Singular Vector Decomposition

compiled by Alvin Wan from Professor Benjamin Recht's lecture

1 Introduction to Unsupervised Learning

Note that today, we will consider X to be $d \times n$, contrary to our usual convention for X. We ask ourselves two questions: Can we compress dimension? Can we compress examples?

First, we have a number of ways to achieve **dimension reduction** (reducing d).

- run time
- storage
- generalization
- interpretability

Second, we have a number of ways to achieve **clustering** (reducing n).

- faster run time
- understanding archetypes
- outlier removal
- segmentation

Most unsupervised learning appeals to matrix factorization. We will factor X $(d \times n)$ into AB, where A is $d \times r$ and B is $r \times n$. Before we explain how this is done, let us consider why this is important. The structure of A and B may give us insight into the data.

We can write X has a linear combination of a_r, the examples. Specifically,

$$X = \begin{bmatrix} x_1 & x_2 \cdots x_n \end{bmatrix} \begin{bmatrix} P_1, P_2 \cdots P_n \end{bmatrix}$$

where $P_i = \begin{bmatrix} a_1, a_2 \cdots a_r \end{bmatrix}^T$, $\sum a_i = 1$ and $a_i \geq 0$. If we could find this factorization, we would have an archetype.

2 (Economy-Sized) Singular Value Decoomposition

To accomplish matrix factorization, we most commonly consider SVD. Every X in $\mathbb{R}^{d\times n}$, where $n > d$, admits a factorization:

$$X = USV^T$$

where U is $d \times d$, S is $d \times d$, and V is $n \times d$. There are a few properties of this decomposition to take note.

1. We also have that $U^TU = I_d$, $V^TV = I_d$, telling us that U, V contain orthogonal vectors.
2. $S = \text{DIAG}(\sigma_i)$, where singular values are ordered along the diagonal from greatest to least, $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_d \geq 0$.

We can rewrite $U = \begin{bmatrix} u_1, u_2 \cdots u_d \end{bmatrix}, V = \begin{bmatrix} v_1, v_2, \cdots, v_d \end{bmatrix}$ and get the following, equivalent, representation for X.

$$X = \sum_{i=1}^{d} \sigma_i u_i v_i^T$$

Now that we've rewritten X, what does it mean to multiply X by some vector z? Like all factorizations, we transform the vector z into a new basis, scale it, and then transform it back into the standard basis. Consider the following.

$$Xz = \sum_{i=1}^{d} \sigma_i u_i (v_i^T z)$$

Consider the vector z in the standard basis. X, in a sense, transforms z and the unit circle its drawn from into another vector z' drawn from an ellipsoid. This allows us to reduce dimensions, because it effectively tells us which directions do not matter.

3 Behavior

We can analyze the behavior of Xz for some vector z using the decomposition. First, consider the case where z is some vector drawn from V, v_i.

$$Xv_i = \sum_{j=1}^{d} \sigma_j u_j (v_j^T v_i) = \sigma_i u_i$$

Let us take the above result and apply the fact that $V^T V = I_d$.

$$X^T X v_i = X^T \sigma_i u_i = \sigma_i X^T u_i = \sigma_i^2 v_i$$

Every singular value of X is the square root of an eigenvalue of $X^T X$ or XX^T. Likewise, each singular vector of X is the eigenvector of $X^T X$ or XX^T.

$$X^T X v_i = \sigma_i^2 v_i$$
$$XX^T u_i = \sigma_i^2 u_i$$

This demonstrates existence, but this is not how we compute these values in practice. This is because squaring the matrix X increases the condition number and decreases accuracy. Note that in practice, we use SVD instead of diagonalization, for purposes of stability.

4 Computation

$$XX^T = (USV^T)(VSU^T) = US^2U^T$$

In the second step, we apply the definition of $V^T V = I_d$. Likewise, we can obtain

$$X^T X = VS^2V^T$$

4.1 Positive, Semi-Definite

A is an positive, semi-definite matrix. This immediately tells us that it has an eigenvalue decomposition and that all of its eigenvalues are non-negative.

$$A = W \Lambda W^T$$

where $WW^T = I$, $\Lambda = \text{DIAG}(\lambda_i)$, and $\lambda_i \geq 0$. How can find W? We already have. This is identical to SVD, when A is positive, semi-definite.

4.2 Symmetric

B is a symmetric matrix.

$$B = W_2 \Lambda_2 W_2^T$$

where $W_2^T W_2 = I$ and the first k diagonal entries of Λ_2 are non-negative but the last $d-k$ are negative, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \cdots \lambda_k \geq 0 > \lambda_{k+1} \geq \cdots \geq \lambda_d$. Consider γ, a diagonal matrix with k leading 1s and $d-k$ -1s. We know that $\Lambda_2 \gamma$ is now positive semi-definite since all negative entries are multiplied by -1. We know that $W_2 \gamma$ is orthogonal, because $(W_2\gamma)(\gamma^T W_2^T) = I$, where $\gamma\gamma^T = I_d$ and per our assumptions, $W_2 W_2^T = I_d$. So, we have a decomposition.

$$B = (W_2\gamma)\Lambda(W_2\gamma)^T$$

5 Eigenvalues v. Singular Values

Consider $C = \begin{bmatrix} 1 & 10^{12} \\ 0 & 1 \end{bmatrix}$. The eigenvalues are 1 and the singular values are $10^{12}, 10^{-12}$. To compute singular values, we can use `scipy.linalg.svd(`$C^T C$`)`. How are they correlated? For arbitrary square matrices, keep in mind that the singular values and eigenvalues have no correlation.

The maximum value of $\|Cz\|$ subject to the constraint that $\|z\| = 1$, is σ_i. More formally, $\max_{\|z\|=1} \|Cz\| = \sigma_i$. Here is why.

$$\begin{aligned} \|Cz\|_2 &= z^T V_C S_C^2 V_C^T z \\ &= \sum_{i=1}^{d} \sigma_i^2 (v_i^T z)^2 \end{aligned}$$

v forms a basis for the orthogonal complement of the null space. To maximize this quantity then, we want $z = v_1$ so that we yield the largest value, which is the largest singular value.

$\sigma_{r+1} = 0 \implies \sigma_{r+2}, \sigma_{r+3} \cdots \sigma_d = 0$ so $rank(X) \leq r$ and X is rank-deficient. We can write $X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$. v_i are a basis for all null(X). We also have that u_i are a basis for range(X).

$$\hat{X} = \begin{bmatrix} u_1, \cdots u_r \end{bmatrix}^T$$

What information are we throwing away? Let us rewrite w.

$$w = (\sum_{i=1}^{r} \alpha_i u_i) + W_\perp$$

where $W_\perp^T u_i = 0$, $i = 1, \ldots d$.