"How Prolific Are You?": A Scientometric Analysis of Researchers

Rahul Yedida[1] and Snehanshu Saha[1]

[1]Department of Computer Science & Engineering, PESIT South Campus

Abstract

Scholars in academia regularly write papers showing the results of their work. To gauge how relevant their work is, several measures are used, including citation count and the h-index. This paper describes the methods used to identify "prolific" authors, whose metrics are significantly higher than their peers, and provides a general discussion of the relationships between different metrics. We also discuss how this analysis is performed even with the high dimensionality of the dataset.

1 Introduction

Scientometrics is the study of measuring science and research. Several measures are currently in use to measure the output of researchers, and each of these individually may not be an indicator of the quality and repute of a scholar's work. Comparing researchers by several metrics at once is cumbersome to do manually, and it would be incorrect to use a function of several metrics to compare how "prolific" a researcher is. In this paper, we first identify linear relationships among the attributes. We assert that this is also useful in finding the "most important" features. Later, we use two algorithms to cluster the records and compare the outputs of both algorithms. Finally, we use these results to identify prolific authors.

The rest of the paper is structured as follows. Section 2 describes the data and shows correlations between the attributes. Section 3 discusses the methods we used to analyse this data.

2 Data

The data contains 618 records of scholars with 25 attributes. There are no missing values, and eight of the attributes are integers. For ease of analysis, we have labeled the attributes as *v1* through *v25*. A sample of the data is shown in the Supplementary Materials section at the end of this paper.

2.1 Preliminary Analysis

For a preliminary analysis of the relationships between the attributes, we sought to

discover monotonic relationships. This was done by computing pairwise Spearman's rank correlation coefficient. This revealed that there were only positive monotonic relationships. 42.08% of these coefficients were above 0.8, and 52% were above 0.7.

A logical next step, then, was to investigate the percentage of linear relationships among these. This was done by computing pairwise Pearson correlation coefficients. To visually understand pairs that had high linear correlations, we displayed this information in a matrix, maintaining only the upper half, and avoiding diagonal entries. This avoids counting self-correlations and counting the same pair twice. Further, when finding multiple linear regression equations, this finds only unique relations between the variables. Figure 1 shows this result. Asterisks indicate a Pearson R value of 0.8 or higher. The numbers at the right indicate how many other variables each variable is highly correlated with.

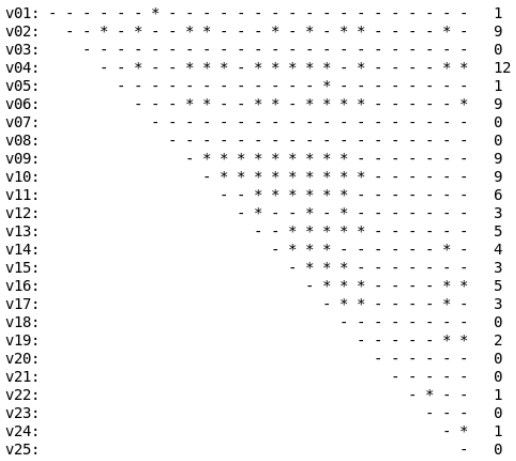

Figure 1: Matrix showing pairs with Pearson R values 0.8 or higher

These correlations provide a starting point to find relationships between the attributes.

3 Methods

This section discusses in detail the methods used to analyse the data.

3.1 Regression Analysis

Our first analysis investigated the precise relationships between the attributes. We looked at linear relationships. Because v4 had the most number of correlations, we started with it.

Multiple regression analysis was performed as follows. For each attribute that the variable being analysed has a high correlation with, we ran an Ordinary Least Squares (OLS) regression model in two cases–with a constant term and without (that is, in a forward step-wise fashion).

In each case, for each variable, we check the p-value to ensure that all variables are statistically significant. Finally, we choose the model with the highest adjusted R^2 value, which avoids the pitfalls of "kitchen sink regression". We also make sure that there are no significant autocorrelations, using the Durbin-Watson test. [1] suggests that a conservative range for the acceptable values for this test is between 1 and 3. In some cases, we chose to pick a model that has a slightly lesser adjusted R^2 value in favor of using lesser variables. This results in a model that still has a high predictive power, yet is simpler. In addition, we also check the residuals vs. fits plots and check that they are not very correlated.

We note here that this analysis may be useful for identifying "important" at-

tributes in a dataset. If a model that predicts a dependent variable through the other attributes yields a high adjusted R^2 value, then we may safely remove this dependent variable, keeping only the statistically significant predictor variables from the regression analysis.

The full regression results can be found in the Supplementary Materials section at the end of this paper. Here, we present a summary of the results of the regression analysis that was performed.

3.1.1 Analysis on the full data

We first looked at v4, and did a step-wise regression analysis. *v6, v9, v11, v13, v14, v17, v19, v24,* and *v25* were statistically significant, and the adjusted R^2 was 0.972. For this model, the Durbin-Watson statistic was 2.06, which suggests very minor negative autocorrelations. The correlations between the residuals and the fitted values was -0.0047. Figure 2 shows the residuals vs. fits plot for *v4*.

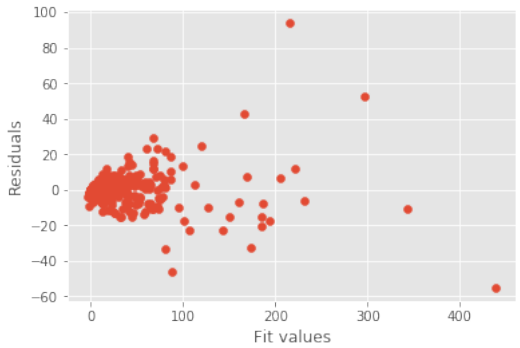

Figure 2: Residuals vs. fits plot for v4

Our next variable of interest was *v2*, having high Pearson R correlations with nine other variables. Only *v9* was not statistically significant. However, we also chose to discard *v24* because it added no predictive power to the model, and the adjusted R^2 for this model was 0.993, and the Durbin-Watson statistic was 1.853, suggesting only minor positive autocorrelations. All the p-values were less than 10^{-3}. The correlation between the residuals and the fits was 0.023. Figure 3 shows the residuals vs. fits plot for *v2*.

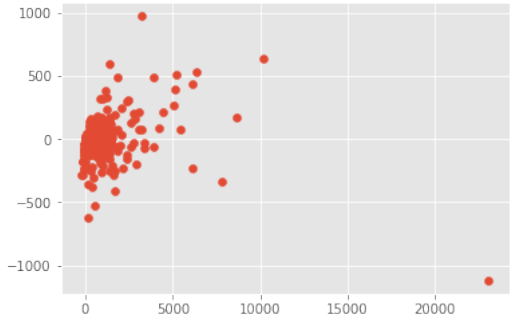

Figure 3: Residuals vs. fits plot for v2

We then looked at *v6*, also having high correlations with nine other variables. *v9* and *v14* were not statistically significant. We also discarded *v25* to obtain a simpler model with an adjusted R^2 value 0.001 less. This model had an adjusted R^2 of 0.913, and a Durbin-Watson statistic of 1.64. All p-values were below 10^{-3}.

Next, we looked at *v9*. Only *v13* was not statistically significant, but we also discarded *v10* and *v11*, which resulted in a model whose adjusted R^2 was 0.986 (a reduction of 0.002 from when the two variables were included). This model had a Durbin-Watson statistic of 1.99.

Finally, for *v10*, all the predictor variables were statistically significant, and we

discarded *v19*, which resulted in a model with the same adjusted R^2. All the p-values were below 10^{-3}, except *v13*, whose p-value was 0.043 (95% C.I [0.07, 0.425]). The Durbin-Watson statistic was 1.861.

It should be noted that in all of the regression results of the above variables, the constant was not statistically significant, and the results reported are those for the models that did not include the constant.

3.1.2 Analysis on subset of the data

Our next step of analysis was to pick a subset of the variables, and redo the same analysis discussed above on this subset. Once again, it turns out that for all the regression equations obtained, the constant was not statistically significant.

The variables we chose were *v9* through *v21* and *v24*. maybe explain why these. Again, we created a matrix of Pearson R correlations, and picked out those that were higher than 0.8. Figure 4 shows this result.

Figure 4: Pearson R correlation matrix for subset of the data

Using this, we redid the same analysis as in the previous subsection. Again, we only discuss the results here, but the full list of experiments is in the Supplementary Materials section.

We first looked at v9. *v13* was not significant when all the variables it was correlated to were considered. However, when *v10* was not considered, it was statistically significant. We chose to sacrifice 0.2% adjusted R^2 in favor of a model that had two fewer predictor variables. The Durbin-Watson value was 1.99.

While predicting *v10*, dropping *v19* resulted in no loss of adjusted R^2. We further dropped *v18*, and dropped *v12* because it was not significant. While dropping *v18* caused a 0.3% drop in adjusted R^2, we were able to use two less variables. The Durbin-Watson statistic was 1.991.

We only looked at two other variables, because the adjusted R^2 for the others was below 0.95. For *v11*, dropping *v17* and *v18* resulted in a 0.2% drop in adjusted R^2, and we were able to use only four variables to get an adjusted R^2 of 0.971 and a Durbin-Watson of 2.081. Finally, for *v13*, all variables were statistically significant, and we discarded *v19* with no loss of adjusted R^2.

Perhaps more interesting than the above results was when we plotted the maximum adjusted R^2 (without discarding any variables) against the number of predictor variables used (on this subset of the data). Figure 5 shows this result, where a logarithm function fits quite well to the data. The equation obtained was

$$\text{Adj. } R^2 = 0.125465 \log n + 0.746654 \quad (1)$$

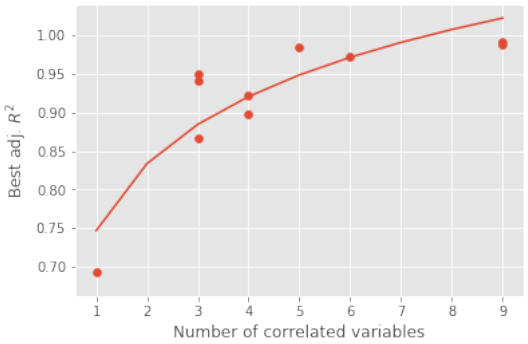

Figure 5: Adjusted R^2 vs. number of predictor variables

The sum of the squared residuals for this fit was 0.01456.

3.2 Cluster Analysis

We next sought to cluster the points, as a first step towards identifying "prolific" authors. Because finding the number of clusters in such high dimensional data is difficult, we chose to use two different algorithms–DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[2] and mean-shift clustering [3]. We picked two subsets of the data to perform this analysis on–the first was the set of variables that were highly correlated with *v4* (denote this subset as S_1)–we chose these because it had an especially high number of correlated variables–and the second was the subset discussed in the previous subsection (which we shall denote as S_2.

Because DBSCAN requires an *eps* parameter, we chose to find a reasonable guess. We did this by setting a threshold that no more than roughly 10% of the points (we chose 60 points) should be in individual custers. We started with an initial *eps* guess of 0.5 and made increments of 0.1 till this condition was satisfied.

For S_1, the value of *eps* we arrived at was 45. Apart from 60 individual clusters, the algorithm identified three clusters of points. The results of this are summarized in Table 1.

Cluster	Number of points
-1	60
0	532
1	20
2	6

Table 1: DBSCAN clustering results on S_1

The cluster -1 represents individual clusters. The rest of the clusters were arbitrarily numbered from zero. Next, we ran mean-shift clustering, which automatically finds clusters by estimating a *bandwidth* parameter. Mean-Shift clustering identified 12 clusters, out of which five were individual clusters. We summarize these results in Table 2.

Cluster	Number of points
-1	5
0	509
1	72
2	19
3	4
4	4
5	3
6	2

Table 2: Mean-Shift clustering results for S_1

We then checked whether the points in

similar-sized clusters in each algorithm were the same. We briefly do this analysis for S_1, but conduct a more thorough investigation for S_2.

We first looked at cluster 0 of both algorithms. Of these, 507 points were common, indicating that the majority of the dataset was clustered into the same cluster by both algorithms. Interestingly, however, this seems to be the only major commonality between the outputs of these algorithms. For example, cluster 1 of DBSCAN and cluster 2 of Mean-Shift, which had a very similar number of points, had no common points at all. Finally, 21 of the points that were marked as individual clusters by DBSCAN were put into cluster 1 by Mean-Shift.

Next, we performed the above analysis for S_2. The value of *eps* we arrived at was 18.6, and DBSCAN identified three clusters apart from sixty individual clusters. A summary is provided in Table 3, again, with cluster -1 meaning individual clusters.

Cluster	Number of points
-1	60
0	541
1	3
2	14

Table 3: DBSCAN clustering results for S_2

Table 4 summarizes the results of mean-shift clustering, which identified seven "real" clusters and eight individual clusters.

Cluster	Number of points
-1	8
0	504
1	82
2	15
3	3
4	2
5	2
6	2

Table 4: Mean-Shift clustering results for S_2

From these two tables, we followed through to track each point in both clusters. The full results of this are given in the Supplementary Materials section, but a summary is as follows. DBSCAN marked 60 points as individual clusters, but this was due to our 10% threshold. On the other hand, mean-shift only identified eight individual clusters. However, mean-shift identified four other clusters that had three points each or less, and all of these were marked as individual clusters by DBSCAN. More surprisingly, a cluster marked by mean-shift as having fifteen points was identified by DBSCAN as fifteen individual clusters. The rest of the 28 individual clusters identified by DBSCAN were a part of mean-shift's second-largest cluster of 82 points. The analysis so far may suggest that we should lower the threshold for DBSCAN, which would increase *eps* (because if fewer points have to be identified as individual clusters, the distance that the algorithm is willing to consider must increase). However, looking at the bigger clusters weakens this position. Mean-shift's largest cluster of 504 points was a proper subset of DBSCAN's largest cluster. A large part of mean-shift's second-

largest cluster, 37 out of 82 points, was also a part of DBSCAN's single largest cluster. This seems to suggest that we should reduce *eps* to make the outputs of both algorithms more similar.

While the two conclusions drawn above may seem contradictory, they are two separate observations. Increasing *eps* would mean DBSCAN would find less clusters as single points. Decreasing *eps* would certainly increase the number of individual clusters, which we certainly do not want, but this would shift some points in the larger clusters around, which may be desirable.

References

[1] A. Field, *Discovering statistics using SPSS*. Sage publications, 2009.

[2] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[3] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.