February, 2012

A Boosted-Trees Method for Name Disambiguation

Jian Wang, *Georgia Institute of Technology - Main Campus*
Kaspars Berzins, *Georgia Institute of Technology - Main Campus*
Diana Hicks, *Georgia Institute of Technology - Main Campus*
Julia Melkers, *Georgia Institute of Technology - Main Campus*
Fang Xiao, *Georgia Institute of Technology - Main Campus*, et al.

A boosted-trees method for name disambiguation

Jian Wang, Kaspars Berzins, Diana Hicks, Julia Melkers, Fang Xiao, & Diogo Pinheiro. (2012). A boosted-trees method for name disambiguation. *Scientometrics*, *93*(2), 391-411. doi: 10.1007/s11192-012-0681-1

Jian Wang, Kaspars Berzins, Diana Hicks, Julia Melkers, Fang Xiao, Diogo Pinheiro

School of Public Policy, Georgia Institute of Technology, 685 Cherry Street, Atlanta, GA 30332-0345, USA

Phone: +1 404 585 0523

Fax: +1 404 385 0504

jianwang@gatech.edu

kberzins@gatech.edu

dhicks@gatech.edu

jmelkers@gatech.edu

fxiao3@gatech.edu

diogo.pinheiro@pubpolicy.gatech.edu

Abstract: This paper proposes a method for classifying true papers of a set of focal scientists and false papers of homonymous authors in bibliometric research processes. It directly addresses the issue of identifying papers that are not associated ("false") with a given author. The proposed method has four steps: name and affiliation filtering, similarity score construction, author screening, and boosted trees classification. In this methodological paper we calculate error rates for our technique. Therefore, we needed to ascertain the correct attribution of each paper. To do this we constructed a small dataset of 4,253 papers allegedly belonging to a random sample of 100 authors. We apply the boosted trees algorithm to classify papers of authors with total false rate no higher than 30% (i.e., 3,862 papers of 91 authors). A one-run experiment achieves a testing misclassification error 0.55%, testing recall 99.84%, and testing precision 99.60%. A 50-run experiment shows that the median of testing classification error is 0.78% and mean 0.75%. Among the 90 authors in the testing set (one author only appeared in the training set), the algorithm successfully reduces the false rate to zero for 86 authors and misclassifies just one or two papers for each of the remaining four authors.

Keywords: *Name disambiguation, Common names, Classification tree, Boosted trees*

Introduction

Since the pioneering work of Cole and Eales (1917) and the advent of Eugene Garfield's Science Citation Index, interest in and use of bibliometrics has grown steadily. In previous decades, computational and data limits meant that work often focused on the national, field or institutional level. However recently attention has turned to the individual level, as evidenced by the attention paid to new measures such as the H-index and strong interest in studying social networks in science (Newman 2001; Hirsch 2005). Accurate analysis at the individual level requires that an individual's oeuvre be correctly identified. This is a challenging problem for a number of reasons. This paper addresses these challenges by developing advanced statistical techniques to identify clean author paper sets.

Clean sets of publications written by focal scientists are typically retrieved by querying on author names. However, there are potential problems: first, a single author may publish under several different names. For example, women in particular may change their names, but others may vary the use of initials or middle names. A second issue concerns common names, that is, different authors may have the same name. On the other hand, the limits of databases exacerbate the difficulty of differentiating names. For example, until 2006 Thomson Reuters Web of Science (WoS) recorded first and middle initials rather than full names, reducing the number of unique names in the data. To deal with these problems, name matching and name disambiguation processes are needed (Kang et al. 2009). This paper seeks to improve name disambiguation and views name matching as a separate process which must be completed before name disambiguation can begin.

The homonymous name problem has recently drawn a lot of attention. Aksnes (2008) showed that 14% (4,362 out of 31,135) Norwegian authors would share the same name if listed in the WoS style. Radicchi et al. (2009) examined the Physical Review publication archive (1893-2006) and estimated that in 92% of the cases, a unique name corresponds to a single author, in other words, 8% of names correspond to multiple individuals. This problem is even more severe for countries where some last names are extremely common, for example, China, Japan, and Korea (Moed 2005; Tang and Walsh 2010). The homonymous name problem challenges not only the validity of individual evaluation, but also the accuracy of the social network analysis. Strotmann et al. (2009) showed that collaboration networks are significantly different before and after name disambiguation.

A number of methods have been proposed for name disambiguation. This study proposes a boosted trees method to discriminate true papers of the focal scientist from false papers of other homonymous authors. The method has four steps: name and affiliation filtering, similarity score construction, author screening, and boosted trees classification. Applying the boosted trees algorithm on low false rate authors' papers (i.e., 3,862 papers of 91 authors), a one-run experiment achieves a testing misclassification error of 0.55%, testing recall of 99.84%, and testing precision of 99.60%. The testing misclassification error of a 50-run experiment has a median 0.78% and mean of 0.75%. The performance of this approach thus exceeds that of previously proposed methods. However, it cannot automatically classify high false rate authors' papers and leaves them to manual checking.

Existing methods

There are generally two ways to approach name disambiguation: one is to treat it as a clustering problem and partition papers of different homonymous authors into clusters of papers of single individuals, and the other is to treat it as a binary classification problem and decide whether each paper is truly written by the focal author.

Many clustering methods have been proposed for name disambiguation, including K-way spectral clustering model (Han et al. 2005b), the SVM-DBSCAN model (Huang et al. 2006), stochastic graph partitioning (Kanani and McCallum 2007; Kanani et al. 2007), heuristic-based hierarchical clustering (Cota et al. 2010), and so on.

Clustering methods rely on pairwise similarities between papers, which can be obtained by either supervised or unsupervised methods. Supervised approaches typically start with building a binary classifier to predict whether two papers are written by the same author, and then use the classifier as the pairwise similarity matrix for clustering. Many classifiers have been used, such as naïve Bayes classifier (Burges 1998; Han et al. 2004), support vector machine (Huang et al. 2006; Yang et al. 2006), maximum entropy (Kanani and McCallum 2007; Kanani et al. 2007), and error-driven training (Culotta et al. 2007). On the other hand, unsupervised approaches develop algorithms to predict the probability that a pair of papers belong to the same individual based on the profile similarity between these two papers without training (Torvik et al. 2005; Torvik and Smalheiser 2009), or uses similarity coefficients, such as counts of common names/words, Jaccard, Jaro, Jaro-Winkler, and cosine, to construct the pairwise similarity matrix directly from some features of the paper (Liben-Nowell and Kleinberg 2007; On et al. 2005; Yin et al. 2007; Tang and Walsh 2010).

Both supervised and unsupervised methods build the similarity matrices indirectly or directly from features of papers and authors. Some studies found that coauthor information was very effective (Torvik et al. 2005; Kang et al. 2009). Tang and Walsh (2010) viewed referencing as a fingerprint of author's cognitive knowledge base and used citation information for name disambiguation. Other features used include: first/middle names, paper title, journal name, words on the first page of paper, commonness of the author name, research themes, funding organizations, and so on (Han et al. 2004; Culotta et al. 2007; Huang et al. 2006; Wooding et al. 2006; Kanani et al. 2007; Han et al. 2005a; Han et al. 2005b; Song et al. 2007).

Taking advantage of more features can improve the algorithm, but clustering requires combining all the feature similarity scores into one single numeric value measuring the overall distance between every two articles. To combine features, some took a linear combination of weighted feature similarity values (Aswani et al. 2006; Lee et al. 2005), and some used the geometric average (Yin et al. 2007). However, taking a weighted average may have problems because of the redundancy or interactive effects in features (Torvik et al. 2005). Several solutions are proposed to deal with this problem, such as probabilistic latent semantic analysis (Hofmann 1999; Song et al. 2007) and latent Dirichlet allocation (Bhattacharya and Getoor 2006, 2007; Blei et al. 2003; Song et al. 2007).

Another challenge confronting clustering is the transitivity problem, that is, different pairwise similarity judgments may give conflicting results. For instance, two papers with very low similarity scores may both have very high similarity score to another paper. A small rate of violation of transitivity may break down the agglomerative clustering. This problem can be avoided by using triplet or higher order comparison among papers to bring in information beyond pairwise comparisons (McCallum and Wellner 2003; Kanani et al. 2007).

Many studies also tried to bring in external information for a better disambiguation. To calculate pairwise similarity, some studies searched for personal publication webpages with both papers (Aswani et al. 2006; Kanani and McCallum 2007; Kanani et al. 2007), some compared two Uniform Resource Locations (URLs) of two webpages containing each paper (McRae-Spencer and Shadbolt 2006; Tan et al. 2006; Yang et al. 2006), and some brought in extra external constraints, for example, an author is unlikely to publish more than 30 papers in a given year (Culotta et al. 2007). Smalheiser and Torvik (2009) provide a useful review of clustering methods proposed for name disambiguation.

The second way to formulate the name disambiguation problem is to treat it as a binary classification problem in which the aim is to discriminate between true papers of a targeted author and false papers of homonymous authors. This approach takes the full set of papers listing the name of a focal author and tries to identify a clean set of papers truly written by the focal author.

D'Angelo et al. (2011) proposed a heuristic approach to map papers to scientists in the Italian university system. They had an external database of over 60,000 scientists in the Italian university system with their name, scientific disciplinary sector (SDS), university, department, and official academic rank information available, and another database of 2001-2007 publications listed in the Thomson Reuters Italian National Citation Report (I-NCR). They mapped papers to focal scientists with several filtering processes using address, WoS-SDS, and shared SDS as filters. The precision and recall of their algorithm was 96.4% and 94.3% respectively.

Onodera et al. (2011) firstly identified 2,595 "source" authors from 629,000 retrieved "source" papers, and then retrieved WoS papers listing the "source" author names (last and first initial). To discriminate between true and false papers, they firstly filtered papers by affiliation address, and then used logistic regression to predict whether the paper was true, using variables measuring how similar the paper was to the "source" paper of the focal scientist. Their algorithm achieved a recall and a precision both about 95%.

These two studies achieve better recall and precision than most clustering approaches. Important for this improved performance are three conditions: First, the studies use an external database about focal scientists (Italian university system database and "source" papers). Second the binary classification problem itself is simpler than clustering, that is, binary classification only needs to make a true or false decision while clustering has to determine the number of clusters and assign papers to a number of candidate clusters. Third, clustering approach confronts more complex situation where several authors are very productive while many are unproductive, while classification for focal authors deals with a relatively more homogenous dataset.

Proposed approach

This study takes the second approach, namely the binary classification approach. It is motivated by our studies on research collaboration/co-authorship networks and attempts to identify a clean publication set for each focal scientist to allow reliable network analyses. The proposed method predicts whether a paper is a true paper of our focal scientist or a false paper of homonymous authors. The real world application of our method to 54,853 papers of 1,315 authors is described in Appendix 1. In this methodological paper we calculate error rates for our technique. Therefore, we needed to independently ascertain the correct attribution of each paper. To do this we constructed a small development dataset of 4,253 papers allegedly belonging to a random sample of 100 American scientists who are part of the larger study. The work described here cleans the development data set and assesses the performance of the algorithms against the independent human judgment of each paper's true status.

Among these 100 authors, we observe several very common English and Chinese last names (such as Smith and Wang). As expected, even after author name and affiliation filtering, the resulting paper set still has a very high false rate. The term *"false rate"* of an author, throughout this paper, refers to the percentage of papers under his/her name that actually were written by somebody else.

To create a cleaned and usable bibliometric data set for each focal author, we designed a four-step method: The first step is downloading paper records from WoS using a name and affiliation match. The second step constructs the pairwise cosine similarity matrices for all papers under each focal author name, and then uses Eigen-decomposition and averaging to construct a variety of numerical similarity scores to measure the distance between one paper and all other papers under the same name. The method assumes that papers written by the same author have characteristics in common while papers written by different authors are less likely to be similar to each other. Affiliation filtering in the first step yields a set of papers with a majority likely to be true, so we can assume that true papers will show a higher similarity to all other papers, while false papers will have lower similarity to all other papers. This difference makes possible classification. In the third step we eliminate a small number of authors with very high false rates. In the fourth step we use boosted trees for classifying true and false papers for the remaining authors.

The effectiveness of the boosted trees classification relies on the assumption that the majority of papers under one author name are true ones, so we only use the boosted trees algorithm for classifying papers of authors with false rate less than 30%. Therefore, an author screening model (step 3) is constructed to decide whether each author is in the low false rate group. Authors in the low false rate group will be included and their papers will be automatically classified by boosted trees, while authors in the high false rate group are excluded and their papers have to be manually cleaned.

The procedure of our method is summarized in Fig. 1. This paper will explain step 1, 2, and 4 (name and affiliation filtering, similarity score construction, and boosted trees classification) first, and then introduce the motivation and design of step 3 (author screening).

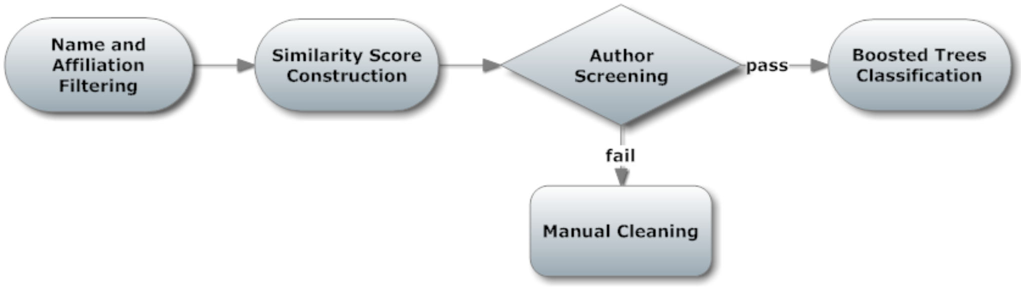

Name and Affiliation Filtering
Similarity Score Construction
Author Screening
pass
Boosted Trees Classification
fail
Manual Cleaning

Fig. 1 Proposed procedure

Name and affiliation filtering

In real-world evaluation practice, external databases concerning focal authors are typical available before collecting bibliometric data. In our context, we have the whole affiliation history information of each author available. In the data retrieval stage, we required a name match (first initial and last name match) and affiliation match (at least one of the author affiliations in WoS record matches one of the affiliations in the focal scientist’s history and publication year not before his/her active career). 4,253 papers matched in total for all 100 authors.

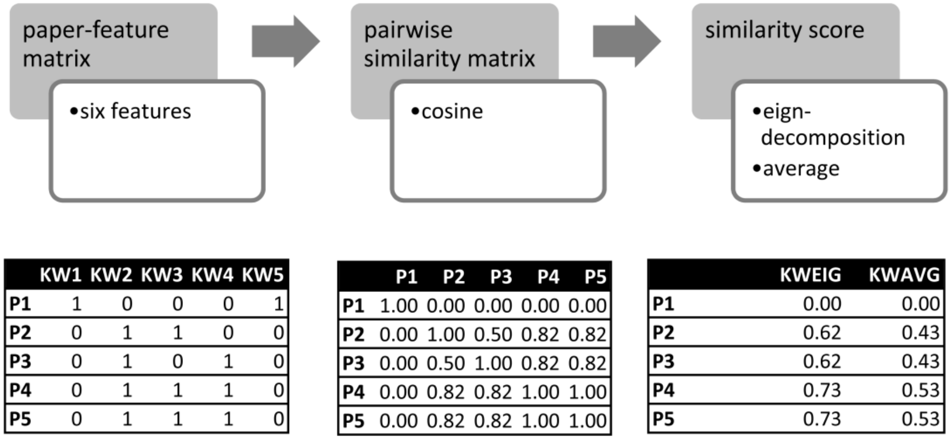
Similarity score construction

Six paper features available in WoS are chosen to measure similarities:

- names of authors
- cited journals
- combined keywords (author keywords + WoS added keywords)
- title words (excluding stop-words)
- abstract words (excluding stop-words)
- subject category

Choice of these features is based on previous literature and data availability in WoS. Affiliation is found to be of particular importance for name disambiguation, but not included here, because we have used affiliation filtering in the first step. Authors with the same name are more likely to be one person, if they coauthor with same coauthors. Coauthor name has been found to be effective for name disambiguation, so we use the "names of authors" information available in WoS. Based on the idea that papers written by the same author should share something similar, a variety of aspects have been used in previous methods. For example, same author may tend to cite the same literature, and references information is widely used in name disambiguation practice. We use "cited journal" but not cited references at the article level, because the former information is much cleaner and more ready for analysis in WoS and the similarity matrix will be too sparse if using cited reference at the article level. Furthermore, many studies try to extract content/meaning information from articles and then measure their content similarities, and features investigated include title, words on the first page, and so on. Following this logic, we take advantage of well-structured "combined keywords," "title words," and "abstract words" data from WoS. Finally, similarity in research themes would help disambiguation, so we use "subject category."

For names of authors, cited journals, combined keywords, title words (excluding stop-words), and abstract words (excluding stop-words), the procedure of constructing similarity score is presented in Fig. 2.



	KW1	KW2	KW3	KW4	KW5
P1	1	0	0	0	1
P2	0	1	1	0	0
P3	0	1	0	1	0
P4	0	1	1	1	0
P5	0	1	1	1	0

	P1	P2	P3	P4	P5
P1	1.00	0.00	0.00	0.00	0.00
P2	0.00	1.00	0.50	0.82	0.82
P3	0.00	0.50	1.00	0.82	0.82
P4	0.00	0.82	0.82	1.00	1.00
P5	0.00	0.82	0.82	1.00	1.00

	KWEIG	KWAVG
P1	0.00	0.00
P2	0.62	0.43
P3	0.62	0.43
P4	0.73	0.53
P5	0.73	0.53

Fig. 2 Similarity score construction

Take "combined keywords" as example, we firstly take all papers under one author, build a matrix (paper-feature matrix) between paper and all keywords used, the ijth element is the number of times the ith paper used jth keyword.

Second, based on paper-feature matrix, we calculate the cosine similarity coefficient between each pair of papers, COS$_{ij}$, which is defined as:

$$COS_{\mathrm{ij}} = \frac{\mathrm{P_i} \cdot \mathrm{P_j}}{\|\mathrm{P_i}\| \|\mathrm{P_j}\|}$$

where P$_i$ is the ith row vector in the paper-feature matrix.

The last step is to project the pairwise similarity matrix into a single numerical similarity score for each paper, indicating its distance to all other papers under the same author name. To project distances into single dimension, we use Principal Coordinates Analysis, a metrical multidimensional scaling method based on the Principal Component Analysis (Johnson and Wichern 2007). That is, distances are projected on the eigenvector of the largest eigenvalue to give the KWEIG measure. Furthermore, in cases when author has diverse publishing profile and KWEIG values wouldn't let us differentiate between true and false papers, we make an alternative projection on the unity vector to give the KWAVG measure.

KWAVG for ith paper is calculated as:

$$KWAVG_i = \frac{1}{n-1}\sum_{j \neq i}^{n} C_{ij}$$

where n is the total number of papers of the author, and C_{ij} is the ij^th element of the pairwise similarity matrix.

KWEIG for i^th paper is calculated as:

$$KWEIG_i = \left|V_i \times E_1\right| \times \left(\frac{\max\limits_{i \in \{1,2,\ldots,n\}} \left\{KWAVG_i\right\}}{\max\limits_{i \in \{1,2,\ldots,n\}} \left\{\left|V_i \times E_1\right|\right\}} \right)$$

where V_i is the pairwise similarity vector of the i^th paper and E_1 is the eigenvector associated with the largest eigenvalue of the pairwise similarity matrix of the author. To normalize the KWEIG value for each paper to between 0 and 1, the signs are removed and transformation is further multiplied by the ratio between $\max\limits_{i \in \{1,2,\ldots,n\}} \left\{KWAVG_i\right\}$ (the largest average similarly score for the author) and $\max\limits_{i \in \{1,2,\ldots,n\}} \left\{\left|V_i \times E_1\right|\right\}$ (the largest Eigen-decomposition value of the similarity vectors for the author).

For subject categories the pairwise similarity matrix is constructed using additional information from the *Similarity Matrix among Subject Categories* constructed by Porter and Rafols (2009), which contains correlations of subject categories as cosines between their co-citation vectors. We define the similarity between two articles as the maximum correlation between their subject categories. Therefore, the ij^th element of the pairwise similarity matrix C_{ij} is calculated as $\max_{k,l} \left\{corr\left(SC_{ik}, SC_{jl}\right)\right\}$, where $k \in \left\{1,2,\ldots, number\ of\ SC\ for\ article\ i\right\}$, $l \in \left\{1,2,\ldots, number\ of\ SC\ for\ article\ j\right\}$, and SC_{im} is the m^th subject category of the i^th article. Then the same procedure as for other features is used to construct a single numerical similarity score from the pairwise similarity matrix using averaging and Eigen-decomposition.

This process gives us twelve similarity scores for each paper:

- names of authors - AUEIG, AUAVG
- cited journals - CJEIG, CJAVG
- combined keywords - KWEIG, KWAVG
- title words - TIEIG, TIAVG
- abstract words - ABEIG, ABAVG
- subject category - SCEIG, SCAVG

In addition to the twelve similarity scores, we supply several other paper level and author level variables for classification:

Paper level:

- number of authors - AUNO
- number of author affiliations - AFNO

Author level:

- number of papers - PUBNO
- FIELD, author's research field: BIOL, CHEM, CS, EAS, or EE
- ASIAN: 1 if surname is Chinese, Korean, or Japanese, 0 otherwise
- Surname commonness - LASTC: use the occurrence frequency count of surnames from 2000 U.S. Census (U.S. Census Bureau 2000)

Boosted trees classification

The boosted trees method is used for classification. Tree-based methods were first proposed by Breiman (1984). The idea is to partition the feature space into two or more sub-regions, and continuously split each sub-region into finer sub-regions. Eventually the whole feature space is partitioned into many fine regions, and all points in the same region are classified as the same class, which is the observed majority class in that region. Two fundamental issues in the classification tree method are how to decide the splitting point (tree growing) and how to control the size of the tree (tree pruning). In tree growing, the analyst chooses to minimize misclassification error, Gini index, or cross-entropy, and the algorithm builds a tree with features and splitting points that minimize the chosen criterion. Tree pruning criteria aim to minimize penalized misclassification error. That is, the objective function to be minimized is not merely the misclassification error, but the misclassification error plus a penalty on tree size. The weight of penalty is typically chosen by cross validation.

Fig. 3 provides a simple illustration. We have 100 observations belonging to two different classes: "in" and "out." We plot all these observations onto a two-dimensional space defined by SCEIG and AUEIG and try to classify these observations by their SCEIG and AUEIG values. A binary classification tree is constructed by partitioning the feature space repeatedly. To minimize misclassification error, we would predict all points in one sub-region as one class, which is the class of the majority points in that region. Therefore, at step 0, we classify all points as "in." At step 1, we try to split the whole feature space by a vertical or horizontal line, and then predict the class of each sub-region using majority rule. Among all possible vertical and horizontal lines, the horizontal line (AUEIG=0.48) minimizes the total misclassification error, so it is selected at step 1. The same procedure is repeated at step 2 and 3. Eventually, we establish a three-step tree to split the whole feature space into four sub-regions and predict all points in the top sub-region as “in,” all points in the bottom left sub-region as “out,” and so on.

The constructed tree can also be presented by the decision tree in Fig. 3. Classification decision follows the procedure laid out by this tree, that is, if one observation has a AUEIG value no less than 0.48, then we predict it as “in,” if not, we go to the next step, if its SCEIG is less than 0.53, then we predict it as “out,” and so on.

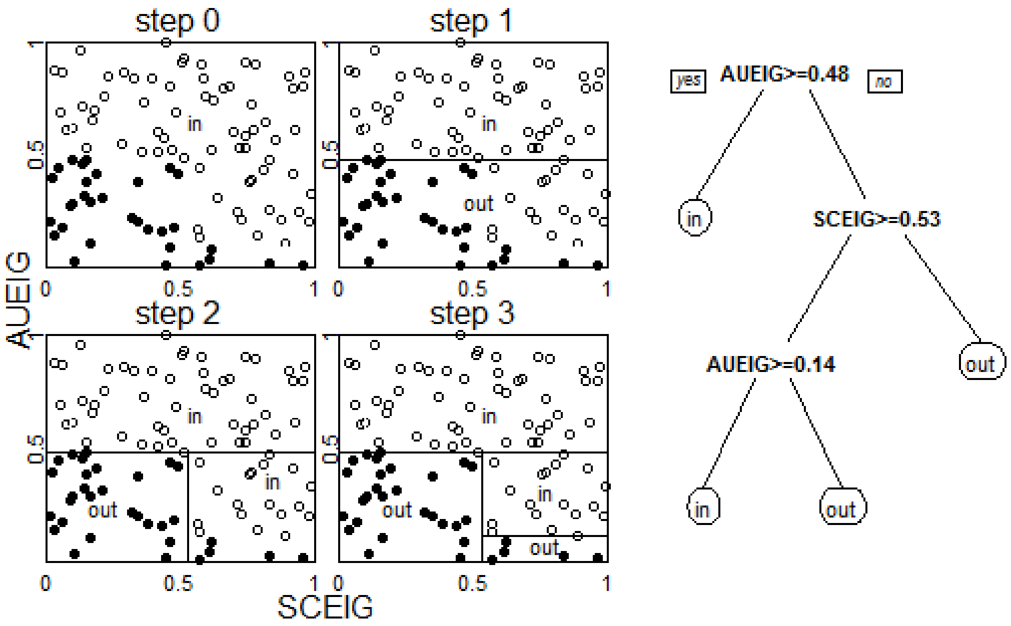


Fig. 3 Classification tree illustration

One extreme is to split the whole feature space into so many sub-regions that each region only has one observation; in this case the total misclassification error would be zero. However, this solution would suffer from the over-fitting problem and has poor prediction power. To address this problem, a penalty on tree size is added into the objective function. That is, to minimize penalized misclassification error instead of only misclassification error. Thanks to Therneau and Atkinson (2010), the “rpart” package is available in R to implement the classification tree method.

The boosted trees method further adds an ensemble layer on top of the classification tree, that is, it sequentially fits a classification tree to reweighted samples of the training data and then takes a weighted majority vote of the sequence of classifications trees for prediction (Friedman et al. 2000). With a committee of a number of trees, boosted trees can achieve lower misclassification errors and higher stability than one single tree. We use the "ada" package in R (Culp et al. 2010) to implement this algorithm, chose the committee size to be 50 trees, and use Gini index as the tree growing criterion. Please refer to Hastie et al. (2009) for an introduction to tree-based methods and boosted trees.

Implementation and results

To use the boosted trees method, we need a training set to train the algorithm before applying it to a dataset. Because in this paper we also are evaluating the algorithm's performance, we apply the algorithm to a small development dataset for which we have manually checked all the 4,253 papers of the focal 100 authors. We randomly assign 2,835 of these papers for training (2/3 of the set) and the remaining 1,418 papers for testing. In a real world application, only the training set would need to be manually checked so the size of the dataset to which the trained algorithm can be applied is limited only by computational capacity.

We code true papers written by the focal scientist as "in," and false papers of homonymous authors as "out." The overall false rate is 7.69%, which is very low because we have already filtered papers by affiliations. However, the false rate varies across scientists: most scientists are lower than 10% while several are as high as 60% (Fig. 4). Therefore, the goal is not only to reduce the overall false rate, but more importantly to reduce the false rate for each individual and remove the structural biases across individuals.

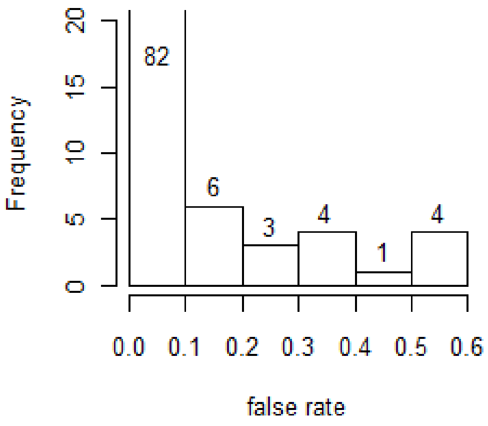


Fig. 4 Histogram of false rate by author

Pre-run results

Exploratory analysis on our development dataset of 4,253 papers published by 100 authors support our assumption that true papers have higher similarity scores, while false ones have very low similarity scores. Fig. 5 illustrates the classification tree that we would use if we were to use one tree for classification. The algorithm in R package “rpart” automatically constructs this tree. This tree recursively split the data into seven regions and predicted all data points in one region to be “in” or “out.” This tree shows that similarity scores are very relevant for classification, and the tree partitions the space at very low values of certain similarity scores many times.

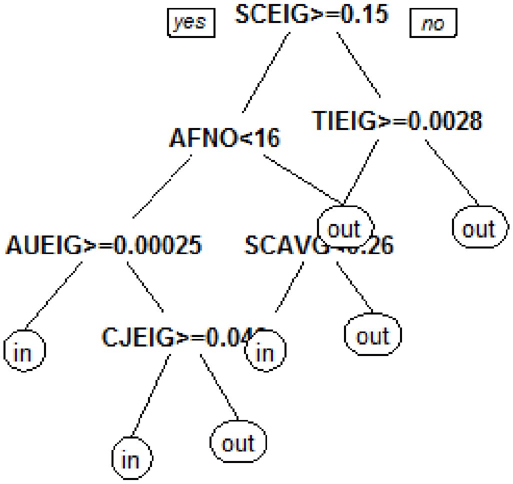


Fig. 5 One tree for paper classification

Although the basic tree approach would stop with this one tree, with the boosted trees approach, we fit 50 trees, and the prediction is based on the weighted majority vote of these 50 trees. For the boosted trees model, the testing recall is 99.69%, the testing precision is 98.79%, and the overall testing misclassification error rate is 1.41%. This is higher than reported previously in the bibliometric literature (D'Angelo et al. 2011; Onodera et al. 2011).

Application scope redefinition

The power of boosted trees classification relies on the assumption that the majority of papers under one author name are true papers while only a few are false. Ideally, we would have authors who publish many similar articles while having only several very different false papers, but this might not always be the case in reality. Our classification algorithm may not have strong power to classify papers of one author with a very high false rate, because the contrast between true and false papers is not sufficient for discrimination. Furthermore, keeping authors with high false rates may also reduce the prediction power on low false rate authors, because high false rate authors' papers may "contaminate" the training set and "mislead" the algorithm. Therefore, one important challenge is to redefine the application scope of this algorithm, that is, to identify the maximum author false rate we can tolerate. In other words, *"application scope"* in this paper means the maximum false rate (author level) allowed for including the author into analysis. For example, application scope 10% means we include only authors with false rate no more than 10% and use only these authors' papers for training and testing. Confronting this decision there is a trade-off: on the one hand, we would like to include only authors with very low false rates to improve the classification accuracy; on the other hand, we would like to include authors with high false rates to save us from time-consuming manual cleaning.

Two criteria are used to balance this trade-off and redefine the application scope: (1) total misclassification error, and (2) power to eliminate bias across authors. Table 1 compares the overall performance of boosted trees on different application scopes. Because the highest author level false rate in our random sample is less than 60%, the 60% group actually refers to the whole dataset of 100 authors' 4,253 papers. We use the previously randomly sampled 2,835 papers for training and the rest 1,418 for testing. Then we reduce the scope step by step. For the 50% scope group, we exclude all papers of 4 authors whose false rates are higher than 50% from both the training and testing set, and only use the remaining training and testing set for analysis. In this table we cannot see a clear decreasing trend of misclassification error as application scope decreases. However, this is just one experiment. Results might be different if we take another random sample of training and testing sets. In addition, it is also important to evaluate the variance of error rates, so that a multiple-run experiment is required.

Table 1 Application scopes comparison

		60%		50%		40%		30%		20%		10%	
		in	out	in	out	in	Out	in	out	in	out	in	Out
Train truth	in	2621	0	2541	2	2537	2	2504	1	2391	0	2294	0
	out	8	206	7	93	14	82	4	77	6	34	3	21
Test truth	in	1301	4	1245	3	1242	4	1231	2	1160	2	1099	0
	out	16	97	15	42	16	39	5	38	6	17	6	6
Train	Author#		100		96		95		91		88		82
	Mis.error		0.28%		0.34%		0.61%		0.19%		0.25%		0.13%
	Recall		100.00%		99.92%		99.92%		99.96%		100.00%		100.00%
	Precision		99.70%		99.73%		99.45%		99.84%		99.75%		99.87%
Test	Author#		99		95		94		90		87		81
	Mis.error		1.41%		1.38%		1.54%		0.55%		0.68%		0.54%
	Recall		99.69%		99.76%		99.68%		99.84%		99.83%		100.00%
	Precision		98.79%		98.81%		98.73%		99.60%		99.49%		99.46%

For each application scope, testing author number is one less than training author number, because there is one author, all whose five papers are sampled in the training set, none in the testing set. "Mis. error": misclassification error rate, that is, the fraction of all papers that are misclassified, for example, testing error of "60%" group is (4+16)/(1301+4+16+97)= 1.41%; "Recall": the fraction of truly-"in" papers that are classified as "in," for example, testing recall of "60%" group is 1301/(1301+4)=99.69%; "Precision": the fraction of classified as "in" papers that are truly "in," for example, testing precision of "one tree prediction" is 1301/(1301+16)=98.79%.

For the multiple-run experiment, for each application scope, we randomly select 2/3 of the papers for training and the rest for testing, one boosted-trees model is fitted, and training and testing misclassification errors are calculated. This process is repeated 50 times to generate 50 training and testing misclassification error observations. Therefore, we can get the empirical distribution of misclassification error at each scope level (Fig. 6). There is roughly an increasing trend of misclassification error as the scope level increases. Among these six groups, we can expect to achieve the lowest misclassification error if we only work on authors with false rate no more than 10%. However, the performance of the 30% scope group is adequate for our project. At this level, the error rate is about 0.75% on average.

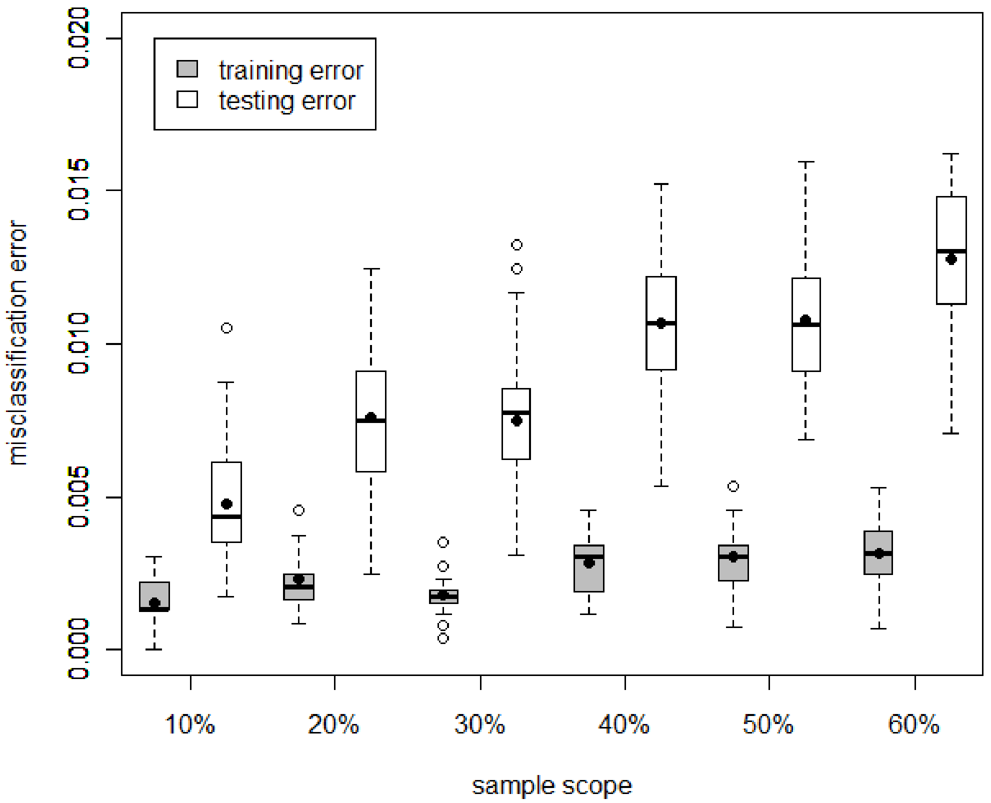


Fig. 6 Misclassification error distribution from a 50-run experiment

The second criterion is whether the algorithm reduces the false rate for all authors. As introduced before, the total false rate of our random sample before boosted trees classification is already very low, most authors have false rate less than 10%. However, these errors are distributed unevenly across authors, with some authors having error rates as high as 60%. For individual evaluation or network analysis, we need to remove this bias across individuals. So we further investigate the one-run experiment results. At each application scope level, we produce the scatter plot of each author's real false rate (in the testing set) against his/her misclassification error rate (Fig. 7). Most authors' false rates are successfully reduced to zero, that is, their paper set is perfectly clean after boosted trees classification. However, errors for several authors are still not removed or reduced to zero at all application scope levels, which are points above the testing error=0 horizontal line on the scatter plots. For scope levels higher than 30%, there are quite a few authors with remaining errors. Furthermore, all 10%, 20%, and 30% groups have four authors with remaining errors.

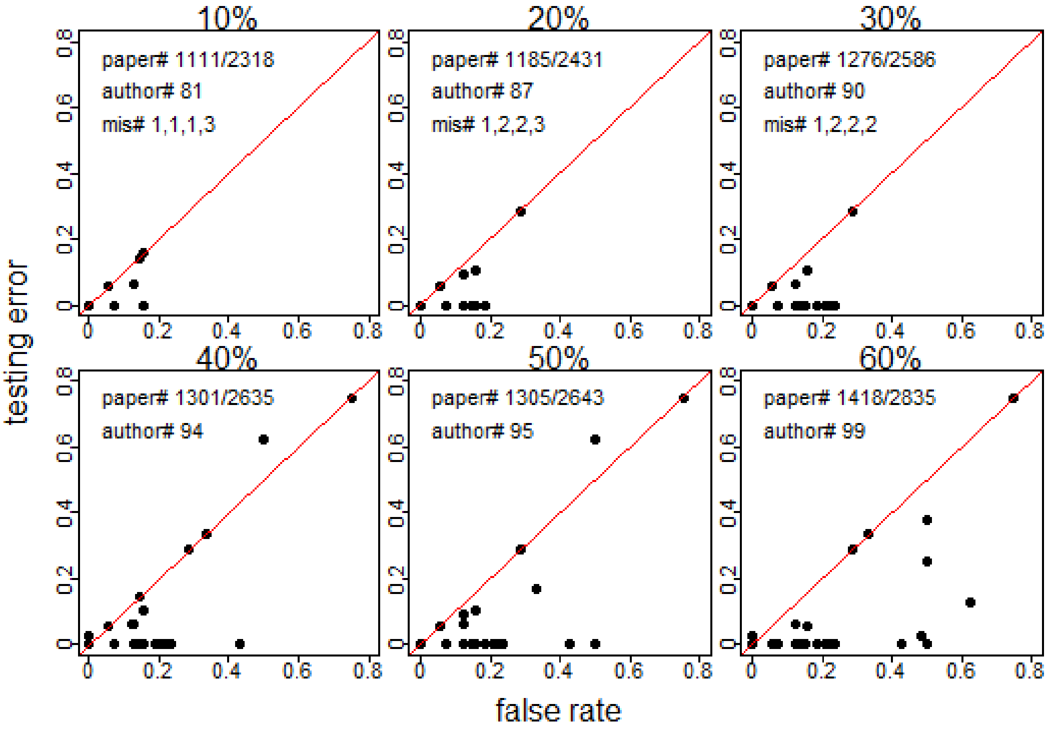


Fig. 7 Performance of the algorithm across authors. One point is one author, its false rate is the original false rate for this author in the testing set (i.e., error rate before implementing boosted trees algorithm to clean the data), and its testing error is the testing misclassification error rate of the algorithm for this author (i.e., error rate after implementing boosted trees algorithm to clean the data). The ideal case is that all points are on the (testing error=0) horizontal line, that is, for each authors, no matter how high his/her original false rate is, his/her paper set will be perfectly cleaned by the algorithm. Points above the (testing error=0) horizontal line and below the 45 degree line indicate that, for these authors, the algorithm improves data quality, but there are still remaining errors. Points on the 45 degree line indicate that, for these authors, the algorithm does not make the paper set “cleaner” nor “dirtier.” Points above the 45 degree line indicate that, for these authors, the algorithm makes the data even “dirtier” than before cleaning.

As an example, among these four challenging authors one biologist has one false paper that is not identified by any of the scope group models except the 60% scope model. This false paper is coauthored by a homonymous author from a hospital and another author from the same graduate school as the focal biologist. The topic of this false paper is somehow similar to the true papers of the focal scientist. Another similar case is that all group models fail to identify false papers about biochemistry from a focal geologist’s papers, 10% group model fails all three papers, 20% to 50% group models fail two papers, and 60% group model fails one. The third case is also similar: all group models fail to identify two false papers about electronics from a computer scientist's true papers. Furthermore, there is also a case in which the algorithm misclassifies true papers as false. For example, there is a computer scientist who participated in high energy physics projects and was listed as one author among hundreds of coauthors. All group models classify two or three of these true high energy papers as false.

Overall, 10%, 20%, and 30% group models achieve very impressive performance in terms of eliminating the bias across authors. They completely clean most authors' paper sets, with only six to eight incorrect predictions. In addition, none of the three groups is clearly superior.

Based on the two criteria (reducing overall error rate and removing individual biases), we define our application scope to be 30%, that is, we will use the boosted trees algorithm to clean authors with false rate no higher than 30%, while leaving higher false rate authors' papers to manual checking. Therefore, in our development dataset we apply the boosted trees method to 91 out of 100 authors and 3,862 out of 4,253 papers, while leaving 391 papers of 9 authors to manual checking. The 50-run experiment shows that the testing classification error of the 30% group is 0.75% on average. The result of one-run experiments shows that the testing misclassification of the 30% group is 0.55%, testing recall is 99.84%, and testing precision is 99.60%, which surpasses the previously published name disambiguation results.

Author screening

Scope redefinition raises one important question: how to predict the false rate of an author, before we actually know the true "in" or "out" value of all his/her papers. Therefore, we need to develop an author screening model to predict the false rate of authors, and each author needs to be screened before sending his/her papers to boosted trees classification. Only low false rate authors' papers are to be classified by boosted trees, while high false rate authors' papers need to be manually cleaned.

The idea of this model is as follows: if the false rate of one author is high, meaning there are a lot of false papers mixed with true papers, then the maximum of the previously defined similarity scores among all his/her papers will probably be low, because no one paper is very similar to all other papers. Similarly, the average may also be low. Therefore, it is reasonable to assume that the distribution characteristics of the paper similarity scores can help to predict if the author has low or high false rate. Consequently we can ground our decision about the author's false rate on the characteristics of the distribution of similarity scores of his/her papers. To assess the distributions we use four statistics: maximum (MAX), minimum (MIN), average (MEAN), and standard deviation (SDEV). The combination of twelve similarity scores and four statistics produces 48 classifiers:

$$\begin{Bmatrix} AU \\ CJ \\ KW \\ TI \\ AB \\ SC \end{Bmatrix} \times \begin{Bmatrix} EIG \\ AVG \end{Bmatrix} \times \begin{Bmatrix} MAX \\ MIN \\ MEAN \\ SDEV \end{Bmatrix}$$

In addition to these 48 classifiers, four author level variables discussed before are also used. They are PUBNO, FIELD, ASIAN, and LASTC.

The goal of this author screening process is to classify authors into two groups: "low" group authors who have false rates no more than 30% and "high" group authors with false rate higher than 30%. For this problem, we are more concerned about misclassifying "high" false rate authors into "low" class, but less concerned about misclassifying "low" group authors into "high" class, because the latter may increase the manual cleaning burden, but the former may reduce the accuracy of boosted trees paper classification. In other words, we are willing to sacrifice some recall (for "low" class) for a higher precision (for "low" class). We would like almost all authors classified as "low" to be actually "low," but can tolerant the situation of many real "low" authors being classified as "high."

To address this concern, we created the variable "CLASS15," coding authors with false rate higher than 15% as "high" and authors with false rate no more than 15% as "low." "CLASS15" is used for training the algorithm. The resulting algorithm will misclassify some authors with false rate higher than 15% as "low," but these misclassified authors will very likely have a real false rate a little bit higher than 15% but still lower than 30%. By reducing the error rate from 30% to 15% for training, we sacrifice recall to achieve the desired higher precision. When evaluating performance, we still use the "CLASS30," which is "low" for authors with false rate no more than 30% and "high" otherwise.

We randomly assign 67 authors for training and the remaining 33 for testing. The same boosted trees algorithm is applied, and one tree example is shown in Fig. 8, which is very simple, but reveals that, as expected, SCAVGMAX is relevant for classifying author classes. In other words, if the maximum of SCAVG (which is the similarity score between one paper and all other papers under the same author name, based on subject categories, and constructed from pairwise similarities by averaging) among all papers is less than 0.75, the paper set likely conflates many false papers rather than containing only a few contaminating papers.

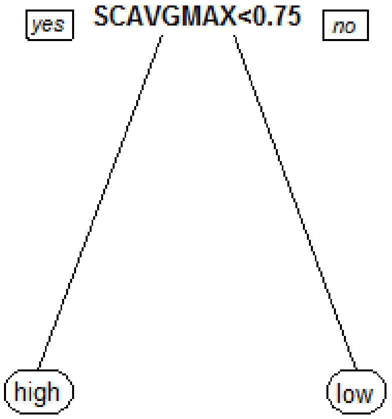


Fig. 8 One tree for author screening

A 50-run experiment is conducted and the distribution of misclassification error, recall, and precision are plotted in Fig. 9. The most important indicator here is the precision (for "low"), as discussed before. Testing precision has a median of 100%, and a mean of 97.64%, which is satisfying. At the same time, we do not want the recall to be too low, which will require too much manual cleaning for the predicted "high" authors. Testing recall has a median of 93.65%, and a mean of 92.78%, which is also satisfying.

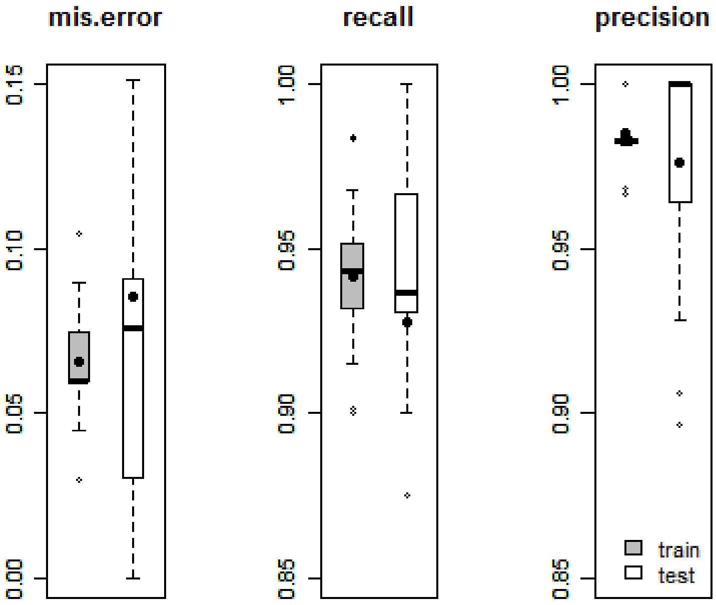


Fig. 9 Author screening model performance

Discussion

This paper proposes a boosted trees method for name disambiguation that will improve the quality and accuracy of bibliometric analysis, particularly in large-scale studies. It includes four steps: name and affiliation filtering, similarity score construction, author screening, and boosted trees classification. We randomly sampled 100 authors and manually checked their 4,253 papers. The results of the manual checking were used for supervised learning by the algorithm. We only apply the boosted trees algorithm to papers of authors with false rate no more than 30%. Therefore, before implementing the boosted trees paper classification, each author is screened to determine if his/her papers can be classified using boosted trees, or whether they will require manual checking. The 50-run experiment showed that the author screening testing precision achieves median of 100%, and mean of 97.64%. In other words, among authors whom we identified as having a low false rate (and therefore whose papers will be automatically classified by boosted trees), only 2.36% of them actually have a high false rate and in fact should not be submitted to the boosted trees algorithm.

Applying the boosted trees algorithm for cleaning papers from authors with no more than 30% false papers (i.e., 3,862 papers of 91 authors), the 50-run experiment shows that the testing misclassification error has a median of 0.78% and mean of 0.75%. The one-run experiment achieves a testing misclassification error of 0.55%, testing recall 99.84%, and testing precision 99.60%. Moreover, among 90 testing authors (one author only appeared in the training set), the algorithm successfully classifies 86 authors' papers 100% correctly, and only for four authors misclassifies one or two papers.

The performance of this algorithm is impressive, but it has several limitations: First, as discussed in the "existing methods" section when comparing classification with clustering, we benefit from having an external database listing known authors and their affiliation history. If authors and/or affiliations are not known in advance, clustering must be used and clustering is more difficult in general and clustering for unknown authors works in a more complex environment.

Second, we assumed that the name and affiliation filtering yielded a 100% recall dataset for each focal author, which however might not be the case. The gathered author affiliation information and author name records might be inaccurate or incomplete. Therefore, the name and affiliation filtering step may reduce our actual recall, which is however not estimated in this paper. Furthermore, we deal with only name disambiguation but not name matching. However, both WoS and Scopus have started to record full author names and links between authors and affiliations, which makes homonymous authors more differentiable. On the other hand, as the amount of available information increases, problem caused by misspelling, spelling difference, and name or affiliation change also become worse. Therefore, the name matching becomes more urgent.

Third, there are still cases this algorithm cannot effectively deal with. It cannot identify false papers that are very similar to true papers, and it may misclassify some true papers which are not similar to the mainstream studies of the focal author. Our method has one assumption that papers published by the same author are similar to each other. However, it is possible, though rare, that an author may switch between two unconnected research streams during his/her career. The algorithm would identify two authors in such cases. Alternatively, it is possible that two homonymous authors work on very similar scientific topics, and the algorithm would identify one author in such cases.

Fourth, an author screening is required before classifying papers, and this additional layer may introduce error. Furthermore, although not found in our study, it is possible that author screening fails. In one extreme case, it would be possible for a homonymous colleague to become a “usurper” if the focal scientist has very few publications, while the homonymous colleague is very productive. In this case, the algorithm will mistakenly view this homonymous colleague as the focal author. Author screening will classify this author into the “low” group, and the paper classification will keep the homonymous colleague’s papers while cleaning out the papers of the focal author.

Fifth, using boosted trees algorithms for paper classification assumes that the majority of the papers are true ones, and this assumption restricts the applicable scope of this algorithm. Some papers will be left for manual cleaning, in this example, 9 out of 100 authors and 391 out of 4,253 papers would need to be manually checked.

This limitation can be addressed by combining the boosted trees algorithm with other approaches proposed previously. If we have more external information about the focal scientists, we could add more layers of filtering to further reduce the false rate of authors, and then their papers can be dealt with by boosted trees classification. Another possibility is to combine this algorithm with clustering methods. Firstly use a “soft” clustering (which allows multiple membership of papers) to cluster papers into groups with very high recall and relatively low precision, and then apply the boosted trees algorithm to refine the cluster to achieve a higher accuracy. Although this method is powerful, combining clustering and boosted trees would have advantages because external information about focal scientists’ affiliation history can be replaced by clustering and therefore is no longer required and the pairwise similarity matrices used in our algorithm require the same data as clustering, so that this method will not introduce much extra computational burden.

In conclusion, we have shown that boosted trees algorithms hold promise for improving name disambiguation processes in bibliometric studies, possibly in combination with other previously proposed solutions. We believe that 99%+ accuracy is possible if the right combination of techniques is used, enabling large scale bibliometrics to move from the national or institutional level to an individual level of analysis. This will serve to improve our understanding of how the science system works and so improve science policy decisions.

Appendix 1: real-world motivation and implementation

This study was motivated by a real-world problem confronting our research on the collaboration networks of American academic scientists. It attempts to identify clean publication sets for each focal scientist and link bibliometric data to survey data, thereby creating an extensive and rich dataset of academic scientists, their networks, and related productivity. From this approach, we hope that this methodological solution can inform other similar issues faced by other researchers.

The data for this study comes from a 2006-2009 NSF-funded national study of academic scientists and engineers in Research I universities in the United States (*Women in Science and Engineering: Network Access, Participation, and Career Outcomes*, Grant # REC-0529642). This NSF funded project involved a two stage online survey, collection of CV data for survey respondents, including a complete history of focal scientists' affiliation information, and collection of lifetime bibliometric data for each focal author, requiring a name and affiliation match. The object of the name disambiguation method presented in this paper was to clean 54,853 papers of 1,315 focal scientists in five disciplines (biology (BIOL), chemistry (CHEM), computer science (CS), earth and atmospheric sciences (EAS), and electrical engineering (EE).

For the development dataset of 100 authors, we firstly excluded scientists with less than five publications, because the similarity score might be unstable if the focal author has very few papers, and then we randomly sampled 100 authors from the remaining 1,255 authors.

The method presented in this paper was implemented to clean the 54,853 papers of 1,315 authors, among them 4,253 papers from 100 authors were manually checked for supervised learning. For the remaining papers, the algorithm automatically classified 44,777 papers of 1,025 authors, 156 papers of 60 authors were left for manual checking because these authors had less than five papers, and 5,667 papers of 130 authors were left for manual checking because these authors were predicted to have high false rates. Overall, the algorithm reduced the labor required for manual data cleaning by about 80% (44,777 out of 54,853 papers were automatically cleaned).

Acknowledgments

This work was supported through a U.S. National Science Foundation grant, "Women in Science and Engineering: Network Access, Participation, and Career Outcomes" (NSF Grant # REC-0529642). We thank the NSF for their support of this work, and the NETWISE team for its collective effort in the development of these data.

References

Aksnes, D. W. (2008). When different persons have an identical author name. How frequent are homonyms? *Journal of the American Society for Information Science and Technology, 59*(5), 838-841.

Aswani, N., Bontcheva, K., & Cunningham, H. (2006). Mining information for instance unification. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Eds.), *The Semantic Web - ISWC 2006* (Vol. 4273, pp. 329-342, Lecture Notes in Computer Science). Berlin / Heidelberg: Springer.

Bhattacharya, I., & Getoor, L. (2006). A latent dirichlet model for unsupervised entity resolution. In J. Ghosh, D. Lambert, D. Skillicorn, & J. Srivastava (Eds.), *Proceedings of the SIAM 6th International Conference on Data Mining* (pp. 47-58). Bethesda, MD: Society for Industrial Mathematics.

Bhattacharya, I., & Getoor, L. (2007). Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD), 1*(1), 1-36.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research, 3*, 993-1022.

Breiman, L. (1984). *Classification and regression trees*. Boca Raton, FL: Chapman & Hall/CRC.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121-167.

Cole, F. J., & Eales, N. B. (1917). The history of comparative anatomy: Part 1.-a statistical analysis of the literature. *Science Progress in the Twentieth Century: A Quarterly Journal of Scientific Work & Thought, 6*, 578-597.

Cota, R. G., Ferreira, A. A., Nascimento, C., Goncalves, M. A., & Laender, A. H. F. (2010). An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. *Journal of the American Society for Information Science and Technology, 61*(9), 1853-1870.

Culotta, A., Kanani, P., Hall, R., Wick, M., & McCallum, A. Author disambiguation using error-driven machine learning with a ranking loss function. In *6th International Workshop on Information Integration on the Web (IIWeb-07), Vancouver, Canada, July 23 2007*

Culp, M., Johnson, K., & Michailidis, G. (2010). ada: an R package for stochastic boosting. http://CRAN.R-project.org/package=ada. Accessed August 01 2011.

D'Angelo, C. A., Giuffrida, C., & Abramo, G. (2011). A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. *Journal of the American Society for Information Science and Technology, 62*(2), 257-269.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Special invited paper. additive logistic regression: A statistical view of boosting. *The Annals of Statistics, 28*(2), 337-374.

Han, H., Giles, L., Zha, H., Li, C., & Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In H. Chen, H. Wactlar, C.-c. Chen, E.-P. Lim, & M. Christel (Eds.), *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 296-305). New York: ACM.

Han, H., Xu, W., Zha, H., & Giles, C. L. (2005a). A hierarchical naive Bayes mixture model for name disambiguation in author citations. In H. M. Haddad, A. Omicini, R. L. Wainwright, & L. M. Liebrock (Eds.), *Proceedings of the 2005 ACM Symposium on Applied Computing* (pp. 1065-1069). New York: ACM.

Han, H., Zha, H., & Giles, C. L. (2005b). Name disambiguation in author citations using a K-way spectral clustering method. In M. Marlino, T. Sumner, & F. Shipman (Eds.), *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 334-343). New York: ACM.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2ed.). New York: Springer.

Hirsch, J. E. (2005). 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, doi:10.1073/pnas.0507655102.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In F. Gey, M. Hearst, & R. Tong (Eds.), *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50-57). New York: ACM.

Huang, J., Ertekin, S., & Giles, C. (2006). Efficient name disambiguation for large-scale databases. *Knowledge Discovery in Databases: PKDD 2006, 4213*, 536-544.

Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis* (6th ed.). Upper Saddle River, N.J.: Pearson Prentice Hall.

Kanani, P., & McCallum, A. Efficient strategies for improving partitioning-based author coreference by incorporating Web pages as graph nodes. In *6th International Workshop on Information Integration on the Web (IIWeb-07), Vancouver, Canada, July 23 2007* (Vol. 23)

Kanani, P., McCallum, A., & Pal, C. Improving author coreference by resource-bounded information gathering from the web. In *20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 6-12 2007* (pp. 429-434): AAAI Press

Kang, I. S., Na, S. H., Lee, S., Jung, H., Kim, P., Sung, W. K., et al. (2009). On co-authorship for author disambiguation. *Information Processing & Management, 45*(1), 84-97.

Lee, D., On, B. W., Kang, J., & Park, S. (2005). Effective and scalable solutions for mixed and split citation problems in digital libraries. In L. Berti-Equille, C. Batini, & D. Srivastava (Eds.), *International Workshop on Information Quality in Information Systems (IQIS 2005)* (pp. 69-76). New York: ACM.

Liben-Nowell, D., & Kleinberg, J. (2007). The link prediction problem for social networks. *Journal of the American Society for Information Science and Technology, 58*(7), 1019-1031.

McCallum, A., & Wellner, B. Object consolidation by graph partitioning with a conditionally-trained distance metric. In *KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation, Washington, DC, August 24-27 2003*: Citeseer

McRae-Spencer, D. M., & Shadbolt, N. R. (2006). Also by the same author: AKTiveAuthor, a citation graph approach to name disambiguation. In G. Marchionini, M. L. Nelson, & C. C. Marshall (Eds.), *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 53-54). New York: ACM.

Moed, H. F. (2005). *Citation analysis in research evaluation*. Dordrecht, the Netherlands: Springer.

Newman, M. E. J. (2001). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America, 98*(2), 404.

On, B. W., Lee, D., Kang, J., Mitra, P., & Acm (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework. In M. Marlino, T. Sumner, & F. Shipman (Eds.), *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 344-353). New York: ACM.

Onodera, N., Iwasawa, M., Midorikawa, N., Yoshikane, F., Amano, K., Ootani, Y., et al. (2011). A method for eliminating articles by homonymous authors from the large number of articles retrieved by author search. *Journal of the American Society for Information Science and Technology, 62*(4), 677-690, doi:10.1002/asi.21491.

Porter, A., & Rafols, I. (2009). Is science becoming more interdisciplinary? Measuring and mapping six research fields over time. *Scientometrics, 81*(3), 719-745, doi:10.1007/s11192-008-2197-2.

Radicchi, F., Fortunato, S., Markines, B., & Vespignani, A. (2009). Diffusion of scientific credits and the ranking of scientists. *Physical Review E, 80*(5), 056103.

Smalheiser, N. R., & Torvik, V. I. (2009). Author Name Disambiguation. *Annual Review of Information Science and Technology, 43*, 287-313.

Song, Y., Huang, J., Councill, I. G., Li, J., & Giles, C. L. (2007). Efficient Topic-based Unsupervised Name Disambiguation. In E. Rasmussen, R. R. Larson, E. Toms, & S. Sugimoto (Eds.), *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 342-351). New York: ACM.

Strotmann, A., Zhao, D., & Bubela, T. (2009). Author name disambiguation for collaboration network analysis and visualization. *Proceedings of the American Society for Information Science and Technology, 46*(1), 1-20, doi:10.1002/meet.2009.1450460218.

Tan, Y. F., Kan, M. Y., & Lee, D. (2006). Search engine driven author disambiguation. In G. Marchionini, M. L. Nelson, & C. C. Marshall (Eds.), *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 314-315). New York: ACM.

Tang, L., & Walsh, J. P. (2010). Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics, 84*(3), 763-784, doi:10.1007/s11192-010-0196-6.

Therneau, T. M., & Atkinson, B. (2010). rpart: recursive partitioning. http://CRAN.R-project.org/package=rpart. Accessed August 01 2011.

Torvik, V. I., & Smalheiser, N. R. (2009). Author name disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data (TKDD), 3*(3), 1-29.

Torvik, V. I., Weeber, M., Swanson, D. R., & Smalheiser, N. R. (2005). A probabilistic similarity metric for Medline records: A model for author name disambiguation. *Journal of the American Society for Information Science and Technology*, *56*(2), 140-158, doi:10.1002/asi/20105.

U.S. Census Bureau (2000). Frequently occurring surnames from Census 2000. http://www.census.gov/genealogy/www/data/2000surnames/index.html. Accessed August 01 2011.

Wooding, S., Wilcox-Jay, K., Lewison, G., & Grant, J. (2006). Co-author inclusion: A novel recursive algorithmic method for dealingwith homonyms in bibliometric analysis. *Scientometrics*, *66*(1), 11-21.

Yang, K. H., Jiang, J. Y., Lee, H. M., & Ho, J. M. (2006). Extracting citation relationships from web documents for author disambiguation. Taipei: Technical Report (TR-IIS-06-017).

Yin, X., Han, J., & Yu, P. S. (2007). Object distinction: Distinguishing objects with identical names. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop* (pp. 1242-1246). Washington, DC: IEEE.