4-2011

A Nonnegative Analysis of Politics

Tim Chartier

Charles D. Wessell
Gettysburg College

Follow this and additional works at: <https://cupola.gettysburg.edu/mathfac>

Part of the [American Politics Commons](#), [Mathematics Commons](#), and the [Numerical Analysis and Computation Commons](#)

Share feedback about the accessibility of this item.

Chartier, Tim, and Charles Wessell. "A Nonnegative Analysis of Politics." *Math Horizons* 47.7 (April 2011), 10-13.

This is the publisher's version of the work. This publication appears in Gettysburg College's institutional repository by permission of the copyright owner for personal use, not for redistribution. Cupola permanent link: <https://cupola.gettysburg.edu/mathfac/25>

This open access article is brought to you by The Cupola: Scholarship at Gettysburg College. It has been accepted for inclusion by an authorized administrator of The Cupola. For more information, please contact cupola@gettysburg.edu.

A Nonnegative Analysis of Politics

Abstract

The article investigates how linear algebra can recover mathematical information from the electronic messages using the Enron Email Sets in Pennsylvania. It states that a term-by-email matrix has been created to cluster algorithm, which allows one to mine through data and discover meaning. Moreover, nonnegative matrix factorization (NMF) enables one to interpret the resulting factorization in terms of the original problem.

Keywords

Presidential elections, political party, data clustering

Disciplines

American Politics | Applied Mathematics | Mathematics | Numerical Analysis and Computation | Political Science

Rapid #: -7480317

Odyssey
IP: 206.107.42.97/ILL

CALL #: [http://0-
www.jstor.org.oasys.lib.oxy.edu/action/showPublicat ...](http://0-www.jstor.org.oasys.lib.oxy.edu/action/showPublicat ...)

LOCATION: **CCO :: Main Library :: JSTOR Arts and Sciences VII
Collection**

TYPE: Article CC: CCG
JOURNAL TITLE: Math horizons
USER JOURNAL
TITLE: Math Horizons
CCO CATALOG
TITLE: Math horizons
ARTICLE TITLE: A Nonnegative Analysis of Politics
ARTICLE
AUTHOR: Tim Chartier and Chuck Wessell
VOLUME: 18
ISSUE: 4
MONTH: April
YEAR: 2011
PAGES: 10-13
ISSN: 1072-4117
OCLC #:
CROSS
REFERENCE ID: [TN:245962][ODYSSEY:206.107.42.97/ILL]
VERIFIED:

BORROWER: **GDC :: Main Library**

This material may be protected by copyright law (Title 17 U.S. Code)
1/23/2014 3:44:52 PM

A Nonnegative Analysis of Politics

Author(s): Tim Chartier and Chuck Wessell

Source: *Math Horizons*, Vol. 18, No. 4 (April 2011), pp. 10-13

Published by: [Mathematical Association of America](http://www.maa.org)

Stable URL: <http://www.jstor.org/stable/10.4169/194762111X12986558508731>

Accessed: 23/01/2014 17:37

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at
<http://www.jstor.org/page/info/about/policies/terms.jsp>

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Mathematical Association of America is collaborating with JSTOR to digitize, preserve and extend access to *Math Horizons*.

<http://www.jstor.org>

A Nonnegative Analysis of Politics

Tim Chartier and Chuck Wessell

A movie rental store categorizes films by genres. Seemingly invisible lines of demarcation between different cliques of friends hamper a sluggish party. An investigator analyzes over half a million emails from Enron looking for insights into the activity of the company. Given appropriate data, a mathematician could build a matrix to form groups for each of these applications. A repository of users' ratings of movies could generate a list of mathematical movie genres. We could mathematically identify potential groups for friendships through numeric answers to a questionnaire. Let's look more carefully at pulling meaning from the Enron email data.

Mining an Email Trail

After Enron filed for Chapter 11 bankruptcy in December 2001, an unprecedented amount of information was released to the public domain through the Federal Energy Regulatory Commission. From this body of information, a research group at Carnegie Mellon University created the Enron Email Sets containing 517,431 email messages from 150 Enron email accounts covering a period from December 1979 through February 2004. The majority of messages spanned three years: 1999, 2000, and 2001. The emails in this corpus reflect the day-to-day activities of what was the seventh-largest company in the U. S. at that time and certain topics of discussion uniquely linked to Enron activities. In particular, the infamous practices of price-fixing, overspeculation, and deceptive accounting, which led to Enron's collapse toward the end of 2001, are all well documented in the emails. Let us see how linear algebra can uncover such information.

First, we form a term-by-email matrix where each row corresponds to a specific word and each column is associated with an email, such that element ij of the matrix equals the number of times term i appears in email j . The term-by-email matrix formed from the Enron emails is shown in figure 1a. Each element of the matrix corresponds to a pixel in the image. A zero element in the matrix corresponds to a white pixel. Each nonzero entry in the matrix is

Figure 1. Pixel plot of the (a) raw Enron term-by-email matrix and (b) reordered Enron term-by-email matrix.

represented by a blue pixel in which the magnitude of the entry is captured by the intensity of the pixel.

Let's apply a mathematical algorithm that reorders the rows and columns of the matrix, so that emails that generally use the same words are grouped together. Such a reordering leads to the matrix in figure 1b. The nice block pattern of the reordered matrix reveals a hidden structure. Each block corresponds to a set of documents that frequently used the same set of terms. Contrasting the two images of figure 1 reveals just how much structure was hidden in the emails.

Let's take a look at the dense submatrices, called *clusters*. The submatrix in the upper left corner of figure 1b contains the word *touchdown*. Other common terms in the cluster are *football*, *longhorn*, *Texas*, *quarterback*, *score*, *redshirt*, *freshmen*, *punt*, and *tackle*. Although the clustering algorithm does not attach meaning to the cluster, a user might label this cluster *Texas Longhorn Football*.

The dataset also contains a small dense cluster of 12 documents that use 447 terms. The following terms commonly appear in this small cluster: *fortune*, *ceo*, *coo*, *top*, *women*, and *powerful*. These terms and abbreviations, in fact, refer to Louise Kitchen, the top-ranking Enron

	ME	MD	MA	MI	MN	MS	MO	MT
Bush-2004	330201	1024703	1071109	2313746	1346695	684981	1455713	266063
Kerry-2004	396842	1334493	1803800	2479183	1445014	458094	1259171	173710
Others-2004	13709	27482	37479	46323	36678	9290	16480	10672
Obama-2008	421923	1629467	1904098	2872579	1573354	554662	1441911	232159
McCain-2008	295273	959862	1108854	2048639	1275409	724597	1445814	243882
Others-2008	13967	42267	68117	88976	61606	10606	41224	16709

Figure 2. Presidential election vote totals for eight states.

employee responsible for energy trading and Enron Online, who was named one of the 50 most powerful women in business by Fortune Magazine in 2001. Looking carefully at the data, it was found that all 12 emails were saved in Louise Kitchen's own private folder. So what appeared to be a small and possibly interesting cluster, after further inspection, is just an "ego cluster."

Note how the structure formed by our clustering algorithm allows us to mine through the data and discover meaning. While there are many clustering methods, we will focus on *nonnegative matrix factorization* (NMF), which identified the clusters in this example. A particularly nice feature of NMF is that the user can interpret, with relative ease, the resulting factorization in terms of the original problem.

A Nonnegative Approach

Linear algebraists love to factor matrices. What distinguishes NMF is that all the matrices involved are nonnegative. That is, each matrix element is greater than or equal to zero. To cluster data into k groups, NMF attempts to express an $m \times n$ matrix A as the product of an $m \times k$ matrix W and a $k \times n$ matrix H , where k is much smaller than either m or n . Since exact equality is almost never possible, the goal of the factorization is to find $A \approx WH$. This is accomplished by following these algorithmic steps:

1. Randomly initialize W and H .
2. Until a maximum iteration count is reached or until $\|A - WH\|$ is less than some prescribed tolerance, update W and H using the rules

$$H = H \cdot (W^T A) ./ (W^T W H + \epsilon)$$

$$W = W \cdot (A H^T) ./ (W H H^T + \epsilon),$$

where $\cdot$ represents element-by-element multiplication. For example,

$$\begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} 8 & 1 \\ 3 & 5 \end{pmatrix} = \begin{pmatrix} 16 & 3 \\ 12 & 5 \end{pmatrix}.$$

Similarly, $./$ represents element-by-element division.

Note that a few details have been omitted in this description

of the NMF algorithm. For instance, what matrix norm will you use to measure the distance between A and WH ?

Second, what value is suitable for ϵ ? (Hint: Think about the purpose ϵ is serving in the formula.) The answers to such questions can involve important insights from linear algebra, and from modeling if A is connected to a physical problem.

At this point an example may be helpful, as it will allow us to see what a typical nonnegative matrix factorization looks like and how it can be used to cluster data. Figure 2 is a matrix showing the vote totals for candidates in the last two presidential elections (the total for all minor candidates is labeled "Others") in the eight states beginning with the letter M.

The NMF algorithm does not generate unique factors W and H , since these are initialized randomly. One possible answer when $k = 2$ is $A \approx WH$, where

$$W = \begin{pmatrix} 1057914 & 2525650 \\ 2725868 & 1298461 \\ 61582 & 18303 \\ 2962308 & 1609474 \\ 977439 & 2392174 \\ 102471 & 43360 \end{pmatrix}.$$

$$H = \begin{pmatrix} 0.1001 & 0.3992 & 0.5397 & 0.6178 & 0.3275 & 0.0404 & 0.2198 & 0.0234 \\ 0.0856 & 0.2395 & 0.2184 & 0.6326 & 0.3969 & 0.2694 & 0.4984 & 0.0943 \end{pmatrix}.$$

The columns of W represent two prototypical states. Since the rows of W represent the same quantities as the rows of A , we see that the state described by column 1 gives more votes to the Democratic Party candidate, while the column 2 state backs the Republican Party candidate. For this small example each column of W is associated with a political party, but if we consider $k > 2$ groups each column will have a more ambiguous meaning.

Each column of A is a linear combination of these two "states" and the columns of H give us the weights to use. For example, Maine's voting behavior can be approximated by

$$\begin{pmatrix} 330201 \\ 396842 \\ 13709 \\ 421923 \\ 295273 \\ 13967 \end{pmatrix} \approx .1001 \begin{pmatrix} 1057914 \\ 2725868 \\ 61582 \\ 2962308 \\ 977439 \\ 102471 \end{pmatrix} + .0856 \begin{pmatrix} 2525650 \\ 1298461 \\ 18303 \\ 1609474 \\ 2392174 \\ 43360 \end{pmatrix}$$

The larger value in each column of H indicates which prototypical state each real state most resembles. For this example, Maine, Maryland, and Massachusetts are more like state 1, while Michigan (just barely), Minnesota, Mississippi, Missouri, and Montana are more like state 2. This is how we use NMF to cluster data, and this idea can be extended to larger values of k and much larger data sets.

Weakness into Strength

Suppose a movie came out last Friday. You aren't sure whether you want to see it so you ask a friend what she thought of it. Not wanting to base your decision on just one opinion you ask some other friends and also read some online reviews. By the end of this process you have probably melded all those different viewpoints so that you have a pretty good idea of whether you want to see the movie or not. Using multiple sources of information when clustering is also appropriate.

The NMF algorithm attempts to find a local minimum for the difference between A and WH . Since the algorithm is initialized with randomly generated W and H , 100 runs of NMF will return 100 slightly different answers, which can result in slightly different clusters. That uncertainty may seem like a weakness of the method, but it can also be viewed as a strength. As in the movie example from the last paragraph, collecting many viewpoints is often a good idea.

Considering the example from the last section, if each time we ran NMF we kept track of which states clustered with other states, after a large number of runs we could form a matrix S where entry S_{ij} was equal to the number of times state i clustered with state j . Here is a sample S produced from 100 runs of NMF.

$$S = \begin{matrix} & \begin{matrix} ME & MD & MA & MI & MN & MS & MO & MT \end{matrix} \\ \begin{matrix} ME \\ MD \\ MA \\ MI \\ MN \\ MS \\ MO \\ MT \end{matrix} & \begin{pmatrix} 0 & 96 & 97 & 30 & 8 & 4 & 3 & 4 \\ 96 & 0 & 99 & 28 & 6 & 2 & 3 & 2 \\ 97 & 99 & 0 & 27 & 7 & 3 & 4 & 1 \\ 30 & 28 & 27 & 0 & 70 & 72 & 69 & 72 \\ 8 & 6 & 7 & 70 & 0 & 92 & 93 & 94 \\ 4 & 2 & 3 & 72 & 92 & 0 & 97 & 96 \\ 3 & 3 & 4 & 69 & 93 & 97 & 0 & 97 \\ 4 & 2 & 1 & 72 & 94 & 96 & 97 & 0 \end{pmatrix} \end{matrix}$$

Figure 3. U.S. states separated into two, three, four, and five clusters.

The states are listed in the alphabetical order of their full names, which is serendipitous in this example since it places the states in a natural order for our final clustering. Our eyes can examine S and do a quick clustering, as only Michigan doesn't cluster with a group of other states at least 90 times. We can now use NMF to cluster S and as a result compute

$$H = \begin{pmatrix} 0.0073 & 0.0000 & 0.0000 & 0.3710 & 0.4565 & 0.4663 & 0.4658 & 0.4686 \\ 0.5618 & 0.5646 & 0.5661 & 0.1960 & 0.0573 & 0.0349 & 0.0352 & 0.0311 \end{pmatrix},$$

which is much more definitive than the one we found before. The final clustering is {Maine, Maryland, Massachusetts} and {Michigan, Minnesota, Mississippi, Missouri, Montana}.

A Presidential State of the Union

Let's now consider the problem of clustering the contiguous U.S. states based on their presidential election voting records. New Mexico and Arizona entered the union in 1912, becoming the 47th and 48th states. Thus we have presidential election data for all 48 states from 1912 through 2008. Our procedure will be the same as described in the last section, only now we'll be clustering 48 states and we'll use data from the 25 presidential elections that took place in this period. We first form A where each row represents a candidate in a particular election. Each election has the two major party candidates and often a fairly well-known third- or fourth-party candidate (for example, Patrick Buchanan in 2000 or Theodore Roosevelt in 1912). Finally, we add a row for each election to accumulate the results for all the minor candidates not represented by any other rows from that year's election.

This results in an 88×48 matrix A on which we will run our clustering algorithm.

Living in a political environment dominated largely by two major parties, choosing $k = 2$ is a natural start. We see what results from $k = 2$ in figure 3a. When examining the map remember that NMF's clusters are not assigned a label corresponding to a particular political party or philosophy. In fact, insight may come from choosing $k > 2$ as seen in figures 3b, 3c, and 3d.

Each of the four clusterings offers a look at the voting behavior of states over the past century and generally will not be a snapshot of the current state of politics in the United States. For instance, regardless of the number of clusters in figure 3, Indiana and Ohio's journeys of presidential voting travel a similar path over the past century.

Clustering can be most useful when it points to unexpected groupings. Do you see any that surprise you in figure 3? In such cases, additional reading or pointed questions to a professor may be needed to form an explanation for an apparent clustering anomaly. Such research can uncover information mined by the clustering algorithm. Returning momentarily to the Enron example, keep in mind that the clustering algorithm identified a dense submatrix with common terms of *fortune*, *ceo*, *coo*, *top*, *women*, and *powerful*. Yet further digging would have uncovered that this set of emails came from Louise Kitchen's private email folder, and all terms referred to her.

As you examine clusters of the United States, remember that the algorithm will assign a cluster to each state even if none of them is an ideal fit. For example, no one who follows contemporary politics would think that the states of Alabama and California have much in common. Yet in figure 3a these two states are in the same cluster because we are looking for only two clusters, and over the past century California's voting patterns had more in common with Southern states than with states of the North and Midwest. Notice that the $k = 3$ clustering shown in figure 3b rectifies this situation by breaking off a few states from each cluster to form a new cluster. You might conclude that this new blue cluster contains states that only weakly belonged to the previous two clusters, and an examination of the H matrix supports that claim.

Ultimately, for students interested in presidential election history these clusterings lead to interesting discussion questions. As k increases, is there a common theme that unites the states in each newly created cluster? What can be made of the fact that as k moves from three to five Nevada

and Florida leave the cluster of Southern states and then return to it? If you were working as a consultant to a presidential candidate, how would you use the $k = 5$ map to advise on allocations of campaign resources? To get a more accurate picture of the current political climate, how many past presidential election results would you include in your data? If you were to break up the past century into shorter time periods in order to see changes in clustering due to political changes in the United States, what would those time periods be? In today's arena of readily available data, clustering is a popular technique that can provide insight into these vast quantities of information.

Here we have clustered political data and posed questions that arose naturally from them. Are there datasets related to other areas that interest you? Apply linear algebra: cluster the data and analyze the results to uncover the hidden structure of the dataset.

About the authors: Tim Chartier has performed mime across the United States, climbed two 14,000-foot mountains in one day, and programmed on one of the world's fastest supercomputers. Chuck Wessell has hit a half-court shot at halftime of a college basketball game, been a contestant on Jeopardy!, won the College Bowl national championship, hiked the Appalachian Trail, run 10 marathons, and completed five Krispy Kreme Challenges.

email: tichartier@davidson.edu
cdwessel@ncsu.edu

10.4169/194762111X12986558508731