State University of New York, College at Geneseo

MATH 345: Numerical Analysis I

Final Project

Singular Value Decomposition Applied to Principal Component Analysis

Authors:
Stephanie Allen
David Clarkson
Shayne O'Brien

Professor:
Dr. Caroline Haddad

FALL 2016

1 Introduction

1.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is a method of decomposing an mxn matrix A into $A = USV^t$ where U is an mxm matrix comprised of the *left singular vectors* $\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u_m}$ of A, V is an nxn matrix comprised of the *right singular vectors* $\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_n}$ of A, and Σ is an mxn rectangular diagonal matrix with the *singular values* $\sigma_1, \sigma_2, ..., \sigma_{Min(n,m)}$. Singular values and singular vectors have similar properties to eigenvalues and eigenvectors, with $A\boldsymbol{v_i} = \sigma_i \boldsymbol{u_i}$ and $A^t \boldsymbol{u_i} = \sigma_i \boldsymbol{v_i}$. They are also related to the composite square matrices A^tA and AA^t, with σ_i being the square roots of the eigenvalues of A^tA and AA^t, $\boldsymbol{v_i}$ consisting of the eigenvectors of A^tA, and $\boldsymbol{u_i}$ being the eigenvectors of AA^t [1, 12, 25].

1.2 Principal Component Analysis

Principal Component Analysis (PCA) is a method of exploratory data analysis for transforming a set of observations of possibly linearly correlated variables into a set of linearly uncorrelated features. This is done by projecting the mxn data matrix X, whose columns are variables and rows the corresponding observations, onto the eigenvectors of the covariance matrix $\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_n}$. This can be done efficiently by finding the singular value decomposition of X and projecting onto its *right singular vectors*, which are the same as the eigenvectors of the covariance matrix. Since higher order principal components of data sets with correlated variables are usually not significant and may thus be discarded without great loss of information, we can use PCA to significantly reduce the dimensionality of a given data set. By doing this, we can provide new perspectives on data sets by making it easier to visualize and explore [24].

1.3 Applications

Given that Singular Value Decomposition is the compression of data based on the number of singular values to be retained, it has wide ranging potential applications. In particular, SVD is used in areas such as machine learning, stock market data analysis, characterization of political positions of congressmen, and the examination of entanglement in quantum computation [6, 11]. Using it, we can completely dissect a matrix for greater insight into where the majority of its information is stored and what happens when we remove parts of this information. In this report, we consider SVD as an application for 1) image compression, and 2) dimensionality reduction of United Nations Development Programme data.

2 Procedure

For our project, we coded a general SVD algorithm from Burden and Faires' 9th edition *Numerical Analysis* textbook. We then used this code as an application for image compression and PCA. Additionally, our group wrote a separate algorithm for PCA using the covariance matrix technique. In the Results section of this paper, we numerically compare PCA using SVD against the covariance matrix technique.

2.1 SVD Generalized Algorithm

In this section, we present pseudocode for SVD, which we based off of the material covered in Section 9.6 of the *Numerical Analysis* textbook for the course [1]. This algorithm is used in the image compression and SVD applied to PCA scripts.

- Find and sort the eigenvalues of A^tA (from largest to smallest). Take the square root of the eigenvalues to get the singular values for A.
- Use $forloops$ to place the singular values on the diagonal of the mxn S matrix.
- Find the *right eigenvectors* of A^tA. Sort the eigenvalues of A^tA, and then sort the *right eigenvectors* according to the order in which the eigenvalues were sorted. These will become the columns of V. Transpose V to get V^t.
- To obtain U:

  - Find the first k columns of U using the k nonzero singular values of A. This is done by solving the formula $u_i = \frac{1}{s_i} A v_i$, where $i = 1, 2, ..., k$ and v_i are the columns of V.
  - Concatenate these k columns using the mxm identity matrix, I_{mxn}.
  - Perform the Gram-Schmidt procedure on the concatenated matrix to produce a set of m orthogonal column vectors. During the process of orthogonalizing each column v to the previous columns, we use the partially orthogonalized v vector. This increases the accuracy of the algorithm and is known as the Modified Gram-Schmidt procedure [16].
  - Remove zero columns from the concatenated matrix which were produced. This is done because the cocatenated matrix must have no more than m orthogonal columns.
  - Normalize the m columns.

- Compute the error of the approximation by calculating the norm of $A - USV^t$ and dividing it by the norm of A.

2.2 Image Compression

Singular Value Decomposition is designed to be used for one dimensional matrices. While one of its useful applications is in image size compression, this data type consists of three dimensions because images are colored using the three additive primary colors red, green, and blue. Each of these requires its own dimension for storage, resulting in a three-dimensional array as the data set for any given image. Images are plotted onto graphs by superimposing the dimensions onto one another. It is important to note here that three-dimensional colored data does not use the same meaning of dimension as in Principal Component Analysis where we reduce the dimensionality of a matrix. Furthermore, in this application, we use SVD in order to reduce the amount of data we much store for an image. According to Noble and Daniel, if we have a matrix A with $rank(A)$, we can choose j number of singular values such that $j < rank(A)$. Using this, we can write $A_j = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t + ... + \sigma_j u_j v_j^t$. This means that we only need to store parts of the U and V matrices [12].

In order to apply SVD to images for compression, we need to significantly alter our existing code to be able to handle a multidimensional array, as opposed to a matrix, input. We do this by separating each of the three primary additive color matrices from one another and apply SVD to each in turn. After this is done, we then reconstruct the matrix for each additive color and concatenate them back together into a single, multidimensional array. To compute the error for our algorithm, we must consider each dimension individually.

When selecting the number of singular values that we would like to remove, k, we use this number for each of the three dimensions so that the image is affected by the change evenly throughout. If we were to specify individual singular values for each of the primary additive color matrices, the output would not be representative of how applying SVD affects individual additive color matrices, rather than the image as a whole. Additionally, MATLAB requires Red-Green-Blue values to lie within the interval [0,1] for them to be plotted. Since the default value range is [0,256], we divide the full array by 256 to convert the values into the required range. If this is not done, the image will not be displayed correctly.

2.3 PCA

2.3.1 Data Description

To test our PCA code, we looked at four Human Development Indicators from the United Nations Development Programme database for 188 countries in the year 2012: life expectancy, gross national product per capita, mean years of schooling, and expected years of schooling (the number of years of education the youth of today should have access to over their lives) [19]. Each column vector $\boldsymbol{x}_i$ in the 188x4 data matrix $\widehat{X}$ represents the set of observations for one of the Human Development variables. A centered set of data was created by subtracting the mean of each column, $\bar{x}_i$, from each variable, $\boldsymbol{x}_i$. The data set was then standardized by dividing the matrix by its standard deviation, s_i, to create the new matrix $X = [\frac{x_i - \bar{x}_i}{s_i}]$ [4, 22]. After these steps were completed, we used our PCA algorithms (one using SVD and one using the covariance matrix) to decrease the dimensions of this data set.

2.3.2 PCA with SVD

We applied the SVD algorithm described in Section 2.1 to our standardized score matrix X. After doing this, we examined the principal components and determined the first two principal components to be sufficiently large for examination, and created a new matrix V which contained the first two right singular vectors/principal components. We then projected X onto the first two principal components to create feature vectors $F = XV^t$, which are each observation's scores on the first two principal components [2, 14, 18, 26]. This reduced our data set from four to two dimensions, at the cost of losing information. The countries which are members of the Organization for Economic Cooperation and Development (OECD) and Non-OECD countries were separated, and a scatterplot was created of their first and second principal component scores [13]. Separating countries in this way made sense because the majority of the "highly developed" countries are OECD members, and PCA can distinguish populations [8, 23].

2.3.3 PCA with Covariance Matrix

A similar procedure was followed for the covariance matrix method as was for the SVD method. The sole difference between these two tests was that instead of calculating the right singular vectors of X using SVD, the eigenvectors of the covariance matrix $\frac{XX^t}{m-1}$ were found instead. The data was then treated identically in that it was projected onto the first two principal components, separated, plotted, and then compared [2, 14, 18, 26].

2.3.4 Considerations

Principal Component Analysis relies on each variable to be measured using the same units. If this is not the case, as it often is not, variables with greater variance will be given more weight in during the component analysis, and relationships between smaller variables will not be accurately reflected or captured. This problem can be avoided by standardizing each variable before applying PCA, as we have done for both methods [22].

Furthermore, for m x n data matrices in which m is significantly greater than n, it is less computationally expensive to use the PCA algorithm in which the eigenvectors of the covariance matrix are found. However, the SVD method is often preferred because it is more numerically stable. While neither of these points affected the quality of our results, for tasks requiring very high precision or working with very large data sets the difference could be sizeable [3].

3 Results

3.1 Image Compression

In this example, we take a proportionally scaled 256x338 pixel .JPG image of a $Canis\ lupus\ familiaris$, or Golden Retriever, and apply SVD. We then visualize how maintaining different numbers of singular values, k, affects image quality. Since the image is 256x338, the original, uncompressed image contains $k = 256$ singular values. In the below subplots, we can see a marked difference between maintaining $k = 4$ versus $k = 256$ singular values, but less so when we compare maintaining $k = 16$ vs. $k = 256$ singular values. When conducting the SVD of this test, we computed the l_2-norm error of the red, green, and blue value matrices to be $1.962103 * 10^{-8}$, $3.202683 * 10^{-8}$, and $1.616040 * 10^{-8}$, respectively.

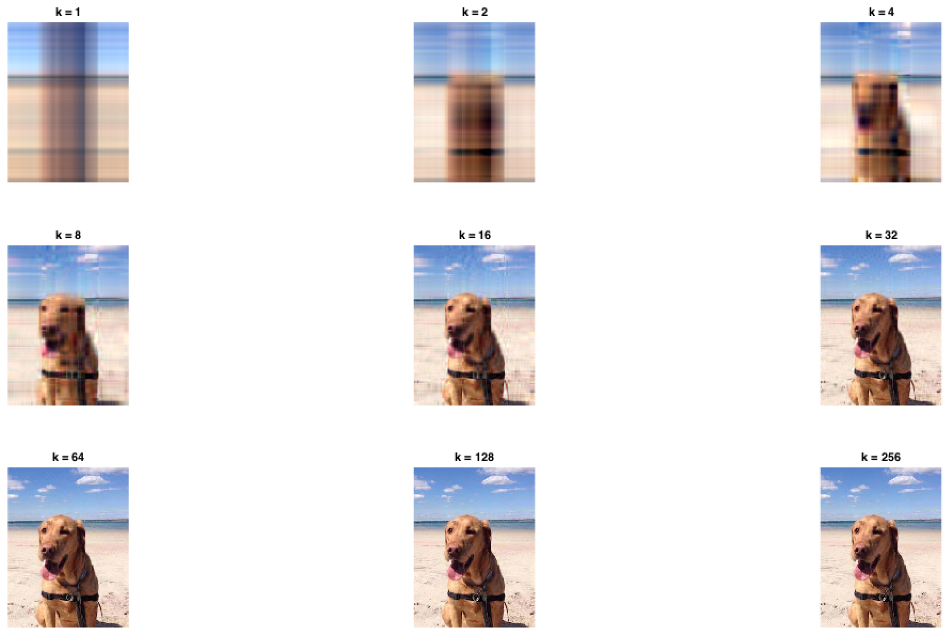


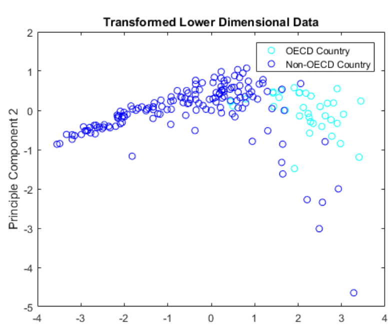
Figure 1: Image Compression for a .JPG image of the *Canis lupus familiaris*, Tucker.

3.2 PCA

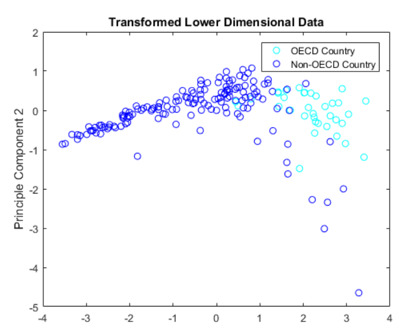
We successfully reduced the dimensions of the four variable human development data set by half to only two variables with acceptable loss from data compression for our purposes: a relative error of 0.317. We did this both by directly calculating a covariance matrix, and more efficiently by calculating the right singular vectors and singular values of X via SVD.

Table 1: Error Calculations for Covariance versus SVD PCA Methods

Method	**Absolute Error**	**Relative Error**
SVD Applied to PCA	7.542083	0.317171
PCA with Covariance Matrix	7.542083	0.317171



(a) Covariance Matrix PCA Graph



(b) SVD PCA Graph

Figure 2: PCA score graphs produced by two separate methods. Notice that there is no discernible difference between the two.

We looked at our data on the principal component axes and noticed a distinct difference between OECD and

Non-OECD countries. OECD countries almost universally had higher scores on the first principal component than the Non-OECD countries.

Table 2: Principal Components of SVD Applied to PCA

Principal Component 1	**Principal Component 2**
0.447325887057755	-0.88876542866507
0.50538977105864	0.180039587323676
0.513467739839755	0.340196219266573
0.529935477413187	0.248894138546581

A glance at the first principal component reveals the reason for this high scoring. OECD Countries are generally developed, industrial nations. The first principal component indicates that all four of these development indicators are positively linearly correlated; since all four development indicators have similar, large, positive values, their scores tend to grow together at roughly the same rate. Industrial nations are likely to score higher on all of these indicators because of their developed infrastructure and strong organization, so we would expect OECD to score very highly on the first principal component [23].

Table 3: Singular Values of A

Singular Values
1.73891276816121
0.70456665802325
0.551531756251022
0.419023783304168

Finally, a look at our singular values shows the reason our relative error is still fairly large. While there is a significant reduction in the information carried by the first component compared with the second component, the second, third, and fourth components all carry similarly large amounts of information in them [4]. This makes this data set relatively hard to compress. Larger data sets containing more human development variables may have lower errors for similar reductions in dimensionality, given that there would be a greater number of development indicators and we expect them to be correlated.

4 Conclusion

In conclusion, we have discussed a few of the applications of SVD — including reducing the storage requirements for images and reducing the number of variables in data sets. Further research would look into finding a data set that would better illustrate the differences between using SVD to do PCA and using the covariance matrix of A to do PCA, would further investigate the applications of SVD, and would look into additional dimensionality reduction techniques.

References

[1] Richard L. Burden and J. Douglas Faires. *Numerical Analysis.* Brooks/Cole Cengage Learning, Boston, MA, 2011.

[2] Robert Collins. Penn state computer vision ii: Principle component analysis. $http://www.cse.psu.edu/\ rtc12/CSE586Spring2010/lectures/pcaLectureShort_6pp.pdf$, 2010.

[3] Frank Dellaert. Georgia tech: Singular value and eigenvalue decompositions. $http://www.cc.gatech.edu/\ dellaert/pub/svd-note.pdf$, 2008.

[4] The Analysis Factor. Three tips for principal component analysis. $http://www.theanalysisfactor.com/tips-principal-component-analysis/$, NA.

[5] Velimir Gayevskiy. Latex templates: Title pages. $http://www.latextemplates.com/template/university-assignment-title-page$, 2016.

[6] L. El Ghaoui. Applications of svd. $https://inst.eecs.berkeley.edu$ $/ ee127a/book/login/l_svd_apps.html$, 2011.

[7] Ricardo Gutierrez-Osuna. Introduction to pattern recognition lecture 5: Dimensionality reduction (pca). $http://courses.cs.tamu.edu/rgutier$ $/cs790_w02/l5.pdf$, NA.

[8] Investopedia. Investopedia: Organization for economic cooperation and development - oecd. $http://www.investopedia.com/terms/o/oecd.asp$, 2016.

[9] Steven Leon. *Linear Algebra with Applications*. Pearson, New York, 2015.

[10] Feifei Li. Svd and pca. $http://www.cs.utah.edu/$ $lifeifei/cis5930/lecture12-a.pdf$, 2008.

[11] Carla D. Martin and Mason A. Porter. The extraordinary svd. $https://arxiv.org/abs/1103.2338$, 2012.

[12] Benjamin Noble and James W. Daniel. *Applied Linear Algebra*. Pearson, 1987.

[13] OECD. Members and partners. $https://www.oecd.org/about/membersandpartners/$, 2016.

[14] University of Pittsburgh. Pca. $https://people.cs.pitt.edu/ milos/courses/cs3750$ $-Fall2007/lectures/PCA.pdf$, Fall 2007.

[15] Martin J. Osborne. using bibtex: a short guide. $https://www.economics.utoronto.ca/osborne/latex/BIBTEX.HTM$, 2015.

[16] Per-Olof Persson. Lecture 5: Gram-schmidt orthogonalization. $https://ocw.mit.edu/$ $courses/mathematics/18-335j-introduction-to-numerical-methods-fall-2010/lecture-notes/MIT18_335JF10_lec10a_hand.pdf$, 2006.

[17] David Poole. *Linear Algebra: A Modern Introduction*. Cengage Learning, CT, USA, 2015.

[18] Victor Powell and Lewis Lehe. Principle component analysis explained visually. $http://setosa.io/$ $ev/principal-component-analysis/$, 2010.

[19] United Nations Development Programme. Human development data (1980-2015). Data from $http://hdr.undp.org/en/data$, 2016.

[20] SHARELaTeX. Algorithms. $https://www.sharelatex.com/learn/algorithms$, NA.

[21] SHARELaTeX. Tables. $https://www.sharelatex.com/learn/Tables$, NA.

[22] Penn State. Alternative: Standardize the variables. $https://onlinecourses.science.psu.edu/$ $stat505/node/54$, NA.

[23] Penn State. Interpretation of the principal components. $https://onlinecourses.science.psu.edu/$ $stat505/node/54$, NA.

[24] Wikipedia. Principle component analysis. $https://en.wikipedia.org/wiki/Principal_component_analysis$, 2016.

[25] Wikipedia. Singular value decomposition. $https://en.wikipedia.org/wiki/Singular_value_decomposition$, 2016.

[26] Meng Zhou. Penn state: Dimensionality reduction applications. $http://citeseerx.$ $ist.psu.edu/viewdoc/download?doi=10.1.1.691.7513\&rep=rep1\&type=pdf$, 2014.