Handout 05: Singular Value Decomposition

Today we are going to work with a particular type of matrix factorization called the singular value decomposition. Start by assuming that we have a matrix A with n rows and p columns such that $n \geq p$. The (thin) singular value decomposition, or SVD, is given by the matrix product:

$$A = UDV^t \tag{5.1}$$

With the following dimensions:

$$A \in \mathbb{R}^{n \times p} \tag{5.2}$$
$$U \in \mathbb{R}^{n \times p} \tag{5.3}$$
$$D \in \mathbb{R}^{p \times p} \tag{5.4}$$
$$V \in \mathbb{R}^{p \times p} \tag{5.5}$$

Furthermore, D is a diagonal matrix with non-negative entries along the diagonal ordered from the largest to the smallest value:

$$D = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_p \end{bmatrix}, \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0. \tag{5.6}$$

The values σ_k are called the *singular values* of the matrix A. Also, V is an orthogonal matrix such that (we showed in Handout 03 that this corresponds to a rotation):

$$V^t V = V V^t = I_p. \tag{5.7}$$

The matrix U is not square, so it cannot be completely orthogonal, but its columns are orthogonal to one another so we have:

$$U^t U = I_p. \tag{5.8}$$

The singular value decomposition exists for any matrix, and so we can use it without any assumptions on the matrix we are working with. This has important geometric implications: **any** linear function can be written as a rotation, a fixed scaling of the components, and another rotation.

SVD and the Normal Equations

If we take the SVD of the data matrix X, we have

$$X = UDV^t. \tag{5.9}$$

Plugging this into the ordinary least squares estimator gives:

$$
\begin{aligned}
\beta &= (X^t X)^{-1} X^t y && (5.10)\\
&= (VD^tU^tUDV^t)^{-1} VD^tU^t y && (5.11)\\
&= (VD(U^tU)DV^t)^{-1} VDU^t y && (5.12)\\
&= (VDI_pDV^t)^{-1} VDU^t y && (5.13)\\
&= (VD^2V^t)^{-1} VDU^t y && (5.14)
\end{aligned}
$$

By taking the fact that a diagonal matrix is its own transpose and using that U^tU is equal to the identity. Note that D^2 is just a matrix with the squared singular values along the diagonal.

Now, notice that the inverse of V is V^t, and vice-versa. Further, the inverse of D^2 is equal to a diagonal matrix with the inverse of the squared singular values along the diagonal (this exists if we assume that $\sigma_1 > 0$). Therefore:

$$
(VD^2V^t)^{-1} = (V^t)^{-1} D^{-2} V^{-1} = VD^{-2}V^t \qquad (5.15)
$$

And we can further simplify the equation for the ordinary least squares estimator:

$$
\begin{aligned}
\beta &= (VD^2V^t)^{-1} VDU^t y && (5.16)\\
&= VD^{-2}V^tVDU^t y && (5.17)\\
&= VD^{-2}DU^t y && (5.18)\\
&= VD^{-1}U^t y. && (5.19)
\end{aligned}
$$

This gives us a compact way to write the ordinary least squares estimator. It is also far more numerically stable to use this formula to compute the estimate β from a dataset. Most importantly, it will yield a lot of intuition for what makes some estimation tasks hard and motivate how we can (partially) address the most challenging regression problems.

SVD in R

In R, you can create the singular value decomposition of a matrix using the function `svd`. To see this, let's construct some simulated data:

```
set.seed(1)
n <- 1e4; p <- 4
X <- matrix(rnorm(n*p), ncol = p)
b <- c(1,2,3,4)
epsilon <- rnorm(n)
y <- X %*% b + epsilon
```

Now, we take the singular value decomposition of the matrix. I will also explicitly extract out and save the matrices U and V as well as the singular values $sigma$:

```
svd_output <- svd(X)
U <- svd_output[["u"]]
V <- svd_output[["v"]]
sigma <- svd_output[["d"]]
```

Now, lets compute the ordinary least square matrix with this data:

```
beta <- V %*% diag(1 / sigma) %*% t(U) %*% y
beta
```

```
          [,1]
[1,] 0.9870134
[2,] 1.9876739
[3,] 3.0045489
[4,] 4.0102080
```

We can verify that this is equivalent to our old form of the estimator by:

```
solve(t(X) %*% X) %*% t(X) %*% y
```

```
          [,1]
[1,] 0.9870134
[2,] 1.9876739
[3,] 3.0045489
[4,] 4.0102080
```

Notice that both are close to the value of `b` in the simulation.

LAB QUESTIONS

1. I showed you how to get a nice equation for β in the ordinary least squares equation. Using the SVD of X, compute a compact formula for the values $\widehat{y} = X\beta$.

2. We glossed over the case where one or more of the singular values is equal to zero. In this question I will show you why we cannot deal with this case in the construction of β. Let V_p denote the last column of V (these columns are called the *right singular vectors*). Argue that:

$$V^t V_p = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ 1 \end{bmatrix} \tag{5.20}$$

Now, assume that $\sigma_p = 0$. Show that (Hint: expand X with the SVD):

$$XV_p = 0. \tag{5.21}$$

Assume that we have a potential candidate β for the regression vector. Show that the fitted values $\widehat{y}$:

$$\widehat{y} = X\beta = X(\beta + a \cdot V_p), \quad \forall a \in \mathbb{R}. \tag{5.22}$$

Explain why this implies that we cannot uniquely determine a value for β according the minimization of the loss function on the training data when $\sigma_1 = 0$.

3. Let X be a matrix with SVD equal to UDV^t and w be a 3-dimensional vector with Euclidean norm equal to one:

$$||w||_2^2 = w^t w = \sum_k w_k^2 = w_1 + w_2 + w_3 = 1. \tag{5.23}$$

It is generally true that we can write the vector w as a weighted sum of the columns of V:

$$w = \sum_k a_k \cdot V_k = a_1 V_1 + a_2 V_2 + a_3 V_3. \tag{5.24}$$

Argue that:

$$\sum_k a_k^2 = a_1^2 + a_2^2 + a_3^2 = 1. \tag{5.25}$$

Then, show that (the middle step is a hint more than anything else):

$$||Xw||_2^2 = ||DV^t w||_2^2 = a_1 \cdot \sigma_1 + a_2 \cdot \sigma_2 + a_3 \cdot \sigma_3. \tag{5.26}$$

Note that all of these results are true in an arbitrary number of dimensions; I just set $p = 3$ so that you could more easily make use of picture arguments. What do these properties this tell you geometrically about the matrix X in terms of the singular values and right-singular vectors?