Nonnegative Factorization of a Data Matrix as a Motivational Example for Basic Linear Algebra

Barak A. Pearlmutter and Helena Šmigoc

Abstract We present a motivating example for matrix multiplication based on factoring a data matrix. Traditionally, matrix multiplication is motivated by applications in physics: composing rigid transformations, scaling, sheering, etc. We present an engaging modern example which naturally motivates a variety of matrix manipulations, and a variety of different ways of viewing matrix multiplication. We exhibit a low-rank non-negative decomposition (NMF) of a "data matrix" whose entries are word frequencies across a corpus of documents. We then explore the meaning of the entries in the decomposition, find natural interpretations of intermediate quantities that arise in several different ways of writing the matrix product, and show the utility of various matrix operations. This example gives the students a glimpse of the power of an advanced linear algebraic technique used in modern data science.

Key words: Nonnegative Matrix Factorization (NMF); Topic Modeling; Data Mining; Matrix Multiplication

1 Introduction

Examples are an essential part of teaching any mathematical subject. They serve a range of purposes, from checking understanding and deepening knowledge to giving a broader view of the subject and its applications. There are an abundance of examples available in the literature, covering every topic of any basic linear algebra course. However, it is not so easy to find examples that give an insight into the

Prof. Barak A. Pearlmutter
Department of Computer Science, Maynooth University, Ireland
e-mail: barak@pearlmutter.net

Dr. Helena Šmigoc
School of Mathematics and Statistics, UCD Dublin, Ireland
e-mail: helena.smigoc@ucd.ie

current development of the subject and are at the same time accessible to students. As the applications of linear algebra are rapidly expanding, and several new developments in the subject are motivated by applications, examples showcasing current applications of the subject are of particular interest.

Because of its utility in other domains, linear algebra is a classical subject routinely taught to students not majoring in mathematics. It is a prerequisite not just for advanced mathematics but also for undergraduate degrees in Engineering, Physics, Computer Science, Biology, Chemistry, Business, Statistics, and the like. Those students in particular benefit from learning from examples, and appreciate seeing interesting applications of the material they are learning. The benefits of using models to introduce mathematical concepts has been studied (Lesh and English, 2005), and models focusing on different concepts from linear algebra are available (Possani et al., 2010; Salgado and Trigueros, 2015; Trigueros and Possani, 2013).

While very simple examples are essential when introducing a topic, examples of applications presented in classrooms often seem contrived. For example, students' knowledge of economics and agriculture is sufficiently sophisticated that simple linear examples of acreage under cultivation invite criticism. On the other hand, it is impossible to bring to the classroom, for instance, deep applications of linear algebra in genetics (Ponnapalli et al., 2011), since most likely neither the instructor nor the students have the necessary background to really understand how they work. To quote Stewart and Thomas (2003),

> While it is true that linear algebra can simplify the solution to many problems, this is only true for those who are very familiar with the subject area. In contrast, the first year university student has a long way to go before being able to see the whole picture.

The press is full of stories about data science: analysis of large corpora of data. Some of these lend themselves naturally to use as motivating examples for various concepts in linear algebra. For example, the Netflix challenge can be viewed as a problem in matrix completion, where a company was highly motivated to recover a low rank decomposition of an almost entirely incomplete matrix of movie ratings.

We present less abstract example, in which matrix multiplication is explicated by examination of a nonnegative decomposition of a term-by-document matrix. This particular example vividly illustrates various views of matrix multiplication (as composition of linear functions; as a sum of outer products of columns with rows; and as a table of inner products of rows with columns), while using only primitive concepts. It also previews and motivates a variety of more advanced concepts (the general algebraic concept of factoring, the notion of rank, approximation and norms, iterative numeric algorithms, constraints like element-wise non-negativity, and column-stochastic matrices), helping sketch the outlines of richer material covered in more advanced courses.

Although intuitive and implemented by a very short algorithm, the technique discussed (NMF) is far from a toy: it has enjoyed a myriad of accessible and engaging applications (Asari et al., 2006; Helleday et al., 2014; Niegowski and Zivanovic, 2014; O'Grady and Pearlmutter, 2008; Ray and Bandyopadhyay, 2016; Smaragdis and Brown, 2003; Wilson et al., 2008). For this reason, the example we present

serves to give a taste of an interesting and accessible application of linear algebra. Although briefly presented in this document for the sake of completeness, we do not suggest attempting to derive the method in the classroom, leaving that too as motivation for the pursuit of more advanced study.

The mathematical notation used below is standard. For example, e_i denotes the vector of appropriate size with i-th entry equal to one and other entries equal to zero. The "discussion point" boxes are intended to be illustrative, and can be used for classroom discussion, project-based learning, or as the basis for assignments.

2 Term-by-Document Matrix: A Small Example

Numeric data organised in a tabular format is something we are all familiar with in our daily lives. Everyone can understand a spreadsheet whose rows are indexed by products, columns by month, and whose entries contain sales. These are the matrices that students entering a linear algebra course have already seen. In data science, tabular data of this sort is known as a "data matrix".

A data matrix of interest in library science is a tabulation of word frequencies by documents. Rows are indexed by words, columns by documents, and the entries of a matrix are the number of times a given word appears in a given document. This particular kind of data matrix is sometimes called a term-by-document matrix. Although this matrix completely ignores the actual arrangement of words within each document (i.e., it is a bag-of-words model), it still contains sufficient information to allow interesting structure to be discovered.

There are several ways in which matrices and matrix multiplication can be introduced in the classroom. Term-by-document matrices can be one of the examples given to the class, starting with a small example that can be given on a board. In the classroom we can show a pre-prepared example, which can be built on by an assignment in which students have the freedom to chose the documents they want to consider. Since the search function in browsers automatically counts the number of times a word appears on a page, such an assignment is not necessarily time demanding.

Here we present an example where the documents are the Wikipedia entries for the four most venomous animals in the world (*Box Jellyfish*[1], *King Cobra*[2], *Marbled Cone Snail*[3], *Blue-Ringed Octopus*[4]) and we consider only five terms (*venom*, *death*, *danger*, *survive*, *Madagascar*). This gives us Table 1.

Going from the table to the matrix

[1] https://en.wikipedia.org/wiki/Box_jellyfish

[2] https://en.wikipedia.org/wiki/King_cobra

[3] https://en.wikipedia.org/wiki/Conus_marmoreus

[4] https://en.wikipedia.org/wiki/Blue-ringed_octopus

Table 1 Term-by-Document Matrix of the Four Most Venomous Animals

		Documents			
		Jellyfish	Cobra	Snail	Octopus
Terms	venom	32	44	1	18
	death	9	3	0	2
	danger	6	4	0	4
	survive	2	0	0	1
	Madagascar	0	0	2	0

$$A = \begin{pmatrix} 32 & 44 & 1 & 18 \\ 9 & 3 & 0 & 2 \\ 6 & 4 & 0 & 4 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

we can lead the discussion in several directions. A representative set of questions is given below. The questions are of course trivial to answer without referring to matrices. The simplicity of the questions makes it easy for students to understand the corresponding matrix operations and motivates them to think about extensions to more involved tasks.

Discussion Point: Determine the frequency of terms appearing in the first document, in the third document, in the first or third document. What is the frequency of terms in all the documents together?

The above questions can all be answered using multiplication of a matrix by a column vector.

$$A\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 32 \\ 9 \\ 6 \\ 2 \\ 0 \end{pmatrix} \qquad A\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

$$A\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 33 \\ 9 \\ 6 \\ 2 \\ 2 \end{pmatrix} \qquad A\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 95 \\ 14 \\ 14 \\ 3 \\ 2 \end{pmatrix}$$

On this small example matrix multiplication, while illustrative, does not help with efficiency of obtaining an answer. However, we can lead the students to think further.

Discussion Point: Can you think about other questions about the set of documents that can be answered using matrix multiplication? (E.g., differences in word frequencies.) How would one extract information from very large datasets? (This is for the computer science students in the class: strategies for assembling, representing, storing, and operating upon a very large data matrix.)

At this point the students can appreciate that in order to get the information[5] about the terms in the i-th document we need to multiply A by e_i, to find the information about the terms in documents i, j and k we need to multiply A by $e_i+e_j+e_k$, or equivalently, add Ae_i, Ae_j, and Ae_k. We can view the matrix A as a transformation that takes information about documents (i, j, k) to information about terms (Ae_i, Ae_j, Ae_k).

$$\text{words} \xleftarrow{A} \text{documents}$$

Furthermore, we can remark that this transformation obeys certain rules

$$A(e_i+e_j+e_k) = Ae_i+Ae_j+Ae_k$$

which can be developed into the definition of linearity. We continue the discussion by presenting the transpose matrix.

$$A^T = \begin{pmatrix} 32 & 9 & 6 & 2 & 0 \\ 44 & 3 & 4 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 \\ 18 & 2 & 4 & 1 & 0 \end{pmatrix}$$

Discussion Point: Which documents contain the third term, the fifth term, the third or the fifth term?

$$A^T \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 0 \\ 4 \end{pmatrix} \qquad A^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \end{pmatrix} \qquad A^T \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 2 \\ 4 \end{pmatrix}$$

Students can see the similarity with the discussion above. To find out how the i-th term is featured in documents we need to multiply e_i by A^T.

[5] The "information" here can be viewed as histograms over either documents or terms, which is something the students should be comfortable with.

$$\text{documents} \xleftarrow{A^T} \text{words}$$

Further questions can be discussed in this framework, touching upon elementary ideas not routinely discussed in the first course on linear algebra, such as non-negativity and sparsity.[6]

Discussion Point: Note that the term Madagascar only appears in the third document. Can we draw any conclusions from this?

Discussion Point: If we were to make a table that includes all the terms that appear in at least one of the four documents, would we expect most of the entries in the matrix to be equal to zero? Why?

Discussion Point: Note that all the elements in the matrix are non-negative integers. Can you think of any other tables with only non-negative integers? How about tables containing only non-negative real elements? Can you think about any other conditions on the entries that are imposed naturally in a particular setting?

Discussion Point: In class, we usually label the rows and columns of a matrix with successive integers: $1,\ldots,n$. These are generally used as "nominal numbers", meaning only their identities are important—like building numbers, course numbers, or social security numbers. And when we write $\sum_{i=1}^{n}$, what we usually mean is really $\sum_{i\in\text{rows}}$. We can change most of our formulas to use this convention. But in actual applications, as in the example here, often the rows and columns have natural labels: names of chemicals, words, documents, people, months, cities, *etc.* When this holds, we can use these labels instead of numbers as indices. And we can freely rearrange the rows and columns, keeping their labels, while still representing the same underlying mathematical object: the same matrix.

This point is illustrated by a term-by-document matrix, which has rows labeled by terms and columns labeled by documents. Let us look at another example. The two tables below contain movie ratings given by four users to five movies:

[6] Sparsity is of particular importance in computer science, where it impacts the representation and manipulation of both matrices and graphs.

Table 2 Labeling Rows and Columns

	Alice	Becky	Cindy	Dora
Alien	4	1	4	5
Animal House	1	5	4	2
Beetlejuice	2	2	5	3
Jaws	5	1	5	5
Life of Brian	1	5	5	1

	Becky	Cindy	Dora	Alice
Animal House	5	4	2	1
Life of Brian	5	5	1	1
Beetlejuice	2	5	3	2
Jaws	1	5	5	5
Alien	1	4	5	4

Discussion Point: Compare the two tables. Do they contain the same information? Can you figure out the principle behind the ordering of rows and columns on the left and on the right?

3 Matrix Factorization

Students are familiar with the idea of factoring an integer as a product of prime numbers. Writing $6 = 3 \times 2$ gives us some information about the number 6. Another example is factoring a polynomial. Writing $x^4 - 10x^3 + 35x^2 - 50x + 24$ as $(x-1)(x-2)(x-3)(x-4)$ uncovers useful information. Both prime factor decomposition and factoring a polynomial are in general hard to do. Given two integers it is straightforward to find their product, but there is no known efficient algorithm for integer factorization.

This concept can, in some sense, be extended to matrices. Given a matrix, we want to write it as a product of two (or more) "simpler" matrices. There are several ways this can be done. A wide range of factorizations of matrices are used in applications, where—depending on the application—different properties of the factors are desired. An example that can be presented in the classroom is given below. The matrix

$$A = \begin{pmatrix} 2 & -1 & 1 & 2 \\ -1 & 1 & -2 & -1 \\ 1 & -2 & 5 & 1 \\ 2 & -1 & 1 & 2 \end{pmatrix}$$

can be factored in several ways:

$$A = \frac{1}{21}\begin{pmatrix} 1 & 1 & -1 & 1 \\ -1 & 0 & 0 & 3 \\ 2 & -1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 7 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 3 & -3 & 6 & 3 \\ 7 & 0 & -7 & 7 \\ -10 & 3 & 1 & 11 \\ 1 & 6 & 2 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 2 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 & 2 & 1 \\ 1 & 0 & -1 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 2 & -1 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -1 & 1 \\ 0 & 1 & -3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The students may not have the mathematical tools to develop the factorizations above, but we can ask them to check their correctness, and to explore properties of the factors.

Demands from applications frequently impose conditions on the factors that are too strong to be satisfied exactly. For example, not every matrix can be written as a product of a column by a row. Or more generally, not every matrix can be written as a product of two low rank matrices. If we are unwilling to relax the conditions, we need to resort to approximate factorizations. Let us consider the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 1.01 \\ 1 & 1 & 1.01 & 1 \\ 1 & 1.01 & 1 & 1 \\ 1.01 & 1 & 1 & 1 \end{pmatrix}.$$

Using elementary tools one can check that A cannot be written as a product of a column and a row.

Discussion Point: Can we find a matrix that is close to the matrix A that can be written as a product of a column by a row?

Students are likely to come up with the following solution:

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix},$$

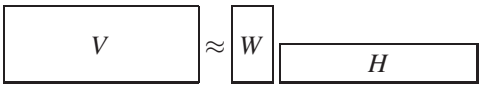
and we may present another one:

$$A_2 = \begin{pmatrix} 1.0025 & 1.0025 & 1.0025 & 1.0025 \\ 1.0025 & 1.0025 & 1.0025 & 1.0025 \\ 1.0025 & 1.0025 & 1.0025 & 1.0025 \\ 1.0025 & 1.0025 & 1.0025 & 1.0025 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1.0025 & 1.0025 & 1.0025 & 1.0025 \end{pmatrix}.$$

Discussion Point: Which solutions is better? What does it mean for a matrix B to be close to A?

This question could be an introduction to the concept of matrix norms.

Factorizations of matrices have been developed that are used in applications, with the aim of uncovering hidden structure. We present a factorization that requires the factors to be non-negative and of a given low rank r. Those conditions are too strong, so the factorization will not be exact. This means that given a matrix V, we obtain matrices W and H so that the matrix $\hat{V} = WH$ is in some sense close to V.



Non-negative matrix factorization, or NMF (Paatero and Tapper, 1994; Lee and Seung, 1999; Wang and Zhang, 2013), is a class of techniques for *approximately* factoring a matrix of non-negative numbers into the product of two such matrices: given an entry-wise non-negative $n \times m$ matrix V, find two entry-wise non-negative matrices W and H, of sizes $n \times r$ and $r \times m$, such that $V \approx WH$. (Even after a value for r has been chosen, and an appropriate measure of similarity of two matrices has been chosen, there can be many possible solutions. However, popular NMF algorithms empirically usually find good solutions, a phenomenon which has been the subject of considerable analysis (Donoho and Stodden, 2004).)

Let us look at the nonnegative matrix factorization of the matrix that corresponds to the left side of Table 2:

$$A = \begin{pmatrix} 4 & 5 & 4 & 1 \\ 5 & 5 & 5 & 1 \\ 5 & 3 & 2 & 2 \\ 4 & 2 & 1 & 5 \\ 5 & 1 & 1 & 5 \end{pmatrix}$$

First we take r to be equal to one. That means that we want to approximate A by a product of a nonnegative column and a nonnegative row. The algorithm returns the following result:

$$W_1 = \begin{pmatrix} 7.137 \\ 8.214 \\ 6.398 \\ 5.974 \\ 6.155 \end{pmatrix} \qquad H_1 = \begin{pmatrix} 0.6709 & 0.4898 & 0.406 & 0.381 \end{pmatrix}$$

$$A - W_1H_1 = \begin{pmatrix} -0.7885 & 1.504 & 1.102 & -1.72 \\ -0.511 & 0.977 & 1.665 & -2.13 \\ 0.7077 & -0.1334 & -0.5976 & -0.4378 \\ -0.008175 & -0.926 & -1.426 & 2.724 \\ 0.8703 & -2.015 & -1.499 & 2.655 \end{pmatrix}$$

Taking $r = 2$ we get:

$$W_2 = \begin{pmatrix} 6.968 & 1.086 \\ 7.908 & 1.364 \\ 3.763 & 3.558 \\ 0.5117 & 6.448 \\ 0 & 7.197 \end{pmatrix} \qquad H_2 = \begin{pmatrix} 0.5171 & 0.6379 & 0.5707 & 0 \\ 0.6658 & 0.1897 & 0.1095 & 0.7133 \end{pmatrix}$$

$$A - W_2H_2 = \begin{pmatrix} -0.3256 & 0.3496 & -0.09564 & 0.2257 \\ 0.002296 & -0.3033 & 0.337 & 0.02677 \\ 0.6857 & -0.07508 & -0.5374 & -0.5376 \\ -0.5576 & 0.4506 & 0.001623 & 0.4004 \\ 0.2088 & -0.365 & 0.2117 & -0.1333 \end{pmatrix}$$

Discussion Point: Compare $A - W_1H_1$ and $A - W_2H_2$. Can you find a nonnegative factorization of A for $r = 4$?

Let us have a closer look at W_2 and H_2. Recall that the rows of W_2 correspond to movies, and the columns of H_2 correspond to users.

Discussion Point: Can we give sensible labels to the columns of W_2, or equivalently, the rows of H_2.

Note that the highest values in the first column of W_2 correspond to movies *Alien* and *Jaws*, while the highest values in the second column correspond to movies *Animal House* and *Life of Brian*. Based on this, we may agree to label the first column "Horror", and the second column "Comedy". The rows of H_2 are labeled correspondingly. Values in H_2 can now be interpreted in the following way. Cindy likes both horror and comedy movies, Dora and Alice prefer horror movies, and Becky likes comedy, but not horror movies. Matrix factorization uncovered genre for our movies.

In the context of the small example, we can look at V as a transformation from "movies" to "people". Now we have the third notion appearing: "genres". The matrix H can be seen as a transformation that takes movies to genres, and W takes genres to people.

$$\text{people} \xleftarrow{W} \text{genres} \xleftarrow{H} \text{movies} \qquad \overbrace{\qquad\qquad}^{V}$$

We may remark to the students that they are justified in finding this example a bit contrived. The is example is too small (and also made up) to be very convincing. We give a larger example based on term-by-document matrix later in the chapter.

The formula for matrix multiplication, $A = BC$,

$$a_{ij} = \sum_{k=1}^{m} b_{ik} c_{kj}$$

can be intimidating to students.

From the point of view of our example, where the rows of V are indexed by "movies" and the columns by "people", we can write down the same formula in the following way:

$$\hat{v}_{\text{movie,person}} = \sum_{g \in \text{genre}} w_{\text{movie},g}\, h_{g,\text{person}}$$

The entry in the matrix $\hat{V}$ that represents a rating of a chosen "movie" to a chosen "person" is computed by summing up the product of how much the person likes genre g and how much the movie is in genre g, over all genres. This process can be depicted graphically:

$$\begin{bmatrix} & \hat{V} & \\ & \bullet & \end{bmatrix} = \begin{bmatrix} & W & \\ & \blacksquare & \end{bmatrix} \begin{bmatrix} & H & \\ & \blacksquare & \end{bmatrix}$$

On the other hand we may notice that $\hat{V}$ is the sum of rank one matrices, each of them giving the contribution of a particular genre.

$$\begin{bmatrix} & \hat{V} & \end{bmatrix} = \sum_{g} \begin{bmatrix} & W_{\bullet,g} & \end{bmatrix} \begin{bmatrix} & H_{g,\bullet} & \end{bmatrix}.$$

This is written as:

$$\begin{aligned} V &= \sum_{g=1}^{r} (g\text{-th column of } W) \cdot (g\text{-th row of } H) \\ &= \sum_{g \in \text{genres}} \text{Ratings Matrix for genre } g \\ &= \text{sum of rank-one per-genre matrices} \end{aligned}$$

This interpretation reinforces the power of the nonnegative matrix factorization. From a bundle of documents, it singles out a particular genre in way that agrees with our intuition in a surprisingly strong way.

Let us go back to the example of movie ratings discussed earlier. We have approximated our matrix A by W_2H_2. Below this product is written as a sum of two rank one matrices:

$$\begin{aligned} W_2H_2 &= \begin{pmatrix} 6.968 & 1.086 \\ 7.908 & 1.364 \\ 3.763 & 3.558 \\ 0.5117 & 6.448 \\ 0 & 7.197 \end{pmatrix} \begin{pmatrix} 0.5171 & 0.6379 & 0.5707 & 0 \\ 0.6658 & 0.1897 & 0.1095 & 0.7133 \end{pmatrix} \\ &= \begin{pmatrix} 6.968 \\ 7.908 \\ 3.763 \\ 0.5117 \\ 0 \end{pmatrix} \begin{pmatrix} 0.5171 & 0.6379 & 0.5707 & 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} 1.086 \\ 1.364 \\ 3.558 \\ 6.448 \\ 7.197 \end{pmatrix} \begin{pmatrix} 0.6658 & 0.1897 & 0.1095 & 0.7133 \end{pmatrix} \\ &= \begin{pmatrix} 3.603 & 4.445 & 3.977 & 0 \\ 4.089 & 5.045 & 4.514 & 0 \\ 1.946 & 2.4 & 2.148 & 0 \\ 0.2646 & 0.3264 & 0.292 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0.7227 & 0.2059 & 0.1189 & 0.7743 \\ 0.9084 & 0.2588 & 0.1495 & 0.9732 \\ 2.368 & 0.6747 & 0.3897 & 2.538 \\ 4.293 & 1.223 & 0.7064 & 4.6 \\ 4.791 & 1.365 & 0.7883 & 5.133 \end{pmatrix} \end{aligned}$$

Supplementary Advanced Material

To minimize the Frobenius norm, $\|V - WH\|_F$, a surprisingly simple method is available. Two updates can be iterated

$$\begin{aligned} H &:= H \odot (W^T V \div W^T WH) \\ W &:= W \odot (VH^T \div WHH^T) \end{aligned}$$

where $\odot$ denotes elementwise multiplication and $\div$ denotes elementwise division. Typically after each update of W its columns are normalized to unit sum.

An enormous number of variations and embellishments of the basic NMF algorithm have been developed, with applications ranging from astronomy to zoology.

4 Example using Module Descriptors

We give an example of factoring a data matrix involving a corpus of documents: module descriptors for 62 mathematics modules that were taught in the School of Mathematics and Statistics, University College Dublin (UCD) in 2015. Module descriptors are relatively short documents that give overviews of the courses. Here are two representative examples of module descriptors.

Numbers and Functions

This module is an introduction to the joys and challenges of mathematical reasoning and mathematical problem-solving, organised primarily around the theme of properties of the whole numbers. It begins with an introduction to some basic notions of mathematics and logic such as proof by contradiction and mathematical induction. It introduces the language of sets and functions, including injective surjective and bijective maps and the related notions of left-, right- and 2-sided inverses. Equivalence relations, equivalence classes. It covers basic important principles in combinatorics such as the Principle of Inclusion-Exclusion and the Pigeonhole Principle. The greater part of the module is devoted to number theory: integers, greatest common divisors, prime numbers, Euclid's algorithm, the Fundamental Theorem of Arithmetic, congruences, Fermat's theorem, Euler's theorem, and arithmetic modulo a prime and applications. The module concludes with some topics from elementary coding theory / cryptography such as the RSA encryption system.

Groups, Rings and Fields

This course will be an introduction to group theory, ring theory and field theory. We will cover the following topics: definition and examples of groups, subgroups, cosets and Lagrange's Theorem, the order of an element of a group, normal subgroups and quotient groups, group homomorphisms and the homomorphism theorem, more isomorphism theorems, definitions of a commutative ring with unity, integral domains and fields, units, irreducibles and primes in a ring, ideals and quotient rings, prime and maximal ideals, ring homomorphisms and the homomorphism theorem, polynomial rings, the division algorithm, gcd for polynomials, irreducible polynomials and field extensions. Time permitting, we may cover the Sylow theorems, solvable groups and further examples of groups.

This set was chosen because the example was developed for the first linear algebra classes at UCD. All the students in those classes were quite familiar with the chosen set of documents, which they need to navigate each semester when choosing and registering for their modules. The number of documents is large enough that we can make a case for needing a computer to help navigate them, but small enough that we can exert some manual control and can be familiar with the entire corpus.

All the words that appear in any of the 62 documents were collected. So-called stop words (common words like "and", "the", and the like), and all the words that appeared fewer than four times, were removed. Words were also down-cased and stemmed, so for example the terms *Eigenvalue* and *eigenvalues* are deemed equivalent. This resulted in a 290×62 matrix V of word counts. Below we show some of the data, starting with lists of the most and the least frequent words.

Table 3 Most and Least Frequent Words

Most Frequent Words	
word	**count**
function	117
theorem	75
linear	72
matrix	66
theory	58
equation	53
mathematical	52
mathematics	52
understand	49
science	48
problem	44

Least Frequent Words				
addition	advanced	arguments	background	behaviour
classify	column	computation	constrained	construct
continuous	definite	depth	described	directional
double	elimination	engineering	evaluate	expressions
flow	foundations	general	importance	independence
induction	integrate	interpret	introduces	known
manipulate	max	maxima	min	minima
nash	nullity	numerical	original	possible
prime	quadratic	range	related	riemann
row	sample	search	significant	solid
special	stock	sum	sylow	together
uncountable	variety			

Using the standard Octave (Eaton et al., 2015) package for NMF, the entry-wise nonnegative matrix V is factored as

$$V \approx WH,$$

where W is an entry-wise non-negative $290 \times r$ matrix and H is an entry-wise non-negative $r \times 62$ matrix. We will see that factoring a matrix in this way reveals a particular structure of the matrix which reveals something about the content of the original documents. Different values for r are used below, and we will see how the information that we obtain changes as we increase r.

For easier interpretation, the entries in each column of W have been permuted so that they appear in descending order, and the term corresponding to each row is shown. (Recall the discussion about labelling rows and columns at the end of Section 2.) We also present a few columns of the matrix H for $r = 3$.

Table 4 W-matrix, $r = 2$

function	23.0	group	17.9
linear	11.1	theorem	15.0
matrix	11.0	theory	6.9
equation	9.9	ring	6.6
derivative	9.4	understand	4.0
calculus	7.8	structure	3.3
differential	6.5	example	3.0
solve	6.1	number	2.9
problem	5.9	isomorphism	2.7
mathematical	5.3	concepts	2.6
science	5.2	homomorphisms	2.5
compute	5.1	sylow	2.3
variable	4.7	subgroups	2.3
applications	4.7	quotient	2.2
integral	4.6	cauchy	2.2

Table 5 W-matrix, $r = 3$

group	18.1	function	27.1	matrix	19.1
theorem	14.8	derivative	11.1	linear	14.9
theory	6.8	calculus	9.2	space	9.5
ring	6.7	equation	6.3	vector	8.7
understand	3.9	differential	5.8	algebra	7.1
structure	3.4	integral	5.7	basis	6.5
number	3.0	problem	5.5	equation	6.4
example	2.9	variable	5.3	compute	5.7
isomorphism	2.6	graph	4.8	system	4.9
homomorphisms	2.6	limit	4.6	rank	3.5
concepts	2.5	solve	4.5	complex	3.5
sylow	2.3	mathematics	4.3	product	3.4
subgroups	2.3	calculate	3.9	number	3.3
applications	2.2	applications	3.8	mathematical	3.3
quotient	2.2	science	3.5	science	3.3
cauchy	2.1	introduction	3.5	solve	3.2
time	2.1	mathematical	3.5	dimensional	3.0
finite	2.1	method	3.4	eigenvalues	2.9
algebraic	2.1	polynomial	3.4	set	2.8
permitting	2.0	differentiation	3.3	eigenvectors	2.7

Table 6 W-matrix, $r = 4$

group	20.5	function	25.2	matrix	19.3	theorem	9.0
theorem	11.3	derivative	11.7	linear	14.8	understand	8.7
ring	6.7	calculus	8.4	space	9.0	question	6.3
theory	4.8	equation	6.7	vector	8.7	complex	6.1
structure	3.9	differential	6.3	algebra	7.1	number	5.6
isomorphism	3.0	problem	5.8	basis	6.5	example	5.6
homomorphisms	2.9	variable	5.5	equation	6.4	concepts	5.4
applications	2.7	graph	4.9	compute	5.7	mathematical	5.2
sylow	2.6	solve	4.8	system	4.8	function	5.1
subgroups	2.6	limit	4.3	rank	3.6	cauchy	4.9
quotient	2.6	applications	4.2	product	3.4	theory	4.6
algebra	2.3	integral	4.1	solve	3.1	integral	3.8
algebraic	2.2	calculate	4.1	science	3.1	demonstrate	3.8
time	2.1	polynomial	3.7	complex	3.0	correctly	3.4
permitting	2.1	differentiation	3.4	eigenvalues	2.9	method	3.4
finite	1.9	science	3.3	dimensional	2.9	series	3.3
lagrange	1.8	mathematics	3.2	eigenvectors	2.8	write	3.3
special	1.7	introduction	3.2	number	2.7	set	3.3
construct	1.5	inverse	3.0	mathematical	2.6	sequence	3.3

Table 7 H-matrix, $r = 3$

Module Names			
Numbers and Functions	Linear Algebra with Applications to Economics	Groups, Rings, and Fields	Differential Equations via Computer Algebra
0.1	0.0	0.3	0.0
0.1	0.0	0.0	0.1
0.0	0.2	0.0	0.0

5 Discussion

Following on from the discussion around the small example presented above, the students understand how the frequency of the terms across all documents is computed. This gives an easy and automated way to derive the most and least frequent words, given in Table 3. While in small example shown the most and least frequent words could easily be found by hand, this is impractical when the matrix becomes large.

Discussion Point: Consider different columns of the matrix W for $r = 2,3$ given in Tables 4 and 5. What do you observe?

Already in the case when $r = 2$, we can see some regularity in the way the terms are grouped into columns. For example, it makes sense that the terms *function*, *derivative*, *differential*, *integral* appear in the same column. In the second column we see the terms *group*, *ring*, *isomorphism*, *homomorphism*, *sylow*, *subgroups*, *quotient*, *cauchy* appearing together.

The factorization is perhaps the most informative for the choice $r = 3$, so let us take a closer look at this case. The terms in the matrix W are grouped in such a sensible way that we can challenge the students to give them titles. Those students who've read ahead a little may suggest *Abstract Algebra* for the first column, while most should find *Calculus* appropriate for the second, and *Linear Algebra* for the third. Things become a little less clear when we consider the matrix W for $r = 4$ (Table 6).

Discussion Point: What are the advantages and disadvantages of choosing r to be small or big?

While higher values of r will make $\hat{V}$ closer to V, they can make it more difficult to interpret the results. An informed choice of r, dependent on the needs of the applications, needs to be made. This problem of "model complexity" has been the subject of a great deal of research in Statistics and Machine Learning.

Discussion Point: In the $r = 3$ case, we were able to give titles to columns of the matrix W. Those titles could be called "topics". The rows of W are indexed by "words" and the columns by "topics". For the multiplication WH to make sense we need to have the rows of H marked by "topics". Let us look at the matrix H given in Table 7 to see if this makes sense.

Representative columns given for the matrix H agree with our prediction that the first row corresponds to Abstract Algebra, the second row to Calculus and the third row to Linear Algebra. For example, the course *Linear Algebra with Applications to*

Economics has the only nonzero entry in the third row, while the course *Differential Equations via Computer Algebra* has the only nonzero entry in the second row.

6 Conclusion

While this is a black box experiment for the students, they are able to appreciate the result and understand the emergence of the topics in an example. The NMF algorithm yields this topic analysis, helping us appreciate the strengths of the method. If we want to bring the discussion further, it can be pointed out how this class of algorithm is used to decompose speech and music into phonemes and notes (Asari et al., 2006; O'Grady and Pearlmutter, 2008; Smaragdis and Brown, 2003), in speech denoising (Wilson et al., 2008) and recognition (Hurmalainen, 2014), in chemistry (Siy et al., 2008) and biomedical sciences (Helleday et al., 2014; Ortega-Martorell et al., 2012; Paine et al., 2016; Ray and Bandyopadhyay, 2016), in the analysis of the cosmic microwave background radiation (Cardoso et al., 2003), etc.

The example presented above can be adapted for classroom needs in various ways. An aspect not discussed here is the potential to turn some of the above ideas into student projects. We are aware that the computational aspects of this may be a big stumbling block, so we are developing a web-based tool to make it easy for students to analyse a set of document in this way. We see a potential for interdisciplinary projects, where students are charged with the task of analyzing a large body of documents on a particular subject, and use linear algebra to reach some conclusions.

In collaboration with Miao Wei,[7] we have created an end-to-end interactive browser-based implementation of the processing pipeline discussed above (taking documents as input and processing them through stemming, the construction of a term-by-document matrix, NMF, and visualization of the resulting factor matrices), which is being made available online.[8]

References

Hiroki Asari, Barak A. Pearlmutter, and Anthony M. Zador. Sparse representations for the cocktail party problem. *Journal of Neuroscience*, 26(28):7477–90, 2006. doi: 10.1523/JNEUROSCI.1563-06.2006.

J.-F. Cardoso, J. Delabrouille, and G. Patanchon. Independent component analysis of the cosmic microwave background. In *Fourth International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 1111–6, Nara, Japan, April 1–4 2003.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004. URL http://books.nips.cc/papers/files/nips16/NIPS2003_LT10.pdf.

[7] Dept of Computer Science, Maynooth University, Ireland, e-mail: davidweimiao@gmail.com.

[8] http://barak.pearlmutter.net/demo/NMF/

John W. Eaton, David Bateman, Sören Hauberg, and Rik Wehbring. *GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations*. Free Software Foundation, 2015. URL http://www.gnu.org/software/octave/doc/interpreter.

Thomas Helleday, Saeed Eshtad, and Serena Nik-Zainal. Mechanisms underlying mutational signatures in human cancers. *Nature Reviews Genetics*, 15:585–98, 2014. doi: 10.1038/nrg3729.

Antti Hurmalainen. *Robust Speech Recognition with Spectrogram Factorisation*. PhD thesis, Tampere University of Technology, Finland, October 2014. URL http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/22512/hurmalainen.pdf.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects with nonnegative matric factorization. *Nature*, 401:788–91, 1999. doi: 10.1038/44565.

Richard Lesh and Lyn D. English. Trends in the evolution of models & modeling perspectives on mathematical learning and problem solving. *ZDM Mathematics Education*, 37(6):487–9, 2005. ISSN 1863-9704. doi: 10.1007/BF02655857.

M. Niegowski and M. Zivanovic. ECG-EMG separation by using enhanced non-negative matrix factorization. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4212–5, August 2014. doi: 10.1109/EMBC.2014.6944553.

Paul D. O'Grady and Barak A. Pearlmutter. Discovering speech phones using convolutive nonnegative matrix factorisation with a sparseness constraint. *Neurocomputing*, 72(1–3):88–101, 2008. doi: 10.1016/j.neucom.2008.01.033.

Sandra Ortega-Martorell, Paulo JG Lisboa, Alfredo Vellido, Margarida Julià-Sapé, and Carles Arús. Non-negative matrix factorisation methods for the spectral decomposition of MRS data from human brain tumours. *BMC Bioinformatics*, 13(1):38, 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-38.

P. Paatero and U. Tapper. Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–26, June 1994. doi: 10.1002/env.3170050203.

M. R. L. Paine, J. Kim, R. V. Bennett, R. M. Parry, D. A. Gaul, M. D. Wang, et al. Whole reproductive system non-negative matrix factorization mass spectrometry imaging of an early-stage ovarian cancer mouse model. *PLoS ONE*, 11(5):e0154837, 2016. doi: 10.1371/journal.pone.0154837.

Sri Priya Ponnapalli, Michael A. Saunders, Charles F. Van Loan, and Orly Alter. A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms. *PLOS ONE*, 6(12):1–11, 12 2011. doi: 10.1371/journal.pone.0028072.

E. Possani, M. Trigueros, J.G. Preciado, and M.D. Lozano. Use of models in the teaching of linear algebra. *Linear Algebra and its Applications*, 432(8):2125–40, 2010. ISSN 0024-3795. doi: 10.1016/j.laa.2009.05.004. URL http://www.sciencedirect.com/science/article/pii/S0024379509002523. Special issue devoted to the 15th ILAS Conference at Cancun, Mexico, June 16-20, 2008.

S. Ray and S. Bandyopadhyay. A NMF based approach for integrating multiple data sources to predict HIV-1-human PPIs. *BMC Bioinformatics*, 8(17), March 2016. doi: 10.1186/s12859-016-0952-6.

Hilda Salgado and María Trigueros. Teaching eigenvalues and eigenvectors using models and APOS theory. *The Journal of Mathematical Behavior*, 39:100–20, 2015. ISSN 0732-3123. doi: 10.1016/j.jmathb.2015.06.005. URL http://www.sciencedirect.com/science/article/pii/S0732312315000462.

Peter W. Siy, Richard A. Moffitt, R. Mitchell Parry, Yanfeng Chen, Ying Liu, M. Cameron Sullards, Alfred H. Merrill, Jr., and May D. Wang. Matrix factorization techniques for analysis of imaging mass spectrometry data. In *8th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2008)*, pages 1–6, 2008.

P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, October 2003. doi: 10.1109/ASPAA.2003.1285860.

Sepideh Stewart and Michael O. J. Thomas. Difficulties in the acquisition of linear algebra concepts. *New Zealand Journal of Mathematics*, 32(Supplementary Issue):207–15, 2003. URL https://www.math.auckland.ac.nz/~thomas/My%20PDFs%20for%20web%20site/21%20Stewart.pdf.

María Trigueros and Edgar Possani. Using an economics model for teaching linear algebra. *Linear Algebra and its Applications*, 438(4):1779–92, 2013. ISSN 0024-3795. doi: 10.1016/j.laa.2011.04.009. URL http://www.sciencedirect.com/science/article/pii/S0024379511003053. 16th ILAS Conference Proceedings, Pisa 2010.

Y. X. Wang and Y. J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–53, June 2013. ISSN 1041-4347. doi: 10.1109/TKDE.2012.51.

Kevin W. Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. Speech denoising using nonnegative matrix factorization with priors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4029–4032, 2008.