR TREND LEARNER: PREDICTING SCHOLAR POPULARITY AS EARLY AND ACCURATE AS POSSIBLE

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais– Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

ADVISOR: MARCOS ANDRÉ GONÇALVES
CO-ADVISOR: JUSSARA MARQUES DE ALMEIDA GONÇALVES

Belo Horizonte
October 2016

© 2016, Masoumeh Nezhadbiglari.
Todos os direitos reservados.

Nezhadbiglari, Masoumeh

N575s Scholar trend learner: predicting scholar popularity as early and accurate as possible / Masoumeh Nezhadbiglari. — Belo Horizonte, 2016
xxii, 44 f. : il. ; 29cm

Dissertações (mestrado) — Universidade Federal de Minas Gerais–Departamento de Ciência da Computação

Orientador: Marcos André Gonçalves
Coorientadora: Jussara Marques de Almeida Gonçalves

1. Computação – Teses. 2. Mineração de dados. 3. Redação acadêmica – Bibliometria.. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

SCHOLARTRENDLEARNER: PREDICTING SCHOLAR POPULARITY AS EARLY AND ACCURATE AS POSSIBLE

MASOUMEH NEZHADBIGLARI

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. MARCOS ANDRÉ GONÇALVES - Orientador
Departamento de Ciência da Computação - UFMG

PROFA. JUSSARA MARQUES DE ALMEIDA GONÇALVES - Coorientadora
Departamento de Ciência da Computação - UFMG

PROF. ALBERTO HENRIQUE FRADE LAENDER
Departamento de Ciência da Computação - UFMG

PROF. FABRÍCIO BENEVENUTO DE SOUZA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 11 de outubro de 2016.

" This dissertation is dedicated to my parents, Mohammad and Maryam, along with my lovely husband, Sajad for their endless love, support and encouragement. Furthermore, I want to thank my sister (Zeinab) for her support and love that helped me to be motivated throughout my study."

Acknowledgments

I would like to give my sincere thanks and appreciation to my adviser, Prof. Marcos André Gonçalves, and my co-adviser, Prof. Jussara M. Almeida for all their help and supports. Also, I would like to declare my special gratitude to my husband Sajad, who has been accompanying me in any circumstance during my study, and to my dear parents, who have always supported and encouraged me to progress.

"Truth and lie are opposite things."
(Unknown)

Abstract

Prediction of scholar popularity has become an important research topic for a number of reasons. In this dissertation, we tackle the problem of predicting the popularity *trend* of scholars by concentrating on making predictions both as *earlier* and *accurate* as possible. In order to perform the prediction task, we first extract the popularity trends of scholars from a training set. To that end, we apply a time series clustering algorithm called K-Spectral Clustering (K-SC) to identify the popularity trends as cluster centroids. We then predict trends for scholars in a test set by solving a classification problem. Specifically, we first compute a set of measures for individual scholars based on the distance between earlier points in their particular popularity curve and the identified centroids. We then combine those distance measures with a set of academic features (e.g., number of publications, number of venues, etc) collected during the same monitoring period, and use them as input to a classification method. One aspect that distinguishes our method from other approaches is that the monitoring period, during which we gather information on each scholar popularity and academic features, is determined on a per scholar basis, as part of our approach. Using total citation count as measure of scientific popularity, we evaluate our solution on the popularity time series of more than 500,000 Computer Science scholars, gathered from Microsoft Azure Marketplace[1]. The experimental results show that our prediction method outperforms other alternative prediction methods. We also show how to apply our method jointly with regression models to improve the prediction of scholar popularity values (e.g., number of citations) at a future time.

Palavras-chave: Trend Cassification, Prediction, Scholar's Popularity.

[1] https://datamarket.azure.com/dataset/mrc/microsoftacademic

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

We have witnessed a steep increase in the volume of scholarly publications, such as scientific articles, conference papers, books and other types of scientific communications in basically all research fields. Such phenomenon is followed by an increasing competition among scientists, as the amount of financial and human resources to produce high quality research is limited. Accordingly, funding agencies and academic departments have relied on some measures of academic success in order to try to better distribute such resources among scholars. One such measure, which aims at assessing the impact of a scholar's research is " popularity", usually quantified by metrics such as overall number of citations [25, 30, 59] or the well-known h-index [20].

In this context, a natural question that arises in many contexts is "How *popular* will a scholar be in the near future or in the long run?" Answering such question is valuable for several goals. From an organization's perspective, knowing the scientific potential of a scholar can be very helpful in decisions for hiring faculty members or for guiding funding agencies in their decision processes. Moreover, academic search engines such as Google Scholar and Microsoft Academic Search or scientific recommender systems (e.g.,[40]) can benefit from such information as a feature for improving their rankings. More importantly, answers for such question, and mainly the factors that influence such answers, can help an individual scholar to better manage her scientific career.

Traditionally, the *total number of citations* has been widely used as a measure of *popularity* for both publications and scientific researchers [25, 30, 50, 59]. Indeed, it has already been argued that citation counts are better indicators of the scientific contribution of researchers than impact factors such as the h-index [25]. Accordingly, we focus on this metric in this dissertation [37].

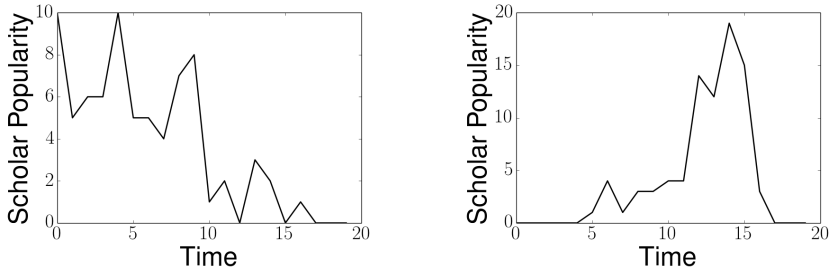
Some prior studies on scholar popularity focus on studying the impact of academic features on popularity [25, 30, 50]. Others aim at developing popularity prediction methods [10, 20]. Among the latter, most attempt to predict the popularity of individual publications. Some studies, for instance, predict the future citation counts of articles based on learning models [13, 60, 61]. Despite such efforts, we are aware of only two previous studies on predicting the popularity of scholars. Acuna et al., [1] use regression models to predict the h-index of scholars at a future time. In [37], the author aims at predicting the scholars' scientific impact in terms of future number of citations and found that the current number of citations is the most reliable feature for such a prediction.

Complementing prior work, we here are interested in predicting the *trend* that the popularity of a scholar will follow in the future (or her popularity curve), as opposed to predicting popularity values at specific future times. Prediction of popularity trends is valuable as it may bring insights into the evolution of the research impact of a scholar. It may also contribute to improving the effectiveness of models to predict future popularity values [47, 63]. Moreover, producing prediction models of scientific impact can also induce interesting services for a digital library, such as a career profile prediction service and expert recommendation.

Another significant contribution of our work is that we aim at solving a trade-off that is inherent to any prediction task. On one hand, we want to make the prediction as early as possible. The sooner we make a prediction, the earlier corrective measures (if any) can be applied[1]. On the other hand, we want to make predictions as accurate as possible. These two goals are often conflicting as one needs to monitor the scholar features for longer periods to guarantee more accurate predictions. So, determining the earliest point in time when prediction can be made with reasonable accuracy is an inherent challenge of the early popularity prediction problem.

Unlike prior work, we here solve this trade-off on a per scholar basis (i.e., the monitoring period is different for different scholars), recognizing that different scholars may present quite different popularity evolution curves. This is better illustrated in Figure 1.1, which shows the popularity curves of two scholars. Scholar A receives most of her citations at the beginning of her academic career, whereas scholar B becomes more popular later on in her profession. Thus, if we monitor both scholars during the same period (e.g., 8 years) to make the prediction, a large portion of the popularity of scholar A would have already passed. Perhaps more accurate and useful predictions could have been made much earlier in scholar A's lifespan. In contrast, predictions before the first 5 years most certainly would not capture the correct trend of scholar B. Thus, the aforementioned trade-off must be solved separately for

[1]We acknowledge that there are also some important boundary events in a scholar career's in which such predictions are useful such as job applications, midterm review before going up for tenure and tenure decision. In any case, our method can be easily adapted to predict in specific points in time.



(a) Scholar A. (b) Scholar B.

Figure 1.1. Popularity evolution of two scholars during 20 years.

each scholar, which implies that determining the duration of the monitoring period for each scholar is part of solving the prediction task. However, the challenge of addressing the trade-off between prediction accuracy and remaining citations after prediction on a per-scholar basis makes this problem much harder than traditional classification tasks.

To tackle the problem of predicting popularity trends of individual scholars, we apply a novel two-step combined learning approach called TrendLearner which was originally proposed for user generated content (UGC) [24] for online content. In this approach the popularity trends of UGC are predicted based on a trade-off between prediction accuracy and remaining interest in the content after prediction. In this dissertation, we adapt TrendLearner to our context of predicting scholar popularity trends by solving the tradeoff between prediction accuracy and remaining citations (or remaining popularity) after prediction. The adaptation consists mainly using features that are specific of our target domain. In other word, we applied the same algorithm of TrendLearner with different features (which associated with scholars) for training classifiers. The idea is to monitor a scholar to determine, for each one, individually, the earliest point in time when prediction can be made with enough confidence (defined by input parameters), producing, as output, the probabilities of each scholar belonging to each class (trend). We also combine the results of this classification task (i.e., the probabilities) with a set of academic associated features, such as number of publications and number of distinct venues, building an ensemble learner. We call our final solution *ScholarTrendLearner*.

1.2 Objectives

The general objective of this dissertation is narrowed down into three specific goals we aim to achieve:

1. **Clustering and classifying scholars based on popularity time series**
   Using state-of-the-art time series clustering techniques, we extract the most common popularity evolution trends followed by scholars. Each discovred trend is represented by a time series centroid.

2. **Extracting and predicting popularity trends of scholars**
   After defining the most correlated features, we exploit the available data to answer the following question: Is it possible to predict how the popularity of individual scholars evolves over time? In other words, we want to know if it is possible to predict the popularity curve (or trend) of each scholar.

3. **Predicting the Number of Citations at a Future time**
   We also investigate whether more effective methods to predict the popularity measures (e.g., citations) of a scholar at a target date can be devised. This is done by exploiting the developed popularity trend prediction models by building specialized models to pre-defined popularity trends. Our results showed that we can indeed improve popularity prediction models using trend prediction models. More importantly, we focus not only on predicting the popularity of a scholar at time $t_t = t_r + \sigma$ but also on the evolution its popularity it may follow after prediction.

1.3 Contributions and Outline of Dissertation

In sum, our main contributions include:

1. Prediction of scholar popularity trends as early and accurate as possible recognizing that different scholars may exhibit quite different popularity trends (as identified by Gonccalves et al. [30]).

2. Determining the best monitoring period for each scholar so as to achieve a good trade-off between prediction accuracy and remaining citation (or popularity) after prediction.

3. The use of ScholarTrendLearner to improve the prediction of popularity metrics (e.g., number of citations), with improvements over the baselines.

The main results of this dissertation were published as a full paper in the proceeding of ACM/IEEE Joint Conference on Digital Libraries (JCDL 2106) [41], the main conference in the field.

The rest of this dissertation is organized as follows. Chapter 2 discusses our related work. We state the target problem and present our approach to solve it in chapter 3. The

dataset and the experimental setup as well as our main experimental results are discussed in Chapter 4. Finally Chapter 5 presents conclusions and directions for future work.

Chapter 2

Related Work

In this chapter, we discuss previous efforts related to our objectives (presented in Chapter 1). In Section 2.1, we discuss works on scientific popularity, influence and related metrics. In Section 2.2, we discuss previous analyses of the temporal evolution of popularity. Finally in Section 2.3, we shift our focus to works on the prediction of scholars' popularity.

2.1 Scientific Popularity

Influence, as a measure of research achievement, has long been discussed. For example, Impact Factor, proposed by Eugene Garfield, is a measure that reflects articles' influence [27, 54]. Journals with higher impact factors are deemed to be more important than those with lower ones [23]. However, impact factor does not reflect the influence of individual papers [16] and hence needs a normalization from the audience of citing sides [65]. Other previous studies aim at measuring the influence of scientific research based on different metrics. These metrics can be classified into two categories: publication level and author level metrics. A number of previous efforts focused on estimating the influence of scientific publications. As an example, Yan et al. implemented a system that takes a series of features of a specific publication as input and predicts its number of citations after a given time period [61].

Regarding the influence of individual scholars, Ding and Cronin [18] proposed the use of weighted citation counts as a measure of the prestige of a scholar, whereas unweighted citation counts should be used as an estimate of the scholar popularity.

The Hirsch-index (h-index), introduced in 2005 by J. E. Hirsch [31], is one of the most popular indicators in information science and informetrics. It combines both productivity and citation impact of a scholar by capturing both the number of publications and the number of citations per publication. Hundreds of articles have been written on the h-index and related indices. The work of L.Egghe is a comprehensive study of h-index that presents

advantages and disadvantages of it and also introduces several h-type indices (also called impact measures) along with applications of these indices [22]. Furthermore, as argued by Leydesdorff [34], the h-index is statistically (using PCA = Principal Components Analysis) compared with non-h-type indices (such as Page Rank, impact factor, Scimago Journal Ranking, network centrality measures, etc.). It is found that the h-index combines the two dimensions (size and impact).

More recently, some approaches have been introduced that improve some limitations of $h-index$, such as $g-index$ [21] and the $h_m-index$ [51]. Unlike $h-index$, $g-index$ depends on the full citation count of very highly cited papers and it can be defined as the number of highly cited articles, such that each of them has an average of g citations. Two years later Schreiber introduced $h_m-index$, a modification of $h-index$ that takes multiple co-authorships into account, by counting each paper only fractionally according to (the inverse of) the number of authors.

Citations indicate the influence of authors, papers and venues. Several works have conducted new retrieval models developed by analysis of citation behaviors can outperform previous approaches. For example Bethard and Jurafsky introduced a model for scientific article retrieval that incorporates a wide variety of important scientific factors, and learns the weights of each of these factors by observing citation patterns [5]. Pao [21] ran a case study where medical professionals gave a description of a topic of interest and an example article, and librarians searched using both keywords and citations. Pao found that searching by citations added an extra 24% relevant articles not found by keyword search [43]. Some previous efforts studied on the relation between citation patterns and impact publications. For instance, Shi, Leskovec, investigated how different citation patterns reflect the scientific impact of the paper. In this study, the authors developed citation projection graphs by investigating citations among publications that a given paper cites [52].

In a different perspective, some previous work focused on quantitative measures of scientific impact. Gonçalves et al. [30] quantified the impact of various academic features (e.g., number of publications, quality of publication venues, properties of the co-authorship network, etc) on scholar popularity by applying regression analysis. The authors also uncovered five profiles (or trends) of scholar popularity evolution. Cason and Lubotsky [9] conducted one of the earliest citation analysis studies with focus on measuring dependences among journals. Pinski and Narin [46] evaluated the influence of journals by taking both the number of citations and the importance of the citing journal into account. As one of the newest study in the research area of quantifying measures of scientific popularity, we can mention work of M.Ausloos et al. [3]. In this study, the role (or weight) of co-authors, has been estimated as the additional value to an author paper's popularity. It is found that an effective h-index can be calculated from the co-authorship popularity matrix (called $H-matrix$)

eigenvalues, through the selection of team partners, but also up to the whole team size.

Finally, in 2015, a very interesting study was performed on the relationship between scholars' breadth of research and scientific impact. Since many existing metrics to evaluate scholars only concern their scientific impact and neglect the importance of the breadth of their research, the authors proposed a new metric based on the existing generalized Stirling metric and compared it to existing metrics, evaluating its relationship to scientific impact[62].

2.2 Evolution and Clustering of Temporal Patterns

Time series data occur in almost all domains, and this fact has created a great interest in time series data mining [4]. There is a plethora of classification algorithms that can be applied to time series. Comprehensively, we can refer the reader to a machine learning book for a description of classification techniques [39]. A recent work suggests that for time series clustering, the choice of the clustering algorithm is much less important than the choice of distance measure used [17]. One of the simplest definition of distance is the Euclidean one, but despite simple, the euclidean distance has major drawbacks, as pointed out by previous studies [4, 63]. For example, this measure fails to account for the shifted behavior of time series.

In order to mitigate the problem of Euclidean distance, Bataista et al. [4] introduced the first complexity-invariant distance (CID) measure for time series and showed that it generally produces significant improvements in classification and clustering accuracy. A measure that is invariant to both shifts and scale is said to be complexity invariant [4]. However, the effectiveness of such a measure in extracting popularity trends has not yet been assessed. Other previous work also make use and extend on the notion of Dynamic Time Warp (DTW) [48, 58]. DTW is not a distance measure. It is an algorithm that finds the optimal alignment between consecutive time series points. In essence, it deals with problems in shifts to align time series and then computes distances using any given distance measure.

Yang et al. [63] proposed a complexity invariant distance measure called d_{ksc}. We note that, in our work, in order to extract trends from popularity time series, we employ the KSC algorithm that uses d_{ksc} which unlike other measures, it can be directly employed in a K-means [39] framework.

We now shift our focus to recent studies of popularity evolution, specially in online content. For instance, Borghol et al. [7] showed how weekly based views can be used to model popularity of user-generated videos. Also, the authors developed a model to determine the number of videos that may exceed a given popularity threshold. More recently, the work of Islam et al. [32] showed that modeling of user-generated video popularity evolution based

on weekly view counts, is still valid even years after upload. Focusing on image content, Cha et al. [12] analyzed the propagation of pictures through Flickr internal OSN. The authors found that popularity (measured in number of favorite markings) of the most popular Flickr pictures exhibit close to linear growth. The authors discussed the importance of social links in the increase in popularity of images, showing that about 50% of favorite markings come from social cascades.

Another interesting study in this context was done by Ratkiewicz et al. [49]. The authors investigated how external events, captured by search volume on Google Trends3 and local browsing (i.e., university/community traffic) affect the popularity of Wikipedia articles. More recently, Khosla et al. [76] compared the use of image and social features for predicting the final popularity values of images.

Regarding clustering objects based on their popularity patterns, there has also been some efforts. For instance, Yang and Leskovec [63] proposed a time series clustering algorithm to identify trends on temporal patterns of popularity evolution. The model proposed by Matsubara et al. [36] provides a unifying analytical framework of the temporal patterns extracted by Crane and Sornette [15] and Yang and Leskovec [63].

2.3 Prediction of Scholar's Popularity

We now focus on previous research that aimed at developing models to predict the popularity of publications or scholars. For example, the authors [8, 10, 35] used measures computed after a paper was published (e.g., number of downloads) to predict its future citation count. Chakraborty et al. also tackled the problem of predicting citation counts of a given article by proposing a two-stage prediction model. The first step of the model fits the pattern of early popularity measures of the article into one of six given patterns. Next, a regression model predicts future citation count of the article based on the subpopulation of scholars (in a training set) who follow the same fitted pattern [13].

Aiming at *predicting* the future popularity of a scholar, Mazloumian [37] examined the predictive capability of citation counts and found that they are reliable predictors of future success (e.g., future citation counts and approval of research grants) for scientists. Acuna et al. proposed a model to predict the future h-index of a scholar based on linear regression with elastic net regularization [1]. The authors evaluated their model on a set of 3,085 neuroscientists. Complementary work of Acuna model, Penner et al. showed that any regression model aimed at "predicting" should avoid using cumulative, nondecreasing, career measures because the retention of past information intrinsic to such measures will yield artificially large coefficients of determination R^2 [45]. On a different direction, Penner

et al. provided evidence that, for the purpose of predicting a scientist's future h-index, linear regression models suffer a variety of flaws and their performance strongly depends upon career age [44].

Van Dijk [57] focused on a slightly different problem: predicting whether a scholar will become a principal investigator (PI). They found that it depends on the number of publications, the impact factor (IF) of the journals in which those papers are published, and the number of papers that receive more citations than average for the journal in which they were published (citations/IF). However, both the scholar's gender and the rank of their university are also of importance, suggesting that non-publication features play a statistically significant role in this process. Hirsch [31] on the other hand focused on an comparison of the predictive power of different metrics, namely, h-index, total citation count, citations per paper, and total paper count. He found that h-index *appears* to be more suitable to predict future achievement than the other metrics but explicitly stated that further studies are required to confirm this.

Compared to these studies, we here focus on a somewhat different problem: predicting the popularity trend (or curve, evolution pattern) of a scholar as early and accurate as possible. Yet, we show that our popularity trend prediction model can be applied to improve regression-based models that predict the future popularity value of a scholar. This is achieved by developing specialized regression models for each trend (as proposed by Pinto et al. [47]).

Unlike all previous studies, this dissertation is the first study that aims at predicting popularity trends for scholars. However, similar efforts in other domains, notably popularity of user generated content (UGC), can be cited. For instance, Nikolov [42] proposed a method that predicts whether a tweet will become a trending topic by applying a binary classification model (trending versus non-trending), learned from a set of objects from each class. Ahmed et al. [2] designed a prediction model in two steps. First they classify UGC objects (e.g., videos) based on their popularity trends and then predict the popularity of that object in the future. Unlike those studies, our method considers the trade-off between remaining citations after prediction and prediction accuracy, adapting a model called TrendLearner, previously proposed to the context of UGC popularity trend predictions [24], to the particular context of scholar popularity trends. Like TrendLearner, our approach determines the duration of the period during which each scholar should be monitored before prediction on a per scholar basis, while other studies considered fixed monitoring periods for all objects such as [47] which predicted the popularity of youtube videos, [55] that aimed to predicting the long time popularity of online conten, and study of [33] that proposed a methodology about macroscopic prediction of the popularity of online contents. Even though in these studies the authors showed the effectiveness of their methods for different monitoring periods, they did

not discuss on methods how to determine such monitoring periods for each individual object.

Prediction of popularity trends has also been studied in social networks and search engines. For instance Vakali et al. [56] designed a cloud-based application named Cloud4Trend to cluster streams of web data and detect the trend of user generated content on Twitter and blogging systems. Golbandi et al. [29] explored a search trend detection algorithm [19] to develop a method for predicting query counts in order to detect search trends.

In sum, to the best of our knowledge, ours is the first work that tackles the prediction of scholar popularity trends as early and accurately as possible recognizing that different scholars may exhibit quite different popularity trends (as identified by Gonçalves [30]). Our solution determines the best monitoring period for each scholar so as to achieve a good trade-off between prediction accuracy and remaining citation (or popularity) after prediction.

Chapter 3

Early Prediction of Popularity Trends of Scholars

In this chapter we describe our scholar popularity trend prediction model, which was adapted from a method that was originally proposed for predicting the popularity of UGC. As mentioned before, we tackle the trade-off between prediction accuracy and the capability of making such prediction as soon as possible, a problem to which we refer to as *early prediction of scholar popularity*. Our model can be summarized into two parts. Firstly the goal is to extract the scholar popularity trends using a training set. Next, we predict the popularity trend (or class) of each scholar in a test set by training a classification method using various scholar features as input.

The rest of this chapter is organized as follow. We formally describe the prediction problem in Section 3.1. In Section 3.2 we present proposed solution (i.e. ScholarTrendLeaner) for early prediction of popularity trends. In Section 3.3 we discuss how predicted trend by ScholarTrendLeaner can improve results of regression-based popularity prediction models.

3.1 Problem Statement

In this dissertation, the early popularity trend prediction problem is defined as follows. Given a training set D_{train} and a test set of scholars D_{test}, the popularity trends are extracted from D_{train}; then a trend which previously extracted is predicted for each scholar in D_{test} using a classifier as early and accurately as possible. Table 3.1 describes the notation used in this chapter. Each scholar x is presented by an n-dimensional time series vector $p_x =< p_{x1}, p_{x2}, ..., p_{xn} >$, where p_{x1} is the acquired popularity (i.e., number of citations) by scholar

x during the i^{th} monitoring time window. We also note the complete dataset used in this chapter is referred to as $D = D_{train} \cup D_{test}$.

Table 3.1. Summary of Notation

Symbol	description
D_{train}	training set
D_{test}	test set
K_i	class i
C_{K_i}	centroid of class i
x	scholar $x \in D_{train}$
p_x	time sereis vector of scholar x
p_{xi}	popularity of x at i^{th} time window
$p_x[a:b]$	a slice of vector p_x from elements a up to b

3.2 Proposed Solution

As we discussed in Chapter 1, we tackle the problem of early predicting popularity trends of individual scholars by applying a new trend classification approach, namely TrendLearner [24]. We adapt TrendLearner to our context of predicting scholar popularity trends by solving the trade-off between prediction accuracy and remaining citations (or remaining popularity) after prediction. Our adaptation of TrendLearner to the scholarly domain has two main steps:

1. In the first step, the popularity trends of scholars are identified by applying a time series clustering algorithm, named K-Spectral Clustering (K-SC) [63]. K-SC extracts popularity trends from a training set based on the centroids of clusters, being agnostic to the volume and length of the time interval.

2. In the second step, a classifier is first built to predict the popularity trend (i.e., class) of each scholar based on distances between her popularity time series and the trends previously extracted by K-SC. This classifier produces as output the probability of the scholar belonging to a particular trend/class. Finally, TrendLearner builds upon this classifier by combining those probabilities with a set of academic features associated with the scholars (e.g., number of publications, number of venues) to an ensemble learner named Extremely Randomized Trees classifier [28].

We here refer to this adaptation of TrendLearner to the scholar domain as ScholarTrendLearner. Compared to the original TrendLearnerm the main differences of ScholarTrendLearner lies in the choice of academic features which are specific of the domain.

Moreover applying such information to a completely different domain is a key contribution of our work. As we will show in Section 4.3, it produces new insights and qualitative resuls which are different from those in TrendLearner [24].

3.2.1 Trend Extraction

Since our popularity prediction model deal with time series, in this section we first discuss common representation of time series in Section 3.2.1.1. Then, we discuss extracting popularity trend of scholars in Section 3.2.1.2.

3.2.1.1 Time Series Representation

In research areas such as Statistics [53] it is common to represent time series using definitions from the stochastic processes literature. Since this is a more general representation, we begin by briefly describing stochastic processes. We then narrow this definition down to the vector representation of time series commonly used in data mining (as well as this dissertation).

A *stochastic process* is denoted as:

$$\{x_{t_i}\}_{i=1}^{\infty} = x_{t1}, x_{t2}, x_{t3}, ..., \tag{3.1}$$

where x_{t_i} are values in $\mathbb{R}$. Each such observation defines the quantity which the time series captures. The values ti represent the points in time (or indexes) for each quantity x_{t_i}. A necessary condition is that $t_1 < t_2 < t_3 t_n$, that captures the nature of a series. It is common for quantities to be observed at uniform lengths from one another, thus making the use of the index variable t_i unnecessary in most applications. Thus, a simpler notation is $\{x_t\}_{t=1}^{\infty}$.

Since the definition of time series based on stochastic process is general definition, in practice we observe a subsequence of the time series. That is, a vector x of observations is observed. In this sense, a time series can be summarized simply as a sequence of data points measured at different times steps [26]. Thus, we define a time series vector as:

$$x = < x_{t_1}, x_{t_2}, x_{t_3}, ..., x_{t_n} > \tag{3.2}$$

where x is an observation vector, again composed of values $x_{t_i} \in \mathbb{R}$. The same comment for uniform indexes apply in this case, thus turning the definition above in the one below:

$$x = < x_1, x_2, x_3, ..., x_n > \tag{3.3}$$

3.2.1.2 Extracting Popularity Trends of Scholars

We identify scholar popularity trends by clustering the popularity time series of scholars in a given training set. To that end, we exploit a clustering algorithm called K-Spectral Clustering (K-SC) as done by Gonçalves et al. [30]. The K-SC algorithm effectively finds temporal patterns based on a time series similarity measure. Similarly to the K-Means algorithm [64], which minimizes the sum of the squared Euclidean distances between the members of the same cluster, K-SC computes cluster centroids by introducing a new distance metric that is invariant to scaling and translation of the time series [63]. That is, given two vectors p_x and p_y that represent the popularity time series of two scholars, the distance $dist(p_x, p_y)$ between both vectors is defined as following:

$$dist(p_x, p_y) = \min_{\alpha,q} \frac{||p_x - \alpha p_{y(q)}||}{||p_x||} \tag{3.4}$$

where $p_{y(q)}$ is the shifted time series p_y by q time units. Note that $dist(p_x, p_y)$ is symmetric in p_x and p_y. For a fixed value of q the optimal distance can be computed by setting its gradient in terms of α equal to zero. Therefore the exact solution for α is $\alpha^* = \frac{p_x^T p_{y(q)}}{||p_{y(q)}||^2}$ which minimized $dist(p_x, p_y)$. However there is no simple manner to find the optimal q. Thus, as in [24, 30], we search for the optimal value of q considering all integers in the range (-n, n), where n is the length of the input vectors p_x and p_y. Note that K-SC requires all time series have the same size n. Thus, we represent each scholar by a vector p_x with 20 elements, that each element represents the scholar popularity (i.e., number of citations) in one year. We discuss more about how we set the value of n in Section 4.1. The detailed description of K-SC algorithm can be found in [63].

Though there exists other clustering methods, such as K-Means and Affinity Propagation [39], we chose to use K-SC as it has some desirable properties for our application. Firstly, we need a time series clustering method compatible with our focus on trends (i.e., popularity evolution patterns), as opposed to specific popularity and time values. Secondly, the euclidean distance used in the aforementioned methods has major drawbacks for this goal, as pointed out by previous studies [4, 63]. For instance, it fails to account for the shifted behavior of time series. K-SC, on the other hand, employs $dist(p_x, p_y)$, which is invariant to time shifts and popularity scale. Thus, it is an algorithm capable of finding the optimal alignment between different time series. Thirdly, it has been shown that K-SC can be very effective on the task of extracting trends from social media [63] and should be easily adaptable to our goals.

Given a number of clusters k and the set of time series to be clustered, K-SC algorithm works as follows:

1. The time series are uniformly distributed to n random clusters K_i, where $i = 1, ..., n$;

2. Cluster centroids are computed based on the members of each cluster. In K-Means based algorithms, the objective is to find centroid c that minimizes:

$$c^* = arg\min_{c} \sum_{p_x \in K_i} dist(p_x, c)^2$$

   We refer the reader to the original K-SC paper for more details on how to find c [63];

3. Each time series vector p_x is assigned to the nearest centroid based on distance metric $dist(p_x, p_y)$;

4. Return to Step 2 until convergence, i.e., until all time series remain within the same cluster in Step 3.

Each cluster's centroid represents the popularity trend that the time series in the cluster follow. So we refer to each cluster as a class K_i, which is represented by centroid C_{K_i}. We discuss how we define the given number of clusters k in Section 3.2.2.

The next step of our method consists of predicting the cluster (or class) to which each scholar in a test set belongs to. Such prediction is performed given the identified cluster centroids (classes) as well as early measures of the scholar popularity (i.e., early points in the scholar popularity curve) and possibly the values of a set of academic features computed over the same monitoring period. We discuss this step of our method in the next section.

3.2.2 Trend Prediction

Given the k centroids (classes) obtained in the previous step using a training set, we now aim at predicting the popularity trend, i.e., determining the class, of each scholar in a given test set as early and as accurate as possible. Hence, we perform our prediction task by building a classifier which monitors the popularity time series (and possibly other academic features) of each scholar x during a monitoring period t_x. As soon as the classifier is "confident enough" that it can determine the class of x, the algorithm stops and returns the detected class. This is performed for each scholar in the test set independently.

We experiment with three classification strategies. The first strategy exploits solely the distances between the popularity curve of each vector p_x (up to the monitoring period t_x) and the centroid of all classes (Section 3.2.2.1). We also explore two other classification strategies that employ a state-of-the-art learning method – extremely randomized trees (ERTrees) [28] – to build a classification algorithm. In one strategy, we use the same computed probabilities as input to the ERTrees. In the other, we experiment with a set of academic features,

whose values are computed over the same monitoring period t_x, as input to ERTrees (Section 3.2.2.2). Finally we combine the two former approaches in a third algorithm by using both probabilities and academic features as input to ERTRee. We show that this algorithm, which we call ScholarTrendLearner, improves the quality of prediction task considering the tradeoff between prediction accuracy and citations after prediction (Section 3.2.2.4).

3.2.2.1 Prediction Based on Class Probabilities

We build a classifier that computes the probability of belonging to class K_i based on the distances between the initial points in the popularity curve of x (captured in vector p_x) and each curve C_{K_i} (denoting the centroid of class K_i). Regarding the shifting invariants in computing the distances, we consider all possible alignments between p_x and C_{K_i}. That is, given a monitoring period t_x, we take a starting time window t_s and vary it from 1 to $|C_{K_i}| - t_x$, where $|C_{K_i}|$ is the number of time windows in C_{K_i}. So given centroid C_{K_i}, the monitoring period t_x and a starting window t_s, this probability is obtained as follows:

$$P(p_x \in K_i | C_{K_i}; t_x, t_s) \propto e(-dist(p_x[1:t_x], C_{K_i}[t_s : t_s + t_x - 1])) \tag{3.5}$$

As already discussed, different popularity time series may need different monitoring periods. given Equation 3.5, the classifier computes the probability of each scholar belonging to each class at the end of each time window, starting with t_x equal to 1, and returns the class K_i with the highest probability For each scholar x, the algorithm stops this procedure once the computed probability exceeds a class-specific threshold $\theta^{[c]}$ or the monitoring period t_x exceeds a maximum limit γ_{max}. Threshold $\theta^{[c]}$ captures the minimum confidence required to state that a scholar belongs to class k. We also consider that a minimum monitoring period is provided for each class, given that different classes, exhibiting different popularity dynamics, may require quite different monitoring periods. This procedure is shown in Algorithm 1.

Algorithm 1 computes probabilities and monitoring periods for all scholars in a given test set D_{test}. The algorithm takes as input the time series of all scholars in D_{test}, the vector C_K with all class centroids, vectors θ with the minimum confidence thresholds for each class, vector γ with the minimum thresholds for monitoring period for each class, and γ_{max}, the maximum threshold for the monitoring period. The output is a vector t with the number of monitored time windows for each scholar and a matrix M with the probabilities of each scholar belonging to each class.

The algorithm begins by initializing matrix M and vector t with 0 in all elements. Starting with a monitoring period t_x equal to the minimum possible for all classes. The algorithm monitors each time series in D_{test}, and computes the probability of time series belonging to each class using function *ComputeProb*. This works as follows: for a given t_x,

the function computes the probability by trying all possible alignments between the initial elements of p_x (up to t_x) and the corresponding cluster centroid. This is done by applying Equation 2 using all possible values of t_s. The algorithm takes the largest of all computed probabilities along with the associated class. The algorithm stops searching for the class of a scholar x when both the computed probability and the monitoring period t_x exceed class-specific thresholds $\theta^{[k]}$ and $\gamma^{[k]}$, respectively. At this point, it saves the identified class in matrix M and the current t_x in vector t. The algorithm repeats this procedure for all scholars in D_{test}, returning matrix M and vector t. Note that matrix M may contain only zeros for some scholars, indicating cases for which the algorithm was not able to predict a class with minimum confidence, within the maximum monitoring period allowed (γ_{max}). This classification strategy exploits only the probabilities of a scholar belonging to each class. We refer to it as *ProbClassifier*.

Algorithm 1 Producing Matrix M and vector t

Require: D_{test}, C_k, θ, γ and γ_{max}
  Initial $t[i] = 0$; $M[i][j] = 0$; $i \leftarrow |D_{test}|$ $t_x \leftarrow min(\gamma)$
  while $(t_x \leq \gamma_{max})$ and $(i > 0)$ **do**
    for all $p_x \in D_{test}$ **do**
      for all $C_{K_i} \in C_k$ **do**
        $P^{[i]} \leftarrow ComputeProb(p_x, C_{k_i}, \theta^{[i]}, t_x)$
      end for
      $prob \leftarrow max(P)$; $k \leftarrow argmax(P)$ ▷ Identify class k with largest probability
      if $(prob > \theta^{[k]})$ and $(t_x \geq \gamma^{[k]})$ **then**
        $t^{[x]} \leftarrow t_x$; $M^{[x]} \leftarrow p$
        $i \leftarrow i - 1$; $D_{test} \leftarrow D_{test} - \{p_x\}$ ▷ Classification of x is done, remove it from D_{test}
      end if
    end for
    $t_x \leftarrow t_x + 1$
  end while
  return t, M
  function $ComputeProb(p_x, C_{k_i}, \theta^{[i]}, t_x)$
    $t_s \leftarrow 1; p \leftarrow 0$
    while $(t_s \leq |C_{k_i}| - t_x)$ and $(p < \theta^{[i]})$ **do**
      $p^{'} \leftarrow e\,(-dist(p_x[1:t_x], C_{K_i}[t_s : t_s + t_x - 1]))$ ▷ Check all possible alignments between the first t_x windows of p_x (monitoring period) and C_{K_i} by varying t_s. Stops once probability exceeds minimum threshold
      $p \leftarrow max(p, p^{'})$ ▷ Take the largest probability (best alignment)
      $t_s \leftarrow t_s + 1$
    end while
    return p
  end function

3.2.2.2 Prediction Using ERTrees

Inspired by [14], we propose to use the probabilities obtained by Algorithm 1 as input features to a feature-learning method. Specifically, we use the estimated probabilities as input to an Extremely Randomized Trees (ERTrees) classifier, a tree-based ensemble method [28]. We refer to this prediction method as *ProbERTrees*. We chose ERTrees as our classification algorithm because of its good accuracy and computational efficiency on large scale datasets. The method builds regression trees according to the classical top-down procedure but randomly choosing the most appropriate features to grow up the trees. A majority voting of the individual regression trees at classification time leads to the final prediction. For more information about ERTrees we refer the reader to the original paper [28]. The description of the Extra-Trees algorithm is given in Section 3.2.2.3.

As a variation of the aforementioned strategy, we also exploited the ERTrees classifier using as input a set of academic features associated with each scholar. These features are presented in Table 1. Note that the feature values used as input are computed over the same monitoring period t_x. We call this approach *FeatERTrees*.

Table 3.2. List of Considered Academic Features

Feature Notation	**Description**
# citations	total number of citations
# publications	total number of publications
# coauthors	total number of coauthors
# venues	total number of distinct venues
$h-index$	h-index of scholar at each year
short impact factor	average yearly number of citations in the last two years
long impact factor	average yearly number of citations in the whole period

3.2.2.3 Extra-Trees Algorithm Description and Rationale

The Extra-Trees method implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset and use averaging to improve the predictive

accuracy and control over-fitting. The two main differences of this method with other tree-based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees. In our case, all features shown in Table 3.2 are numerical features. So the Extra-Trees splitting procedure for numerical input features is given by Algorithm 3.2.2.3[1].

Algorithm 2 Extra-Trees splitting algorithm (for numerical features)

Split_a_node(S)
Input: the local learning subset S corresponding to the node we want to split
Output: a split [a < ac] or nothing
– If stop split(S) is TRUE then return nothing.
– Otherwise select K features $a_1, ..., a_K$ among all non constant (in S) candidate features;
– Draw K splits $s_1, ..., s_k$, where $s_i =$**pick a random split**$(S, a_i), \forall_i = 1, ..., K$;
– Return a split s_* such that Score$(s_*, S) = max_{i=1,...,k}$ Score(s_i, S)

Pick a random split(S, a)
Input: a subset S and an attribute a
Output: a split
– Let a^S_{max} and a^S_{min} denote the maximal and minimal value of a in S;
– Draw a random cut-point a_c uniformly in $[a^S_{max}, a^S_{min}]$;
– Return the split $[a < a_c]$.

Stop split(S)
Output: a subset S
Output: a boolean
– If $|S| < n_{min}$, then return TRUE; – If all features are constant in S, then return TRUE; – If the output is constant in S, then return TRUE; – Otherwise, return FALSE.

This algorithm has two parameters: K and n_{min} , the former is the number of features (or features) randomly selected at each node. The parameter K may be chosen in the interval $[1, ..., n]$, where n is the number of features. The latter is the minimum sample size for splitting a node. It is used several times with the (full) original learning sample to generate an ensemble model. The predictions of the trees are aggregated to yield the final prediction, by majority vote in classification problems and arithmetic average in regression problems [28].

[1] complete Extra-Trees algorithm, for numerical and categorical features, is presented by Geurts et al. [28]

3.2.2.4 ScholarTrendLearner

Finally, we propose to “merge" the two former approaches by combining the probability features with “domain-specific" features. In other words, we use as input features to the ERTrees classifier both the set of probabilities taken from matrix M and the scholar's associated feature values (Table 1). We refer to this approach as ScholarTrendLeaner.

3.3 Applying Results to Regression-Based Predictive Models

In this section, we show how to apply the predicted trends to estimate the popularity of a scholar at a given future time. For this, we exploit ideas first proposed in [47], which builds specialized regression popularity prediction models for Web objects (in the case, YouTube videos) with similar popularity patterns. In that work, the authors demonstrated that the specialized models can, at classification time, reduce the prediction errors. We combine our predicted trends with the state-of-the-art (ML) and MRBF regression-based models proposed in [47] to predict the future popularity of scholars.

Before describing the prediction models, we discuss the regression models commonly used to predict popularity of Scholars in Sections 3.3.1. Next, the used performance criterion by ML and MRBF models (i.e, the mean Relative Squared Error (mRSE)), described in Section 3.3.2. Finally, we describe the ML and MRBF prediction models in Sections 3.3.3 and 3.3.4 respectively.

3.3.1 Linear Regression Models

Ordinary Least Squares (OLS) linear regression models have been adopted as a means build models of popularity prediction. An OLS regression is defined as follows:

$$y_{t+h} = X_t^T \Theta + \varepsilon \tag{3.6}$$

where X_t is a matrix of multiple time series column vectors (also called the covariate matrix), each with observations up to reference time t, y_{t+h}, is the response, and ε is the error of the model. The notation, X^T represents a matrix transpose. Solving the OLS equation for Θ will define the prediction model, that is, the parameter Θ, that minimizes the Mean Squared Error (MSE):

$$mse(y) = n^{-1} \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{3.7}$$

However, if applied to heavy-tailed data this model may fail to produce accurate predictions. One of the premises for linear regression is that of independence between the errors, ε, and the response y_{t+h}. Due to the heavy-tailed nature of popularity, this premise is violated. For example, if we take the MSE equation above and apply it to a heavy-tailed distribution, it is better to reduce the error on the most popular objects since any arithmetic mean is biased towards higher values. Thus, a correlation between errors and parameters will exist in the model. To understand this, note that MSE will try to get the most popular content correct, since they have a large impact in the mean error. Given that only a handful of these objects exist, the model may be wrong for the majority of content.

In order to mitigate this behavior, the Mean Relative Squared Error (MRSE) was suggested [47, 55], being given by:

$$mrse(y) = n^{-1} \sum_{i=1}^{n} \left(\frac{y_i - \bar{y}}{y_i}\right)^2 \tag{3.8}$$

In order to create such a model, we can slightly change the OLS equations. That is, the new equation will have the form:

$$1 = T_t^T \Theta + \varepsilon, \tag{3.9}$$

where 1 is a vector of ones, and T_t is a matrix composed of column vectors with the following normalization:

$$t^j = < \frac{x_1}{y}, \frac{x_2}{y}, ..., \frac{x_n}{y} >, \tag{3.10}$$

which is the original time series vector divided by the response variable. The proof for that this model minimizes MRSE can be found inis given by Pinto et al. [47]. We use the above models for our prediction task.

3.3.2 Performance Criterion

I this section, we introduce the performance criterion (i.e., mRSE) adapted to evaluate the performance of the ML and MRBF prediction models. Let $N(x,t)$ be the total number of citations received by scholar x up to day t (N(x,0)=0), and $\hat{N}(x,t_r,t_t)$ be the total number of citations predicted for scholar x at target date t_t based on data from the first t_r time windows. Given a training set C, the mRSE for this prediction is defined as:

$$mRSE = \frac{1}{|C|} \cdot \sum_{x \in C} \left(\frac{\hat{N}(x,t_r,t_t)}{N(x,t)} - 1\right)^2 \tag{3.11}$$

As shown in equation 3.11, the relative squared error is adapted instead of absolute quadratic error. Because, relative errors tend to be more relevant and meaningful than absolute ones, particularly given the great variability in popularity across different scholars [11].

3.3.3 Multivariate Linear (ML) Model

The original ML model is a multivariate linear regression model that receives the popularity (number of citations in our case) acquired by an object (scholar) at multiple given points in time up to a given reference date t_r and predicts the popularity of the scholar at a target date t_t ($t_r < t_t$). We assume each time window be one year, although in this model there is no assumption on the rate of time windows.

More formally, the ML model is described as follows. let $g_i(x)$ be the number of citations received by scholar x on the i-th time window, therefore, $g_i(x) = N(x,i) - V(x,i-1)$, thus the *feature vector* $G_{t_r}(x)$ is defined as:

$$G_{t_r}(x) = (g_1(x), g_2(x), ..., g_{t_r}(x))^T \tag{3.12}$$

and the popularity of scholar x at t_t is estimated as:

$$\hat{N}(x,t_r,t_t) = \theta_{(t_r,t_t)} \cdot G_{t_r}(x) \tag{3.13}$$

where $\theta_{(t_r,t_t)} = (\theta_1, \theta_2, ..., \theta_{t_r})$ is the vector of *model parameters* and depends only on t_r and t_t. Given equation 3.11 and training set C, we can compute the optimal values for the elements of vector θ_{t_r,t_t} as the ones that minimizes the mRSE on C by solving the following optimizatin problem:

$$argmin_{\theta_{(t_r,t_t)}} \frac{1}{|C|} \cdot \sum_{x \in C} \left(\frac{\theta_{(t_r,t_t)} \cdot G_{t_r}(x)}{N(x,t)} - 1 \right)^2 \tag{3.14}$$

which is an Ordinary Least Squares (OLS) problem. If n be the number of scholars in the training set and p be the number of model parameters, it is considered that $n \geq p$. One possible drawback of ML model is that the number of parameters is not fixed and it increases linearly with t_r. however, this would not be an issue in practice because we considered that the ScholarTrendLearner does not observe the popularity of scholars for very long time and it predicts the popularity as early as possible.

3.3.4 MRBF Model

In the ML model, different weights are assigned to different time windows in the observed history of the scholar. The MRBF model is an extension of the ML model that includes additional features to measure the similarity between the popularity curves of the object and known examples from a training set, based on Radial Basis Functions (RBFs). Using these extra features, some particular aspect of certain group of scholars can be captured.

For measuring the similarity between videos, the Radial Basis Function (RBF) is used that is a real-valued function whose value depends only on the distance between its inputs and a given point, the *center*. The Gaussian RBF that capture similarity between a scholar x_c as center and a target scholar x is as follows:

$$RBF_{x_c}(x) = e^{\left(-\frac{||g(x)-g(x_c)||^2}{2.\sigma^2}\right)} \tag{3.15}$$

where σ is a parameter and $g(x)$ is the ML model feature vector for scholar x. A number of scholars from training set are selected randomly to be centers for RBF features. Then for each scholar x, the value of $RBF_{x_c}(x)$ is computed and used as one of the features in the prediction model. This model is called MRBF model and defined as:

$$\hat{N}(x,t_r,t_t) = \theta_{(t_r,t_t)} \cdot G(x) + \sum_{x_c \in C} \omega_{x_c} \cdot RBF_{x_c}(x) \tag{3.16}$$

where C is the set of scholars chosen as centers and ω_{x_c} is the model weight associated with the RBF feature for x_c used for prediction purposes. Notice that the MRBF model as defined in Equation 3.16 is mathematically equivalent to:

$$\hat{N}(x,t_r,t_t) = \theta^*_{(t_r,t_t)} \cdot G^*_{t_r}(x) \tag{3.17}$$

where θ^* is the θ vector with the ω_{x_c} parameters appended to it and $G^*_{t_r}(x)$ is the $G(x)$ vector with the values of the corresponding RBF functions appended to it i.e., the RBF features can be simply treated as additional features in the original feature and parameter vectors. Equation 3.17 is in exactly the same format as Equation 3.13 that describes the ML model. Thus, the optimization problem can be solved using the same OLS technique. Because of extra features, to reduce the risk of over-fitting training set, this optimization problem can also be solved using ridge regression technique.

In order to use the MRBF model, we should set the parameter σ and also the number of scholars that chosen as centers. We experimented with many values for σ and finally chose the value that provided the lowest prediction error in a cross-validation set, i.e., $\sigma = 0.06$. For the number of RBF features, we considered 100 and 500 centers and finally because

of computational cost we selected 100 centers uniformly at random from the training set. However there is very little variation in prediction error between different numbers of centers.

3.3.5 Model Specialization

The different popularity trends (i.e., cluster centroids) display different behaviors, and thus it is likely that exploring particular aspects of each trend can lead to improved prediction accuracy. Because of this reason we exploit mRSE measure to assess the prediction accuracy of the ML and MRBF models applied to the context of scholar popularity, considering two approaches, also proposed in [47]: a general and a specialized model. In the former, the parameters of the regression model are configured using the whole set of objects in training set, while in the latter, the parameters are trained using only specific information of each cluster previously identified by the first step of our ScholarTrendLeaner method. We consider the monitoring time window t_x previously set by our model (see Table 2) as the reference date t_r and $t_t = t_r + \delta$ as target date that considering δ equals to 1 and 4, thus $t_t = t_x + 1,4$, meaning one and four years in the future, respectively.

Chapter 4

Dataset and Experiments

In this Chapter, we first present a brief characterization of the used datasets in Section 4.1. then we discuss our experimental setup in Section 4.2. Finally the results of the trend prediction model are presented in Section 4.3.

4.1 Dataset

We evaluate our prediction models using an experimental research dataset developed by Microsoft Academic named Microsoft Azure Marketplace (MAM)[1]. MAM indexes 19,856,190 scholars covering a total of over 39 million publications by those scholars. The dataset also contains other types of information such as publication venues for journals and conferences as well as keywords and references for each publication.

In order to exclude very inactive scholars, we restricted our dataset to authors having at least 10 publications. After applying this filtering, we obtained data about roughly 1,500,000 scholars. Using this data, we produced the yearly time series of the number of citations for each scholar, considering the period between 1995 and 2014 (20 years). We note that MAM records are very sparse before and after this period. Thus, we set the parameter n, the length of the popularity time series of the K-SC algorithm (see Section 3.2.1) equal to 20 elements[2]. Then we eliminated time series containing more than 80% elements equal to 0 indicating that the corresponding scholars were not very popular throughout their academic careers. After this filtering, we were left with 500,000 scholars, over which we evaluate our prediction models. For each scholar we also extracted the time series associated with the other features

[1]The dataset is of public use and can be downloaded from https://datamarket.azure.com/dataset/mrc/microsoftacademic.

[2]Note that scholars with fewer years of activity or no citations in some of those years have elements equal to 0 in their corresponding popularity vector.

shown in Table 3.2, namely: total number of publications, distinct venues and coauthors as well as the value of *h*-index, short impact factor (the average number of citations of articles published in the last two years), and long impact factor (average number of citations received by papers published so far).

4.2 Experimental Setup

Since our popularity prediction problem is formulated as a clustering task combined with a classifier algorithm, we here discuss how we defined the input parameters of the K-SC algorithm, Algorithm 1 and the Extremely Randomized Trees classifier. Similarly to other clustering algorithms, K-SC requires the number of clusters k as input. To set such parameter, we relied on the β_{CV} clustering quality metric [38]. The β_{CV} is defined as the ratio of the coefficient of variation (CV)[3] of the intra-cluster distances to the CV of the inter-cluster distances. The intra-cluster distance is the distance between a cluster member and its centroid, and the inter-cluster distance is the distance between different cluster centroids. The general purpose of the clustering task is to group elements so as to obtain high similarity among members of the same cluster, and low similarity across members of different clusters. Thus, the idea behind the β_{CV} heuristic is to minimize the variance of the intra-cluster distances while maximizing the variance of the inter-cluster distances. The value of β_{CV} should be computed for increasing values of k. We select the lowest value of k after which the value of β_{CV} becomes stable, implying that intra and inter-cluster distances are stable as well. This indicates that the clustering process converged. After applying the β_{CV} heuristic in our data, we found that the β_{CV} value stabilized around $k = 5$ as shown in Figure 4.1, which was used as an input parameter for the K-SC algorithm.

[3]The ratio of the standard deviation to the mean.

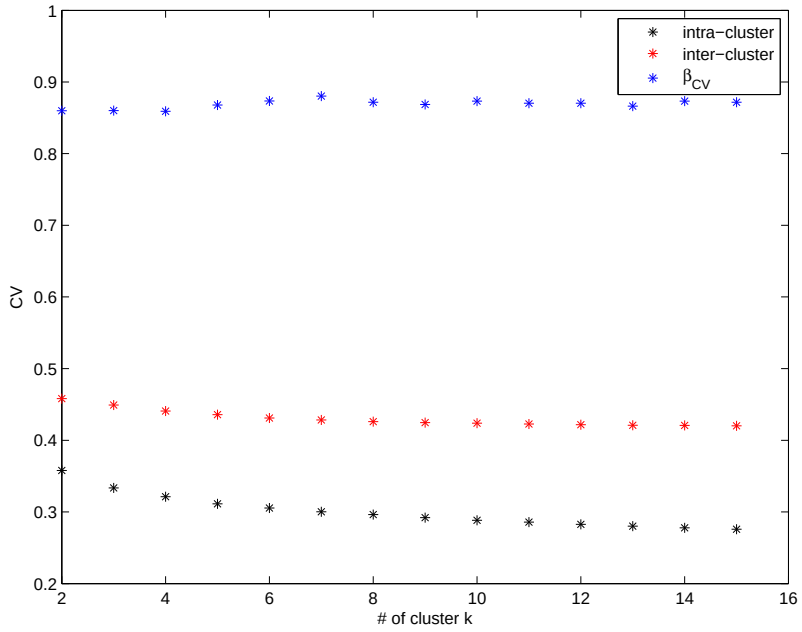


Figure 4.1. β_{CV} clustering quality metric.

Regarding the parameters of Algorithm 1, namely vectors θ and γ, we adopt the same parameterization approach as in [24]. Notably, we apply an One-Vs-All classification (OVA) algorithm [6] for all classes separately. OVA is implemented for all scholars in the training set that are classified previously by considering different values for $\gamma^{[i]}$ from 1 up to γ_{max}. We select the smallest value of $\gamma^{[i]}$ (the minimum monitoring period for class K_i) for which the classification performance exceeds a given target (e.g., classification above random chance, meaning Micro-F1 above 0.5). Using the selected value for $\gamma^{[i]}$, the minimum confidence $\theta^{[i]}$ is the average probability computed for all scholars in class K_i. Regarding parameter γ_{max}, we set its value equal to the total number of points in the popularity time series (i.e., 20), as done in [24]. Table 4.1 shows the best parameter values obtained following the aforementioned procedure for each of the five identified classes. These parameter values were used by all four classification approaches described in Section 3. We note that in order to be able to compare those approaches under fair conditions, all of them monitor each scholar x until t_x, the monitoring time produced as output of ScholarTrendLearner.

Finally, the Extremely Randomized Trees classifier has three parameters K, M and n_{min}. Parameter K determines the strength of the feature selection process and was set to the square root of the total number of features. The averaging strength parameter M denotes the number of trees in the ensemble, set to 20, a default value suggested by the majority of the works on ERTrees. We then apply cross-validation within the training set to choose the

smoothing strength parameter n_{min}, the minimum number of samples required for splitting a node, considering values equal to 1, 2, 4, 8, 16, 32. For more information concerning the parameterization of Extremely Randomized Trees we refer to [28].

4.3 Experimental Results

In this section we present our experimental results regarding the extraction and prediction of scholar popularity trends. The results are assessed by means of a 5-fold cross validation[4] such that the original dataset is randomly partitioned into 5 equally sized folds. One of the folds is used as test set (D_{test}) whereas the remaining four folds are used as training set for learning the model (with one of the training folds used as validation set for parameter setting). The cross-validation process is then repeated 5 times with each of the five folds used exactly once as the test data. The results from the five test folds are then averaged to produce a single prediction result.

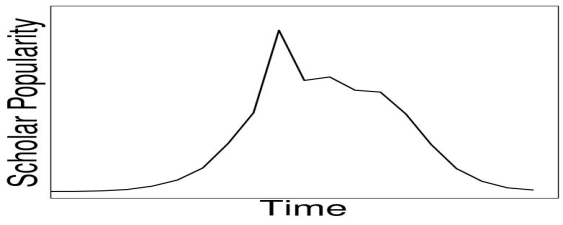
Figure 4.2 shows the centroids (i.e., popularity trends) of the five clusters for scholars in the training set[5]. The figure also presents, for each cluster, the percentage of scholars belonging to it as well as the average number of citations of them. Absolute values in both axes are omitted to emphasize the scale and time shifting (x-axis) invariants of the algorithm.

Classes K_1, K_3 and K_4 correspond to scholars who managed to become increasingly popular over time, acquiring more and more citations over time. Particularly, the popularity of scholars identified in cluster K_4 is roughly stable over a longer period of time. Moreover, those scholars tend to be the most popular ones, on average. Scholars in clusters K_1 and K_3 exhibit a sharper increase (particularly those in K_1) towards their popularity peak, having also a smaller total number of citations, on average, compared to scholars in K_4. Although there are similarities between these two patterns, the main difference regards the popularity growth rates. Note that the sharp decay in the late of career in these clusters exhibits that recent publications have fewer citations. Also, recall that each centroid represents an average popularity curve for all scholars in the cluster. By manually inspecting the popularity curves for various scholars in these three clusters, we found that the decay at the end was not clear in several individual curves, although others did exhibit it.

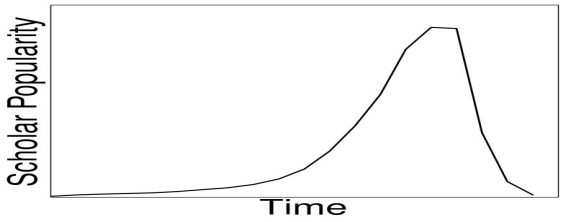
In contrast, scholars in clusters K_0 and K_2 are the least popular ones. They reached the popularity peak more quickly rather than other three clusters. In other word these scholars grow in popularity, experiencing a clear peak, but fail to remain popular afterwards. The main differences between scholars in K_0 and K_2 are the rates of popularity growth and decay before and after the peak, which are much sharper for scholars in K_2. As mentioned, these

[4] A standard technique for estimating the performance of predictive models.

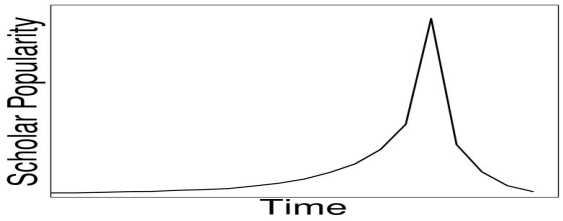
[5] The same clusters were found in all five training sets.



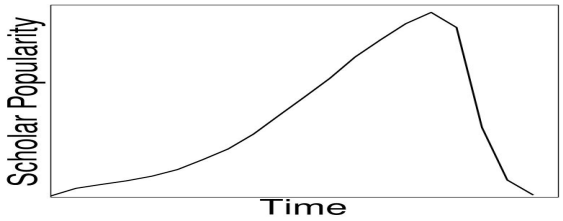
(a) K_0 (13.35% of scholars; Avg.# of citations = 331)



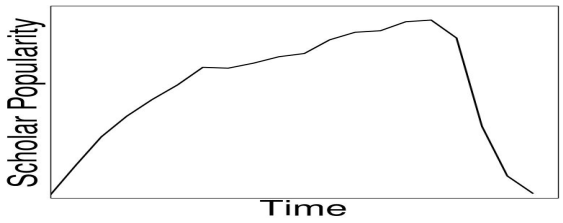
(b) K_1 (24.62% of scholars; Avg.# of citations = 367)



(c) K_2 (10.32% of scholars; Avg.# of citations = 77)



(d) K_3 (31.53% of scholars; Avg.# of citations = 965)



(e) K_4 (20.15% of scholars; Avg.# of citations = 1050)

Figure 4.2. Scholar popularity trends extracted by K-SC.

popularity profiles are similar to those identified in [30]. However, unlike in that work, we here take a step further and try to predict the popularity trend of each scholar and use such trend predictions to improve the prediction of popularity values.

Recall that we here investigate four trend prediction strategies: ScholarTrendLearner, ProbClassifier, ProbERTrees and FeatERTrees. As discussed in Section 3.2.2.1, ProbClassifier is trained with probabilities and assigns the class (i.e., popularity trend) with largest probability (based on the distance to the closest cluster centroid) in matrix M to a scholar. ProbERTrees predicts a class for scholars by training an extremely randomized trees learner using only probabilities as input features. FeatERTrees predicts the class of a scholar by training an extremely randomized trees learner using only the scholar features. ScholarTrendLeaner, in turn, uses both sets of features as input to the learner. As mentioned, all methods use the monitoring period optimized by ScholarTrendLearner (presented in Table 4.1), to ensure a fair comparison[6]. All results are averages over all test sets, along with 95% confidence intervals. Before discussing our popularity trend prediction results, we note that, as shown in Table 4.1, classes K_3 and K_4 require more monitoring time windows (i.e., larger value of γ_i), as these classes experience some fluctuations that may be confused as peaks. Thus, it is harder to determine whether the scholar belongs to one of those classes. On the other hand, classes K_0 and K_1 and K_2 exhibit sharper peaks, requiring somewhat shorter monitoring periods, making it easier to classify scholars in these classes.

Table 4.1. Best values for parameters θ and γ (average results across all training sets)

Cluster	θ	γ
K_0	0.279	13
K_1	0.228	13
K_2	0.226	13
K_3	0.229	14
K_4	0.226	14

Table 4.2 show the results of the four prediction approaches in terms of prediction accuracy. Since our prediction problem is a classification task, we assess prediction accuracy using the Micro and Macro F1 scores. These measures are computed based on precision and recall measures. Precision for a class K_i, $p(i)$, is the number of correctly classified scholars out of those assigned to K_i by the classifier, while the Recall of class K_i, $r(i)$, is the number of correctly classified objects that should have been classified to that class. The F1 measure

[6]All features are computed considering this optimized time window.

of class K_i, F1(i) is computed as:

$$F1(i) = \frac{2p(i) \times r(i)}{p(i) + r(i)}$$

Macro F1 is the average F1 across all classes, whereas Micro F1 is computed based on global precision and recall, calculated for all classes.

Table 4.2. Evaluation of Scholar Popularity Trend Prediction Methods (averages and 95% confidence intervals)

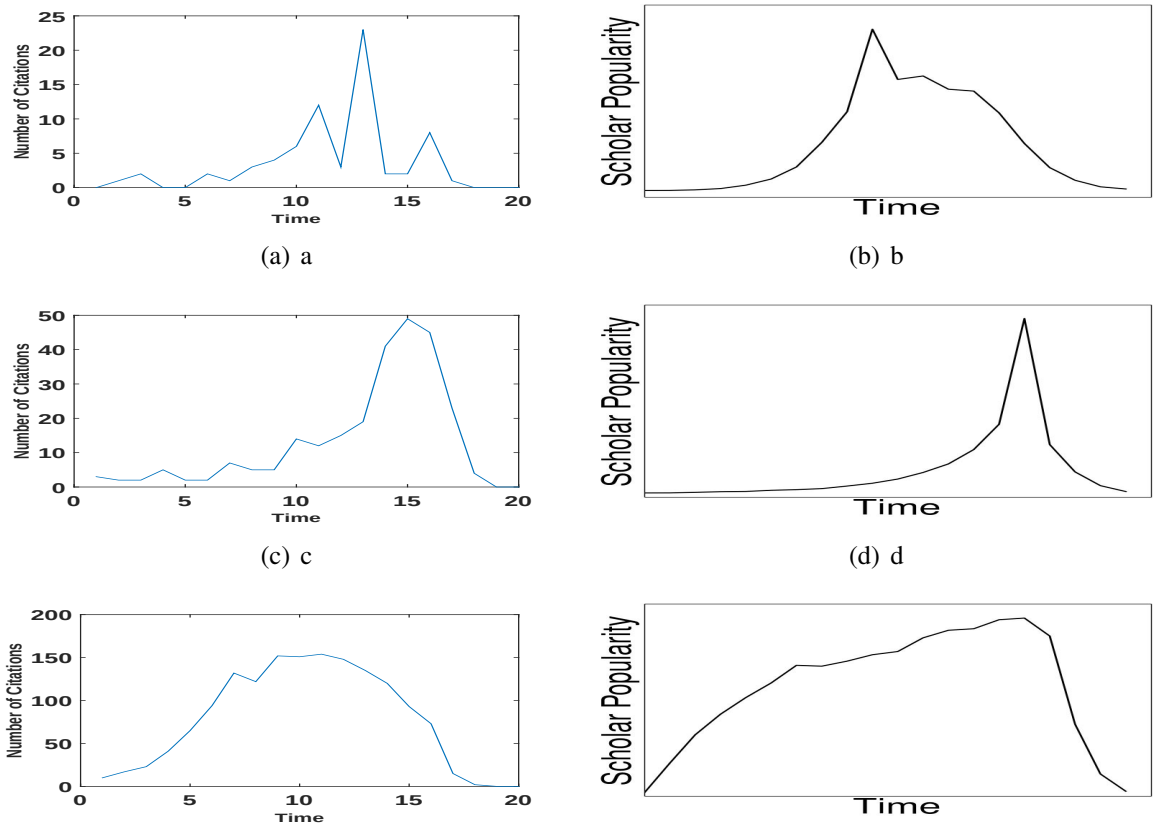
	Macro F1 Score	**Micro F1 Score**
ProbClassifier	0.743± 0.006	0.785± 0.003
ProbERTrees	0.731±0.005	0.803±0.004
FeatERTrees	0.352±0.005	0.459±0.005
ScholarTrendLearner	0.754± 0.007	0.814± 0.003

As shown in Table 4.2, ScholarTrendLearner improves the results over the other three alternative classification methods. Combining probabilities and scholar features brings explicit advantages over using either set of probabilities or scholar features separately. As shown in Table 4.2, the average improvements of ScholarTrendLearner over the other prediction approaches in Micro and Macro $F1$ reach up to 36% and 39%, respectively. In order to illustrate the results obtained with ScholarTrendLearner, Figure 4.3 shows the true popularity curve and the *predicted* trend of three example scholars. Note that the predictions, which match the correct classes, capture reasonably well the popularity dynamics of all three scholars.

As a note regarding prediction effectiveness, recall that, as discussed in Section 3.2.2, ScholarTrendLearner may not be able to properly identify the class of a scholar within the maximum monitoring period allowed (γ_{max}). In such cases, the algorithm produces no prediction result. However, we found that this happened for only a small fraction of scholars in our dataset (10%), which exhibit quite different popularity curves which could not be matched (with enough confidence) to any identified cluster.

Recall that ScholarTrendLearner was derived from TrendLearner [24], a model proposed to predict popularity trends of UGC. When comparing our results to those of the original method, we find:

- Our results have much higher Micro and especially Macro $F1$ values.



(a) a (b) b

(c) c (d) d

(e) e (f) f

Figure 4.3. True popularity curve (left) and predicted (right) for three example scholars

- Moreover, unlike observed in [24], the use of only scholar features as input to ERTrees (FeatERTrees) proved to be a much worse approach, compared to the others.

- A third difference is that the improvements of ScholarTrendLearner over using only probabilities or scholar features, though statistically significant, are somewhat less impressive than the corresponding gains of the original TrendLearner on UGC.

Such differences reflect the idiosyncrasies of each particular domain, and indicate that the adaptation of the original technique to a very different domain (the scientific one) does produce new important insights, being a significant contribution of our work.

We now turn our attention to how early the prediction of ScholarTrendLearner is made, as one of our goals is to make the predictions as early as possible. Recall that we assess this measure based on remaining popularity (citation) after the prediction. Figure 4.4 presents the complementary cumulative distribution of the fraction of remaining citations (RC) after prediction, produced for all scholars in the five test sets. In this graph, the x-axis represents the fraction of total citation counts after prediction while the y-axis represents the fraction

of scholars with remaining citations higher than the corresponding value in the x-axis. We note that for around 48% of the scholars, the prediction could only be made after all the citations had been received (remaining citations equal to 0). This reflects the diversity of popularity profiles across the scholars and indicates that the prediction task we are tackling is quite hard. However, for 21% of the scholars, ScholarTrendLearner was able to make predictions before more than 50% of their citations were still to be received. For 40% of the scholars, this fraction is still quite significant (30%). Thus, there is a significant diversity in the required monitoring periods produced by ScholarTrendLearner. We also observe that for more than 45% of scholars, ScholarTrendLearner could make predictions before half of the maximum monitoring time windows (i.e., before 10 years). Other scholars required longer monitoring periods. In sum, the diversity of the results shown in Figure 4.4, considering both remaining citations and required monitoring period confirms the necessity of personalizing the monitoring period on a per-scholar basis, which our ScholarTrendLearner approach addresses.

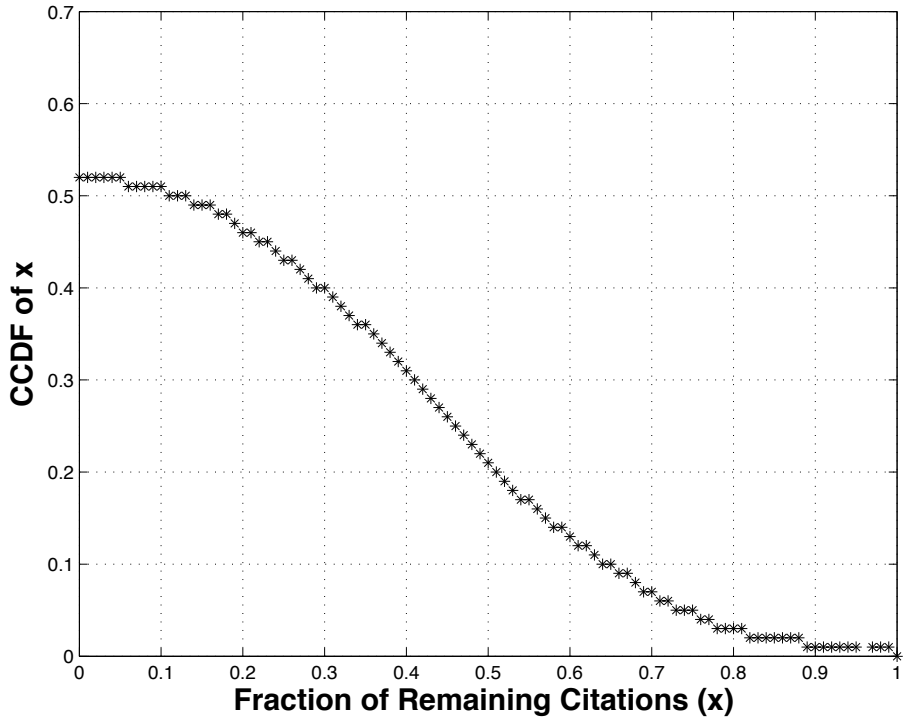


Figure 4.4. Remaining citations after prediction

4.3.1 Prediction Results

Table 4.3 shows the average mRSE along with 95% confidence intervals produced by the two prediction approaches: general and specialized. We notice that for both $\delta = 1$ and $\delta = 4$ the two MRBF models are much better than the ML results, which is consistent with [47]. We also note that, again for both values of δ, the specialized models have a tendency to generate better results than the general ones, mainly for the "specialized MRBF" vs. "general MRBF" case, though statistical superiority cannot be guaranteed, mostly due to the high variance in the results. Finally, notice that, as expected, the accuracy for predicting farther in the future is smaller than for just one year later. Yet, the errors, in general can be considered very small, meaning that our predictions are quite accurate, especially for the "specialized MBRF" model.

Table 4.3. Prediction Errors mRSE for ML and MRBF models (Averages and confidence intervals; $\delta = 1, 4$)

Regression Model	δ=1	δ=4
General ML	0.043±0.000	0.282±0.005
Specialized ML	0.040±0.006	0.268±0.047
General MRBF	0.019±0.003	0.062±0.018
Specialized MRBF	0.018±0.004	0.047±0.009

Chapter 5

Conclusions and Future Research Directions

In this dissertation, we introduced ScholarTrendLearner, a supervised prediction model that estimates the popularity trends of scholars using a combination of distance-based and associated academic features. Unlike previous work, our approach focuses on predicting popularity trends of scholars. We also focused on the natural trade-off between accurate predictions and the remaining citations after prediction.

Our main contribution is a method that aims at reducing prediction time while keeping prediction accuracy as high as possible. According to the dynamic nature of scientific popularity in scholar career's content, an efficient prediction solution is able to determine the monitoring periodd, targets and prediction dates automatically. We here provided such a solution as ScholarTrendLearner. Our method determines the monitoring time window on a per scholar basis, considering the diversity on the popularity of scholars throughout their careers.

Our experimental results show that high Macro and MicroF1 values can be obtained (above 0.75 and 0.81, respectively), with statistically significant gains over the alternative approaches. We could also achieve a good tareoff between early prediction and accuracy. For instance, we could reliably predict the popularity of more than 20% of the scholars in our dataset before 50% of the total number of citations obtained by them in their entire career is acquired. Furthermore, we concluded that combining our predicted popularity trends with two recently proposed regression based prediction models (ML and MRBF) can lead to highly accurate popularity predictions.

As possible directions for future work, we note that different academic features may affect differently the prediction effectiveness. Thus, we intend to investigate the effectiveness of ScholarTrendLearner using different subsets of features as well as possibly new academic

features. Moreover, our present study is focused only in one knowledge area: Compute Science. Scholars from different areas may exhibit different popularity evolution patterns and idiosyncrasies. Therefore, a natural follow-up study is to apply our proposed methods to other knowledge areas, including a thorough comparison of the trends discovered for each of them. Another aspect we plan to investigate is the impact of data quality issues (missing information, name ambiguity, etc) on our prediction results. We also want to study the impact in our methods of changing the target popularity metric to h-index as there is evidence (though not strong yet) that it may have better predictive power than other metrics [31].

Finally, as a long term goal we want to generalize the TrendLeaner algorithm to work in any area (not only UGC and the Scholarly domains, as we have done) in which optmizing the tradeoff between early prediction and accuracy is an important goal.

Bibliography

[1] Daniel E Acuna, Stefano Allesina, and Konrad P Kording. Future impact: Predicting scientific success. *Nature*, 489(7415):201–202, 2012.

[2] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 607–616. ACM, 2013.

[3] Marcel Ausloos. Assessing the true role of coauthors in the h-index measure of an author scientific impact. *Physica A: Statistical Mechanics and its Applications*, 422: 136–142, 2015.

[4] Gustavo EAPA Batista, Eamonn J Keogh, Oben Moses Tataw, and Vinícius MA de Souza. Cid: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28(3):634–669, 2014.

[5] Steven Bethard and Dan Jurafsky. Who should i cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 609–618. ACM, 2010.

[6] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[7] Youmna Borghol, Siddharth Mitra, Sebastien Ardon, Niklas Carlsson, Derek Eager, and Anirban Mahanti. Characterizing and modelling popularity of user-generated videos. *Performance Evaluation*, 68(11):1037–1055, 2011.

[8] Tim Brody, Stevan Harnad, and Leslie Carr. Earlier web usage statistics as predictors of later citation impact. *Journal of the American Society for Information Science and Technology*, 57(8):1060–1072, 2006.

[9] Hulsey Cason and Marcella Lubotsky. The influence and dependence of psychological journals on each other. *Psychological Bulletin*, 33(2):95, 1936.

[10] Carlos Castillo, Debora Donato, and Aristides Gionis. Estimating number of citations using author reputation. In *String processing and information retrieval*, pages 107–117. Springer, 2007.

[11] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2007.

[12] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM, 2009.

[13] Tanmoy Chakraborty, Suhansanu Kumar, Pawan Goyal, Niloy Ganguly, and Animesh Mukherjee. Towards a stratified learning approach to predict future citation counts. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 351–360. IEEE Press, 2014.

[14] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[15] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[16] Jordan D Dimitrov, Srini V Kaveri, and Jagadeesh Bayry. Metrics: journal's impact factor skewed by a single paper. *Nature*, 466(7303):179–179, 2010.

[17] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[18] Ying Ding and Blaise Cronin. Popular and/or prestigious? measures of scholarly esteem. *Information processing & management*, 47(1):80–96, 2011.

[19] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 11–20. ACM, 2010.

[20] Yuxiao Dong, Reid A Johnson, and Nitesh V Chawla. Will this paper increase your h-index?: Scientific impact prediction. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 149–158. ACM, 2015.

[21] Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152, 2006.

[22] Leo Egghe. The hirsch index and related impact measures. *Annual review of information science and technology*, 44(1):65–114, 2010.

[23] Alan Fersht. The most influential journals: Impact factor and eigenfactor. *Proceedings of the National Academy of Sciences*, 106(17):6883–6884, 2009.

[24] Flavio Figueiredo, Jussara M Almeida, Marcos André Gonçalves, and Fabrício Benevenuto. Trendlearner: Early prediction of popularity trends of user generated content. *arXiv preprint arXiv:1402.2351*, 2014.

[25] Tove Faber Frandsen and Jeppe Nicolaisen. Effects of academic experience and prestige on researchers' citing behavior. *Journal of the American Society for Information Science and Technology*, 63(1):64–71, 2012.

[26] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[27] Eugene Garfield. Impact factors, and why they won't go away. *Nature*, 411(6837): 522–522, 2001.

[28] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[29] Nadav Golbandi Golbandi, Liran Katzir Katzir, Yehuda Koren Koren, and Ronny Lempel Lempel. Expediting search trend detection via prediction of query counts. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 295–304. ACM, 2013.

[30] Glauber D Gonçalves, Flavio Figueiredo, Jussara M Almeida, and Marcos A Gonçalves. Characterizing popularity: A case study in the computer science research community. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 57–66. IEEE Press, 2014.

[31] Jorge E Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences of the United States of America*, 102(46): 16569–16572, 2005.

[32] M Aminul Islam, Derek Eager, Niklas Carlsson, and Anirban Mahanti. Revisiting popularity characterization and modeling of user-generated videos. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, pages 350–354. IEEE, 2013.

[33] Jong Gun Lee, Sue Moon, and Kave Salamatian. An approach to model and predict the popularity of online contents with explanatory factors. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 623–630. IEEE, 2010.

[34] Loet Leydesdorff. How are new citation-based journal indicators adding to the bibliometric toolbox? *Journal of the American Society for Information Science and Technology*, 60(7):1327–1336, 2009.

[35] Cynthia Lokker, K Ann McKibbon, R James McKinlay, Nancy L Wilczynski, and R Brian Haynes. Prediction of citation counts for clinical articles at two years using data available within three weeks of publication: retrospective cohort study. *BMJ*, 336 (7645):655–657, 2008.

[36] Yasuko Matsubara, Yasushi Sakurai, B Aditya Prakash, Lei Li, and Christos Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 6–14. ACM, 2012.

[37] Amin Mazloumian. Predicting scholars' scientific impact. *PloS one*, 7(11):e49246, 2012.

[38] Daniel A Menascé and Virgilio Almeida. *Capacity Planning for Web Services: metrics, models, and methods*. Prentice Hall PTR, 2001.

[39] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[40] Cristiano Nascimento, Alberto HF Laender, Altigran S da Silva, and Marcos André Gonçalves. A source independent framework for research paper recommendation. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 297–306. ACM, 2011.

[41] Masoumeh Nezhadbiglari, Marcos André Gonçalves, and Jussara M. Almeida. Early prediction of scholar popularity. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, June 19 - 23, 2016*, pages 181–190, 2016.

[42] Stanislav Nikolov. *Trend or no trend: a novel nonparametric method for classifying time series*. PhD thesis, Twitter Inc, 2012.

[43] Miranda Lee Pao. Term and citation retrieval: A field study. *Information Processing & Management*, 29(1):95–112, 1993.

[44] Orion Penner, Raj K Pan, Alexander M Petersen, and Santo Fortunato. The case for caution in predicting scientists' future impact. *arXiv preprint arXiv:1304.0627*, 2013.

[45] Orion Penner, Raj K Pan, Alexander M Petersen, Kimmo Kaski, and Santo Fortunato. On the predictability of future impact in science. *Scientific reports*, 3, 2013.

[46] Gabriel Pinski and Francis Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing & Management*, 12(5):297–312, 1976.

[47] Henrique Pinto, Jussara M Almeida, and Marcos A Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 365–374. ACM, 2013.

[48] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.

[49] Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Traffic in social media i: paths through information networks. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 452–458. IEEE, 2010.

[50] Pentti Riikonen and Mauno Vihinen. National research contributions: A case study on finnish biomedical research. *Scientometrics*, 77(2):207–222, 2008.

[51] Michael Schreiber. To share the fame in a fair way, hm modifies h for multi-authored manuscripts. *New Journal of Physics*, 10(4):040201, 2008.

[52] Xiaolin Shi, Jure Leskovec, and Daniel A McFarland. Citing for high impact. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 49–58. ACM, 2010.

[53] Robert H Shumway and David S Stoffer. Time series analysis and its application with r examples. *University of California, Davis, CA*, 2006.

[54] Yang Sun and C Lee Giles. *Popularity weighted ranking for academic digital libraries*. Springer, 2007.

[55] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[56] Athena Vakali, Maria Giatsoglou, and Stefanos Antaris. Social networking trends and dynamics detection via a cloud-based framework design. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1213–1220. ACM, 2012.

[57] David van Dijk, Ohad Manor, and Lucas B Carey. Publication metrics and success on the academic job market. *Current Biology*, 24(11):R516–R517, 2014.

[58] Taras K Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, 4(1):52–57, 1968.

[59] Erjia Yan and Ying Ding. Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, 60(10):2107–2118, 2009.

[60] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. Citation count prediction: Learning to estimate future citations for literature. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1247–1252. ACM, 2011.

[61] Rui Yan, Congrui Huang, Jie Tang, Yan Zhang, and Xiaoming Li. To better stand on the shoulder of giants. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 51–60. ACM, 2012.

[62] Shiyan Yan and Carl Lagoze. Understanding the relationship between scholars' breadth of research and scientific impact. *iConference 2015 Proceedings*, 2015.

[63] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177–186. ACM, 2011.

[64] Mohammed J Zaki and Wagner Meira Jr. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[65] Michel Zitt. Citing-side normalization of journal impact: A robust variant of the audience factor. *Journal of Informetrics*, 4(3):392–406, 2010.