Text Mining Methods on Mathematical Documents: A Case Study in Fields Medalists

By

TREVOR C. CHAN

SENIOR THESIS

Submitted in partial satisfaction of the requirements for Highest Honors for the degree of

BACHELOR OF SCIENCE

in

APPLIED MATHEMATICS

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA,
DAVIS

Approved:

Jesús A. De Loera
March 2016

1 Introduction

With current technological advances, text classification has become a widely explored problem in the field of machine learning. in this case study we will be examining the effectiveness of such text classification methods on mathematical papers by Fields Medalist authors. it is our hope that analysis techniques give results that allow us as readers to recognize authors of each paper without having to read the papers themselves. The success of such methods would also mean that concepts important to specific authors can be conceptualized by a machine, which aids in our ability to learn about a given Fields Medalist without reading about him or her. Furthermore, news analysts have been applying these very machine learning methods to find patterns in news articles. In such cases, words that are deemed relevant to a topic or are heavily covered in the news are used to summarize the important events at a particular time.

It should be noted that this method utilizes certain features, such as the length and content of each paper, to automatically categorize the documents with good accuracy. We proceed by first parsing the papers into a text summarization matrix, which keeps track of the frequency of each word in every paper. This matrix is then subjected to two different methods, namely LASSO, a convex optimization method, and a network centrality method. It is my hope that analysis of these two methods, along with the creation of a third, hybrid method for text summarization will provide us a good way to differentiate which method works under a given circumstance.

2 Modeling and Solving the Problem

2.1 Data

We start by doing some basic preprocessing of our data. Our dataset consists of 185 Fields Medal Papers for 20 Fields Medal professors who work on various topics such as topology, partial differential equations, statistics, and so on. The texts are converted to lowercase, and punctuations and numbers are removed. Additionally, certain low-information words, known as stopwords, such as “for” and “the” are ignored.

After preprocessing the data, we form X, termed the text summmarization matrix. We let $D := \{1, 2, \ldots, n\}$ be the set of documents indexed by the integers, and let P be the set of one-word and two-word phrases in our dataset. Then, we have $X \in \mathbb{R}^{n\times p}$, where $n = |D|$ and $p := |P|$ with $|\cdot|$ being the cardinalities. The elements of X are denoted by x_{ij} which is the number of times phrase $j \in P$ appears in document $i \in D$. The columns are said to be the *features* and the rows are the *documents*. Hence, X is a features-document matrix, which we simply call our features matrix.

However, it may be preferrable to rescale X to reduce the probability that our feature selection methods will select an insignificant term. The two rescaling approaches that will be compared are known as the L^2 and the term frequency-inverse document frequency (*tf-idf*) rescaling techniques, as seen in [1]. These methods help reduce the variance and weight of high-frequency features (see [2]).

Let $X^{(l)} = [x^{(l)}_{ij}] \in \mathbb{R}^{n\times p}$ be a L^2 rescaled version of X when its columns are normalized under the L^2 norm:

$$x^{(l)}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}}. \tag{1}$$

Let $X^{(t)} = [x_{ij}^{(t)}] \in \mathbb{R}^{n \times p}$ be a *tf-idf* rescaled version of X when it is rescaled as follows:

$$x_{ij}^{(t)} := \frac{x_{ij}}{q_i} \log(\frac{n}{d_j}), \text{ where } q_i = \sum_{j=1}^{p} x_{ij} \text{ and } d_j = \sum_{i=1}^{n} \mathbf{1}\{x_{ij} > 0\}, \tag{2}$$

where $\mathbf{1}\{x_{ij} > 0\}$ is an indicator function equal to 1 when $x_{ij} > 0$ and equal to 0 otherwise.

Now, we form a vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ for the classification. To do this, we define a subject of interest known as the *query*. In our case, we are interested in certain professors, so our query is a professor's name. If document $i = 1, 2, \ldots, n$ is written by the professor of interest, then $y_i \overset{set}{=} 1$, otherwise $y_i \overset{set}{=} -1$. For simplicity, we denote the set of documents related to the query (i.e. written by the professor of intersest) as D_+.

2.2 Corpus-Based Stop Word Lists

We now introduce the concept of stop words, or low information words we can remove from the features matrix without negatively impacting our selection of sparse summarizers. Typical stop words such as "the" or "and" are necessary grammatically, but typically add very little information. For example, most internet search engines have developed lists of several hundred typical stop words that are automatically removed from search queries prior to running the search algorithm. The reason is words like "the" show up far too often to help determine if a particular webpage relates to the query or not.

Unfortunately, in subject specific corpora like ours, these pre-made stop word lists often contain words that are rarely, if ever, used, such as the word "anybody". At the same time, there are many words that in a mathematical context can be thought of as stop words, such as "function", even though the same words would be very informative in a corpus consisting of a random selection of English documents. Below we describe a method that deals with these issues by creating a corpus-specific stop word list, allowing us to eliminate thousands of terms from our features matrix. See [3] for full details of the method.

Creating the Stop Word List Building the stop word list takes two steps. First we create a small list of representative stop words. In the second step we use our representative list to find the rest of the stop words in the corpus. From a machine learning point of view, the representative list is our training data we use to classify the rest of the stop words.

3 Feature Selection Methods

We now discuss the feature selection methods for obtaining our summarizers. The two feature selection methods we use are *least absolute shrinkage and selection operator (LASSO),* and *L1-penalized logistic regression (L1LR).* These different methods will reveal some important words that a single method on its own would not have found. We note that we discuss the four methods using the unweighted matrix X, however, they can also be applied to $X^{(l)}$ and $X^{(t)}$.

L1-penalized Linear Regression (LASSO) is a widely-used feature selection model which is based off of Least Square Linear Regression (LSLR). LSLR is a well-known method used to find a linear relationship between an explanatory variable and a response variable (see [4]). In our case, each y_i in the $\mathbf{y}$ vector is a response variable, and the entries in the corresponding i^{th} row of the X matrix are the p explanatory variables. That is, the $\mathbf{y}$ vector is the response vector and the X matrix is the explanatory matrix. Assuming a linear relationship between $\mathbf{y}$ and X, we write $\mathbf{y}$ as a linear combination of the p columns of X plus some error terms:

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\gamma}.$$

The coefficient vector $\boldsymbol{\beta} := [\beta_1, \beta_2, \ldots, \beta_p]^T$ is the parameter of interest. Thus, by linear relationship, we mean that $\mathbf{y}$ is linear in $\boldsymbol{\beta}$ (see [5]). The intercept vector $\boldsymbol{\gamma}$ is constant. The way to estimate $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ is through minimizing the sum of square of the error terms:

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) = \arg\min_{\boldsymbol{\beta},\boldsymbol{\gamma}} ||\boldsymbol{y} - X\boldsymbol{\beta} - \boldsymbol{\gamma}||^2.$$

A naive approach for choosing a small number of features is by to select the $\hat{\beta}_i > \epsilon$ for a selected $\epsilon > 0$. The corresponding features can be selected as the summarizer. However, this result from LSLR is difficult to interpret, because an excessive number of coefficients will potentially end up statistically significant. Since our goal is to find a sparse set of features to summarize the documents, we could simply choose the first few features corresponding to the most significant coefficients from the LSLR result. Yet, those features are not representative empirically. In theory, the reason is that the result from LSLR is largely influenced by the outliers since all of the p phrases are included in the linear combination. To minimize the effect of outliers, we penalize them by adding a L_1-norm penalty term: $\lambda \sum_{j=1}^{p} |\beta_j|$. The resulting objective function is called the LASSO:

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) = \arg\min_{\boldsymbol{\beta},\boldsymbol{\gamma}} ||\boldsymbol{y} - C\boldsymbol{\beta} - \boldsymbol{\gamma}||^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \text{ where } \lambda > 0. \tag{3}$$

Here, λ is a penalty parameter which can force $\hat{\boldsymbol{\beta}}$ to be sparse. The higher the λ, the more sparse $\hat{\boldsymbol{\beta}}$ is. Implicit in (3) is that there is a trade-off between the sum of square of the errors and the nonzero coefficients in $\hat{\boldsymbol{\beta}}$. If a coefficient's existence cause the sum of square of the errors to be large, it will be regularized to zero. Due to this trade-off, most of the coefficients will turn out to be zero as desired [4]. Note that the intercept γ is not penalized. If it is, the intercept will get close to zero, which indicates that the number of times when y_i takes on 1 equals to the number of times when y_i takes on -1. An easy way to see this is to consider a special case where the C matrix is a zero matrix. Then, the intercept is the average of the y_i's. This average being close to zero indicates that y_i takes on 1 or -1 for approximately an equal number of times. In our case, this is not true since y_i's typically equal -1, so we do not penalize $\boldsymbol{\gamma}$ (see [1]).

L1-penalized Logistic Regression (L1LR) is another type of regression analysis, but is specifically designed for classification in which the response variable is categorical. Our case is a binary logistic regression since each entry of the response vector $\mathbf{y}$ only takes on 1 or -1. The parameter of interest is the probability of y_i being 1, written as $P(y_i = 1)$, in each sample (see [6]). We would first try to directly write $P(y_i = 1)$ as a linear combination of the features represented by the i^{th} row of X:

$$P(y_i = 1) = \boldsymbol{x_i}^T\boldsymbol{\beta} + \gamma, \tag{4}$$

where $\boldsymbol{x_i} = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$, and γ is a constant (see [6]).

One problem for this model is that the right hand side can take values greater than 1 or less than 0, while the probability on the left hand side cannot. A remedy is to take the natural logarithm of the probability ratio:

$$\ln \frac{P(y_i = 1)}{P(y_i = -1)}.$$

This is called $logit(P(y_i = 1))$, which can match the range of the right hand side of (4). Moreover, $logit(P(y_i = 1))$ is symmetric, i.e. $logit(P(y_i = 1)) = -logit(P(y_i = -1))$ as explained in [6].

Now we can model $logit(P(y_i = 1))$ as a linear combination of the features:

$$\ln \frac{P(y_i = 1)}{P(y_i = -1)} = \boldsymbol{x_i}^T\boldsymbol{\beta} + \gamma.$$

Since it is more intuitive to think in terms of probabilities, we solve for $P(y_i = 1)$ by substituting $P(y_i = -1)$ with $1 - P(y_i = 1)$:

$$\ln \frac{P(y_i = 1)}{1 - P(y_i = 1)} = \boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma.$$

$$\frac{P(y_i = 1)}{1 - P(y_i = 1)} = \exp(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma).$$

$$P(y_i = 1) = \frac{1}{1 + \exp(-(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))}.$$

This is called the *logistic model*. Similarly, we can also solve for $P(y_i = -1)$:

$$P(y_i = -1) = \frac{\exp(-(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))}{1 + \exp(-(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma)}.$$

The general form of the probability for y_i is

$$P(y_i) = \frac{\exp(y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))}{1 + \exp(y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))}. \tag{5}$$

If $\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma = 0$, $P(y_i)$ is 0.5, which means that it is equally possible for y_i to take 1 or -1 . If $\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma = 1$, $P(y_i = 1)$ and $P(y_i = -1)$ are 0.73 and 0.27 respectively. If $\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma = -1$, the reverse holds. When $\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma > 1$, $P(y_i = 1)$ quickly converges to 1. When $\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma < -1$, $P(y_i = 1)$ quickly converges to 0. The region in between is ambiguous, meaning that y_i is not strongly likely to take one side [7].

In order to estimate the parameter $\boldsymbol{\beta}$ from this logistic model, we will use the technique of maximizing the log-likelihood. Likelihood is the probability that a particular sample can occur given different values of the parameter in the model (see [8]). In our case, the sample is the set of documents D, and the parameter is $\boldsymbol{\beta}$. Since D is given, the parameter $\boldsymbol{\beta}$ that can maximize the likelihood of D can be found. For simplicity, each document in D is assumed to be independent, meaning that the occurrence of one document does not affect the probability of the others (see [8]). Then the likelihood function for D is

$$\prod_{i=1}^{n} P(y_i).$$

By taking the logarithm, we obtain the so-called log-likelihood function:

$$\begin{aligned}
\log \prod_{i=1}^{n} P(y_i) &= \sum_{i=1}^{n} \log P(y_i) \\
&= \sum_{i=1}^{n} \log \frac{\exp(y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))}{1 + \exp(y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))} \\
&= \sum_{i=1}^{n} \log \frac{1}{1 + \exp(-y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))} \\
&= -\sum_{i=1}^{n} \log(1 + \exp(-y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))).
\end{aligned}$$

We maximize the log-likelihood function or, equivalently, minimize the negative of it:

$$\arg\min_{\beta, \gamma} \sum_{i=1}^{n} \log(1 + \exp(-y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))).$$

In order to force $\boldsymbol{\beta}$ to be sparse, the L_1-norm penalty term is added (see [1]). Thus, the complete objective function for L1LR is

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) = \arg\min_{\boldsymbol{\beta}, \boldsymbol{\gamma}} \sum_{i=1}^{n} \log(1 + \exp(-y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma))) + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{6}$$

Although the original purpose of L1LR is to classify the new samples based on the features selected, we are only interested in the selected features themselves. Though these features can be used for classification, we will only consider them as summarizers. As shown above, L1LR, in theory, may give better results because of the binary nature of our experiment.

3.1 Convex Optimization

We now explain how the *L1LR* and *LASSO* objective functions can be solved by noting that they are convex. A set is convex if the line segment connecting two arbitrary points of the set completely lies inside the set. Examples of a convex set in $\mathbb{R}^2$ include squares, circles, ellipsoids and so on [9]. Mathematically, if a set $V \subseteq \mathbb{R}^n$ is convex, for any $x, y \in V$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta x + (1-\theta)y \in V.$$

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if the domain of f, written as $D(f)$, is a convex set and if for all $x, y \in D(f)$ and $0 \leq \theta \leq 1$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Function $-f$ is then concave [10]. Convex sets and a convex function $f : D(f) \to \mathbb{R}$ can be connected through the epigraph of the function f, which is defined as

$$epi(f) = \{(x, y) | x \in D(f), f(x) \leq y\}.$$

This can be used to show that a function is convex, according to the following theorem.

Theorem 1. *A function $f : D(f) \to \mathbb{R}$ is a convex function if and only if* epi(f) *is a convex set.*

The penalty term in the LASSO and L1LR objective functions is a sum of the absolute functions, which can be written as $f(z) = |z|$. Since $epi(f(z))$ is a convex set, $f(z)$ is a convex function. The penalty term, which sums the absolute functions, is also convex due to the following theorem.

Theorem 2. *If $f, g : D(f) \to \mathbb{R}$ are convex functions, then the sum of f and g is also a convex function.*

Next theorem is useful to prove that the complete LASSO and L1LR objective functions are convex.

Theorem 3. *Let $f : D(f) \to \mathbb{R}$ have a continuous second derivative. f is convex on the convex set $D(f)$ if and only if $f''(x) \geq 0$ for all $x \in D(f)$.*

The LASSO objective function without a penalty term is

$$\arg\min_{\beta,\gamma} ||\mathbf{y} - X\boldsymbol{\beta} - \boldsymbol{\gamma}||^2, \tag{7}$$

which is convex. Equation (7) can be viewed as a sum of functions in the form of $f(z) = z^2$. Since $f(z)'' = 2$, (7) is convex by Theorem 2 and 3.

The L1LR objective function without a penalty term can also be proved to be convex. Let us rewrite the log-likelihood function as

$$\log \prod_{i=1}^{n} P(y_i) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma)))$$
$$= -\sum_{i=1}^{n} f(\boldsymbol{x_i}^T \boldsymbol{\beta} + \gamma),$$

where f is the *logistic loss function* in a form of

$$f(z) = 1 + \exp(-z).$$

Then,

$$f(z)' = -\frac{\exp(-z)}{1 + \exp(-z)}$$
$$f(z)'' = \frac{\exp(-z)}{(1 + \exp(-z))^2}.$$

Since $f(z)'' > 0$, the logistic loss function is convex by Theorem 1. The log-likelihood function is then concave. Thus, we take the negative of it to make it convex so that we can use convex optimization methods to solve it [10]. If we add the penalty term, the complete LASSO and L1LR objective functions are proved to be convex.

Convex optimization The standard form of a general optimization problem is

$$\text{minimize } f_0(x)$$
$$\text{subject to } f_i(x) \geq 0, i = 1, 2, ..., m.$$

The optimal solution is the $x \in \mathbb{R}^n$ that can minimize the objective function $f_0(x)$ with constraints. Note that the constraints can also be strictly equal.

The convex optimization problem requires that the objective function and the inequality constraint functions are convex, and that the equality constraint functions are linear. An important feature described in the following theorem distinguishes the convex optimization problems from the general optimization ones [10].

Theorem 4. *If $f_0 : D(f) \to \mathbb{R}$ is a convex function, then every local optimum is a global optimum.*

Consequently the global minimum of the LASSO and L1LR objective functions are guaranteed. Also, note that our objective functions are unconstrained. One method to solve an unconstrained convex optimization problem is Newton's method [10]. According to Newton's method, the optimal solution for our case is the $\boldsymbol{\beta}$ such that

$$\nabla f(\boldsymbol{\beta}) = 0, \text{ where } \nabla f(\boldsymbol{\beta}) = (\frac{\partial}{\partial \beta_1} f(\boldsymbol{\beta}), ..., \frac{\partial}{\partial \beta_p} f(\boldsymbol{\beta}))^T$$

Here $f(\cdot)$ refers to the LASSO and L1LR objective functions without the penalty term, because the Newton's method requires differentiability. Since the penalty term makes the objective functions not differentiable. To solve the objectives including the penalty terms, an efficient alternative can be to use the cyclical coordinate descent method which can be seen in [11].

4 Network Centrality

To contrast the methods presented by convex optimization, we now examine a different approach to text analysis, namely that of network theory. With networks, we can hope to exploit the ”community structures” that appear to be ever present in mathematical papers, as well as gain a visual understanding of how concepts presented by each author are related.

4.1 Preliminary Network Theory

Basic network theory dictates a collection of nodes and edges. We can model a given mathematical paper with the nodes and edges representing nouns and verbs respectively. This establishes an acceptable framework with which we can derive meaning. We can first examine how to approach network theory mathematically. Given a network with a set of nodes V and a set of edges E, we can use a matrix to describe the way in which our network N behaves. Take, for example three nodes A, B, and C representing three friends, all of whom wish to give gifts to one another. We label each edge 1 to denote whether one of the friends has given a gift to the other. Suppose then that A gives B and C a gift. As such we can generate the following matrix,

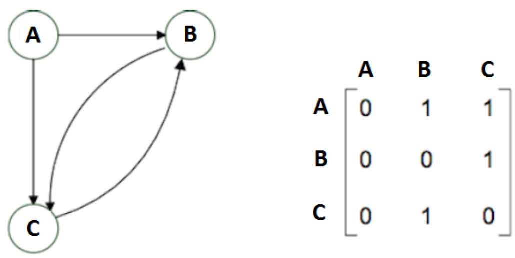

Figure 1: Building an adjacency matrix

We note here that this represents a directed matrix, as the act of A giving B a gift does not necessarily imply that B gives A a gift. With this simple understanding of how a network works, we can now construct our network.

4.2 Building our Network

To establish our network of Field’s Medalists, we first note that our graph will be undirected. That is to say the edges between each node will represent the distance each node is from the other. For our experiments, we chose a distance of 50, as it allowed for many nodes to remain unconnected, yet shows distinctly when two nodes are related. A strictly frequency-based approach, this method of counting gives us an idea of whether concepts are related. i.e. if two words appear closely very frequently, it is likely they are related. We present a few of the generated networks below.

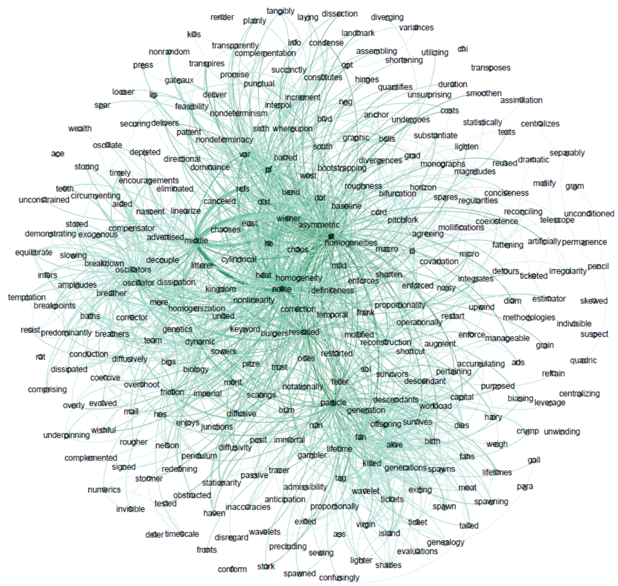

Figure 2: Network of Martin Hairer

From this network we can observe that notable concepts of Field's Medalist Martin Hairer include "noise", "homogeneities", and "particles". This is indicated by the amount of edges it has connecting it to other nodes.

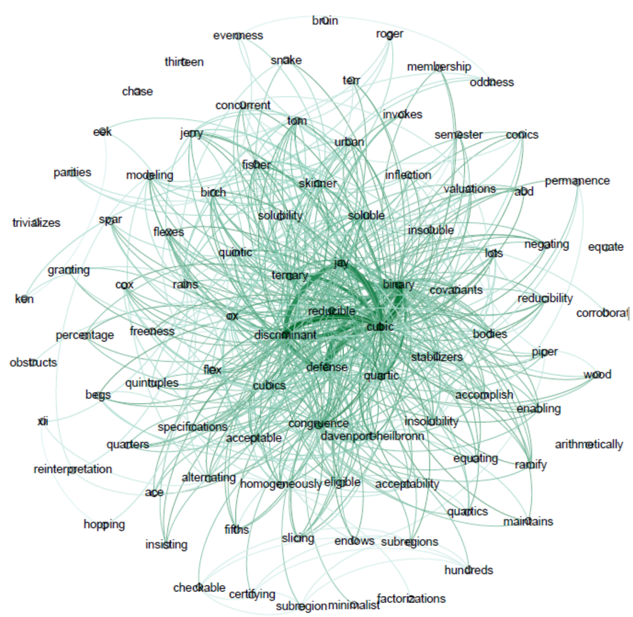

Figure 3: Network of Manjul Bhargava

Likewise, Manjul Bhargava's network indicates that important concepts to him include "binary", "cubic", "descriminant", and "reducible", as they appear to be central nodes by which the majority of other words connect to each other. This important observation leads us to examine the idea of a community established by concepts in more depth.

4.3 Communities

This notion of "community" is explored by an algorithm called community detection that exploits significant groupings among a set of nodes [12]. With this algorithm we can classify nodes into naturally occurring communities of a network.

An intuitive and very visual approach to community detection involves simply identifying nodes with higher edge weights outside the group. A key observation would be to look for groups of nodes that are internally well connected but sparsely connected to other nodes.

To measure the innerconnectivity, we apply a tool called modularity. To understand this tool, we consider a random network split into two networks with matching total edge weights. Formulating these networks as matrices, A_1 and A_2. we have that modularity will be the difference between the number of edges between the same group of nodes in A_1 and A_2. The expected number of edges between a node i and a node j is

$$\frac{k_i k_j}{2m}$$

, where k_i is the degree of node i and k_j the degree of node j. m is the total weight of edges in A. We can use this tool to create a matrix of modularity values, B, where each entry in B will be

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

. This will then allow us to partition our graph into two communities. Using this method, we can inductively achieve additional partitioning. For every entry in B we have the difference in edge weights between A_1 and A_2. Determining a partition requires that we find the eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ of our modularity matrix. Taking the largest eigenvalue λ_{big} of B, we examine the value of its corresponding eigenvector ν_{big}. If its jth value ,$\nu_{big_j} > 0$, node j belongs to the first of the two communities. And vice versa if $\nu_{big_j} < 0$, it belongs to the second. Iteratively applying this process generates two distinct communities.

4.4 Centrality

Having successfully categorized a set of communities within our network we now turn our attention to finding the most "important" nodes of our network. We do this with a concept called a centrality measure. To briefly explain, we can characterize "importance" of a node by the influence it has on other nodes in the network. We will discuss four types of centrality measures used to classify important nodes, weighted degree centrality, closeness centrality, betweenness centrality, and an algorithm called PageRank.

4.4.1 Weighted Degree Centrality

The weighted degree of a node n is the sum of the weight of all its edges. For a unweighted network, this is the number of edges for any given node. The weighted degree of any node n is given by

$$deg(n) = \sum_{i=1}^{|} E|w_i$$

where $|E|$ is the number of edges of n and w_i is the weight of the ith edge. Thus we can also attain the average weighted degree,

$$Avg_deg = \sum_{j=1}^{|} N|\frac{deg(n_j)}{N}$$

where $|N|$ is the number of nodes in the network. Average weighted degree gives an idea as to how our network is connected on average. Weighted degree centrality measures which nodes are most important based on their degrees.

4.4.2 Closeness Centrality

Next we measure importance using closeness centrality. Closeness centrality is a measure of the average distance from a given node to all other nodes. In network theory, the distance between two nodes is a measure of how many edges are between them on the shortest path. If two nodes are connected by an edge, the distance between them is 1. If there are two edges on the shortest path between them, the distance is 2, and so on. Let n_i and n_j the ith and jth nodes in a network. Let $\delta(ni; nj)$ be the distance between nodes i and j. Again, let $|N|$ denote the number of nodes in the network. Then, every node n_i is assigned a closeness

$$C_{n_i} = \sum_{j=1}^{|} N| \frac{\delta(n_i, n_j)}{N}$$

Here, notice that the lower the closeness value is, the more important a node is. This is because a node with lower closeness is closer (has a shorter average distance) to all other nodes in the network. In this way, it is more central. To make the measure consistent with other centrality measures, we change the closeness C_{n_i} to $(1 - C_{n_i})$ for each node. Then, like the other measures, higher closeness value refers to a more important word.

4.4.3 Betweenness Centrality

Betweenness centrality measures the number of shortest paths passing through a given node. That is, if we look at all the shortest path between all pairs of nodes, betweenness centrality of node ni measures the proportion of those paths which must pass through ni to those that do not. The betweenness B of node ni is given by

$$B_{n_i} = \sum_{j,k=1}^{|} N| \frac{\sigma(n_j n_k(n_i))}{\sigma(n_j n_k)}$$

where $\sigma(n_j n_k)$ is the number of shortest paths between nodes nj and n_k, and $\sigma(n_j n_k(n_i))$ is the number of shortest paths between n_j and n_k which pass through node n_i. The higher the proportion of shortest paths passing through a node, the more important that node is by betweenness centrality.

4.4.4 PageRank

Utilized by Google, PageRank is an algorithm which uses probabilities to determine the importance of nodes. It was developed by Larry Page for the purpose of ranking webpage importance based on the structure of hyperlinks between them. Consider a network A, and suppose that a random walker is moving between the nodes of A. Intuitively, the PageRank of a node n_i in A, call it $PR(n_i)$, is the likelihood that the random walker will land on n_i. Since it is a probability, the PageRank of a node will between 0 and 1. If a node has .5 PageRank, there is a 50 percent chance of the walker landing on that node. Consequently, a node's importance is boosted if that node has high ranking neighbors. Those high ranking neighbors, in turn, are also boosted if they have high ranking neighbors. In this way, the ranking of a single node is in balance with all other nodes in the network. Let B_{n_i} be the set $\{b1, b2, ..., b_n\}$ of all nodes which are connected to n_i.

Then, the PageRank of node n_i is given by

$$PR(n_i) = \sum_{b_j \in B_i} \frac{PR(b_j)}{L(b_j)} \text{ for } j = 1, 2, ..., N$$

where $PR(b_j)$ is the PageRank of b_j , and $L(b_j)$ is the number or edges emanating from b_j . This tells as that the PageRank of a node is the average sum of the PageRank of its neigh- boring nodes. Notice that the PageRank of node n_j is dependent on the PageRank of all neighboring nodes to n_j . Then, our definition is recursive. To establish a starting point, we choose one node n_k to give a

fixed PageRank value to, then define all other PageRank values in terms of $PR(n_k)$. We then normalize to ensure that the probabilities are still between 0 and 1.

5 Experimental Results

We implemented the convex optimization of LASSO with and without a weighted matrix. $tfidf$ and $l2$ weightings were incorporated into the term-frequency matrix. We also implemented the aforementioned centrality measures as well as the PageRank algorithm to see how different the results were. Below we list the top ten words returned by each method.

It should be noted that the convex optimization methods produced similar results to the network centrality measures. As we lack a control set, it is hard to say as to the accuracy in our methods for classifying concepts, but this correlation would seem a likely indicator that the terms discovered are important concepts to the authors.

5.1 Results From Features Matrix Using LASSO Without Weighting

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	modular	decoupling	lemme	sparse	noise	graded	provably	entropy	magnetic
2	binary	monster	vorticity	schema	progressions	baseline	symbols	equalizers	bowen	monotone
3	ternary	acted	cubes	suite	subgraph	oscillator	constructible	provable	symbolic	neutron
4	davenport	fake	inflation	springer	complexity	reconstruction	determinant	formalizing	marker	membrane
5	congruence	reflection	paraboloid	stack	tao	wiener	gather	definable	eigenfunctions	excitations
6	soluble	cusps	swirl	perverse	progression	bifurcation	symbol	subcategory	regulator	realizably
7	solubility	algebras	stream	champ	bipartite	particle	subdivision	correspondent	packets	anisotropy
8	discriminant	leech	caps	traces	discrepancy	rescaled	determinants	syntactic	expansive	monotonicity
9	cubics	enveloping	invoking	simples	approximately	oscillators	width	fullness	periodic	excitation
10	reducible	reflective	overlapping	demonstration	output	south	triangulated	conjunctions	centrally	branching

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	escaping	cylinders	mark	discrete	nonstandard	landau	morphism	soup	node	jordan
2	reciprocity	pants	quantum	observable	absorption	collision	universe	loop	origami	idempotent
3	mandelbrot	recognizable	quiver	observables	commensurate	damping	morphisms	loops	origamis	residually
4	valence	twists	virtual	percolation	patterns	plasma	equivalences	surely	child	sandwich
5	tall	twist	dots	summability	magnitude	grazing	pull	nested	swap	commutators
6	basin	cylinder	slope	interface	obeys	nonlinear	contextual	clusters	tiled	homomorphic
7	prim	stratum	framing	crossing	averaged	datum	solid	intensity	transverse	sandwiches
8	cascade	multicomponent	varieties	configuration	uncountable	modes	homotopy	outer	exchange	king
9	collars	subsurface	vacuum	harmonic	ode	curvature	universes	carpet	controls	primeness
10	multipliers	earthquake	envelopes	conformal	avoids	stat	univalent	excursions	parent	linearizing

5.2 Results From Features Matrix Using LASSO With L^2 Weighting

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	modular	snew	group	human	dot	pdo	subobject	entropi	magnet
2	binari	monster	decoupl	schema	program	kepsilon	grade	cuspidal	bowen	poset
3	selmer	hopf	cube	homomorphism	hypergraph	nodelabl	rem	provabl	lindenstrauss	pca
4	ternari	bialgebra	axisymmetr	isomorphism	quasirandom	aboveright	symbol	ramifie	quasimod	excit
5	davenport	cocommut	vortic	geometriqu	bohr	renormalis	polylogarithm	effectiv	nonexpans	monoton
6	solubl	bicharact	inflat	hitchin	szemeredi	akernel	subdivis	troncatur	subconvex	neutron
7	heilbronn	heegner	paraboloid	composant	colour	middl	holonom	supercoher	toral	anisotropi
8	congruenc	coxet	wellposed	defini	freiman	oscil	fukaya	comodul	linnik	membran
9	flex	multilinear	swirl	consideron	progress	baselin	polytop	subcategori	marker	compound
10	bhargava	liftabl	stream	produit	subgraph	nodeanchor	projector	full	homoclin	bramson

1	quasifuchsian	cylind	nakajima	discret	nonstandard	landau	simplici	loop	origami	jordan
2	metriz	submanifold	ptcolor	fermion	concat	collis	morphism	soup	rauzi	mccrimmon
3	escap	pant	yangian	interfac	wigner	damp	fibrat	sure	veech	idempot
4	mandelbrot	subsurfac	quantum	configur	absorpt	ricci	unival	excurs	node	sandwich
5	renormaliz	recogniz	quiver	summabl	elliott	boltzmann	cofibr	nest	zorich	kurosh
6	teichmu	circumfer	virtual	percol	gower	vlasov	contextu	arm	child	ezelmano
7	hubbard	stratum	slope	conform	helmholtz	pme	constr	surround	roth	mathucsdedu
8	nehari	multicompon	quasimodular	hexagon	longpath	plasma	presheaf	jump	multicon	zelmanov
9	pluriharmon	earthquak	frame	quad	morawetz	mode	pull	carpet	birkhoff	cheng
10	tall	subbundl	baxter	spinor	carleman	thmconc	kan	cluster	swap	kostrikin

5.3 Results From Features Matrix Using LASSO With tf-idf Weighting

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	modular	snew	group	hypergraph	dot	pdo	subobject	entropi	magnet
2	binari	monster	decoupl	pour	quasirandom	kepsilon	grade	cuspidal	bowen	poset
3	selmer	hopf	cube	par	spars	aboveright	rem	provabl	lindenstrauss	pca
4	ternari	bialgebra	axisymmetr	schema	bohr	renormalis	symbol	chtouca	quasimod	excit
5	davenport	cocommut	vortic	homomorphism	szemeredi	akernel	polylogarithm	ramifie	subconvex	monoton
6	solubl	bicharact	inflat	numero	colour	baselin	subdivis	effectiv	nonexpans	neutron
7	heilbronn	heegner	paraboloid	isomorphism	freiman	nodeanchor	fukaya	troncatur	linnik	membran
8	congruenc	multilinear	wellposed	connex	progress	hairer	holonom	supercoher	marker	anisotropi
9	bhargava	coxet	swirl	geometriqu	subgraph	wiener	polytop	subcategori	unipot	bramson
10	flex	liftabl	stream	hitchin	tao	deff	projector	comodul	archimedean	compound

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	quasifuchsian	cylind	nakajima	discret	nonstandard	landau	simplici	loop	origami	jordan
2	escap	submanifold	ptcolor	fermion	concat	collis	morphism	soup	rauzi	sandwich
3	metriz	pant	yangian	interfac	wigner	damp	fibrat	sure	veech	idempot
4	mandelbrot	subsurfac	quantum	configur	absorpt	ricci	unival	excurs	node	mccrimmon
5	renormaliz	recogniz	quiver	summabl	elliott	boltzmann	cofibr	nest	zorich	kurosh
6	teichmu	circumfer	virtual	percol	gower	vlasov	contextu	arm	child	kostrikin
7	nehari	stratum	slope	conform	helmholtz	plasma	presheaf	surround	roth	cheng
8	hubbard	multicompon	quasimodular	hexagon	morawetz	mode	constr	jump	multicon	zelmanov
9	tall	earthquak	frame	quad	longpath	thmconc	pull	carpet	birkhoff	ezelmano
10	pluriharmon	subbundl	baxter	spinor	carleman	graze	myproof	cluster	exchang	burnsid

5.4 Results From Weighted Degree Centrality Measure

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	vertex	decoupling	schema	program	dot	determinant	syntactic	entropy	magnetic
2	binary	algebras	cube	tel	human	middle	pro	conservative	periodic	scattering
3	discriminant	modular	cubes	springer	move	noise	rem	subcategory	aligned	neutron
4	defense	monster	inflation	tore	moves	refs	symbols	provably	symbolic	axis
5	jay	acted	vorticity	son	reasoning	asymmetric	conference	provable	eigenfunctions	excitations
6	quartic	enveloping	paraboloid	car	mathematicians	dist	intellectual	equalizers	regulator	anisotropy
7	ternary	fake	swirl	champ	progressions	baseline	symbol	formalizing	beta	monotone
8	congruence	super	overlapping	plat	output	var	invertible	fullness	aperiodic	polarized
9	davenport-heilbronn	relaxed	stream	es	tao	bend	zeta	conjunctions	marker	excitation
10	reducible	cusps	invoking	grand	humans	st	logarithmic	unites	xi	membrane

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	trees	cylinders	quantum	discrete	nonstandard	landau	morphism	loop	cm	jordan
2	mandelbrot	parallel	varieties	conformal	commensurate	damping	morphisms	soup	node	sandwich
3	cascade	cylinder	quiver	harmonic	absorption	kt	functor	law	pt	commutators
4	escaping	twist	chamber	crossing	avoids	interaction	universe	nested	swap	nil
5	bifurcates	saddle	slope	percolation	energies	nonlinear	homotopy	loops	controls	residually
6	reciprocity	geodesic	polarization	observables	magnitude	collision	types	origin	interval	idempotent
7	repelling	shortest	dots	observable	dichotomy	plasma	rules	outermost	transverse	sandwiches
8	cardioid	twists	virtual	interface	averaged	temps	pull	surely	exchange	king
9	codes	pants	framing	configuration	obeying	curvature	equivalences	clusters	algorithm	linearizing
10	branched	recognizable	envelopes	martingale	patterns	ff	bx	outer	parabolic	nonidentical

5.5 Results From Closeness Centrality Measure

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	vertex	erroneous	son	human	pencil	graded	dresser	flattening	magnetic
2	discriminant	algebras	indulge	car	program	separably	rem	elevation	similarities	scattering
3	binary	modular	medians	grand	mathematicians	st	complexes	poser	tripling	neutron
4	reducible	acted	superfix	es	humans	noise	logarithm	syntactic	contractions	axis
5	congruence	monster	thirds	tore	move	rescaled	invertible	reunions	flatten	gaps
6	defense	fake	illustrating	premier	output	particle	symbols	premieres	luckily	anisotropy
7	jay	lane	shoots	tel	reasoning	heat	endowed	unites	contractive	experiments
8	quartic	leech	beautifully	om	progressions	homogeneity	constructible	provably	extracts	measurements
9	ternary	mill	kl	rapport	trying	gram	symbol	glands	entropy	membrane
10	skinner	cusps	median	dire	highly	quadric	story	motifs	quantifiable	branching

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	amplifying	geodesic	quantum	discrete	afforded	landau	homotopy	surely	sublist	dear
2	transports	shortest	varieties	conformal	availability	damping	universe	law	sublists	friend
3	repelling	pants	virtual	harmonic	charger	nonlinear	morphism	origin	interval	jordan
4	tableaux	stratum	quiver	observable	pip	fast	morphisms	loop	vertical	sandwich
5	mandelbrot	cylinder	dots	observables	polio	interaction	types	loops	exchange	compelled
6	trees	saddle	concretely	configurations	pop	plasma	functor	clusters	conjugacy	king
7	branched	parallel	chamber	percolation	shell	velocity	univalent	sites	ea	nil
8	multipliers	cylinders	vacuum	configuration	thermodynamics	kt	operations	soup	return	homomorphic
9	escaping	twist	slope	crossing	war	kinetic	axiom	conditionally	algorithm	sandwiches
10	quadratics	subspaces	brevity	interface	wham	collision	rules	outer	parabolic	commutators

5.6 Results From Betweenness Centrality Measure

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	vertex	decoupling	ram	human	st	graded	syntactic	entropy	magnetic
2	congruence	algebras	plates	son	program	noise	rem	repose	periodic	branching
3	binary	modular	ensured	car	progressions	rescaled	logarithm	premieres	eigenfunctions	correlated
4	discriminant	acted	alphabets	tore	sparse	particle	complexes	unites	duke	attractive
5	rains	theta	aleph	es	move	heat	pol	commencer	untempered	membrane
6	skinner	reflection	sieving	grand	mathematicians	ho	polygons	provably	el	scattering
7	reducible	monster	invoking	tel	tells	pf	constructible	subcategory	division	neutron
8	quartic	cusps	toga	premier	trying	middle	invertible	conservative	normalizations	tail
9	ternary	leech	execute	tame	humans	fan	symbol	fullness	aspect	axis
10	alternating	unimodular	inflation	springer	progression	correction	story	formalizing	explicate	monotone

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	mandelbrot	geodesic	quantum	conformal	magnitude	landau	homotopy	law	interval	jordan
2	repelling	shortest	virtual	discrete	obeys	damping	morphism	surely	unstable	compelled
3	trees	pants	varieties	harmonic	absorption	fast	operations	sites	exchange	commutators
4	reciprocity	stratum	gauge	crossing	hardy	nonlinear	morphisms	loop	return	residually
5	tableaux	subspaces	concretely	percolation	averaged	curvature	universe	origin	tiled	king
6	quadratics	saddle	partitions	interfaces	counterexample	collision	univalent	jump	homeomorphism	sandwich
7	depicts	cylinder	wedge	configurations	obeying	transport	types	loops	vertical	nil
8	branched	parallel	quiver	observable	host	interaction	functor	clusters	parabolic	homomorphic
9	multipliers	twist	slope	configuration	strengthen	concentration	rules	readily	conjugacy	sandwiches
10	tractable	cylinders	dots	observables	nonstandard	kinetic	libraries	intensity	origami	conflict

5.7 Results From PageRank

	Bhargava	Borcherds	Bourgain	Chau	Gowers	Hairer	Kontsevich	Lafforgue	Lindenstrauss	Louis
1	cubic	vertex	adjustments	son	stilted	depleted	chords	syntactic	affects	staggered
2	discriminant	algebras	aiding	car	cautious	spawned	tab	provably	annoyances	upright
3	binary	modular	allied	tore	idiomatically	ass	tensors	conservative	conflicting	caption
4	reducible	acted	ate	grand	spot	wealth	complicating	subcategory	cork	isosceles
5	congruence	monster	attributes	tel	stares	leverage	undiscovered	repose	diagonalized	recherches
6	defense	fake	bracketed	es	solicited	refrain	tilted	dresser	disintegrated	reports
7	jay	enveloping	divisibilities	premier	cognitive	rat	prepares	provable	illuminate	ye
8	ternary	theta	hajj	dire	grammatical	disregard	intrinsically	formalizing	integrations	satellite
9	quartic	leech	infinitude	rapport	repetitive	admissibility	factorizes	commencer	quantifies	sinusoidal
10	skinner	cusps	randomization	schema	lexical	spar	detects	equalizers	reprise	rugged

	McMullen	Mirzakhani	Okounkov	Smirnov	Tao	Villani	Voevodsky	Werner	Yoccoz	Zelmanov
1	repelling	geodesic	annoying	metrically	glossing	controllable	invent	compensation	hyperspace	jordan
2	trees	shortest	exemplify	unimodal	opaque	ancient	imitated	surprises	quaternions	commutators
3	mandelbrot	pants	sting	reshuffle	profitably	investment	thoroughly	displace	capes	sandwich
4	reciprocity	saddle	cation	reasonings	meager	strain	morphism	triadic	attested	residually
5	branched	stratum	enunciated	ascribes	spreads	apparatus	homotopy	angel	occasions	dear
6	quadratics	cylinder	everyday	emphasizes	hp	lax	morphisms	bibliographical	concatenating	friend
7	multipliers	twist	tautology	aw	delicately	frontal	universe	concatenations	rename	homomorphic
8	escaping	cylinders	analogies	stripe	swapped	balances	functor	sticks	supplementary	king
9	cascade	parallel	customarily	pressing	lame	questioned	types	waiting	tetrahedron	compelled
10	respecting	subspaces	regularize	farthest	cautioned	equates	operations	penalizes	interval	nil

References

[1] L. Miratrix, B. Gawalt, J. Jia, B. Yu, and L. E. Ghaoui, "Summarizing large-scale, multiple-document news data: sparse methods and human validation," tech. rep., UC Berkeley, Berkeley, CA 94720, May 2011.

[2] B. L. Monroe, M. P. Colaresi, and K. M. Quinn, "Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict," *Political Analysis*, vol. 16, pp. 372–403, Oct. 2008.

[3] e. a. M. Junod

[4] J. Neter, M. Kutner, W. Wasserman, and C. Nachtsheim, *Applied Linear Statistical Models*. McGraw-Hill, 2004.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[6] R. A. Johnson and D. W. Wichern, *Applied Multivate Statistical Analysis*. Pearson Education, 2007.

[7] K. Koh, S.-J. Kim, and S. P. Boyd, "An interior-point method for large-scale l_1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 8, pp. 1519–1555, 2007.

[8] G. G. Roussas, *An Introduction to Probability and Statistical Inference*. Academic Press, 2003.

[9] R. Vanderbei, *Linear Programming Foundations and Extensions*. Springer, 3rd ed., 2008.

[10] A. Jacobs, "Statistical analysis of newspaper headlines with optimization, url: http://repository.tudelft.nl/view/ir/."

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33.1, 2010.

[12] M. Newman, "Modularity and community structure in networks," *PNAS*, p. 103, 2006.