Singular Value Decomposition Tutorial

Kirk Baker

March 29, 2005 (Revised January 14, 2013)

Contents

1 Acknowledgments

Several people have been kind enough to point out errors in the original document or otherwise provide encouragement, including Xiaofei Lu, John McNally, Maiko Sell, Li Xue, don quitoxe, Juda Djatmiko, Dio Coumans, Mansi Radke, Roberto Mirizzi, Gursimran singh, Ryan Angilly, Mark Wong-VanHaren, Harry Podschwit, Laxmi, Paul Hurt, Pritesh Patel, and B. L. Deekshatulu. I tried to fix all the typos, but haven't gotten around to the more substantive suggestions yet. Thank you all.

I wrote this as an assignment for an NLP seminar taught by Chris Brew. Thank you can never suffice.

2 Introduction

Most tutorials on complex topics are apparently written by very smart people whose goal is to use as little space as possible and who assume that their readers already know almost as much as the author does. This tutorial's not like that. It's more a manifestivus for the rest of us. It's about the mechanics of singular value decomposition, especially as it relates to some techniques in natural language processing. It's written by someone who knew zilch about singular value decomposition or any of the underlying math before he started writing it, and knows barely more than that now. Accordingly, it's a bit long on the background part, and a bit short on the truly explanatory part, but hopefully it contains all the information necessary for someone who's never heard of singular value decomposition before to be able to do it.

3 Points and Space

A point is just a list of numbers. This list of numbers, or *coordinates*, specifies the point's position in space. How many coordinates there are determines the *dimensions* of that space.

For example, we can specify the position of a point on the edge of a ruler with a single coordinate. The position of the two points $0.5cm$ and $1.2cm$ are precisely specified by single

coordinates. Because we're using a single coordinate to identify a point, we're dealing with points in one-dimensional space, or 1-space.

The position of a point anywhere in a plane is specified with a pair of coordinates; it takes three coordinates to locate points in three dimensions. Nothing stops us from going beyond points in 3-space. The fourth dimension is often used to indicate time, but the dimensions can be chosen to represent whatever measurement unit is relevant to the objects we're trying to describe.

Generally, space represented by more than three dimensions is called *hyperspace*. You'll also see the term *n-space* used to talk about spaces of different dimensionality (e.g. 1-space, 2-space, ..., n-space).

For example, if I want a succinct way of describing the amount of food I eat in a given day, I can use points in n-space to do so. Let the dimensions of this space be the following food items:

Eggs Grapes Bananas Chickens Cans of Tuna

There are five categories, so we're dealing with points in 5-space. Thus, the interpretation of the point $(3, 18, 2, 0.5, 1,)$ would be "three eggs, eighteen grapes, two bananas, half a chicken, one can of tuna".

4 Vectors

For most purposes, points and vectors are essentially the same thing[1], that is, a sequence of numbers corresponding to measurements along various dimensions.

Vectors are usually denoted by a lower case letter with an arrow on top, e.g. $\vec{x}$. The numbers comprising the vector are now called *components*, and the number of components equals the dimensionality of the vector. We use a subscript on the vector name to refer to the component in that position. In the example below, $\vec{x}$ is a 5-dimensional vector, $x_1 = 8$, x_26, etc.

$$\vec{x} = \begin{pmatrix} 8 \\ 6 \\ 7 \\ 5 \\ 3 \end{pmatrix}$$

Vectors can be equivalently represented horizontally to save space, e.g. $\vec{x} = [8, 6, 7, 5, 3]$ is the same vector as above. More generally, a vector $\vec{x}$ with n-dimensions is a sequence of n numbers, and component x_i represents the value of $\vec{x}$ on the i^{th} dimension.

[1] Technically, I think, a vector is a function that takes a point as input and returns as its value a point of the same dimensionality.

5 Matrices

A matrix is probably most familiar as a table of data, like Table 1, which shows the top 5 scorers on a judge's scorecard in the 1997 Fitness International competition.

Contestant	Round 1	Round 2	Round 3	Round 4	Total	Place
Carol Semple-Marzetta	17	18	5	5	45	1
Susan Curry	42	28	30	15	115	3
Monica Brant	10	10	10	21	51	2
Karen Hulse	28	5	65	39	132	5
Dale Tomita	24	26	45	21	116	4

Table 1: 1997 Fitness International Scorecard. Source: Muscle & Fitness July 1997, p.139

A table consists of rows (the horizontal list of scores corresponding to a contestant's name), and columns (the vertical list of numbers corresponding to the scores for a given round). What makes this table a matrix is that it's a rectangular array of numbers. Written as a matrix, Table 1 looks like this:

$$\begin{bmatrix} 17 & 18 & 5 & 5 & 45 & 1 \\ 42 & 28 & 30 & 15 & 115 & 3 \\ 10 & 10 & 10 & 21 & 51 & 2 \\ 28 & 5 & 65 & 39 & 132 & 5 \\ 24 & 26 & 45 & 21 & 116 & 4 \end{bmatrix}$$

The size, or dimensions, of a matrix is given in terms of the number of rows by the number of columns. This makes the matrix above a *"five by six"* matrix, written 5×6 matrix.

We can generalize the descriptions made so far by using variables to stand in for the actual numbers we've been using. Traditionally, a matrix in the abstract is named A. The maximum number of rows is assigned to the variable m, and the number of columns is called n. Matrix *entries* (also called *elements* or *components*) are denoted by a lower-case a, and a particular entry is referenced by its row index (labeled i) and its column index (labeled j). For example, 132 is the entry in row 4 and column 5 in the matrix above, so another way of saying that would be $a_{45} = 132$. More generally, the element in the i_{th} row and j_{th} column is labeled a_{ij}, and called the ij-entry or ij-component.

A little more formally than before, we can denote a matrix like this:

5.1 Matrix Notation

Let m, n be two integers ≥ 1. Let a_{ij}, $i = 1, ..., m$, $j = 1, ..., n$ be mn numbers. An array of numbers

$$A = \begin{bmatrix} a_{11} & ... & a_{1j} & ... & a_{1n} \\ . & & . & & . \\ a_{i1} & ... & a_{ij} & ... & a_{in} \\ . & & . & & . \\ a_{m1} & ... & a_{mj} & ... & a_{mn} \end{bmatrix}$$

is an $m \times n$ matrix and the numbers a_{ij} are elements of A. The sequence of numbers

$$A_{(i)} = (a_{i1}, ..., a_{in})$$

is the i_{th} row of A, and the sequence of numbers

$$A^{(j)} = (a_{1j}, ..., a_{mj})$$

is the j_{th} column of A.

Just as the distinction between points and vectors can blur in practice, so does the distinction between vectors and matrices. A matrix is basically a collection of vectors. We can talk about row vectors or column vectors. Or a vector with n components can be considered a $1 \times n$ matrix.

For example, the matrix below is a word×document matrix which shows the number of times a particular word occurs in some made-up documents. Typical accompanying descrip-

	Doc 1	Doc 2	Doc 3
abbey	2	3	5
spinning	1	0	1
soil	3	4	1
stunned	2	1	3
wrath	1	1	4

Table 2: Word×document matrix for some made-up documents.

tions of this kind of matrix might be something like "high dimensional vector space model". The dimensions are the words, if we're talking about the column vectors representing documents, or documents, if we're talking about the row vectors which represent words. High dimensional means we have a lot of them. Thus, "hyperspace document representation" means a document is represented as a vector whose components correspond in some way to the words in it, plus there are a lot of words. This is equivalent to "a document is represented as a point in n-dimensional space."

6 Vector Terminology

6.1 Vector Length

The length of a vector is found by squaring each component, adding them all together, and taking the square root of the sum. If $\vec{v}$ is a vector, its length is denoted by $|\vec{v}|$. More concisely,

$$|\vec{v}| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

For example, if $\vec{v} = [4, 11, 8, 10]$, then

$$|\vec{v}| = \sqrt{4^2 + 11^2 + 8^2 + 10^2} = \sqrt{301} = 17.35$$

6.2 Vector Addition

Adding two vectors means adding each component in $\vec{v_1}$ to the component in the corresponding position in $\vec{v_2}$ to get a new vector. For example

$$[3, 2, 1, -2] + [2, -1, 4, 1] = [(3 + 2), (2 - 1), (1 + 4), (-2 + 1)] = [5, 1, 5, -1]$$

More generally, if $A = [a_1, a_2, ...a_n]$ and $B = [b_1, b_2, ...b_n]$, then $A+B = [a_1+b_1, a_2+b_2, ...a_n+b_n]$.

6.3 Scalar Multiplication

Multiplying a scalar (real number) times a vector means multiplying every component by that real number to yield a new vector. For instance, if $\vec{v} = [3, 6, 8, 4]$, then $1.5 * \vec{v} = 1.5 * [3, 6, 8, 4] = [4.5, 9, 12, 6]$. More generally, *scalar multiplication* means if d is a real number and $\vec{v}$ is a vector $[v_1, v_2, ..., v_n]$, then $d * \vec{v} = [dv_1, dv_2, ..., dv_n]$.

6.4 Inner Product

The *inner product* of two vectors (also called the *dot product* or *scalar product*) defines multiplication of vectors. It is found by multiplying each component in $\vec{v_1}$ by the component in $\vec{v_2}$ in the same position and adding them all together to yield a scalar value. The inner product is only defined for vectors of the same dimension. The inner product of two vectors is denoted $(\vec{v_1}, \vec{v_2})$ or $\vec{v_1} \cdot \vec{v_2}$ (the dot product). Thus,

$$(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$

For example, if $\vec{x} = [1, 6, 7, 4]$ and $\vec{y} = [3, 2, 8, 3]$, then

$$\vec{x} \cdot \vec{y} = 1(3) + 6(2) + 7(8) + 3(4) = 83$$

6.5 Orthogonality

Two vectors are *orthogonal* to each other if their inner product equals zero. In two-dimensional space this is equivalent to saying that the vectors are perpendicular, or that the only angle between them is a 90° angle. For example, the vectors $[2, 1, -2, 4]$ and $[3, -6, 4, 2]$ are orthogonal because

$$[2, 1, -2, 4] \cdot [3, -6, 4, 2] = 2(3) + 1(-6) - 2(4) + 4(2) = 0$$

6.6 Normal Vector

A *normal vector* (or *unit vector*) is a vector of length 1. Any vector with an initial length > 0 can be normalized by dividing each component in it by the vector's length. For example, if $\vec{v} = [2, 4, 1, 2]$, then

$$|\vec{v}| = \sqrt{2^2 + 4^2 + 1^2 + 2^2} = \sqrt{25} = 5$$

Then $\vec{u} = [2/5, 4/5, 1/5, 1/5]$ is a normal vector because

$$|\vec{u}| = \sqrt{(2/5)^2 + (4/5)^2 + (1/5)^2 + (2/5)^2} = \sqrt{25/25} = 1$$

6.7 Orthonormal Vectors

Vectors of unit length that are orthogonal to each other are said to be *orthonormal*. For example,

$$\vec{u} = [2/5, 1/5, -2/5, 4/5]$$

and

$$\vec{v} = [3/\sqrt{65}, -6/\sqrt{65}, 4/\sqrt{65}, 2/\sqrt{65}]$$

are orthonormal because

$$|\vec{u}| = \sqrt{(2/5)^2 + (1/5)^2 + (-2/5)^2 + (4/5)^2} = 1$$

$$|\vec{v}| = \sqrt{(3/\sqrt{65})^2 + (-6/\sqrt{65})^2 + (4/\sqrt{65})^2 + (2/\sqrt{65})^2} = 1$$

$$\vec{u} \cdot \vec{v} = \frac{6}{5\sqrt{65}} - \frac{6}{5\sqrt{65}} - \frac{8}{5\sqrt{65}} + \frac{8}{5\sqrt{65}} = 0$$

6.8 Gram-Schmidt Orthonormalization Process

The Gram-Schmidt orthonormalization process is a method for converting a set of vectors into a set of orthonormal vectors. It basically begins by normalizing the first vector under consideration and iteratively rewriting the remaining vectors in terms of themselves minus a

multiplication of the already normalized vectors. For example, to convert the column vectors of

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 2 & 3 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

into orthonormal column vectors

$$A = \begin{bmatrix} \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{6} & \frac{2}{3} \\ 0 & \frac{2\sqrt{2}}{3} & \frac{-1}{3} \\ \frac{\sqrt{6}}{3} & 0 & 0 \\ \frac{\sqrt{6}}{6} & \frac{-\sqrt{2}}{6} & \frac{-2}{3} \end{bmatrix},$$

first normalize $\vec{v_1} = [1, 0, 2, 1]$:

$$\vec{u_1} = [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}].$$

Next, let

$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} = [2, 2, 3, 1] - [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}] \cdot [2, 2, 3, 1] * [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$$

$$= [2, 2, 3, 1] - (\frac{9}{\sqrt{6}}) * [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$$

$$= [2, 2, 3, 1] - [\frac{3}{2}, 0, 3, \frac{3}{2}]$$

$$= [\frac{1}{2}, 2, 0, \frac{-1}{2}]$$

Normalize $\vec{w_2}$ to get

$$\vec{u_2} = [\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3}, 0, \frac{-\sqrt{2}}{6}]$$

Now compute $\vec{u_3}$ in terms of $\vec{u_1}$ and $\vec{u_2}$ as follows. Let

$$\vec{w_3} = \vec{v_3} - \vec{u_1} \cdot \vec{v_3} * \vec{u_1} - \vec{u_2} \cdot \vec{v_3} * \vec{u_2} = [\frac{4}{9}, \frac{-2}{9}, 0, \frac{-4}{9}]$$

and normalize $\vec{w_3}$ to get

$$\vec{u_3} = [\frac{2}{3}, \frac{-1}{3}, 0, \frac{-2}{3}]$$

More generally, if we have an orthonormal set of vectors $\vec{u_1}, .., \vec{u_{k-1}}$, then $\vec{w_k}$ is expressed as

$$\vec{w_k} = \vec{v_k} - \sum_{i=1}^{k-1} \vec{u_i} \cdot \vec{v_k} * \vec{u_i}$$

7 Matrix Terminology

7.1 Square Matrix

A matrix is said to be *square* if it has the same number of rows as columns. To designate the size of a square matrix with n rows and columns, it is called *n-square*. For example, the matrix below is 3-square.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

7.2 Transpose

The *transpose* of a matrix is created by converting its rows into columns; that is, row 1 becomes column 1, row 2 becomes column 2, etc. The transpose of a matrix is indicated with a superscripted ^{T}, e.g. the transpose of matrix A is A^T. For example, if

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

then its transpose is

$$A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

7.3 Matrix Multiplication

It is possible to multiply two matrices only when the second matrix has the same number of rows as the first matrix has columns. The resulting matrix has as many rows as the first matrix and as many columns as the second matrix. In other words, if A is a $m \times n$ matrix and B is a $n \times s$ matrix, then the product AB is an $m \times s$ matrix.

The coordinates of AB are determined by taking the inner product of each row of A and each column in B. That is, if $A_1, ..., A_m$ are the row vectors of matrix A, and $B^1, ..., B^s$ are the column vectors of B, then ab_{ik} of AB equals $A_i \cdot B^k$. The example below illustrates.

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} B = \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix} AB = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 16 \\ 0 & 26 \end{bmatrix}$$

$$ab_{11} = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = 2(3) + 1(-1) + 4(1) = 9$$

$$ab_{12} = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = 2(4) + 1(4) + 4(2) = 16$$

$$ab_{21} = \begin{bmatrix} 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = 1(3) + 5(-1) + 2(1) = 0$$

$$ab_{22} = \begin{bmatrix} 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = 1(2) + 5(4) + 2(2) = 26$$

7.4 Identity Matrix

The identity matrix is a square matrix with entries on the diagonal equal to 1 and all other entries equal zero. The diagonal is all the entries a_{ij} where $_i =_j$, i.e., $a_{11}, a_{22}, ..., a_{mm}$. The n-square identity matrix is denoted variously as $I_{n \times n}$, I_n, or simply I. The identity matrix behaves like the number 1 in ordinary multiplication, which mean $AI = A$, as the example below shows.

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} AI = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$ai_{11} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 2(1) + 0(4) + 0(6) = 2$$

$$ai_{12} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 2(0) + 4(1) + 6(0) = 4$$

$$ai_{13} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 2(0) + 4(0) + 6(1) = 6$$

$$ai_{21} = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 1(1) + 3(0) + 5(0) = 1$$

$$ai_{22} = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 1(0) + 3(1) + 5(0) = 3$$

$$ai_{23} = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 1(0) + 3(0) + 5(1) = 5$$

$$= \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix}$$

7.5 Orthogonal Matrix

A matrix A is orthogonal if $AA^T = A^TA = I$. For example,

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & -4/5 \\ 0 & 4/5 & 3/5 \end{bmatrix}$$

is orthogonal because

$$A^TA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & -4/5 \\ 0 & 4/5 & 3/5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & 4/5 \\ 0 & -4/5 & 3/5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

7.6 Diagonal Matrix

A diagonal matrix A is a matrix where all the entries ai_{ij} are 0 when $i \neq j$. In other words, the only nonzero values run along the main dialog from the upper left corner to the lower right corner:

$$A = \begin{bmatrix} a_{11} & 0 & ... & 0 \\ 0 & a_{22} & & 0 \\ . & & . & . \\ 0 & ... & & a_{mm} \end{bmatrix}$$

7.7 Determinant

A determinant is a function of a square matrix that reduces it to a single number. The determinant of a matrix A is denoted $|A|$ or $det(A)$. If A consists of one element a, then $|A| = a$; in other words if $A = [6]$ then $|A| = 6$. If A is a 2×2 matrix, then

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

For example, the determinant of

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}$$

is

$$|A| = \begin{vmatrix} 4 & 1 \\ 1 & 2 \end{vmatrix} = 4(2) - 1(1) = 7.$$

Finding the determinant of an n-square matrix for $n > 2$ can be done by recursively deleting rows and columns to create successively smaller matrices until they are all 2×2 dimensions, and then applying the previous definition. There are several tricks for doing this efficiently, but the most basic technique is called *expansion by row* and is illustrated below for a 3×3 matrix. In this case we are expanding by row 1, which means deleting row 1 and successively deleting columns 1, column 2, and column 3 to create three 2×2 matrices. The determinant of each smaller matrix is multiplied by the entry corresponding to the intersection of the deleted row and column. The expansion alternately adds and subtracts each successive determinant.

$$\begin{vmatrix} -1 & 4 & 3 \\ 2 & 6 & 4 \\ 3 & -2 & 8 \end{vmatrix} = (-1)\begin{vmatrix} 6 & 4 \\ -2 & 8 \end{vmatrix} - (4)\begin{vmatrix} 2 & 4 \\ 3 & 8 \end{vmatrix} + (3)\begin{vmatrix} 2 & 6 \\ 3 & -2 \end{vmatrix} =$$

$$-1(6 \cdot 8 - 4 \cdot -2) - 4(2 \cdot 8 - 4 \cdot 3) + 3(2 \cdot -2 - 3 \cdot 6) =$$

$$-56 - 16 - 66 = -138$$

The determinant of a 4×4 matrix would be found by expanding across row 1 to alternately add and subtract 4 3×3 determinants, which would themselves be expanded to produce a series of 2×2 determinants that would be reduced as above. This procedure can be applied to find the determinant of an arbitrarily large square matrix.

7.8 Eigenvectors and Eigenvalues

An *eigenvector* is a nonzero vector that satisfies the equation

$$A\vec{v} = \lambda\vec{v}$$

where A is a square matrix, λ is a scalar, and $\vec{v}$ is the eigenvector. λ is called an *eigenvalue*. Eigenvalues and eigenvectors are also known as, respectively, *characteristic roots* and *characteristic vectors*, or *latent roots* and *latent vectors*.

You can find eigenvalues and eigenvectors by treating a matrix as a system of linear equations and solving for the values of the variables that make up the components of the eigenvector. For example, finding the eigenvalues and corresponding eigenvectors of the matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

means applying the above formula to get

$$A\vec{v} = \lambda\vec{v} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in order to solve for λ, x_1 and x_2. This statement is equivalent to the system of equations

$$2x_1 + x_2 = \lambda x_1$$

$$x_1 + 2x_2 = \lambda x_2$$

which can be rearranged as

$$(2 - \lambda)x_1 + x_2 = 0$$

$$x_1 + (2 - \lambda)x_2 = 0$$

A necessary and sufficient condition for this system to have a nonzero vector $[x_1, x_2]$ is that the determinant of the coefficient matrix

$$\begin{bmatrix} (2 - \lambda) & 1 \\ 1 & (2 - \lambda) \end{bmatrix}$$

be equal to zero. Accordingly,

$$\begin{vmatrix} (2 - \lambda) & 1 \\ 1 & (2 - \lambda) \end{vmatrix} = 0$$

$$(2 - \lambda)(2 - \lambda) - 1 \cdot 1 = 0$$

$$\lambda^2 - 4\lambda + 3 = 0$$

$$(\lambda - 3)(\lambda - 1) = 0$$

There are two values of λ that satisfy the last equation; thus there are two eigenvalues of the original matrix A and these are $\lambda_1 = 3, \lambda_2 = 1$.

We can find eigenvectors which correspond to these eigenvalues by plugging λ back in to the equations above and solving for x_1 and x_2. To find an eigenvector corresponding to $\lambda = 3$, start with

$$(2 - \lambda)x_1 + x_2 = 0$$

and substitute to get

$$(2-3)x_1 + x_2 = 0$$

which reduces and rearranges to

$$x_1 = x_2$$

There are an infinite number of values for x_1 which satisfy this equation; the only restriction is that not all the components in an eigenvector can equal zero. So if $x_1 = 1$, then $x_2 = 1$ and an eigenvector corresponding to $\lambda = 3$ is $[1, 1]$.

Finding an eigenvector for $\lambda = 1$ works the same way.

$$(2-1)x_1 + x_2 = 0$$

$$x_1 = -x_2$$

So an eigenvector for $\lambda = 1$ is $[1, -1]$.

8 Singular Value Decomposition

Singular value decomposition (SVD) can be looked at from three mutually compatible points of view. On the one hand, we can see it as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation. This ties in to the third way of viewing SVD, which is that once we have identified where the most variation is, it's possible to find the best approximation of the original data points using fewer dimensions. Hence, SVD can be seen as a method for data reduction.

As an illustration of these ideas, consider the 2-dimensional data points in Figure 1. The regression line running through them shows the best approximation of the original data with a 1-dimensional object (a line). It is the best approximation in the sense that it is the line that minimizes the distance between each original point and the line. If we drew a perpendicular line from each point to the regression line, and took the intersection of those lines as the approximation of the original datapoint, we would have a reduced representation of the original data that captures as much of the original variation as possible. Notice that there is a second regression line, perpendicular to the first, shown in Figure 2. This line captures as much of the variation as possible along the second dimension of the original data set. It does a poorer job of approximating the orginal data because it corresponds to a dimension exhibiting less variation to begin with. It is possible to use these regression lines to generate a set of uncorrelated data points that will show subgroupings in the original data not necessarily visible at first glance.

These are the basic ideas behind SVD: taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the

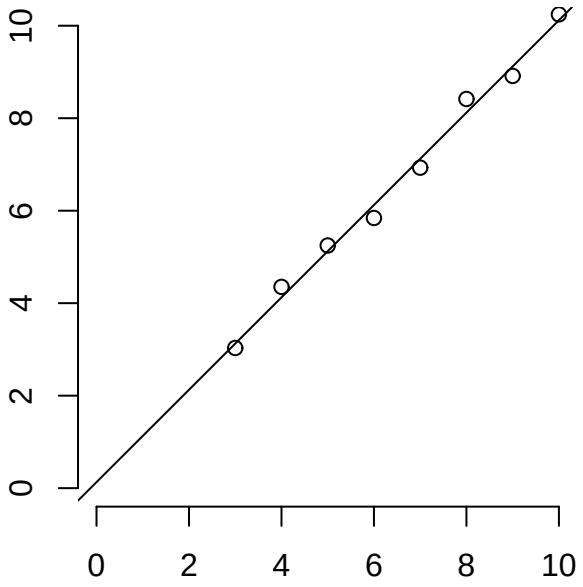

Figure 1: Best-fit regression line reduces data from two dimensions into one.

original data more clearly and orders it from most variation to the least. What makes SVD practical for NLP applications is that you can simply ignore variation below a particular threshhold to massively reduce your data but be assured that the main relationships of interest have been preserved.

8.1 Example of Full Singular Value Decomposition

SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V. The theorem is usually presented something like this:

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T$$

where $U^T U = I, V^T V = I$; the columns of U are orthonormal eigenvectors of AA^T, the columns of V are orthonormal eigenvectors of $A^T A$, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order.

The following example merely applies this definition to a small matrix in order to compute its SVD. In the next section, I attempt to interpret the application of SVD to document classification.

Start with the matrix

$$A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

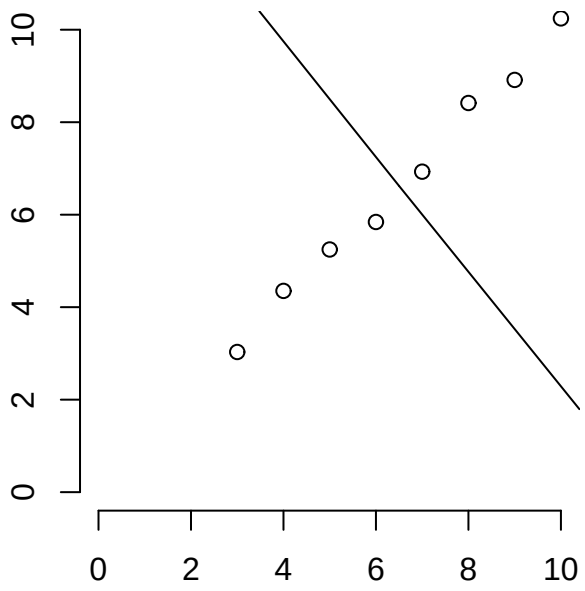

Figure 2: Regression line along second dimension captures less variation in original data.

In order to find U, we have to start with AA^T. The transpose of A is

$$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$$

so

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

Next, we have to find the eigenvalues and corresponding eigenvectors of AA^T. We know that eigenvectors are defined by the equation $A\vec{v} = \lambda\vec{v}$, and applying this to AA^T gives us

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We rewrite this as the set of equations

$$11x_1 + x_2 = \lambda x_1$$

$$x_1 + 11x_2 = \lambda x_2$$

and rearrange to get

$$(11 - \lambda)x_1 + x_2 = 0$$

$$x_1 + (11 - \lambda)x_2 = 0$$

Solve for λ by setting the determinant of the coefficient matrix to zero,

$$\begin{vmatrix} (11-\lambda) & 1 \\ 1 & (11-\lambda) \end{vmatrix} = 0$$

which works out as

$$(11 - \lambda)(11 - \lambda) - 1 \cdot 1 = 0$$

$$(\lambda - 10)(\lambda - 12) = 0$$

$$\lambda = 10, \lambda = 12$$

to give us our two eigenvalues $\lambda = 10, \lambda = 12$. Plugging λ back in to the original equations gives us our eigenvectors. For $\lambda = 10$ we get

$$(11 - 10)x_1 + x_2 = 0$$

$$x_1 = -x_2$$

which is true for lots of values, so we'll pick $x_1 = 1$ and $x_2 = -1$ since those are small and easier to work with. Thus, we have the eigenvector $[1, -1]$ corresponding to the eigenvalue $\lambda = 10$. For $\lambda = 12$ we have

$$(11 - 12)x_1 + x_2 = 0$$

$$x_1 = x_2$$

and for the same reason as before we'll take $x_1 = 1$ and $x_2 = 1$. Now, for $\lambda = 12$ we have the eigenvector $[1, 1]$. These eigenvectors become column vectors in a matrix ordered by the size of the corresponding eigenvalue. In other words, the eigenvector of the largest eigenvalue is column one, the eigenvector of the next largest eigenvalue is column two, and so forth and so on until we have the eigenvector of the smallest eigenvalue as the last column of our matrix. In the matrix below, the eigenvector for $\lambda = 12$ is column one, and the eigenvector for $\lambda = 10$ is column two.

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Finally, we have to convert this matrix into an orthogonal matrix which we do by applying the Gram-Schmidt orthonormalization process to the column vectors. Begin by normalizing $\vec{v_1}$.

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = \frac{[1,1]}{\sqrt{1^2 + 1^2}} = \frac{[1,1]}{\sqrt{2}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

Compute

$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} =$$

$$[1,-1]-[\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}]\cdot[1,-1]*[\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}]=$$

$$[1,-1]-0*[\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}]=[1,-1]-[0,0]=[1,-1]$$

and normalize

$$\vec{u_2}=\frac{\vec{w_2}}{|\vec{w_2}|}=[\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}}]$$

to give

$$U=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}}\end{bmatrix}$$

The calculation of V is similar. V is based on A^TA, so we have

$$A^TA=\begin{bmatrix}3 & -1\\ 1 & 3\\ 1 & 1\end{bmatrix}\begin{bmatrix}3 & 1 & 1\\ -1 & 3 & 1\end{bmatrix}=\begin{bmatrix}10 & 0 & 2\\ 0 & 10 & 4\\ 2 & 4 & 2\end{bmatrix}$$

Find the eigenvalues of A^TA by

$$\begin{bmatrix}10 & 0 & 2\\ 0 & 10 & 4\\ 2 & 4 & 2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\lambda\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$$

which represents the system of equations

$$10x_1+2x_3=\lambda x_1$$

$$10x_2+4x_3=\lambda x_2$$

$$2x_1+4x_2+2x_3=\lambda x_2$$

which rewrite as

$$(10-\lambda)x_1+2x_3=0$$

$$(10-\lambda)x_2+4x_3=0$$

$$2x_1+4x_2+(2-\lambda)x_3=0$$

which are solved by setting

$$\begin{vmatrix}(10-\lambda) & 0 & 2\\ 0 & (10-\lambda) & 4\\ 2 & 4 & (2-\lambda)\end{vmatrix}=0$$

This works out as

$$(10-\lambda)\begin{vmatrix}(10-\lambda) & 4\\ 4 & (2-\lambda)\end{vmatrix}+2\begin{vmatrix}0 & (10-\lambda)\\ 2 & 4\end{vmatrix}=$$

$$(10-\lambda)[(10-\lambda)(2-\lambda)-16]+2[0-(20-2\lambda)]=$$

$$\lambda(\lambda-10)(\lambda-12)=0,$$

so $\lambda=0,\lambda=10,\lambda=12$ are the eigenvalues for A^TA. Substituting λ back into the original equations to find corresponding eigenvectors yields for $\lambda=12$

$$(10-12)x_1+2x_3=-2x_1+2x_3=0$$

$$x_1=1,x_3=1$$

$$(10-12)x_2+4x_3=-2x_2+4x_3=0$$

$$x_2=2x_3$$

$$x_2=2$$

So for $\lambda=12$, $\vec{v_1}=[1,2,1]$. For $\lambda=10$ we have

$$(10-10)x_1+2x_3=2x_3=0$$

$$x_3=0$$

$$2x_1+4x_2=0$$

$$x_1=-2x_2$$

$$x_1=2,x_2=-1$$

which means for $\lambda=10$, $\vec{v_2}=[2,-1,0]$. For $\lambda=0$ we have

$$10x_1+2x_3=0$$

$$x_3=-5$$

$$10x_1-20=0$$

$$x_2=2$$

$$2x_1+8-10=0$$

$$x_1=1$$

which means for $\lambda=0$, $\vec{v_3}=[1,2,-5]$. Order $\vec{v_1}$, $\vec{v_2}$, and $\vec{v_3}$ as column vectors in a matrix according to the size of the eigenvalue to get

$$\begin{bmatrix}1 & 2 & 1\\ 2 & -1 & 2\\ 1 & 0 & -5\end{bmatrix}$$

and use the Gram-Schmidt orthonormalization process to convert that to an orthonormal matrix.

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = [\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$$

$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} = [2, -1, 0]$$

$$\vec{u_2} = \frac{\vec{w_2}}{|\vec{w_2}|} = [\frac{2}{\sqrt{5}}, \frac{-1}{\sqrt{5}}, 0]$$

$$\vec{w_3} = \vec{v_3} - \vec{u_1} \cdot \vec{v_3} * \vec{u_1} - \vec{u_2} \cdot \vec{v_3} * \vec{u_2} = [\frac{-2}{3}, \frac{-4}{3}, \frac{10}{3}]$$

$$\vec{u_3} = \frac{\vec{w_3}}{|\vec{w_3}|} = [\frac{1}{\sqrt{30}}, \frac{2}{\sqrt{30}}, \frac{-5}{\sqrt{30}}]$$

All this to give us

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

when we really want its transpose

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

For S we take the square roots of the non-zero eigenvalues and populate the diagonal with them, putting the largest in s_{11}, the next largest in s_{22} and so on until the smallest value ends up in s_{mm}. The non-zero eigenvalues of U and V are always the same, so that's why it doesn't matter which one we take them from. Because we are doing full SVD, instead of reduced SVD (next section), we have to add a zero column vector to S so that it is of the proper dimensions to allow multiplication between U and V. The diagonal entries in S are the singular values of A, the columns in U are called left singular vectors, and the columns in V are called right singular vectors.

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$

Now we have all the pieces of the puzzle

$$A_{mn} = U_{mm} S_{mn} V^T_{nn} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\sqrt{12}}{\sqrt{2}} & \frac{\sqrt{10}}{\sqrt{2}} & 0 \\ \frac{\sqrt{12}}{\sqrt{2}} & \frac{-\sqrt{10}}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

8.2 Example of Reduced Singular Value Decomposition

Reduced singular value decomposition is the mathematical technique underlying a type of document retrieval and word similarity method variously called *Latent Semantic Indexing* or *Latent Semantic Analysis*. The insight underlying the use of SVD for these tasks is that it takes the original data, usually consisting of some variant of a word×document matrix, and breaks it down into linearly independent components. These components are in some sense an abstraction away from the noisy correlations found in the original data to sets of values that best approximate the underlying structure of the dataset along each dimension independently. Because the majority of those components are very small, they can be ignored, resulting in an approximation of the data that contains substantially fewer dimensions than the original. SVD has the added benefit that in the process of dimensionality reduction, the representation of items that share substructure become more similar to each other, and items that were dissimilar to begin with may become more dissimilar as well. In practical terms, this means that documents about a particular topic become more similar even if the exact same words don't appear in all of them.

As we've already seen, SVD starts with a matrix, so we'll take the following word× document matrix as the starting point of the next example.

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix}$$

Remember that to compute the SVD of a matrix A we want the product of three matrices such that

$$A = USV^T$$

where U and V are orthonormal and S is diagonal. The column vectors of U are taken from the orthonormal eigenvectors of AA^T, and ordered right to left from largest corresponding eigenvalue to the least. Notice that

$$AA^T = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} 2 & 1 & 5 & 7 & 0 \\ 0 & 6 & 0 & 0 & 10 \\ 8 & 0 & 7 & 8 & 0 \\ 6 & 1 & 4 & 5 & 0 \\ 0 & 7 & 0 & 0 & 7 \end{bmatrix} = \begin{bmatrix} 104 & 8 & 90 & 108 & 0 \\ 8 & 87 & 9 & 12 & 109 \\ 90 & 9 & 90 & 111 & 0 \\ 108 & 12 & 111 & 138 & 0 \\ 0 & 109 & 0 & 0 & 149 \end{bmatrix}$$

is a matrix whose values are the dot product of all the terms, so it is a kind of dispersion matrix of terms throughout all the documents. The singular values (eigenvalues) of AA^T are

$$\lambda = 321.07, \lambda = 230.17, \lambda = 12.70, \lambda = 3.94, \lambda = 0.12$$

which are used to compute and order the corresponding orthonormal singular vectors of U.

$$U = \begin{bmatrix} -0.54 & 0.07 & 0.82 & -0.11 & 0.12 \\ -0.10 & -0.59 & -0.11 & -0.79 & -0.06 \\ -0.53 & 0.06 & -0.21 & 0.12 & -0.81 \\ -0.65 & 0.07 & -0.51 & 0.06 & 0.56 \\ -0.06 & -0.80 & 0.09 & 0.59 & 0.04 \end{bmatrix}$$

This essentially gives a matrix in which words are represented as row vectors containing linearly independent components. Some word cooccurence patterns in these documents are indicated by the signs of the coefficients in U. For example, the signs in the first column vector are all negative, indicating the general cooccurence of words and documents. There are two groups visible in the second column vector of U: *car* and *wheel* have negative coefficients, while *doctor*, *nurse*, and *hospital* are all positive, indicating a grouping in which *wheel* only cooccurs with *car*. The third dimension indicates a grouping in which *car*, *nurse*, and *hospital* occur only with each other. The fourth dimension points out a pattern in which *nurse* and *hospital* occur in the absence of *wheel*, and the fifth dimension indicates a grouping in which *doctor* and *hospital* occur in the absence of *wheel*.

Computing V^T is similar. Since its values come from orthonormal singular vectors of A^TA, arranged right to left from largest corresponding singular value to the least, we have

$$A^TA = \begin{bmatrix} 79 & 6 & 107 & 68 & 7 \\ 6 & 136 & 0 & 6 & 112 \\ 107 & 0 & 177 & 116 & 0 \\ 68 & 6 & 116 & 78 & 7 \\ 7 & 112 & 0 & 7 & 98 \end{bmatrix}$$

which contains the dot product of all the documents. Applying the Gram-Schmidt orthonormalization process and taking the transpose yields

$$V^T = \begin{bmatrix} -0.46 & 0.02 & -0.87 & -0.00 & 0.17 \\ -0.07 & -0.76 & 0.06 & 0.60 & 0.23 \\ -0.74 & 0.10 & 0.28 & 0.22 & -0.56 \\ -0.48 & 0.03 & 0.40 & -0.33 & 0.70 \\ -0.07 & -0.64 & -0.04 & -0.69 & -0.32 \end{bmatrix}$$

S contains the square roots of the singular values ordered from greatest to least along its diagonal. These values indicate the variance of the linearly independent components along each dimension. In order to illustrate the effect of dimensionality reduction on this data set, we'll restrict S to the first three singular values to get

$$S = \begin{bmatrix} 17.92 & 0 & 0 \\ 0 & 15.17 & 0 \\ 0 & 0 & 3.56 \end{bmatrix}$$

In order for the matrix multiplication to go through, we have to eliminate the corresponding row vectors of U and corresponding column vectors of V^T to give us an approximation of A using 3 dimensions instead of the original 5. The result looks like this.

$$\hat{A} =$$

$$\begin{bmatrix} -0.54 & 0.07 & 0.82 \\ -0.10 & -0.59 & -0.11 \\ -0.53 & 0.06 & -0.21 \\ -0.65 & 0.07 & -0.51 \\ -0.06 & -0.80 & 0.09 \end{bmatrix} \begin{bmatrix} 17.92 & 0 & 0 \\ 0 & 15.17 & 0 \\ 0 & 0 & 3.56 \end{bmatrix} \begin{bmatrix} -0.46 & 0.02 & -0.87 & -0.00 & 0.17 \\ -0.07 & -0.76 & 0.06 & 0.60 & 0.23 \\ -0.74 & 0.10 & 0.28 & 0.22 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} 2.29 & -0.66 & 9.33 & 1.25 & -3.09 \\ 1.77 & 6.76 & 0.90 & -5.50 & -2.13 \\ 4.86 & -0.96 & 8.01 & 0.38 & -0.97 \\ 6.62 & -1.23 & 9.58 & 0.24 & -0.71 \\ 1.14 & 9.19 & 0.33 & -7.19 & -3.13 \end{bmatrix}$$

In practice, however, the purpose is not to actually reconstruct the original matrix but to use the reduced dimensionality representation to identify similar words and documents. Documents are now represented by row vectors in V, and document similarity is obtained by comparing rows in the matrix VS (note that documents are represented as row vectors because we are working with V, not V^T). Words are represented by row vectors in U, and word similarity can be measured by computing row similarity in US.

Earlier I mentioned that in the process of dimensionality reduction, SVD makes similar items appear more similar, and unlike items more unlike. This can be explained by looking at the vectors in the reduced versions of U and V above. We know that the vectors contain components ordered from most to least amount of variation accounted for in the original data. By deleting elements representing dimensions which do not exhibit meaningful variation, we effectively eliminate noise in the representation of word vectors. Now the word vectors are shorter, and contain only the elements that account for the most significant correlations among words in the original dataset. The deleted elements had the effect of diluting these main correlations by introducing potential similarity along dimensions of questionable significance.

9 References

Deerwester, S., Dumais, S., Landauer, T., Furnas, G. and Harshman, R. (1990). "Indexing by Latent Semantic Analysis". Journal of the American Society of Information Science 41(6):391-407.

Ientilucci, E.J., (2003). "Using the Singular Value Decomposition". http://www.cis.rit.edu/~ejipci/research.htm

Jackson, J. E. (1991). *A User's Guide to Principal Components Analysis.* John Wiley & Sons, NY.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA.

Marcus, M. and Minc, H. (1968). *Elementary Linear Algebra.* The MacMillan Company, NY.

perfectly