Predicting How Many Citations an Academic Paper Will Receive

Kyler Siegel

December 14, 2013

1 Introduction

The goal of this project is to determine to what extent an academic paper's success, as measured by the number of citations it receives, can be inferred solely from the paper's bibliography, i.e. its list of references and the number of citations each reference received. The basic intuition is that perhaps high caliber papers tend to have high caliber bibliographies, or perhaps there are certain non-obvious signatures of a bibliography that are characteristic of well-cited papers. We attempt to train a supervised machine learning algorithm on a large set of paper citation data. If successful, one could feed any paper's bibliography (along with appropriate citation data for each reference, which is generally available through online databases) into a previously trained machine learning algorithm to get a good prediction of how successful the paper will be, without ever actually reading the paper.

2 The Data

We gathered academic paper citation data from the online database Scopus. For each of 34,326 papers we formed a feature vector which includes the number of times each reference in the bibliography was cited, along with the year of publication and general subject category of the paper. The papers were taken from the broad categories of arts and humanities, biology, psychology, mathematics, and physics. For each subject we gathered data for papers from 20-30 top journals, dating from the nineties to the present. The data was scraped from the web using a fairly simple python program involving the module *mechanize* and regular expression functionality. We removed from the dataset any paper which either received 0 citations or had 0 references which received citations, the logic being that these papers probably represented incomplete or inaccurate citation data. After this removal the dataset contains 27,416 papers. Since different papers can different numbers of references, we padded each paper by zeroes so that all the vectors have the same length. The largest number of references turned out to be 82, so after padding all vectors have this length.

3 Running Machine Learning Algorithms

With the data in feature vector format, we ran various machine learning algorithms for regression on the data set. The algorithms we used, primarily adapted from the *sklearn* python module, are as follows:

- linear regression
- ridge regression (a generalized linear model)
- lasso regression (a generalized linear model)
- elastic net regression (a generalized linear model)
- support vector regression wth linear kernel
- support vector regression with radial basis function (rbf) kernel
- random forest regression
- support vector machine for classification with the y values discretized into 2 parts, using medians to assign values to each part of the resulting partition
- same as above, but with 3 parts
- same as above, but with 4 parts.

4 Results

In order to evaluate the success of each algorithm, we used ten-fold hold out cross validation. Namely, we randomly divided the data into 10 parts and tested the algorithms on one of the parts after training on the union of the other 9 parts. To measure the test accuracy, we used two measures: R^2 and the average absolute fractional error E. Here R^2 is defined by $1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$, where $SS_{\text{tot}} = \sum_i (y_i - \overline{y})^2$, $SS_{\text{res}} = \sum_i (y_i - f_i)^2$, with y_i the observed values, f_i the corresponding predicted values, and $\overline{y}$ the mean of the observed values. Moreover, we define E to be the average over all test points of $\frac{|f_i - y_i|}{|y_i|}$. The average absolute fractional error has the advantage of being scale invariant and easy to interpret, with a larger value indicating larger average percent error. However, it does not it does not account for variance in the data. Namely, i the y_i are closely concentrated near $\overline{y}$, it is easy to achieve a small value for E simply by picking all the f_i's to be $\overline{y}$. On the other hand, R^2 accounts for this by taking into account the variance of the data, at the cost of being slightly more difficult to make intuitve sense of.

The resulting accuracy data is as follows:

	LR	Ridge	Lasso	EN	Linear SVR	RBF SVR	RF	SVM 2 labels	SVM 3	SVM 4
R^2	0.153	0.154	0.164	0.157	0.068	0.078	0.080	0.021	0.078	0.119
E	4.431	4.430	4.319	3.941	1.936	1.558	2.744	1.587	1.501	1.753

In terms of R^2, the generalized linear models perform the most accurately, with the random forest regressor performing almost as well. In particular, Lasso performs the best. The support vector machine based algorithms performed considerable worse. On the other hand, with respect to E the discretized support vector machine algorithms performed the best, followed by support vector regression, whereas the generalized linear models have much larger E values. In terms of E the discretized support vector machine with 3 labels performs the best.

5 Conclusion

Unfortunately, the largest R^2 value of any of the machine learning algorithms we tested is 0.164, which is probably not large enough to make any reliable predictions in most applications. This likely indicates that the citation number of a paper is not easily predicted in terms of the paper's bibliography, and indeed one should actually should the paper to better judge its quality. On the othe rhand, it is also possible that gathering more data may improve the learning algorithms. In particular, since our data was mostly gathering from high end journals, we may have missed parts of the feature space containing journals with lower impact factors.