A Mathematical Analysis Of Mathematical Salaries and More

Victoria Chayes Tong Cheng Terence Coelho Quentin Dubroff Dodam Ih
Joe Olsen Jason Saied Yukun Yao Doron Zeilberger Tianhao Zhang

August 18, 2019

Abstract

In this final project of Experimental Mathematics Class (Spring 2019), we use the data of the tenured and tenure-track faculty in the Rutgers math department as a case study to demonstrate the statistical and mathematical relationships among several variables, e.g., the number of publications and citations, the rank of professorship and of course, the salaries. Different statistical tools, including simple and multi-variable regression, logistic regression, lasso and ridge regression, neural network and unsupervised learning, are exploited so that the results obtained from various methods can be easily compared.

1 Introduction

This paper is a final project of Experimental Mathematics Class at Rutgers in Spring 2019 taught by Prof. Doron Zeilberger. In this project, we collect public data online of the tenured and tenure-track faculty members at the department of mathematics at Rutgers University-New Brunswick.

In our data set, there are 59 professors. The variables are Lastname, Firstname, Rank (Assistant Professor=1, Associate Professor=2, Full Professor=3, Distinguished Professor=4), Number of Publication, Number of Citations, H-Index, Base Salary, AMS Fellow (Yes=1, No=0), the Year of Ph.D. Awarded.

The data of names and ranks are from the website of Rutgers math department. The numbers of publications, citations and h-index are from MathSciNet. The data of salaries are also publicly available online from the following link: `https://php.app.com/agent/rutgersemployees/search`. The binary data on whether a professor is an AMS fellow is from the website of AMS. And the data of the year when the Ph.D. degree was awarded can be easily obtained from Mathematics Genealogy.

As a class project, we intend to study the statistical, and of course, mathematical relationship between these variables we collect. Especially, much of our study focuses on the prediction of salaries and ranks with the data of numbers of publications, citations and h-indices.

Accompanying Maple package and data files

This article is accompanied by multiple Maple packages available from the front of this article

`https://sites.math.rutgers.edu/ yao/DAMF.html` ,

where the readers can also find the raw data file.

2 Exploratory Data Analysis

At first, we would like to do exploratory data analysis for mean, variance, percentiles (with histograms) and the co-variance matrix of the fields [NumberOfPublications, NumberOfCitations, H-index, Salary, YearOfPhD]. Of course, our silicon servant and Maple computed the mean and standard deviations for each field across all professors:

Field	Mean	Standard Deviation
Rank	3.153	0.971
Number of Publications	71.661	62.223
Number of Citations	1027.526	1119.278
h-index	14.271	7.783
Base Salary	158295.424	45632.948
Year of PhD	1989.051	15.105

Table 1: Means and standard deviations across all professors (n=59)

The following tables are means and standard deviations across professors in each rank and various percentiles for each field and the covariance matrix. All these results are obtained via Maple rather than Python or R.

Field	Mean	Standard Deviation
Number of Publications	13.167	6.094
Number of Citations	50.000	31.332
h-index	4.000	1.633
Base Salary	99083.333	2863.807
Year of PhD	2013.833	1.772

Table 2: Means and standard deviations across Assistant Professors (n=6)

Field	Mean	Standard Deviation
Number of Publications	26.833	18.685
Number of Citations	145.667	88.009
h-index	6.000	2.000
Base Salary	125592.333	27075.982
Year of PhD	2003.167	12.020

Table 3: Means and standard deviations across Associate Professors (n=6)

Field	Mean	Standard Deviation
Number of Publications	42.500	21.381
Number of Citations	593.100	426.532
h-index	12.400	4.271
Base Salary	134656.750	13648.669
Year of PhD	1991.550	10.749

Table 4: Means and standard deviations across Professors (n=20)

Field	Mean	Standard Deviation
Number of Publications	116.222	64.911
Number of Citations	1762.519	1239.229
h-index	19.778	6.768
Base Salary	196231.148	39484.961
Year of PhD	1978.556	9.199

Table 5: Means and standard deviations across Distinguished Professors (n=27)

Field	5-th	10-th	25-th	50-th	75-th	90-th	95-th
Publications	9.300	15.000	26.167	58.000	106.833	139.467	156.400
Citations	35.200	71.133	203.000	632.000	1456.000	2552.733	3541.300
h-index	3.300	5.000	8.167	13.000	18.833	26.733	27.700
Base Salary	97200.000	102000.000	126681.500	143601.000	190641.333	214756.200	264902.800
Year of PhD	1967.300	1969.000	1977.000	1989.000	2000.833	2011.733	2014.700

Table 6: Various percentiles

	Rank	Publications	Citations	h-index	Base Salary	Year of PhD
Rank	0.959	37.484	635.636	5.544	33293.589	-11.473
Publications	37.484	3938.469	52316.612	358.731	210680.394	-624.879
Citations	635.625	52316.612	127348.301	8189.872	3594213.250	-9719.269
h-index	5.544	358.731	8189.872	61.615	258841.538	-72.617
Base Salary	33293.589	210680.394	3594213.250	258841.538	2202648298.421	-480151.487
Year of PhD	-11.473	-624.879	-9719.2689	- 72.617	-480151.487	232.084

Table 7: The covariance matrix

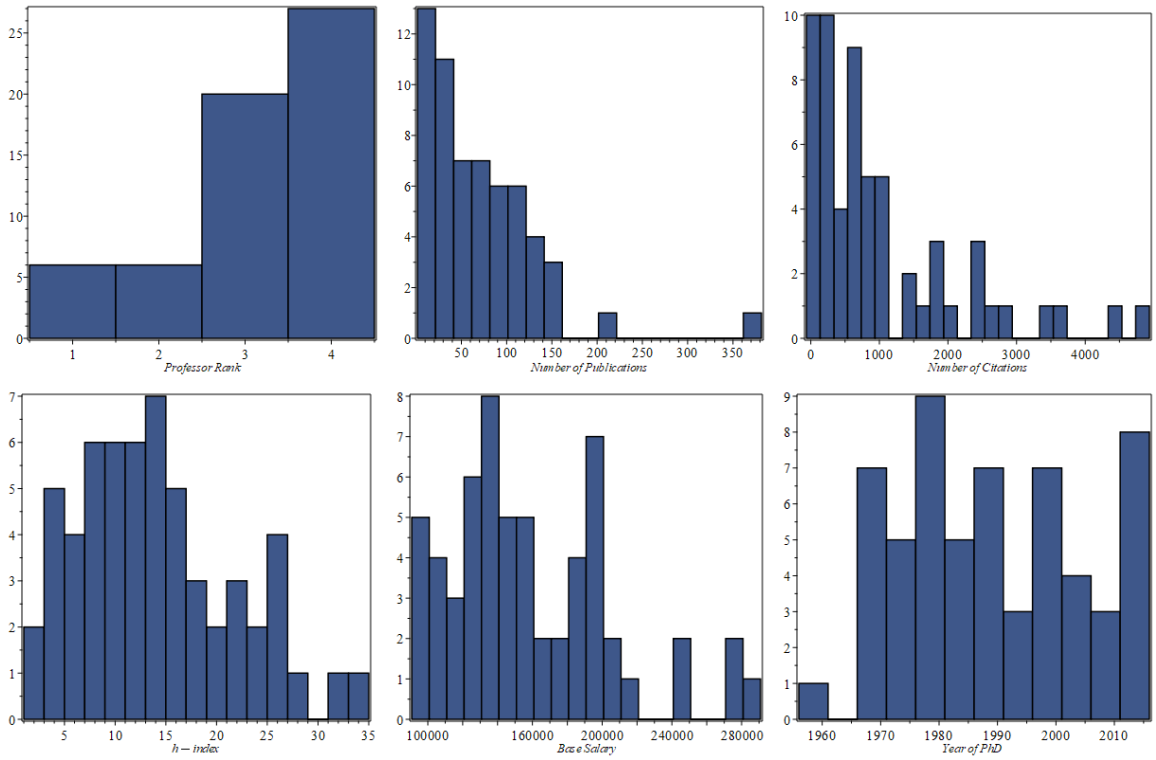

Figure 1: Histograms for each data field

	Rank	Publications	Citations	h-index	Base Salary	Year of PhD
Rank	1	0.609886	0.574939	0.721209	0.724367	-0.769026
Publications	0.609886	1	0.738458	0.728221	0.715299	-0.653597
Citations	0.574939	0.738458	1	0.924239	0.678393	-0.565147
h-index	0.721209	0.728221	0.924239	1	0.702617	-0.607262
Base Salary	0.724367	0.715299	0.678393	0.702617	1	-0.671558
Year of PhD	-0.769026	-0.653597	-0.565147	-0.607262	-0.671558	1

Table 8: The correlation matrix

3 Simple Regression to Predict Salaries

In this section, we'd like to use simple linear regression model to predict the salaries of math professors. We try to use (i) the number of publications, (ii) the number of citations and (iii) both (i) and (ii) to do the prediction.

3.1 The Number of Publications

Using the built-in function *Fit* in Maple and the database *MR* in *DATA.txt* we have built already, it follows that the coefficient in model $y = ax + b$ is $[698.117924677398, 108267.583295999]$.

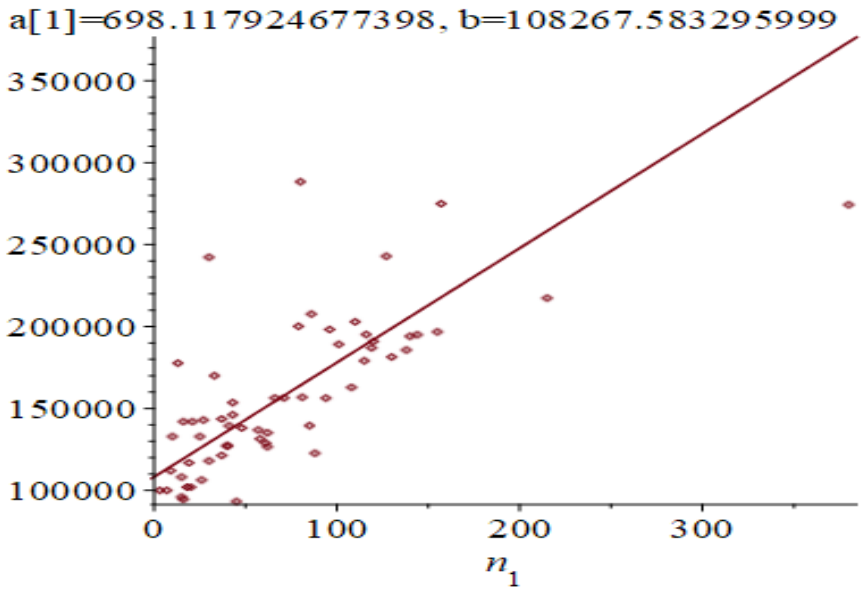

Figure 2: The simple regression model to predict salaries from the number of publications

where x denotes the **number of publications** and y denotes the **salary**.

3.2 The Number of Citations

Similarly, using the built-in function *Fit* in Maple and the database *MR DATA.txt*, it follows that the coefficient in model $y = ax + b$ is $[39.9464245189426, 117249.456948536]$.

The following is a graph of the model

where x denotes the **number of citations** and y denotes the **salary**.

3.3 The Numbers of Publications and Citations

Using the built-in function *Fit* in Maple and the database *MR*, it follows that the coefficient in model $y = a_1x_1 + a_2x_2 + b$ is $[552.468114188644, 10.9647258343195, 107438.462275095]$.

The following is a graph of the model where x_1 denotes the **number of publications**, x_2 denotes the **number of citations** and y denotes the **salary**.

4 Multi-Variable Regression to Predict Salaries

In this section, we develop multi-variable linear regression models for the dependent variable *Base Salary*. In particular, we will be comparing two regression models. In the first model, we take

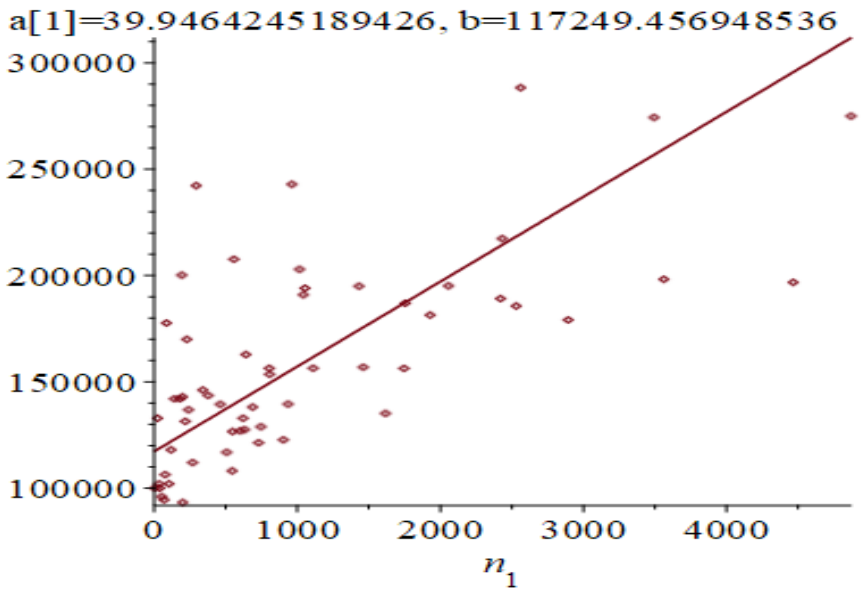

Figure 3: The simple regression model to predict salaries from the number of citations

the independent variables to be *Number of Publications*, *Number of Citations*, and *H-index*. In the second model, we include the same three independent variables in addition to a fourth variable *Year of PhD*.

For the obvious reasons, an R1 University would most likely value the research efforts of the professors the University employs. Does a research-intensive university, however, compensate productive research efforts? To gain insight into this question, we examine a linear multiple-regression model considering three independent variables: *Number of Publications*, *Number of Citations*, and *H-index*. These three variables are often-used proxies for researcher productivity and research impact, and so these three variables would presumably have some relationship with the base salary a professor earns.

To examine this relationship, we used Maple's Statistical capabilities to develop a multiple-regression model with *Base Salary* as the dependent variable. The code and data used to develop the model is available at the project website.

Using Maple's Fit(f,X,Y,v) procedure, we produced the following regression model:

There are a few important observations to make about the model above:

First, note that of the three variables of interest, the only one that achieved a level of significance is the Number of Publications ($p < .005$). On the other hand, both the H-Index and the Number of Citations were far from the standard thresholds for statistical significance. Based on our sample of 59 professors from a single R1 University, we fail to reject the hypothesis that the *Number of Citations* and the *H-Index* do not predict *Base Salary*.

Second, the model we've generated here accounts for approximately 58% of the variation in the sample. Our model was a fairly good predictor of *Base Salary* at the institution from which we

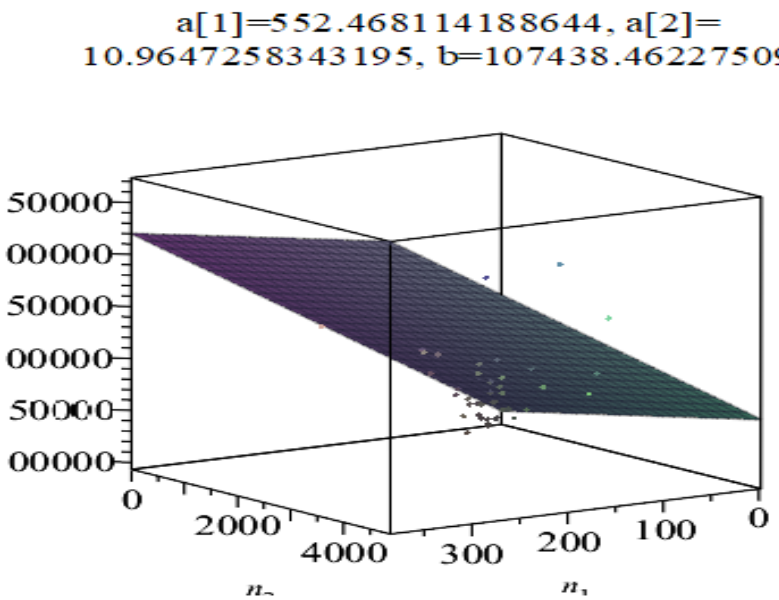

Figure 4: The simple regression model to predict salaries from the numbers of publications and citations

Model: y=102213.93+323.73663x[1]+0.19726192 x[2]+2289.8942x[3]				
Coefficients	Estimate	Standard Error	t	P>|t|
NumOfPubs	**323.737**	**98.2510**	**0.3.29500**	**0.00172617**
NumOfCites	0.197262	9.80409	0.0201204	0.984020
HIndex	2289.89	1387.26	1.65065	0.104509
Constant	102213.93	11871.3	8.61019	0.00000
R-Squared	0.581963			
Adjusted R-Squared:	0.559161			

Table 9: The Three Variable Regression Model considering *Number of Publications*, *Number of Citations*, and *H-Index*. Statistically significant predictors are printed in bold.

collected the data. Given that only one of the three variables was a statistically significant predictor, however, could we potentially do better? In the next section, we explore the accuracy of the model using one additional variable: The year the Ph.D. was received.

Folk wisdom indicates that with increasing age comes increasing wisdom, though the authors of this paper have no convincing empirical evidence that this claim is true. It may be the case, however, that with age of a terminal research degree comes an increase in *Base Salary*. We added this fourth variable, *Year of PhD*, to explore how our model was affected by the amount of time a professor has spent in their career outside of the PhD program. Below is the model generated by Maple. As before, the Maple code and the data is available at the project website.

We again make some interesting observations about this model.

Model: y=1.88663*10^6+220.97760*x[1]+2.4400278*x[2]+1542.0680*x[3]-889.20919*x[4]				
Coefficients	Estimate	Standard Error	t	P>|t|
NumOfPubs	**220.97760**	**102.186**	**2.16250**	**0.0350271**
NumOfCites	2.4400278	9.80409	0.0201204	0.790132
HIndex	1542.0680	1387.26	1.65065	0.260633
PHDYear	**-889.20919**	**351.619**	**-2.52890**	**0.0143961**
Constant	1.88663×10^6	11871.3	8.61019	0.00991043
R-Squared	0.626229			
Adjusted R-Squared:	0.598542			

Table 10: The Four Variable Regression Model considering *Number of Publications*, *Number of Citations*, *H-Index*, and *Year of PhD*. Statistically significant predictors are printed in bold.

As before, we see that *H-Index* and *Number of Citations* do not rise to the level of statistical significance in our model. On the other hand, *Number of Publications* and *Year of PhD* are both statistically significant predictors of base salary in our data ($p < .05$). The fact that *Year of PhD* negatively correlates with *Base Salary* supports the claim that professors generally earn more the later they are in their career.

In this four variable model, we've explained approximately 63% of the variance in the data, capturing more variance than the three variable model. As before, our model is a fairly good predictor of base salary. Moreover, the increase in the amount of variance captured is most likely significant. comparing the adjusted R-squared value, we see that the improvement is more than what would be expected by chance alone. Thus, we have reason to believe that *Year of PhD* is an acceptable variable to add to our model.

5 Logistic Regression to Predict Rank

The file `EM19-Sec5.txt` contains methods for doing logistic regression, and also methods to get the data in the proper format.

`NormPrep(D,Descs,Target,k,L)` reorganizes a data set in the usual format as [[Descriptive Features],Target Feature] (where the chosen descriptive features are given by Descs and the chosen target feature is given by Target). Instead of using the Target feature directly, we change it to 0 or 1: it will be 1 if it has value $\geq k$, and 0 otherwise. We then normalize the i-th descriptive feature by dividing by $L[i]$. The $L[i]$ should be chosen as approximate upper bounds for the values of the descriptive features, so that the new descriptive features take values approximately between 0 and 1.

`LogRegress(D,rate,K,x,tolerance,numTries)` takes a data set D in the format outputted by `NormPrep` and uses gradient descent to find the weights of the logistic regression function. The algorithm stops when it runs more than numTries iterations or when the sum of squared errors is less than tolerance.

First we calculated the data sets using publications, citations, and h-index to predict rank, normalizing by 400, 5000, and 30 respectively. We let k vary from 2 to 4. The command to do this is:

```
for k from 2 to 4 do

S[k]:=NormPrep(MR, [4,5,6],3,k,[400, 5000, 30]);

od:
```

We then ran logistic regression on each of those data sets with the following commands:

```
for k from 2 to 4 do

W[k]:=LogRegress(S[k], 2.0, 1.0, x, 0.01, 100000);

od;
```

Note that we made the tolerance very low: we decided to essentially discard the tolerance, instead recalculating each set of weights 100,000 times. we chose a step size of 2.0 because experimentally, it seemed to give smaller errors faster for the $k = 4$ case.

My output has the form [sum of squared errors, regression function]. For $k = 2, 3, 4$ respectively, we obtained:

- $[1.949937354, 1/(1. + \exp(2.016456500 - 28.67062169 * x[1] - 154.0659487 * x[2] + 10.21190166 * x[3]))]$.
- $[2.405193364, 1/(1. + \exp(8.323091810 + 10.98823741 * x[1] + 18.38531024 * x[2] - 42.23399707 * x[3]))]$.
- $[5.233781077, 1/(1. + \exp(8.135517535 - 34.32290447 * x[1] + 7.142634696 * x[2] - 7.459781090 * x[3]))]$.

Note that after 100,000 tries, the error for the $k = 4$ case is still very high compared to the others. (Further, with fewer than 10,000 tries one can get the errors for $k = 2$ and $k = 3$ to be below 3.0.) This seems to indicate that the distinction between rank 4 and the rest (distinguished or non-distinguished professor) is not as easily predicted by number of publications, number of citations, and h-index.

6 Lasso Regression to Predict Salaries

Lasso Regression is an extension of linear regression that tries to remove "redundancy" by pushing unnecessary coefficients close to 0. For instance, if we are trying to determine a way to guess a variable z using a linear function in the variables v, x, y and in all our data, $v \approx 2x + 3y$, we may want to make the coefficient of v zero since we really can't measure its impact based on our data.

Recall that for standard linear regression, we seek to find a vector $\vec{w}$ such that we minimize

$$\sum_i (\vec{w} \cdot \vec{x}_i - y_i)^2.$$

If we wish to add a constant vector, we can append 1 to the end of each element of our data set.

For Lasso regression (with parameter λ), we seek to minimize

$$\sum_i (\vec{w} \cdot \vec{x}_i - y_i)^2 + \lambda |w|$$

As we make λ larger, we will see some coefficients push to 0; in the limit all coefficients will become 0 (which is meaningless). In practice, one will use split the data into training data, validation data and testing data. The training data is used to compute the $\vec{w}$ (which depends on λ) and the validation data is used to determine which value of λ is best. The test data is used to measure how good the final choice is.

The method we use in this paper is coordinate descent with soft thresholding. This is one of the older methods but its complexity suffices.

In this section of the project, we seek to estimate a professor's salary based on their number of citations and publications and a professor's rank based on these numbers and the year they graduated from a Ph.D. program. One can expect these inputs to be strongly correlated, so Lasso regression may be preferred to standard linear regression, but we need a lot more data to know this for sure.

Unfortunately, coordinate descent does not converge when appending a 1 to the end of our data points [This is something I will look more into. It makes no sense and I haven't been able to fully verify the derivation of the method used;‘ when that is done it may shed some light]. But it does converge without. Since we are not using a constant term, we use years since earning a Ph.D. instead of the year they earned a Ph.D.

First, we approximate $\vec{w}$ with $\vec{x} = \{num\ publications,\ num\ citations\}$ and y representing the base salary. We use error tolerance .01

For $\lambda = 10^i$, we have:

i	$\vec{w}$
-3	$[1204.802738, 20.48713152]$
0	$[1204.802735, 20.48713167]$
1	$[1204.802702, 20.48713341]$
2	$[1204.802380, 20.48715046]$
3	$[1204.799146, 20.48732224]$
4	$[1204.766829, 20.48904015]$
5	$[1204.443636, 20.50621793]$
6	$[1201.211721, 20.67799697]$
7	$[1168.892572, 22.39578733]$
8	$[845.7010724, 39.57369030]$
9	$[0, 82.08941057]$
10	$[0, 49.05144060]$

Next, we approximate $\vec{w}$ with $\vec{x} = \{$*num publications, num citations, 2019-year of Ph.D.*$\}$ and y representing the rank (recall that 1 for Asst. Professor, 2 for Assoc. Professor, 3 for Professor, 4 for Distinguished Professor). We use error tolerance .01.

For $\lambda = 10^i$, we have:

i	$\vec{w}$
-3	$[0.02917311079, 0.001673821023, 0.09394253627]$
0	$[0.02917217085, 0.001673817356, 0.09393501152]$
1	$[0.02916370288, 0.001673784318, 0.09386722104]$
2	$[0.02916370288, 0.001673784318, 0.09386722104]$
3	$[0.02907902313, 0.001673453938, 0.09318931622]$
4	$[0.02823222572, 0.001670150141, 0.08641026800]$
5	$[0.01976425160, 0.001637112171, 0.01861978578]$
6	$[0, 0.001306732472, 0]$
7	$[0, 0, 0]$

7 Ridge Regression to Predict Salaries

In this section, we are going to find whether our data have multicolinearity or not with Ridge regression. Additionally we will use this method to predict salary from [NumberOfPublications, NumberOfCitations] and predict the rank from [NumberOfPublications, NumberOfCitations, YearOfPhD].

7.1 Ridge Regression

Here we consider a general situation to introduce our method which uses Ridge regression to predict. For all observations i, we denote the i-th observation on the k-th explanatory variable by X_{ki} and the outcome variable by Y_i. The we can explore the issue caused by multicolinearity by examining the process of attempting to obtain estimation for the parameters of the multiple

regression equation

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki}.$$

Further we get

$$Y = X\beta^T$$

where

$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{k1} \\ 1 & X_{12} & \dots & X_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1N} & \dots & X_{kN} \end{bmatrix}$$

is an $N \times (k+1)$ matrix, where N is the number of observations and k is the number of explanatory variables.

Here we want to get the approximate values for β. We apply the Tikhonov regularization which is also known as ridge regression in statistics. Then the explicit formula of β given by this method is

$$\beta := (X^T X + \Gamma^T \Gamma)^{-1} X^T Y$$

where in most situations $\Gamma = \alpha I$. Here α is to be determined and I is the identity matrix. Moreover, α is always determined by cross-validation method. Further, we use 70 percent of the data set to train and the remaining data to test.

7.2 Error

It is a good way to use MSE to test the errors. In our problem, it is indeed an accessible way, but when we use it to test the prediction of salary, we found it extremely upset because of the large difference (around 10^8). It will become a big challenge for us to judge if our method is appropriate or not. We can't say it is definitely an inappropriate way because of the big difference, since we have to notice that the value of Y, i.e., the salary, is large, so some perturbation will be amplified. Thus we have to look for a new method to test our error and we adopt the new formula for error as follows

$$E := \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\frac{Y_i - \hat{Y_i}}{Y_i})^2}.$$

7.3 Prediction for Salary

In this part we use [NumberOfPublications, NumberOfCitations] to construct a linear prediction for salary. By apply our algorithm. First input

```
Main(0.7, 1, [4, 5, 7],2)
```

where 0.7 means we use 70% to train and use the remaining to test and 1 means we select α from $[0, 1]$ where we choose 0.1 to be initial step and the 2 means *way* that is if $way = 1$ then output a

picture if $way = 2$ then output solution combined by error and β.. It will output the data and the corresponding error. And the output is as following

$$[[[102536.973741645, 523.186015435636, 27.1832340825623]], 0.00331248298977668]$$

which means the error the this model is only 0.331%. Further we can write the formula for our model.

$$\begin{aligned} Rank :=& 102536 \\ &+ 523.186 * NumberOfPublications \\ &+ 27.1832 * NumberOfCitations \end{aligned}$$

And the picture which is consists of the initial value of Y and the $\hat{Y}$ is as Figure1.

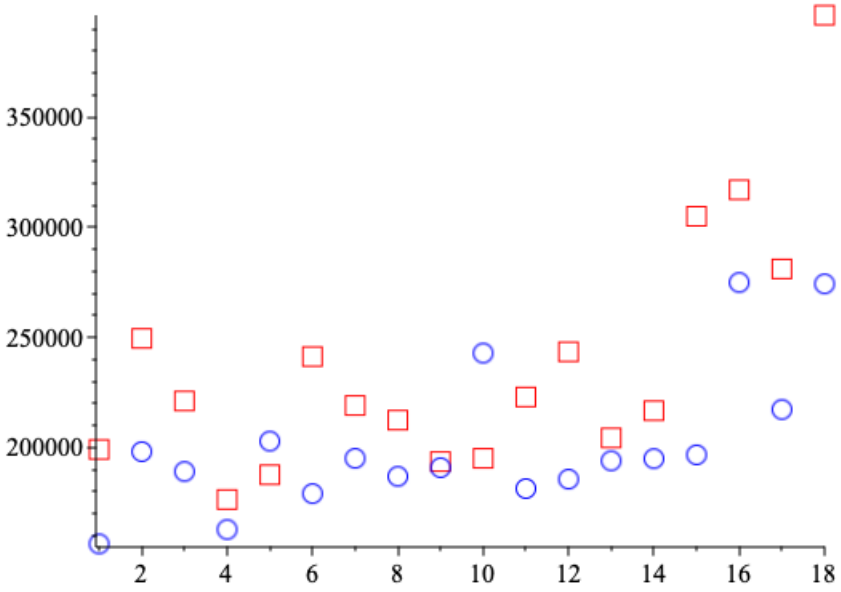

Figure 5: $\hat{Y}$ in red&box and Y is in blue&circle

7.4 Prediction for Rank

In this part we use [NumberOfPublications, NumberOfCitations, YearOfPhD] to construct a linear prediction for rank. By apply our algorithm. First input

```
Main(0.7,1,[4,5,9,3],2)
```

The meaning of parameters are exactly the same as above.

And the output is the following list

$$[[[65.75303125, 0.01315241331, 0.0002925868813, -0.03189812789]], 0.005672106778]$$

which means the error the this model is only 0.567%. Further we can write the formula for our model.

$$\begin{aligned} Rank :=& 65.753 \\ &+ 0.01315 * NumberOfPublications \\ &+ 0.00029 * NumberOfCitations \\ &- 0.031898 * YearOfPhD \end{aligned}$$

And the picture which is consists of the initial value of Y and the $\hat{Y}$ is as Figure2

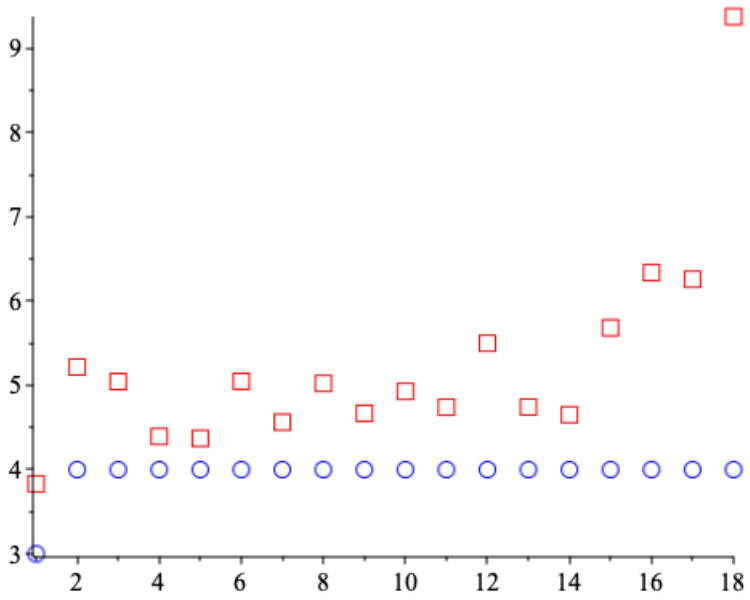

Figure 6: $\hat{Y}$ in red&box and Y is in blue&circle

8 Neural Network to Predict Rank

Artificial neural networks (ANNs) are used most often to extract complex relationships within a data set. While our data set is currently small, we thought that it would be interesting to explore how well an ANN classifies our data. In this section, we will work with the following problem: Given a professor, who is represented by a list of length three containing the number of publications, the number of citations on these publications, and h-index, we would like to predict what rank this professor has. In order to use an ANN for this task, our final model should output a vector which has length of the number of possible rankings whose elements are between zero and one and whose entries sum to 1. We will begin with a simple linear classifier, and after experimenting with this simple model, we may/will see if adding non-linearity through a second layer allows us to more accurately model our data set.

8.1 Methodology

One of the simplest forms of an ANN is a one-layer linear classifier. We shall follow the article `http://cs231n.github.io/neural-networks-case-study/#linear` from the Stanford cs231n course with several modifications. The model explained in this article is known as a *soft-max linear classifier*. In this model, our data undergoes a linear transformation from some k-dimensional real space to N-dimensional real space, where k is the number of descriptive features of the data and N is the number of target features. We interpret this N-dimensional vector as a list of unnormalized log probabilities, and we apply the soft-max function which element-wise exponentiates and normalizes this vector to obtain a list of probabilities.

We will train our neural network with hand-labeled (by the Rutgers Mathematics promotion committee) data, which is a list of professors and current rankings in the format [descriptive feature 1, descriptive feature 2,..., descriptive feature k, rank]. Descriptive features may be chosen from the following: number of citations, number of publications, h-index, salary, AMS fellowness, and year of receiving PhD. Before training the ANN, we do the following preprocessing on our training set data: For each descriptive feature F, we transform F so that it has mean zero and standard deviation one. In addition, since our model predicts probabilities, we convert the number professor rank into a length-four vector (probability distribution), which is a one at position i if the professor is of rank i and zero otherwise. This is known as a *one-hot* encoding of the target feature. Using this encoding of the target feature, we can compute how far wrong our model's current prediction is from the truth. To this end, we use the cross-entropy loss function. For two probability distributions p, the true distribution, and q, the test distribution, on a base set X, the cross-entropy $L(p, q)$ is defined as

$$L(p, q) = \sum_{x \in X} -p(x) \log(q(x)).$$

Using our one-hot encoding of the target feature, our loss for a single piece of data is thus

$$-\log(q(x_i)),$$

where i is the correct label for this piece of data. We sum over all of the training data to get the loss for a single iteration (epoch) of training. Using the loss function, we back-propagate the error after each epoch to update the weights of our ANN. In addition, we also update the ANN weights with a small amount of *regularization*, which keeps the weights closer to zero. The purpose of this is to prevent over-fitting our data and to encourage use of all target features by the ANN.

8.2 Results

We start with training the neural net using all available numerical descriptive features other than salary. We permute the data after extracting the relevant fields and take the first 45 entries to train the ANN. The rest we set aside for testing. After experimenting with hyper-parameters, we find that the network seems to converge after 200 epochs. Below is a plot of the cross-entropy loss for each epoch of training on this data set.

To test the trained network, we let the network's prediction of a given professor rank be the argmax of the list of probabilities. We may now evaluate the accuracy on the training set and find that the

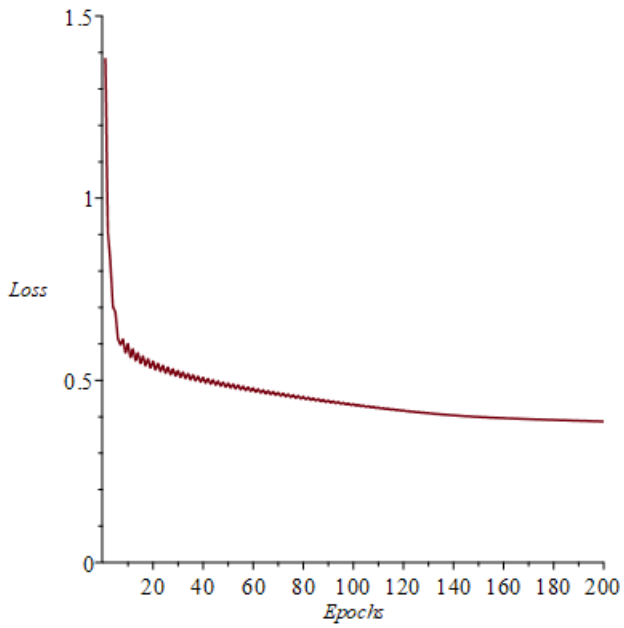

Figure 7: The cross-entropy loss for each epoch of training

ANN predicts professor rank correctly 12 out of 14 times! The list of predictions by the network is [4, 4, 4, 4, 2, 4, 3, 4, 3, 3, 4, 3, 1, 3], and the correct rankings are [4, 4, 4, 4, 2, 4, 4, 4, 3, 3, 4, 3, 1, 2].

It is interesting to see how our model performs with using fewer descriptive features. It seems that the number of publications, the number of citations, and the h-index are particularly important criteria, so we use these to train the ANN. Using the same hyper-parameters, the neural net trains well after 200 epochs. The loss curve is similar, yet we find that the ANN predicts the ranking correctly only 9 out of 14 times. Here is the list of predictions [3, 2, 3, 1, 3, 3, 4, 3, 3, 4, 4, 3, 4, 1] and the true rankings [3, 3, 3, 1, 2, 3, 4, 4, 4, 4, 4, 3, 4, 2]. It is instructive to plot the data to see how it is clustered. Below is a plot of the three-dimensional data where each points color corresponds to the ranking of the professor. Magenta corresponds to assistant professor, blue to associate professor, green to professor, and orange to distinguished professor.

One can see that the data is not linearly separable, and in fact does not seem to be separable by any simple non-linear model. Future investigation could include finding a small number of parameters which allow for a linear separation of the data or seeing how well a non-linear model predicts professor rankings. To experiment, one can follow the help screen in a maple worksheet after reading `EM19-Sec8.txt`.

9 Unsupervised Clustering For Predictive Analysis

The method used in this section was an unsupervised clustering algorithm developed by the 2015 UCLA Applied Math REU Hyperspectral Imagery research team [1],[2]. Hyperspectral images are

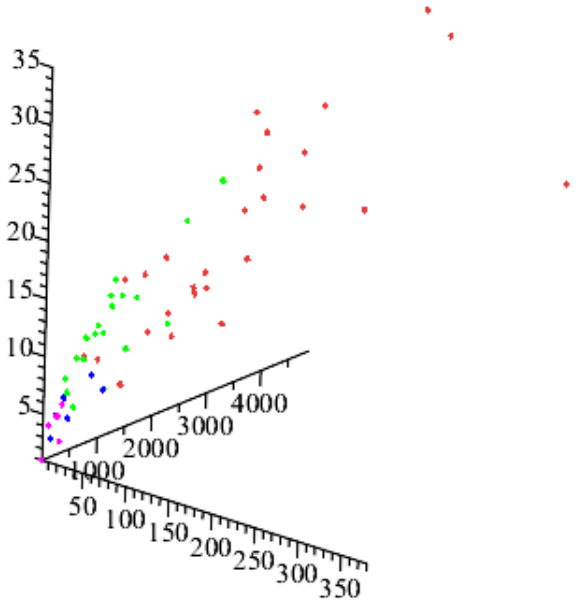

Figure 8: The three-dimensional data where each points color corresponds to the ranking of the professor.

"photos" taken with special sensors that allow for each pixel to have measurements from hundreds of different wavelengths. Thus instead of each pixel being identified by an RGB value, it would instead by identified by a vector of data called its *spectral signature*. The spectral signature of a pixel can identify its material content, i.e., grass or brick or asphalt, or even differentiate clear chemical gas spills from clean air. Oftentimes, there are only five or six distinct materials in a hyperspectral image to try to identify. Analysis of hyperspectral imagery therefore is focused on taking hundreds of datapoints, each in the form of spectral signatures of pixels, and sorting them into n groups, where n is the number of distinct materials expected to be in the image. The following terminology will be borrowed from the hyperspectral lexicon: each sorted group is called a *cluster*, and the average vector of a cluster is its *centroid*.

Most hyperspectral imagery analysis techniques require previous knowledge of the materials present in the image (such as what the spectral signature of each material is) beforehand in order to properly cluster the image. The NLTV algorithm developed by the 2015 UCLA Hyperspectral team was amongst a rarer class of algorithms that would cluster the data in an unsupervised manner. Simply input the image, and the clustered image would be output. As such, it seemed like an incredibly promising method to adapt for the dataset of professors and their salaries. There are four natural "clusters" within the dataset: Assistant Professor, Associate Professor, Professor, and Distinguished Professor. If the datapoint "signatures" of publications, citations, h-index, base salary, AMS Felloship, and year of PhD were truly distinct between these four classifications of professorship, then an unsupervised clustering method meant for vectors of data ought to predict

9.1 The Algorithm

The core of the sorting algorithm comes from the minimization of an energy functional

$$E(u) = \| \bigtriangledown u \|_{L1} + \lambda \langle u, f \rangle, \tag{9.1}$$

where $u : \Omega \to [0,1]^n$ is the *labeling function* on the data, n is the number of clusters it is being sorted into, and Ω is the domain of the data, and f is a fidelity function. The inspiration comes from the imaging process technique of *total variation* introduced by Rudin et al in 1992 [6] for noise reduction, which corresponds to the minimization of the gradient of u. In highly noisy images or datasets where adjacent pixels do not matter, simply calculating the gradient directly does not give as pertinent information. Therefore, we turn to the theory of nonlocal operators introduced by [4],[5], Zhou and Schölkopf and adapted to image processing by Osher and Gilboa [7].

Let Ω be a region in $\mathbb{R}^k$, and $u : \Omega \to \mathbb{R}$ be a real function. Then the non-local derivative is defined as

$$\frac{\partial u}{\partial y}(x) := \frac{u(y) - u(x)}{d(x,y)}, \quad \text{for all } x, y \in \Omega \tag{9.2}$$

where d is a positive distance between x and y. With the following non-local weight defined as 9.3, we can re-write the non-local derivative as 9.4.

$$w(x,y) = d^{-2}(x,y) \tag{9.3}$$

$$\frac{\partial u}{\partial y}(x) = \sqrt{w(x,y)}(u(y) - u(x)) \tag{9.4}$$

Then the non-local gradient $\bigtriangledown_w u$ for $u \in L^2(\Omega)$ as a function from Ω to $L^2(\Omega)$ is the collection of all partial derivatives

$$\bigtriangledown_w u(x)(y) = \frac{\partial u}{\partial y}(x) = \sqrt{w(x,y)}(u(y) - u(x)). \tag{9.5}$$

Note that here, "distance" can either refer to the standard Euclidean distance

$$d(x,y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}, \tag{9.6}$$

the cosine distance

$$d(x,y) = 1 - \frac{x \cdot y}{||x||||y||}, \tag{9.7}$$

or a linear combination of them.

The non-local energy functional we are trying to minimize takes the form of

$$E(u) = \| \bigtriangledown_w u \|_{L1} + \lambda \sum_{i=1}^{n} |u_i(x) g(x) - c_i|^2, \tag{9.8}$$

where $\| \bigtriangledown_w u \|_{L1}$ is the L^1 norm on the space $L^2(\Omega, L^2(\Omega))$ defined as

$$\| v \|_{L^1} := \int_\Omega \| v(x) \|_{L^2} \, dx = \int_\Omega \Big| \int_\Omega v(x)(y)^2 dy \Big|^{\frac{1}{2}} dx \tag{9.9}$$

and the fidelity function is explicitly given by $\lambda \sum_{i=1}^n |u_i(x)g(x) - c_i|^2$, where $g(x)$ is the datapoint and c_i is the ith cluster centroid. We explicitly discretize the labeling function and nonlocal operators, $u = (u_1, u_2, \ldots, u_n)$ is a matrix of size $m \times n$, where m is the number of datapoints and n is the number of clusters. Each u_i takes values between 0 and 1, $\sum_{i=1}^n u_{ki} = 1$ for all $k \in 1, ..., m$. Then $(\bigtriangledown_w u_l)_{i,j} = \sqrt{w_{i,j}}((u_l)_j - (u_l)_i)$ is the nonlocal gradient of u_l; $(\mathrm{div}_w v)_i = \sum_j \sqrt{w_{i,j}} v_{i,j} - \sqrt{w_{j,i}} v_{j,i}$ is the divergence of v at i-th datapoint; and the discrete L^1 norm of $\bigtriangledown_w u_l$ are defined as:

$$\| \bigtriangledown_w u_l \|_{L^1} = \sum_i \left(\sum_j (\bigtriangledown_w u_l)_{i,j}^2 \right)^{\frac{1}{2}}. \tag{9.10}$$

The functional 9.8 is convex, so a global minimum exists. However, calculating $\| \bigtriangledown u \|_{L^1}$ via gradient descent involves calculating $\mathrm{div}(\frac{\bigtriangledown u}{|\bigtriangledown u|})$, which is highly unstable because $| \bigtriangledown u |$ can be equal to zero. In 2011, Chambolle and Pock introduced a first-order primal dual algorithm, which they proved converged to a saddle point with a rate of $O(1/N)$ in finite dimensions for the complete class of convex problems [8]. This was used as an inspiration to craft a saddle point solution with respect to u,$\overline{u}$, and p. Full motivation and description can be found in [1], [2],[3]. The algorithm is as follows:

Primal-Dual Iterations
• Iterations ($n > 0$): Update $u^n, p^n, \bar{u}^n$ as follows:

$$\begin{cases} p^{n+1} = \mathrm{proj}_P(p^n + \sigma \bigtriangledown_w \bar{u}^n) \\ u^{n+1} = \arg\min_u \delta_U(u) + \frac{1}{2} \| (I + \tau F)^{\frac{1}{2}} u - (I + \tau F)^{-\frac{1}{2}} (u^n + \tau \mathrm{div}_w p^{n+1}) \|^2 \\ \bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n) \end{cases}$$

where F is the discretized fidelity function matrix with the inbuilt weight λ.

The overall sorting algorithm is then:

Nonlocal Total Variation Unsupervised Clustering
• Initiate parameters.
• Calculate weight matrix.
• Set n random datapoints as the first iteration of centroids, set
$u^0 = \overline{u}^0 = \mathrm{Matrix(m,n,1/}m)$ and p^0 zeroed out.
while not converge do
Inner Loop: Primal Dual Algorithm to find minimizing u.
Outer Loop: Threshold u into an assignment function, and use the new sorting of the data to update the centroids.
end

There are two main changes between the algorithm written for this project, and the algorithm developed in 2015. Firstly, the calculation of the weight matrix is done directly between all datapoints in this project; the original hyperspectral algorithm used a "patch" distance to filter for noise, and employed an approximate nearest neighbor search to save computational time. Secondly, a smart simplex clustering method instead of directly threshholding was developed for the hyperspectral with inspiration from [9]. While a coded version in maple has been submitted, it was not used in the analysis of the data as it makes the outerloop of the algorithm far more computationally expensive, for no increase in convergence time in a dataset as clean as this one.

There are a number of parameters involved in the algorithm, but the two most vital ones are λ, which determines the weight given to the minimization of the fidelity function vs the gradient of u, and the choice of Euclidean vs Cosine distance for the creation of the weight matrix and fidelity distance calculations. The value for λ ought to be comparatively large to prioritize tight sorting. Euclidean vs Cosine vs a linear combination is something that should be tailored to the dataset. In hyperspectral imagery, oftentimes using the Cosine distance instead of the Euclidean distance could account for differences in direct lighting on the materials present in the image. Here, it is an option because some of the fields (ie h-index or AMS Fellow: 0/1) have a smaller range of values, and some of the fields (ie salary or number of citations) have a much larger range of values, so that field does not dominate. The data can also be "normalized" via dividing by the max value of each of the fields before clustering.

9.2 Analysis Of Data

A thorough analysis of the data with different parameters and subfields is done in this section. The number of assignments to each cluster and centroids of the cluster are reported. The fields are as follows: Rank (Assistant Professor=1, Associate Professor=2, Professor=3, Distinguished Professor=4), Number of Publications, Number of Citations, MRhIndex, Base Salary, AMSfellow(Yes=1,No=0), and Year of PhD.

For the clustering algorithm applied to the entirety of the numeric data collected, a few interesting patterns emerged. Oftentimes the lower ranks (Assistant Professor and Associate Professor) are grouped together, and a "pure" Distinguished Professor and mixed Distinguised Professor and Professor groups are output. When data is normalized first, clusters are entirely determined by AMS yes or no, and depending on the weighting of λ there is either one entirely "no" cluster or two entirely "no" clusters.

Test 1

Subfields: All (Rank, number of publications, number of citations,h-index, base salary, AMS Fellow, year of PhD).

Parameters: Euclidean distance, $\lambda = 10^6$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	14	2	24.9	260.5	7.1	106652	.14	2002
Centroid 2	23	3.17	52	658.3	12.9	141536	.57	1991
Centroid 3	17	3.82	112.4	1744.3	19.5	192245	.82	1980
Centroid 4	5	4	154.8	2436.2	22.8	264564	.80	1971

Test 2

Subfields: All (Rank, number of publications, number of citations,h-index, base salary, AMS Fellow, year of PhD).

Parameters: Euclidean distance, data originally normalized $\lambda = 10^{15}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	11	1.45	21	82.3	5	112367	0	2011
Centroid 2	15	3.3	50.7	572	11.8	147092	0	1985
Centroid 3	21	3.5	68.3	770	14.1	158551	1	1988
Centroid 4	12	4	150.1	2914	26.2	213953	1	1974

Test 3

Subfields: All (Rank, number of publications, number of citations,h-index, base salary, AMS Fellow, year of PhD).

Parameters: Euclidean distance, data originally normalized, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	26	2.5	38.1	364.8	8.9	132401	0	1996
Centroid 2	10	3	40.5	432.8	10.9	138789	1	1993
Centroid 3	13	3.9	98.4	1194.4	17.8	177696	1	1983
Centroid 4	10	4	155.3	3128.5	27	219906	1	1974

Test 4

Subfields: All (Rank, number of publications, number of citations,h-index, base salary, AMS Fellow, year of PhD).

Parameters: Cosine distance, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	12	1.6	21.4	100.5	5.1	107371	0.08	2008
Centroid 2	16	3.3	51.4	756.4	14	134510	.56	1989
Centroid 3	17	3.5	70	528.4	11.7	178819	.59	1985
Centroid 4	14	3.9	139.9	2738	25.6	204207	.93	1977

Test 5

Subfields: All (Rank, number of publications, number of citations,h-index, base salary, AMS Fellow, year of PhD).

Parameters: Cosine distance, data originally normalized, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	26	2.5	38.1	364.8	8.9	132401	0	1996
Centroid 2	10	3	40.5	432.8	10.9	138789	1	1993
Centroid 3	11	4	96	1077	16.8	179301	1	1982
Centroid 4	12	3.9	148.1	2913	26.4	211400	1	1977

It is also possible to run tests on limited data. Here we use cosine distance on non-normalized data. Once more, the algorithm skewed towards lumping associate and assistant professors together, and making sub-clusters of distinguished professors. The pair of subsets that best subcategorized centroids were [H-index, Year of PhD].

Test 6

Subfields: Number of publications,base salary

Parameters: Cosine distance, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	27	2.4	27.4	308	8.7	124973	.30	2000.
Centroid 2	16	3.6	74.8	1105.9	15.9	169494	.68	1981
Centroid 3	14	4	121.1	2049.3	21.6	197257	.86	1979
Centroid 4	2	4	297.5	2962.5	25	245834	1	1966

Test 7

Subfields: Number of citations,base salary

Parameters: Cosine distance, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	23	2.2	24.2	220	7.3	122726	.22	2001
Centroid 2	21	3.6	66.5	1094.8	16.3	163636	.71	1985
Centroid 3	11	4	125.3	1849.7	20.8	192613	.82	1977
Centroid 4	4	4	224	3056.5	25.3	240409	1	1973

Test 8

Subfields: h-index, year of PhD

Parameters: Cosine distance, $\lambda = 10^{20}$.

	# Elem.	Rank	Pub.	Cit.	H-Ind.	B. Salary	AMS	Year of PhD
Centroid 1	17	2.1	28.1	310.2	7.8	118049	.24	2003
Centroid 2	15	3.3	59.1	904.7	13.5	156430	.47	1983
Centroid 3	19	3.6	88.4	1115	16.4	175803	.79	1986
Centroid 4	8	4	148	2574.4	24.5	205736	.86	1978

The overall pattern emerges that NLTV Unsupervised clustering is indeed good at picking out four clusters that do correlate to rank, but consistently there is more variation between Distinguished Professors within the given data than Assistant and Associate Professors, which skews the clustering.

10 Summary

In this class project, multiple mathematical and statistical methods are used to predict the ranks and salaries of a faculty member from other independent variables such as the number of publications and the number of citations. Various results are obtained. The future work could be the extension of this project to other math departments in the United States or even to faculties of other fields. It would be interesting to see the criteria to be a professor and how much they earn in different universities and different fields.

References

[1] Wei Zhu, Victoria Chayes, Alexandre Tiard, Stephanie Sanchez, Devin Dahlberg, Da Kuang, Andrea Bertozzi, Stanley Osher, Dominique Zosso, *Nonlocal total variation with primal dual algorithm and stable simplex clustering in unspervised hyperspectral imagery analysis*. Technical report, CAM report 15-44, UCLA, 2015.

[2] Wei Zhu, Victoria Chayes, Alexandre Tiard, Stephanie Sanchez, Devin Dahlberg, Andrea L Bertozzi, Stanley Osher, Dominique Zosso, Da Kuang, *Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm*. IEEE Transactions on Geoscience and Remote Sensing, Vol 55 Issue 5 pg 2786-2798, 2017.

[3] Wei Zhu, *Nonlocal Variational Methods in Image and Data Processing*. PhD thesis, UCLA, 2017.

[4] D. Zhou and B. Schölkopf. "Regularization on discrete spaces". Springer, Berlin, Germany, pp. 361-368.

[5] D. Zhou and B. Schölkopf. "Discrete regularization". MIT Press, Cambridge, MA, pp. 221-232.

[6] L. I. Rudin, S. Osher, E. Fatemi. "Nonlinear total variation based noise removal algorithms." Physica D. 60: 259268, 1992.

[7] Guy Gilboa, Stanley Osher. "Nonlocal Operators with Applications to Image Processing." SIAM Multiscale Model. Simul. 7(3), 1005-1028, 2008

[8] Antonin Chambolle and Thomas Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging." Springer, Journal of Mathematical Imaging and Vision. 40(1), 120-145, 2011.

[9] Nicolas Gillis, Da Kuang and Haesun Park. "Hierarchical Clustering of Hyperspectral Images Using Rank-Two Nonnegative Matrix Factorization." IEEE, Transactions on Geoscience and Remote Sensing. 53(4), 2066-2078, 2015.

[10] Leslie Valiant. "Probably Approximately Correct." Basic Books, 2013.

[11] D. Kelleher, Brian Mac Namee, Aoife D'Arcy. "Fundamentals of Machine Learning for Predictive Data Analytics." The MIT Press, 2015.

Contact information of the corresponding author:

Yukun Yao, Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA.
Email: `yao at math dot rutgers dot edu` .