Singular Value Decomposition

Peter N Robinson

July 30, 2019

In this appendix we will present an intuitive explanation of the Singular Value Decomposition (SVD), which is intended to help readers unfamiliar with the SVD follow our presentation of the CONiFER algorithm in chapter **??**. Section 1 offers a brief review of the SVD, and section 3 presents an application of the SVD to dimensionality reduction that is related but in a sense inverse to the CoNIFER algorithm.

1 Singular Value Decomposition

The SVD is one of the most widely used matrix decompositions. Among its many applications are information retrieval, political science, computer dating, pseudoinverse, bioinformatics, and many other areas. For simplicity, we will assume that all matrices in this section contain only real numbers.

The SVD is a matrix factorization that is closely related to the eigenvalue-eigenvector factorization of a symmetric matrix:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{1}$$

where the eigenvalues are in the diagonal matrix $\mathbf{\Lambda}$ and the eigenvector matrix $\mathbf{Q}$ is orthogonal: $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, because we can choose the eigenvectors of a symmetric matrix to be orthonormal. Note that right-multiplying $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ by $\mathbf{Q}$ leads to $\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$, which is just a restatement of the fact that the columns of $\mathbf{Q}$ are the eigenvectors of $\mathbf{A}$, $\mathbf{A}\mathbf{q_i} = \lambda_i\mathbf{q_i}$

Note that the formula $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ can be written as the sum of outer products of the eigenvectors of $\mathbf{A}$ weighted by their corresponding eigenvectors:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \sum_i \lambda_i \mathbf{q}_i^T \mathbf{q}_i \tag{2}$$

The SVD can be regarded generalization of this factorization to arbitrary matrices. It factorizes an arbitrary $m \times n$ matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3}$$

In this equation, $\mathbf{U}$ is an orthogonal $m \times m$ matrix whose columns form an orthonormal basis $\mathbf{U} = \{\mathbf{u_1}, \cdots, \mathbf{u_m}\}$. $\mathbf{V}$ is an $n \times n$ orthogonal matrix whose columns also form an orthonormal basis $\mathbf{V} = \{\mathbf{v_i}, \ldots, \mathbf{v_n}\}$. The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$.

$\mathbf{\Sigma}$ is an $m \times n$ diagonal matrix with elements $\sigma_1, \ldots, \sigma_r$ on the diagonal and whose off-diagonal entries are all zero. These are the *singular values* of $\mathbf{A}$, and r is the rank of $\mathbf{A}$. These r singular values are the square roots of the nonzero eigenvalues of both $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$.

Note that $\mathbf{\Sigma}$ is not necessarily a square matrix. Assuming there are no zero-valued singular values, the diagonal is completed as much as possible given the dimensions of the matrix, e.g.,

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}_{m \times n} \qquad \mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \end{pmatrix}_{m \times n}$$

Just as we can reconstruct a symmetric matrix $A \in \mathbf{R}^{n \times n}$ on the basis of its eigendecomposition

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \tag{4}$$

where the λ_i are the eigenvalues of $\mathbf{A}$ and the $\mathbf{u}_i$ are the corresponding orthonormal eigenvectors, we can reconstruct a matrix $\mathbf{X} \in \mathbf{R}^{m \times n}$ with $\text{rank}\,(\mathbf{X}) = r \leq \min(m, n)$ by the SVD:

$$\mathbf{X} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{5}$$

The columns of $\mathbf{U}$ are called the left singular vectors $\{\mathbf{u}_k\}$, those of $\mathbf{V}$ are the right singular vectors $\{\mathbf{u}_k\}$.

2 Truncated SVD

For the truncated SVD, we do not reconstruct the entire original matrix, but instead only use the k column vectors of $\mathbf{U}$ and the k row vectors of $\mathbf{V}^T$ corresponding to the k largest singular values to reconstruct part of the data in the original matrix. The rest of the matrix is discarded.

$$\mathbf{A} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{u}_i^T \tag{6}$$

3 SVD for data dimensionality reduction

SVD, and the closely related Principle Components Analysis (PCA) can be used for data dimensionality reduction. In many cases, data are measured in a high dimensional space, but actually lie within or near a simpler, lower dimensional manifold. SVD can be used to extract the most important, characteristic dimensions od highest variability from the data, and to discard remaining dimensions to both save space and in some applications "denoise" the data. Consider the example of a straight line in 3 dimensional space that is measrued with minor errors. The data can be generated and ploted in R as follows, and the result is shown in Figure 1.

```
library(plot3D)
t<-seq(0,1,by=0.01)
x<-0+t*1.2*t + runif(101,min=-0.05,max=0.05)
y<-0+t*0.9*t + runif(101,min=-0.05,max=0.05)
z<-0+t*1*t + runif(101,min=-0.05,max=0.05)
scatter3D(x, y, z, colvar = NULL, col = "blue",
          pch = 19, cex = 0.5,bty="g",
          ticktype = "detailed",nticks=5)
```

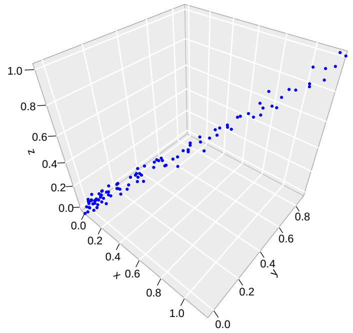

Figure 1: A straight line in three dimensional space, ranging from (0,0,0) to (1,1,1) with certain measurement errors.

We can arrange the 101 observations of points on the line with x, y, and z coordinates into an 101×3 matrix $\mathbf{A}$. We perform an SVD on A with the `svd` command.

```
dat<-data.frame(x=x,y=y,z=z)
A<-as.matrix(dat)
s<-svd(A)
```

The variable s is actually an object whereby `s$u` represents the 101×3 matrix (U), `s$d` is a vector that contains the diagonal elements of the 3×3 matrix $\mathbf{\Sigma}$, and `s$v` represents the 3×3 matrix $\mathbf{U}$. To reconstruct the entire original matrix from $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$, as we have seen, we can sum up the outer products of the components of $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{A} = \mathbf{U\Sigma V}^T = \sum_{i=1}^{3} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{7}$$

If we posit that our data is sufficently well described if we use truncate the SVD to a dimensionality of one (which seems a good assumption, because our data are arranged as a straight line, after all), then we can reconstructthe following truncated matrix.

$$\mathbf{A}_1 = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T \tag{8}$$

We do this in R as follows, and the resulting plot shows that $\mathbf{A}_1$ does indeed capture a large part of the data (Figure 2). In fact, the resulting line is seems to display less deviations from a straight line, i.e., the capture the most important characteristics of the data while reducing noise.

```
A1 <- (s$u[,1]*s$d[1]) %*% t(s$v[,1])
x<-A1[,1]
y<-A1[,2]
z<-A1[,3]
scatter3D(x, y, z, colvar = NULL, col = "blue",
          pch = 19, cex = 0.5,bty="g",
          ticktype = "detailed",nticks=5)
```

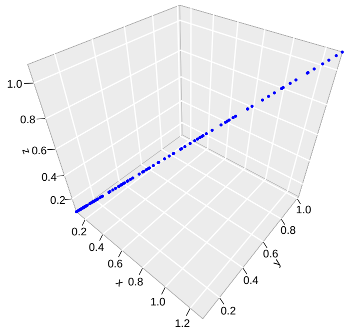

Figure 2: The straight line reconstructed from the truncated SVD.

A classic application of dimnesionality reduction with the SVD is for image compression. This can be easily demonstrated by reading a PGM (Portable

GreyMap) graphics file into R and performing SVD on the corresponding matrix. For demonstration purposes, we chose a picture of the Taj Mahal.[1]

The following command reads the file, performs an SVD, and reconstructs partial matrices using 5, 25, and 50 singular values.

```
taj <- read.pnm('tajmajal.pgm')
M<-matrix(taj@grey,nrow=taj@size[1],ncol=taj@size[2])
s <- svd(M)
u<-s$u
d<-s$d
v<-s$v
i<-5
taj5 <- u[,1:i] %*% diag(d[1:i]) %*% t(v[,1:i])
i<-25
taj25 <- u[,1:i] %*% diag(d[1:i]) %*% t(v[,1:i])
i<-50
taj50 <- u[,1:i] %*% diag(d[1:i]) %*% t(v[,1:i])
```

The results of the calculations can be displayed with the image command (only shown here for hte original matrix), and all four images are displayed in Figure 3.

```
image(M,col=grey.colors(255),xaxt='n',yaxt='n')
```

[1] If you want to try this with an image of your own, you can export gray-scale PGM images with the graphics program Gimp on linux.

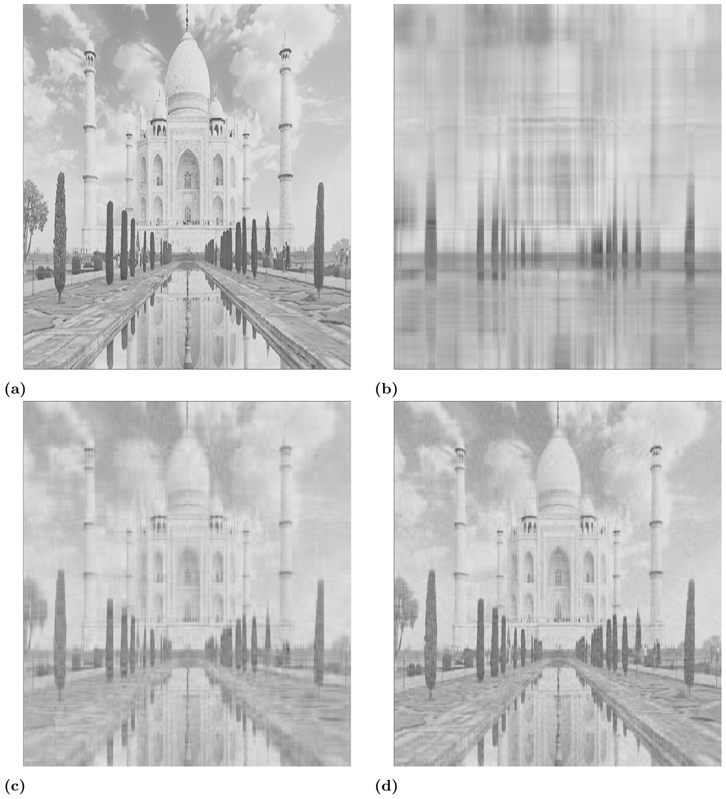

Figure 3: The Taj Mahal. (a) Original image. (b)–(d) Reconstructed images with (b) 5, (c) 25, and (d) 50 of the original 320 dimensions.

4 SVD Analysis for CNV Calling

In chapter **??**, we present an algorithm for characterizing a certain class of structural variant in exome data, CoNIFER (copy number inference from exome reads) [**?**]. One often sees the SVD (and the related Principle COmponents Analysis) used for dimensionality reduction, where only the top few singular vectors (or eigenvectors) are retained under the assumption that they contain the interesting part of the data. For a number of bioinformatics algorithms, the assumption is different: the top singular vectors are removed under the assumption that they contain most of the bias that we would like to get rid of. For readers who would like to learn more about the SVD, we recommend the book by Professor Gilbert Strang of the Massachusetts Insitute of Technology shown below, as well as his free online courses on the subject, which at the time of this writing could be found at the MIT OpenCourseWare website.[2]

References

[1] Gilbert Strang Introduction to Linear Algebra, Fifth Edition. Wellesley-Cambridge Press, 2016.

[2] `https://ocw.mit.edu/index.htm`