Stat 315c: Transposable Data
Singular Value Decomposition (review)

Art B. Owen

Stanford Statistics
The SVD is a core technique in many matrix data analyses. It is used to do least squares computations in a most reliable way. It is also useful in theoretical analysis of matrices. We’ll use it at first to understand some classical methods. Then we revisit it as an 'end in itself'
Definition

SVD

The matrix $A_{m \times n}$ can be written $A = U \Sigma V'$ where

- $U_{m \times m}$ is orthogonal
- $V_{n \times n}$ is orthogonal, and
- $\Sigma_{m \times n}$ is diagonal

with singular values $\Sigma_{jj} \equiv \sigma_j$ where

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_{\min(m,n)} = 0$$

Cols of $U$ (resp $V$) are left (right) singular vectors

A matrix is what a matrix does (F. Gump)

$$Ax = U \Sigma V' x$$

Rotate then stretch then rotate
Properties

Skinny SVD

\[ A = \tilde{U}\tilde{\Sigma}\tilde{V}' \]
\[ \tilde{U} = \text{First } r \text{ columns of } U \]
\[ \tilde{\Sigma} = \text{Upper } r \times r \text{ submatrix of } \Sigma \]

Outer product representation

\[ A = \sum_{i=1}^{r} \sigma_i u_i v_i' \] so \( A \) has rank \( r \)
\[ u_i = \text{Column } i \text{ of } U \]
\[ v_j = \text{Column } j \text{ of } V \]
Reduced rank approximations

Best rank $k \leq r$ approx to $A$

$$\hat{A}_k = \sum_{i=1}^{k} \sigma_i u_i v_i'$$

Minimizes Frobenius norm

$$\| A - \hat{A}_k \|_F^2$$

where

$$\|X\|_F^2 = \sum_{i} \sum_{j} X_{ij}^2$$

$$\|\hat{A}_k\|_F^2 = \sum_{i=1}^{k} \sigma_i^2$$

$$\| A - \hat{A}_k \|_F^2 = \sum_{i=k+1}^{\min(n,m)} \sigma_i^2$$
**Norms and conditions**

**Matrix norms**

\[ \|A\|_2 \equiv \max_{\|x\|=1} \|Ax\| = \sigma_1 \]

\[ \|A\|_F = \sum_{i=1}^{r} \sigma_i^2 \]

**Condition: numerical difficulties bounded in terms of \( \kappa \)**

\[ \min_{\|x\|=1} \|Ax\| = \sigma_{\min(n,p)} \]

\[ \kappa(A) = \frac{\max_{\|x\|=1} \|Ax\|}{\min_{\|x\|=1} \|Ax\|} = \frac{\sigma_1}{\sigma_{\min(m,n)}} \]
Sums of squares and eigendecomposition

Symmetric matrix $A$

$$A = P \Lambda P', \quad \text{Eigen vectors in cols of } P, \ \Lambda = \text{diag(e vals)}$$

Matrix squaring

$$X = U \Sigma V'$$
$$X'X = V \Sigma' U' U \Sigma V'$$
$$= V \Sigma' \Sigma V' \quad \text{so} \quad \Lambda(X'X) = \Sigma' \Sigma \quad P(X'X) = V$$
$$XX' = U \Sigma \Sigma' U' \quad \text{so} \quad \Lambda(XX') = \Sigma \Sigma' \quad P(XX') = U$$

$X_{n \times p}$ $n \geq p$

$$\Sigma' \Sigma = \text{diag}(\sigma^2_1, \ldots, \sigma^2_p) \quad \Sigma \Sigma' = \text{diag}(\sigma^2_1, \ldots, \sigma^2_p, 0, \ldots, 0)$$

When $X = U \Sigma V'$

$$X' = V \Sigma' U' \quad X \text{ and } X' \text{ have same (nonzero) singular values}$$
Computing the SVD

Cost (WLOG $n \leq m$)

- $O(m^2n + mn^2 + n^3)$ for $U$, $\Sigma$, $V$
- $O(mn^2 + n^3)$ for $\tilde{U}$, $\tilde{\Sigma}$, $\tilde{V}$

Further savings if only $\Sigma$ needed
Big savings possible if only low rank approx needed

Computing a truncated SVD $k \ll \min(m, n)$

- Very roughly $O(mnk)$
- Exact cost seems unknown
- Can study empirically
- In R the cost is the same for all $k$
- matlab does better
Applications

Fitting regressions

\[ \hat{Y} = HY \text{ where } H = X(X'X)^{-1}X' = \tilde{U}\tilde{U}' \]

Computing principal components

Apply to variance or correlation matrix

Correspondence Analysis

We’ll see

Latent semantic indexing

Dimension reduction in information retrieval
Further references

Golub and van Loan “Matrix Computations”