Linear Algebra and Matrix Methods
Singular Value Decomposition (SVD), Principal Component Analysis (PCA)

Various decompositions so far

We have seen many ways to factor a “nicely behaved” matrix A:

- invertible: $A = LDU$
- symmetric: $A = LDL^t$
- diagonalizable: $A = S\Lambda S^{-1}$
- similar: $A = MJM^{-1}$, J Jordan form, M invertible

(We will discuss Jordan form in detail in the next section.)

Recall the Four Fundamental Subspaces

Recall the four fundamental subspaces generated by an $m \times n$ matrix A:

- **row space**

$$C(A^t) \subseteq \mathbb{R}^n,\ dim(C(A^t)) = r = rank(A)$$

- **null space**

$$N(A) \subseteq \mathbb{R}^n,\ \dim(N(A)) = n - r,\ N(A) \perp C(A^t)$$

- **column space**

$$C(A) \subseteq \mathbb{R}^m,\ \dim(C(A)) = r = rank(A)$$

- **left null space**

$$N(A^t) \subseteq \mathbb{R}^m,\ \dim(N(A^t)) = m - r,\ N(A^t) \perp C(A).$$

SVD: Σ holds the "singular values" of A

We will now decompose A into the **singular value decomposition**

$$A = U\Sigma V^t, \quad \text{or } AV = U\Sigma,$$

where Σ is diagonal starting in the upper-left corner, and its entries σ_i are called the **singular values** of A, written in descending order:

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > 0.$$

These σ_i are square roots of shared eigenvalues of A^tA and AA^t.

Σ only has these r nonzero values; it has zeroes elsewhere.

SVD generalizes the other decompositions

$$A = U\Sigma V^t, \quad \text{or } AV = U\Sigma,$$

Writing Σ in this way, U and V have the following column forms:

- $U = (u_1\, u_2\, \cdots\, u_m)$ is $m \times m$ and orthogonal ($U^t = U^{-1}$).

  $\{u_1, ..., u_r\}$ are an orthonormal basis for $C(A)$;

  $\{u_{r+1}, ..., u_m\}$ are an orthonormal basis for $N(A^t)$.

- $V = (v_1\, v_2\, \cdots\, v_n)$ is $n \times n$ and orthogonal ($V^t = V^{-1}$).

  $\{v_1, ..., v_r\}$ are an orthonormal basis for $C(A^t)$;

  $\{v_{r+1}, ..., v_n\}$ are an orthonormal basis for $N(A)$.

SVD: derivation

Why does this work?

AA^t and A^tA are symmetric and positive semidefinite, so they can both be diagonalized with nonnegative real eigenvalues.

Thus, we have orthogonal eigenvector matrices U and V and diagonal eigenvalue matrices Λ_{AA^t}, Λ_{A^tA} such that

$$AA^t = U\Lambda_{AA^t}U^t$$

$$A^tA = V\Lambda_{A^tA}V^t.$$

List the eigenvalues in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$.

SVD: derivation

We can see that Λ_{AA^t} are Λ_{A^tA} are deeply related.

Let $\lambda_i > 0$ be an eigenvalue from AA^t, and $\vec{y}$ its eigenvector. Then

$$AA^t\vec{y} = \lambda_i\vec{y} \implies A(A^t\vec{y}) = \lambda_i\vec{y} \implies A^tA(A^t\vec{y}) = \lambda_i(A^t\vec{y}),$$

implying that $\vec{x} = A^t\vec{y}$ is an eigenvector of A^tA with eigenvalue λ_i.

Thus, $\vec{y}$ is a column of V and $\vec{x}$ is a column of U, with the same eigenvalue λ_i.

Repeat the argument with A^tA to see that $\vec{y} = A\vec{x}$.

SVD: derivation

Thus, AA^t and A^tA share all nonzero eigenvalues. If A is $m \times n$,

- Λ_{AA^t} is $m \times m$,
- Λ_{A^tA} is $n \times n$.

Thus, the $m \times n$ "pseudodiagonal" matrix $\Sigma = (\sigma_{ij})$ such that

- $\sigma_{ii} = \sqrt{\lambda_i}$ for $i = 1, 2, ..., r = rank(A)$
- $\sigma_{ij} = 0$ elsewhere

allows us to rewrite the diagonalizations of AA^t and A^tA as

$$AA^t = U\Sigma\Sigma^t U^t$$
$$A^tA = V\Sigma^t\Sigma V^t.$$

SVD: derivation

Inserting clever uses of the identity matrix via the orthogonal matrices U and V displays

$$AA^t = U\Sigma\Sigma^t U^t = (U\Sigma V^t)(V\Sigma^t U^t) = (U\Sigma V^t)(U\Sigma V^t)^t$$

$$A^t A = V\Sigma^t\Sigma V^t = (V\Sigma^t U^t)(U\Sigma V^t) = (U\Sigma V^t)^t(U\Sigma V^t).$$

These both display the SVD

$$A = U\Sigma V^t.$$

SVD eigenpairs

We know

$$\lambda_1 = \sigma_1^2 \geq \cdots \geq \lambda_r = \sigma_r^2 > 0$$

are the shared eigenvalues of AA^t and A^tA. Their eigenvectors are:

- AA^t: $\{u_1, ..., u_r\}$
- A^tA: $\{v_1, ..., v_r\}$.

We do not need eigenvectors associated to zero eigenvalues.

Example: SVD computation

To compute the SVD of the matrix

$$A = \begin{pmatrix} -5 & 2 & 0 \\ 1 & -1 & -4 \\ -3 & 9 & 6 \\ 18 & 0 & -12 \end{pmatrix},$$

we first compute AA^t and A^tA:

$$AA^t = \begin{pmatrix} 29 & -7 & 33 & -90 \\ -7 & 18 & 12 & -30 \\ 33 & 12 & 126 & -126 \\ -90 & -30 & -126 & 468 \end{pmatrix}, \quad A^tA = \begin{pmatrix} 359 & -38 & -230 \\ -38 & 86 & 50 \\ -230 & 50 & 196 \end{pmatrix}.$$

Example: SVD computation

AA^t and A^tA are symmetric and positive semidefinite; they have nonnegative real eigenvalues.

We order them from greatest to least.

matrix	eigenvalues
AA^t	538.330, 86.885, 15.7853, 0
A^tA	538.330, 86.885, 15.7853

Clearly, the first three eigenvalues are shared.

Example: SVD computation

We know the singular values of A: the square roots of the shared eigenvalues of AA^t and A^tA.

$$\begin{aligned} \sigma_1 &= \sqrt{538.330} & \geq \sigma_2 &= \sqrt{86.885} & \geq \sigma_3 &= \sqrt{15.7853} \\ &= 23.202 & &= 9.321 & &= 3.973. \end{aligned}$$

Thus, the singular value matrix Σ is

$$\Sigma = \begin{pmatrix} 23.202 & 0 & 0 \\ 0 & 9.321 & 0 \\ 0 & 0 & 3.973 \\ 0 & 0 & 0 \end{pmatrix}.$$

Example: SVD computation

With the eigenvectors normalized, we can build U and V:

$$U = \begin{pmatrix} -0.1849145 & -0.02048505 & -0.81776325 & 0.54465609 \\ 0.14084102 & 0.16139879 & -0.56607528 & -0.79603582 \\ -0.30891704 & -0.92656823 & -0.0464587 & -0.20948311 \\ 0.92224763 & -0.33911962 & -0.09307863 & 0.16060372 \end{pmatrix}$$

$$V^t = \begin{pmatrix} 0.80133938 & -0.14183833 & -0.58115151 \\ -0.32835045 & -0.91634888 & -0.2291085 \\ 0.50004117 & -0.37441503 & 0.78087913 \end{pmatrix}.$$

Note that, depending on how you compute, some eigenvectors may need to be negated to make the decomposition work.[*]

[*] In Python 3, `numpy.linalg.svd(A)` will do it all.

SVD as a sum of outer products

As the $\sigma_i > 0$ are scalars,

$$A = U\Sigma V^t = \sum_{i=1}^{r} \sigma_i u_i v_i^t$$

is a sum of **outer products**: $m \times n$ rank-one matrices[†] $u_i v_i^t$.

The relative sizes of the σ_i determine the weight that each contributes to the sum.

[†] u_i, a column, is $m \times 1$; v_i^t, a row, is $1 \times n$. Their product is $m \times n$.

Pseudoinverses

One big problem in linear algebra is that non-square matrices (and some square matrices) are non-invertible.

Recall the other word we used for a non-invertible matrix: **singular**.

We can use the SVD to generalize the notion of an "inverse matrix" with the **pseudoinverse**.

Pseudoinverses

If A has an inverse A^{-1}, then the SVD of A yields:

$$\begin{aligned} A = U\Sigma V^t \implies \quad & AV = U\Sigma \\ \implies \quad & Av_j = \sigma_j u_j,\ j = 1, 2, ..., r \\ \implies \quad & v_j = \sigma_j A^{-1} u_j \\ \implies \quad & A^{-1} u_j = \sigma_j^{-1} v_j. \end{aligned}$$

The psuedoinverse of A, denoted A^+, is the $n \times m$ matrix satisfying

$$A^+ = V\Sigma^+ U^t,$$

where Σ^+ is the rectangular diagonal matrix with the shape of Σ^t, whose nonzero entries are σ_j^{-1}.

Pseudoinverse = Inverse if A invertible

Thus, we have

$$Av_j = \begin{cases} \sigma_j u_j & j = 1, 2, ..., r \\ 0 & j = r+1, ..., n, \end{cases}$$

$$A^+ u_j = \begin{cases} \sigma_j^{-1} v_j & j = 1, 2, ..., r \\ 0 & j = r+1, ..., m. \end{cases}$$

If A^{-1} exists, it should be clear that $A^+ = A^{-1}$.

We can use A^+ to project and solve in the following ways:

- AA^+ projects onto $C(A)$: $\quad AA^+ = A(A^tA)^{-1}A^t$.
- A^+A projects onto $C(A^t)$: $\quad A^+A = A^t(AA^t)^{-1}A$.
- The least squares solution to $A\vec{x} = b$ is $\hat{x} = A^+b$.

Example: Pseudoinverse

Using A from the previous example,

$$A = \begin{pmatrix} -5 & 2 & 0 \\ 1 & -1 & -4 \\ -3 & 9 & 6 \\ 18 & 0 & -12 \end{pmatrix},$$

we easily compute the 3×4 matrix Σ^+:

$$\Sigma^+ = \begin{pmatrix} \frac{1}{23.202} & 0 & 0 & 0 \\ 0 & \frac{1}{9.321} & 0 & 0 \\ 0 & 0 & \frac{1}{3.973} & 0 \end{pmatrix} = \begin{pmatrix} 0.043 & 0 & 0 & 0 \\ 0 & 0.107 & 0 & 0 \\ 0 & 0 & 0.252 & 0 \end{pmatrix}.$$

Example: Pseudoinverse

Transposing U and V^t given earlier, we get $A^+ = V\Sigma^+U^t$:

$$A^+ = \begin{pmatrix} -0.10858647 & -0.07206592 & 0.016123 & 0.03208348 \\ 0.08020869 & 0.03661807 & 0.09735563 & 0.03647179 \\ -0.15559023 & -0.11875274 & 0.02138086 & -0.03305866 \end{pmatrix}$$

Note that the 3×3 product

$$A^+A = I,$$

but the 4×4 product

$$AA^+ \neq I.$$

Principal Component Analysis: Dataset as Matrix

The SVD can be used to process a large quantity of sampled data to determine the correlations between the different values collected.

Consider a dataset contain n samples, each with m measurements.

We expect that $m \ll n$.

Denote measurement i in sample j by d_{ij}.

Call the matrix of all these measurements $D = (d_{ij})$.

Principal Component Analysis: Centering the Data

Compute the **sample mean vector** $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_n \end{pmatrix}$:

μ_i is the sample mean of measurement i across all n samples.

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} d_{ij}.$$

The dataset will be centered (made mean zero): set

$$a_{ij} = d_{ij} - \mu_i.$$

Then the matrix $A = (a_{ij})$ is the centered version of the data.

Principal Component Analysis: Sample Variance

We can use the SVD of A to compress the data while keeping "most" of the "variance" of the data.

Using the centered $m \times n$ data matrix A, compute the $m \times m$ **sample covariance matrix** across the samples[‡]:

$$S = \frac{1}{n-1} A A^t.$$

The point of this reduction is that, if $m \ll n$, we can represent "most" of the data collected by using fewer than m eigenpairs.

[‡] Variance is a measure of the *spread* of data or a probability distribution, and *not* of central tendency; as such, variance does not change if the mean μ changes. Also, the $n-1$ instead of n makes this covariance matrix *unbiased*.

Principal Component Analysis: Covariance Eigenproblem

Our sample covariance matrix

$$S = \frac{1}{n-1} AA^t$$

has $rank(A) = r \leq m \ll n$ positive eigenvalues

$$\lambda_1 = \sigma_1^2 \geq \lambda_2 = \sigma_2^2 \geq \cdots \lambda_r = \sigma_r^2$$

with associated orthonormal eigenvectors

$$u_1, u_2, ..., u_r$$

coming from the SVD of A.

Principal Component Analysis: Covariance Eigenproblem

To understand these calculations, we state the diagonalization of S:

$$S = X \Lambda X^{-1}.$$

The eigenvalues of S, in Λ, are the eigenvalues of $\frac{1}{n-1} A A^t$.

Thus, if $X = U$ is orthogonal, then the shared eigenvalues of A and A^t are of the form

$$\sqrt{(n-1)\lambda_i} = \sigma_i \sqrt{n-1}.$$

We also compute the eigenvectors of $\frac{1}{n-1} A^t A$ to build V. Then, the SVD of A can be written

$$A = U(\sqrt{n-1}\,\Sigma) V^t.$$

Principal Component Analysis: Variance Percentages

Compute the **total variance**[§] of the dataset A:

$$T = \sum_{i=1}^{r} \lambda_i = \sum_{i=1}^{r} \sigma_i^2 .$$

Then the eigenpair (λ_i, u_i) contributes a fraction of the total variance: precisely

$$\frac{\lambda_i}{T} = \frac{\sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2}$$

of it.

[§] As $\lambda_i = \sigma_i^2$ is considered a variance, σ_i is a standard deviation.
We track the overall spread of the data set via variance, but others may use standard deviation.

Principal Component Analysis: Projecting via the PCA

Thus, ordering the eigenvalues from greatest to least gives the **principal components** of the eigenproblem:

The larger the eigenvalue, the more weight the eigenvector has in determining the overall "direction" of the sampled data.

By recoordinatizing the data in terms of the principal components, we can compress the data into a smaller-dimensional space while retaining most[¶] of the information present.

Note: some, possibly all, intuition of the data will be lost.

[¶] "Most" is subject to threshold whim; often, 95% is used as a threshold.

Principal Component Analysis: Projecting via the PCA

For any matrix M, let M_k denote the matrix of M's first k columns.

If we want to retain, say, 95% of the variance of the original data, figure out how many eigenvalues (starting from the largest) add up to 0.95 T, and project onto those eigenvectors.

For example, if the first k eigenvalues break the 95% threshold,

$$A \approx U(\sqrt{n-1}\,\Sigma_k)V^t,$$

where the first $k$columns are nonzero.

If we select $k = 2$ or $k = 3$, the projection AV_k can be graphed to visualize how these points may cluster, if classification of the data is intended.

Example: Computing the PCA

Here is an example of real-world data undergoing PCA to get a reduction of dimension.

The following table represents the $m = 5$ aggregated grades[‖] of a class of $n = 31$ students:

Student	0	1	2	…	28	29	30
HW	0.933	1.000	0.978	…	0.889	1.000	1.000
Quiz	0.805	0.985	0.999	…	0.878	0.983	0.962
Exam1	0.590	1.040	0.690	…	0.750	0.990	1.030
Exam2	0.741	0.800	0.918	…	0.871	0.953	0.800
FinalExam	0.547	0.853	0.780	…	0.680	0.867	0.733

Can we "compress" this data via SVD?

[‖] percentages written as decimals

Example: Computing the PCA

First, we subtract off the mean vector values

HW	0.840
Quiz	0.876
Exam1	0.812
Exam2	0.858
FinalExam	0.735

per row and call the resulting 5×31 matrix of centered data A.

Example: Computing the PCA

We then compute the 5×5 covariance matrix $S = \frac{1}{30}AA^t$,

and the 31×31 matrix $\frac{1}{30}A^tA$.

Since A is 5×31, we know that the SVD of A, and thus the diagonalization of S, will have at most 5 nonzero eigenvalues.

Example: Computing the PCA

S has the nonzero eigenvalues, in descending order,

$\lambda_1 = 0.210 > \lambda_2 = 0.028 > \lambda_3 = 0.025 > \lambda_4 = 0.012 > \lambda_5 = 0.004$

with associated eigenvectors $\vec{u}_1$, ..., $\vec{u}_5$.

The total variance of the data is

$$T = \sum_{i=1}^{5} \lambda_i = 0.279.$$

Example: Computing the PCA

These eigenvalues and eigenvectors can be used to reconstruct A with varying levels of resolution.

The 85% threshold is broken at $k = 2$, since

$$\frac{0.21 + 0.028}{0.279} = 0.85068.$$

If we are OK with 85% of the variance being represented in the PCA, then we can use the approximation

$$A \approx U(\sqrt{30}\,\Sigma_2)V^t.$$

Remember that only the first 2 columns of U and V actually matter in this computation, since Σ is diagonal.

Example: Visualizing the PCA

We can visualize the 31 points of data, at what might be considered "85% of variance represented", in the compressed 2-dimensional space, by using the projection

$$AV_2,$$

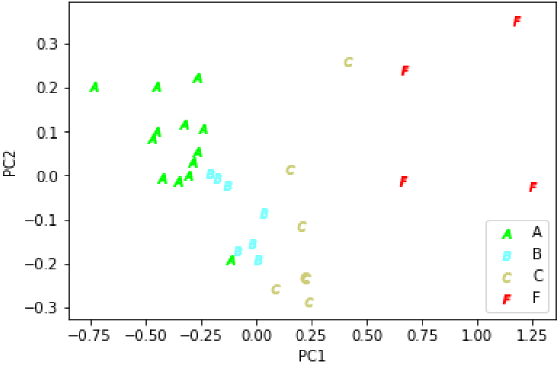
where V_k is the first k columns of V.

We plot the resulting 2×31 matrix as points on the plane,

- with axes labeled "PC1" and "PC2",
- and points (PC1, PC2) colored by final letter grade**, a categorical measurement left out of the initial PCA.

**Although final letter grade is computed numerically, we are using the letter as a categorical variable for classification purposes in this illustration.

Example: Visualizing the PCA



It is clear how we can classify the different final grades via clusters on this graph, even though most of the information about the 5 given grades has been compressed down to 2 dimensions.

Engineering Issue: Standardize the Data

In some cases the different parameters present in a single sample are wildly different[††], and it makes more sense to consider a standardized view of the samples: instead of

$$a_{ij} = d_{ij} - \mu_i,$$

we use

$$a_{ij} = \frac{d_{ij} - \mu_i}{s_i},$$

where, of all samples of the parameter i,

$$\mu_i = \text{sample mean and } s_i \text{ the sample standard deviation.}$$

Note that in this case the matrix S is a **correlation matrix**.

[††] Imagine covariance between height in feet vs weight in pounds. A change of 1 in each unit means drastically different things.