AMOD 5450H: Data Mining

Singular Value Decomposition

Submitted by: Nicholas Hopewell

Submitted to: Dr. Sabine McConnell

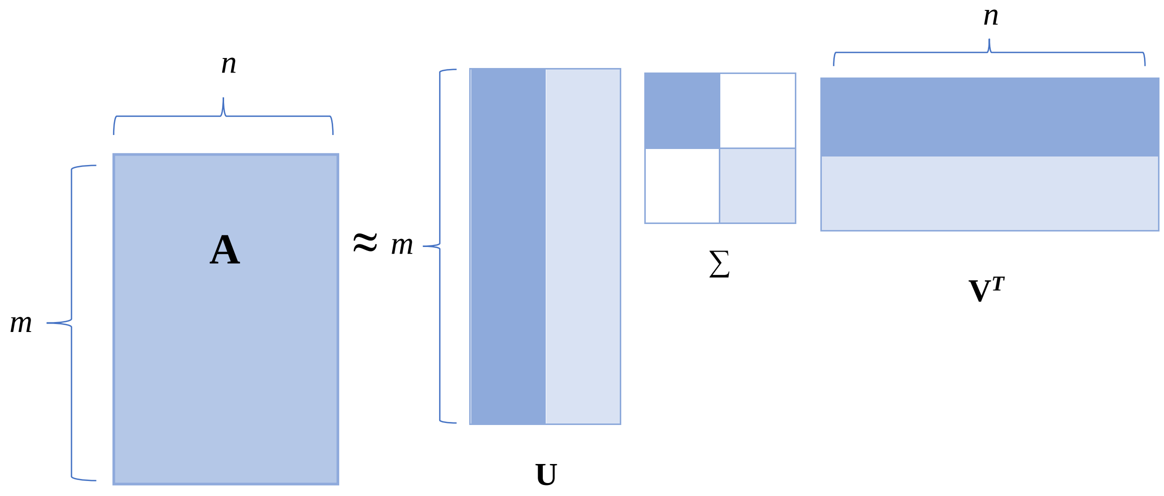
January 2, 2018

The goal of this paper is to describe singular value decomposition in a way which is understandable to individuals not of a mathematics background. Singular value decomposition (SVD) is a commonly used linear algebra technique which has practical applications in many areas including computer science, data analysis, and statistics. Singular value decomposition is used as a method of reducing the dimensionality of a data matrix. The dimensionality of data refers to its number of attributes; dimensionality reduction is often a necessary step when working with data which contain a large number of attributes. One of the primary benefits of reducing the dimensionality of a dataset is that it allows datamining algorithms to achieve better results in less time and required memory (Tan, Steinbach, & Kumar, 2005). This improved performance after dimensionality reduction has to do with the fact irrelevant and redundant features have been, at least partially, eliminated. Another reason algorithms tend to preform better on data of lower dimensions is because of something referred to as the 'curse of dimensionality.' This term simply refers to the idea that most analyses become much more difficult as the dimensionality of data increases. At high, dimensions classification becomes unreliable with low accuracy (due to increased sparseness of the data), and the distance measures used for clustering data become less meaningful and result in clusters of inferior quality (Tan, Steinbach, & Kumar, 2005).

Since SVD is used to reduce the dimensions of a matrix, a data matrix must be taken as input. A data matrix is simply a format to organize a set of numbers as a structure of rows and columns. Any matrix can be described in terms of its number of rows and columns where *m* refers to the number of rows and *n* refers to the number of columns. The product of the rows and columns is equal to the size of the matrix. Matrices are often called '*m* x *n* matrices' for this reason (Tan, Steinbach, & Kumar, 2005). In many circumstances, a matrix is represented as the uppercase letter A (or some other uppercase letter). The transpose of a matrix is simply the result of swapping the rows and the columns of the matrix and is often denoted as A^{T} (Tan, Steinbach, & Kumar, 2005). Below is an example of a matrix and its transpose.

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 7 & 6 & 2 \end{bmatrix} \; A^T = \begin{bmatrix} 3 & 7 \\ 4 & 6 \\ 5 & 2 \end{bmatrix}$$

The underlying goal of SVD is to take an input matrix of values and represent it as a product of three different matrices. The theorem states that given any *m* x *n* real matrix, the matrix can be expressed by the following equation: A = U∑V^{T}. A ‘real matrix’ is simply one that contains real numbers (those which fall on a number line) rather than complex numbers (combination of real and imaginary numbers). This formula provides a standard format for any real matrix input to be decomposed into. Below is a way to visualize the equation where matrix A is a product of the matrices U, ∑, and V^{T}.



The columns of matrix U contain what are called the left singular vectors while the columns of matrix V^T contain the right singular vectors (Tomasi, 2013: Strang, 2016). U and V^T are orthonormal because they have a Euclidean value of one which means that the sum of the squared values of each column equals one (the 'normal' part of orthonormal), and because their columns are orthogonal. The columns are orthogonal because if one takes any column from U and a corresponding column from V^T and multiples them together, adds the resulting products, the result will equal a value of zero (Tomasi, 2013: Strang, 2016). This is called taking a dot product; a dot product of two non-zero vectors is equal to zero only in the case where those vectors are perpendicular (Tan, Steinbach, & Kumar, 2005). As seen in the visual example above, the angle between U and V^T is geometrically 90 degrees making them orthogonal. In addition, matrix ∑ contains non-zero numbers only on the diagonal of the matrix, and these values are all positive numbers and are sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots \geq 0$) and are referred to as the singular values.

A description of these terms cannot be fully fleshed-out without going into the algebra; this description would not be very beneficial and is unnecessary to understand what the numbers represent (the representation is the important part). Very briefly, the right singular vectors come from the eigenvectors of A^TA (which are the columns of matrix V^T), and the left singular vectors (the columns of matrix U) come from the eigenvectors of AA^T (Tan, Steinbach, & Kumar, 2005). The singular values of matrix ∑ are the square roots of the non-zero eigenvalues of A^TA and AA^T (Tan, Steinbach, & Kumar, 2005). Eigenvalues and eigenvectors represent very similar things in comparison to singular vectors and singular values in that they capture the structure of a matrix by factoring (decomposing) it into the product of these three matrices (Tan, Steinbach, & Kumar, 2005). To understand eigenvalues, what is most important is that the relative size of an eigenvalue gives information about how important a concept is to the data. Thus, it is best to think about the singular values as the strength of these concepts, ordered in terms of magnitude to reflect the strength of each concept in decreasing fashion. Another way to think about this is that relatively large eigenvalues reflect that a certain concept accounts for a substantial proportion of explained variance in the original data. To understand what is meant by ‘concepts’ it is best to go over an example.

Imagine we had a matrix which contained values of reader ratings of different novels. Each column of the matrix represented one novel while each row represented a reader. Readers rated how much they enjoyed each novel on a scale from 0 – 5 where 0 = the novel was of no interest and 5 = the novel was very interesting. For transparency purposes I should mention that I am going to make up all the data as well as the values of the three matrices U, ∑, and V^T simply to make the process more understandable. Suppose this is the novel ratings matrix:

	Howl’s Moving Castle	The Lord of the Rings	The Lion, the Witch and the Wardrobe	Neuromancer	Ringworld
Reader 1	1	2	1	0	0
Reader 2	3	3	4	0	0
Reader 3	4	4	5	0	0
Reader 4	4	5	5	0	0
Reader 5	2	0	0	4	4
Reader 6	0	0	0	5	4
Reader 7	0	0	1	3	2

The same data in a rectangular square bracket matrix:

$$Novel\ Matrix\ (A) = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 \\ 3 & 3 & 4 & 0 & 0 \\ 4 & 4 & 5 & 0 & 0 \\ 4 & 5 & 5 & 0 & 0 \\ 2 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 1 & 3 & 2 \end{bmatrix}$$

If this matrix were decomposed into the long matrix U, the diagonal matrix $\sum$, and the wide matrix V^T, in some sense we would look to discover concepts which underlie the original data. In this example, the matrix is quite small and not very complex. If, for instance, the data were very complex (highly-dimensional) it would be necessary to reduce this complexity down into a smaller number of predominant concepts. These concepts should be more intuitive to understand and capture an adequate amount of the total information in a simpler number of dimensions (i.e., account for an adequate amount of explained variance in the data). If the data had a lot of irrelevant information and redundant attributes which repeated information in some way, focusing on the underlying concepts would avoid these issues. Overall, this information can be reduced into a smaller amount of information while still retaining as much of the original information as necessary (especially the essential information).

But what exactly are these ‘concepts’ which can be extracted from the data matrix? Closely inspecting the original data (in particular, the square bracket matrix) it is quite easy to see, in this example, that the preferences of these readers can be simply summarized as two groups of readers: those who like fantasy novels and those who like science-fiction novels. Readers 1, 2, 3 and 4 all rated fantasy novels as interesting and science-fiction novels as uninteresting. On the other hand, readers 5, 6, and 7 all rated fantasy novels as uninteresting and science-fiction novels as interesting. To help visualize these clustering's of movie preferences:

$$Novel\ Matrix\ (A) = \begin{bmatrix} \boxed{\begin{matrix} 1 & 2 & 1 \\ 3 & 3 & 4 \\ 4 & 4 & 5 \\ 4 & 5 & 5 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} & \boxed{\begin{matrix} 4 & 4 \\ 5 & 5 \\ 3 & 2 \end{matrix}} \end{bmatrix}$$

So, the novels fall into fantasy and science-fiction novels and the readers fall into fantasy and science-fiction lovers. These concepts are sometimes referred to as latent variables or dimensions, latent factors (in factor analysis), or principal components (in principal components analysis). Singular value decomposition allows these latent dimensions, or concepts, to be extracted from the original data (Tan, Steinbach, & Kumar, 2005). Inputting this matrix into the SVD algorithm would produce something like this (recall that I am making up these values simply for learning purposes):

$$\mathbf{A = U\sum V}^{\boldsymbol{T}}$$

$$\underset{\mathbf{A}}{\begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{4} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{5} & 0 & 0 \\ \mathbf{4} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ \mathbf{2} & 0 & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & 0 & \mathbf{1} & \mathbf{3} & \mathbf{2} \end{bmatrix}} = \underset{\mathbf{U}}{\begin{bmatrix} .\mathbf{16} & .03 & -.01 \\ .\mathbf{42} & .08 & -.02 \\ .\mathbf{57} & .09 & -.05 \\ .\mathbf{67} & .13 & -.06 \\ .17 & -.\mathbf{49} & .\mathbf{61} \\ .05 & -.\mathbf{81} & .05 \\ .04 & -.\mathbf{39} & .\mathbf{29} \end{bmatrix}} \text{ x } \underset{\sum}{\begin{bmatrix} \mathbf{13} & 0 & 0 \\ 0 & \mathbf{8.1} & 0 \\ 0 & 0 & \mathbf{1.7} \end{bmatrix}} \text{ x } \underset{\mathbf{V}^{\boldsymbol{T}}}{\begin{bmatrix} .\mathbf{58} & .\mathbf{64} & .\mathbf{57} & .09 & .08 \\ 0.9 & .06 & .14 & -.\mathbf{71} & -.\mathbf{66} \\ .-\mathbf{52} & .04 & -.\mathbf{41} & .08 & .11 \end{bmatrix}}$$

The above example result of a singular value decomposition can be interpreted in a couple steps. Firstly, the columns of matrix U can be thought of as the concepts or factors previously explained: we can think of the first column as the fantasy concept, the second column as the science-fiction concept, and the third and final column as a less meaningful concept which happens to be noise in the data (determined by looking at the magnitude of the third value of matrix ∑). As seen in this matrix, the first four readers most strongly associate with the first concept (fantasy), and the last three readers strongly associate with the second concept (science-fiction). Also notice that some of the science-fiction readers associate with the noise concept, this is simply because they rated some interest in certain fantasy novels while the fantasy readers rated all science-fiction novels as not interesting. Interpreting matrix U in such a way, it is useful to think about this matrix as a record, instance, or in this case, reader to concept similarity matrix. Therefore, if A is an *m* x *n* matrix, it is helpful to think of U as an *m* x *r* matrix where r is equal to the number of concepts or factors, ranked in order of their strengths.

The second matrix, ∑, contains the singular values which denote the strength of each of these concepts as well as the noise in the data. Recall that the singular values sit along the diagonal of the matrix and are represented as σ_1 to σ_N. It appears that the strength of the fantasy concept is quite strong which makes sense as fantasy readers are more represented in the matrix compared to science-fiction readers. The strength of the science-fiction concept is again quite strong, but not as strong as the fantasy concept, and the final singular value represents noise in the data. This matrix can be thought of as an indicator of how present a certain concept /factor appears to be within the original data. Therefore, if A is an *m* x *n* matrix, it is helpful to think of ∑ as an *r* x *r* matrix (where r is equal to the number of concepts or factors).

Finally, matrix V^T can be thought of as an attribute to concept matrix. Recall the attributes (columns) of the matrix were five different novels: Howl's Moving Castle; The Lord of the Rings; The Lion, the Witch and the Wardrobe; Neuromancer; and Ringworld. Looking at this matrix, the first three novels strongly associate with the first concept (fantasy) and the last two novels strongly associate with the second concept (science-fiction). Notice also that the first and third novels associate with the third concept which was actually just noise in the data. This is because reader five and reader seven, who were both science-fiction fans, rated those two novels as somewhat interesting despite them being fantasy novels. Therefore, if A is an m x n matrix, it is helpful to think of V^T as an n x r matrix (where r is equal to the number of concepts or factors).

In conclusion, after the singular value decomposition is applied to a matrix, one can easily and understandably interpret the overall concepts or factors which are present in the data. The strength of these concepts also becomes apparent as well the noise which has also been modeled by SVD. SVD also allows the association between each record and each concept, as well as the association between each attribute and each concept, to be interpreted. Importantly though, notice how the SVD gives standardized values between -1 and 1; this standardization allows for any matrix to be decomposed in such a way and allows clear understanding of the strengths of factors and their associations. Finally, it is important to realize that some investigation and domain knowledge is necessary to interpret the results of SVD as was done above. If one does not first have knowledge about what the attributes mean and some domain knowledge about how they might be related or connected in diverse ways for different circumstances, interpreting the results of SVD would not be straight forward.

Singular value decomposition, as previously mentioned, has many applications across various domains. One of the more common applications of SVD is in imagine processing and compression (Lijie, 2012: Hladnik, 2013: Ogden & Huff, 1997). Any image with any number of pixels is essential a matrix of numbers of equivalent dimensions. For instance, a 1000 x 2000-pixel image can be represented as a 1000 x 2000 matrix, and thus, can reduced using SVD. Each entry of the matrix would correspond to a pixel of the image and its numerical value would be associated with the colour or shade of the pixel. Recall that the singular values represent the strength of a concept. This understanding is useful in the example of novel readers but for the example of image processing, singular values can be thought of as directions which are magnified the most by the images matrix. A more straightforward way to understand this idea is that the first singular value captures the most information about the image, while the second singular value captures the second most information about the image and so on (in decreasing order).

Extending this idea, if an image needed to be compressed (represented with less information/pixels), SVD could produce what is known as a 'low-rank approximation' of the matrix by specifying a small number of singular values to retain (might refer to this value as 'k'). For increased values of k, the SVD algorithm would iterate a greater number of times before converging on the approximation, producing a clearer and clearer image with each iteration (at the cost of a reduced compression ratio). Therefore, if one needed a very compressed image, a small number singular values could be taken, resulting in a very blurry image mainly capturing the colours and shades of the original image but no specific details. Herein lies the trade off of SVD: retaining a lower number of singular values reduces the dimensionality of the matrix by a lot (while retaining only the most important information about the image and dropping the rest), and retaining a greater number of singular values results in less dimensionality reduction but more of the original information being modelled (Lijie, 2012: Hladnik, 2013). No matter what the decision, SVD can be used to significantly reduce the storage cost of image. This application is useful in many ways, one of which is that it can allow facial recognition technologies (and imagine recognition in general) to perform tasks with considerably less computation time required (Lijie, 2012: Hladnik, 2013: Ogden & Huff, 1997).

Another application of SVD is in web searching and recommendation systems. Recommendation systems is an area of data analysis which as grown rapidly over the years. As online transactions increase, recommendation systems bring more and more interesting products to the attention of consumers (Sarwar, Karypis, Konstan, Riedl, 2002). These systems are increasingly under pressure to scale to meet the massive amounts of available consumer data which businesses have access to. With huge growth of consumer data as well as products available online, these systems also need to constantly improve the quality of their recommendations. Solving these challenges is not an easy task because the shorter a recommendation system searches for recommendations the more scalable it will be but, as result, the recommendations will also be of lower quality. Researchers have shown that SVD can be used for intelligent information retrieval through incorporating dimensionality reduction methods into recommendation systems (Sarwar, Karypis, Konstan, Riedl, 2002). This is because when making recommendations based off of the searches of a consumer and their purchase preferences, the larger trends and directions of the consumers behaviour is most important to the process. Therefore, the massive amount of consumer information, including web searching, can be represented quite adequately with a few singular values. This process effectively reduces the amount of information a recommendation algorithm must work with by orders of magnitude while still retaining the most relevant information for making consumer recommendations.

These are only a couple of the various applications of singular value decomposition. Other applications include tasks such as digital cryptography and watermarking (Benhocine, Laouamer, Nana, & Pascu, 2013), improving clustering of data (Drineas, Frieze, Kannan, Vempala, & Vinay, 2004), and signal processing of noisy signals (Tufts, Kumaresan & Kirsteins, 1982). Singular value decomposition has many advantages in terms of information decomposition and the reduction of computational cost which are necessary for many applications and learning algorithms (these have already been discussed). That being said, the largest and most notable *limitation* of SVD is that, although it produces reduced information which can be worked with more quickly, the process of reducing that information is very computationally expensive (Sarwar, Karypis, Konstan, & Riedl, 2002). This limitation has to do with the fact that SVD is preforming matrix decomposition on the original data; if the original data is massive, this decomposition process may be very intensive and optimal for all tasks.

References

Benhocine A, Laouamer L, Nana L, Pascu AC (2013). New images watermarking scheme based on singular value decomposition. J Inf Hiding Multimedia Signal Process 4(1):9–18

Hladnik, A (2013). Image compression and face recognition: two image processing applications of principal component analysis. International Circular of Graphic Education and Research, 6, 56-61.

Lijie, C (2012). "Singular Value Decomposition Applied To Digital Image Processing", Division of Computing Studies, Arizona State University Polytechnic Campus, Mesa, Arizona 85212, pp. 1 – 15.

Ogden CJ, Huff T (1997). The singular value decomposition and its application in image processing. Lin. Algebra-Maths- 45, College of Redwoods.

P. Drineas, A. Frieze, R. Kannan, S. Vempala, V. Vinay. Clustering large graphs via the singular value decomposition. Mach. Learn. 56**,** 9–33 (2004)

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J. Incremental SVD-Based Algorithms for Highly Scaleable Recommender Systems. Proceedings of the Fifth International Conference on Computer and Information Technology (2002)

Strang, G (2016). *Introduction to Linear Algebra*. 5th ed. Wellesley-Cambridge Press. ISBN: 9780980232776

Tomasi, C (2013). Orthogonal matrices and the singular value decomposition. https://www.cs.duke.edu/courses/fall13/compsci527/notes/svd.pdf

Tufts DW, Kumaresan R, Kirsteins I: Data adaptive signal estimation by singular value decomposition of a data matrix**.** *Proceedings of the IEEE* 1982,70(6):684-685.