Singular Value Decomposition

and Mahalanobis Distance

A Complete Guide: Geometry, Algebra, and Computation

Abstract

This document explains Singular Value Decomposition (SVD) from first principles, builds up the geometric intuition step by step with a fully worked numeric example, and then shows precisely how SVD gives rise to the Mahalanobis distance. The connection is this: SVD decomposes a covariance matrix into a rotation and a scaling; Mahalanobis distance is exactly Euclidean distance after you undo both the rotation and the scaling. Every step includes both symbolic derivation and concrete numbers so you can verify each calculation by hand.

Contents

1 What SVD Is

1.1 The Core Idea

Every matrix $\mathbf{A}$, regardless of shape or content, does one thing geometrically: it transforms vectors. SVD tells you that *every* such transformation, no matter how complicated, is secretly just three simple operations performed in sequence:

$$\textbf{rotate} \longrightarrow \textbf{scale} \longrightarrow \textbf{rotate}$$

SVD makes this explicit by factoring $\mathbf{A}$ into three matrices that perform exactly these three operations:

Singular Value Decomposition

Any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be written as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

where:

- $\mathbf{U} \in \mathbb{R}^{m \times m}$: orthogonal matrix of **left singular vectors** (output directions)
- $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$: diagonal matrix of **singular values** $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$
- $\mathbf{V} \in \mathbb{R}^{n \times n}$: orthogonal matrix of **right singular vectors** (input directions)

"Orthogonal" means $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$ — transposing gives the inverse, so these matrices represent pure rotations (and possibly reflections).

1.2 What Each Factor Does Geometrically

When you apply $\mathbf{A}$ to a vector $\mathbf{x}$, the three factors act in sequence as $\mathbf{A}\mathbf{x} = \mathbf{U}(\mathbf{\Sigma}(\mathbf{V}^\top\mathbf{x}))$:

1. $\mathbf{V}^\top\mathbf{x}$: **rotate** the input vector into the coordinate system defined by the right singular vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots$ These are the "natural input directions" of the transformation.
2. $\mathbf{\Sigma}(\cdot)$: **scale** each component i by σ_i. Directions with large σ_i get stretched; directions with small σ_i get compressed; if $\sigma_i = 0$, that direction is annihilated.
3. $\mathbf{U}(\cdot)$: **rotate** the scaled result back into the output space. The columns $\mathbf{u}_1, \mathbf{u}_2, \ldots$ are the "natural output directions."

The singular values σ_i measure how much $\mathbf{A}$ stretches the i-th direction. They are always non-negative.

1.3 The Outer Product Form

There is an equivalent way to write SVD that reveals structure more clearly:

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $r = \text{rank}(\mathbf{A})$. Each term $\sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ is a **rank-1 matrix** — a simple stretched "slab." The full matrix $\mathbf{A}$ is the sum of these slabs, ordered by importance (σ_1 is largest). Truncating the sum after k terms gives the best rank-k approximation to $\mathbf{A}$ — this is the foundation of image compression, PCA, and noise reduction.

2 A Fully Worked Numerical Example

2.1 The Matrix

Let us compute the SVD of

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}.$$

2.2 Step 1 — Find Singular Values from $\mathbf{A}^\top\mathbf{A}$

The singular values are the square roots of the eigenvalues of $\mathbf{A}^\top\mathbf{A}$ (equivalently, $\mathbf{A}\mathbf{A}^\top$ for the left side). Since $\mathbf{A}$ is symmetric here, $\mathbf{A}^\top = \mathbf{A}$, so:

$$\mathbf{A}^\top\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}.$$

Find eigenvalues via the characteristic polynomial $\det(\mathbf{A}^\top\mathbf{A} - \lambda\mathbf{I}) = 0$:

$$(10-\lambda)(5-\lambda) - 25 = 0 \implies \lambda^2 - 15\lambda + 25 = 0.$$

Using the quadratic formula:

$$\lambda = \frac{15 \pm \sqrt{225 - 100}}{2} = \frac{15 \pm \sqrt{125}}{2} = \frac{15 \pm 5\sqrt{5}}{2}.$$

So:

$$\lambda_1 = \frac{15 + 5\sqrt{5}}{2} \approx 13.09, \qquad \lambda_2 = \frac{15 - 5\sqrt{5}}{2} \approx 1.91.$$

The singular values are:

$$\sigma_1 = \sqrt{\lambda_1} \approx 3.618, \qquad \sigma_2 = \sqrt{\lambda_2} \approx 1.382.$$

2.3 Step 2 — Find Right Singular Vectors V

For each eigenvalue, solve $(\mathbf{A}^\top\mathbf{A} - \lambda_i\mathbf{I})\mathbf{v} = \mathbf{0}$.

For $\lambda_1 \approx 13.09$: $(10 - 13.09)v_1 + 5v_2 = 0 \implies -3.09\,v_1 + 5\,v_2 = 0$, giving $v_2/v_1 = 0.618$. Normalising:

$$\mathbf{v}_1 \approx \begin{bmatrix} 0.851 \\ 0.526 \end{bmatrix}.$$

For $\lambda_2 \approx 1.91$: Orthogonal to $\mathbf{v}_1$:

$$\mathbf{v}_2 \approx \begin{bmatrix} -0.526 \\ 0.851 \end{bmatrix}.$$

Therefore:

$$\mathbf{V} = \begin{bmatrix} 0.851 & -0.526 \\ 0.526 & 0.851 \end{bmatrix}, \qquad \mathbf{V}^\top = \begin{bmatrix} 0.851 & 0.526 \\ -0.526 & 0.851 \end{bmatrix}.$$

2.4 Step 3 — Find Left Singular Vectors U

For each i: $\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i$.

For $i = 1$:

$$\mathbf{A}\mathbf{v}_1 = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0.851 \\ 0.526 \end{bmatrix} = \begin{bmatrix} 3(0.851) + 1(0.526) \\ 1(0.851) + 2(0.526) \end{bmatrix} = \begin{bmatrix} 3.079 \\ 1.903 \end{bmatrix}.$$

Dividing by $\sigma_1 \approx 3.618$: $\mathbf{u}_1 \approx [0.851,\ 0.526]^\top$.

For $i = 2$:

$$\mathbf{A}\mathbf{v}_2 = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -0.526 \\ 0.851 \end{bmatrix} = \begin{bmatrix} -0.727 \\ 0.176 \end{bmatrix}.$$

Dividing by $\sigma_2 \approx 1.382$: $\mathbf{u}_2 \approx [-0.526,\ 0.851]^\top$.

2.5 Step 4 — Assemble the Decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \underbrace{\begin{bmatrix} 0.851 & -0.526 \\ 0.526 & 0.851 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} 3.618 & 0 \\ 0 & 1.382 \end{bmatrix}}_{\mathbf{\Sigma}} \underbrace{\begin{bmatrix} 0.851 & 0.526 \\ -0.526 & 0.851 \end{bmatrix}}_{\mathbf{V}^\top}.$$

Verification

Compute $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and check it equals $\mathbf{A}$:

$$\mathbf{U}\mathbf{\Sigma} = \begin{bmatrix} 0.851 \times 3.618 & -0.526 \times 1.382 \\ 0.526 \times 3.618 & 0.851 \times 1.382 \end{bmatrix} = \begin{bmatrix} 3.079 & -0.727 \\ 1.903 & 1.176 \end{bmatrix}.$$

$$(\mathbf{U}\mathbf{\Sigma})\mathbf{V}^\top = \begin{bmatrix} 3.079 & -0.727 \\ 1.903 & 1.176 \end{bmatrix} \begin{bmatrix} 0.851 & 0.526 \\ -0.526 & 0.851 \end{bmatrix} = \begin{bmatrix} 3.000 & 1.000 \\ 1.000 & 2.000 \end{bmatrix} = \mathbf{A}.\checkmark$$

2.6 Geometric Visualisation

Figure 1 shows what the SVD of $\mathbf{A}$ does to the unit circle. The unit circle (all vectors of length 1) gets mapped to an ellipse. The right singular vectors $\mathbf{v}_1, \mathbf{v}_2$ define the "natural input directions" — when you apply $\mathbf{A}$ to a vector along $\mathbf{v}_i$, the output points exactly along $\mathbf{u}_i$, scaled by σ_i. Every other input direction produces an output that is a mixture.

3 From Data to Covariance: the SVD Connection

Before getting to Mahalanobis distance, we need to understand how SVD of the *data matrix* relates to eigendecomposition of the *covariance matrix*. This is a key bridge.

3.1 The Data Matrix and Covariance

Suppose you have n observations of a d-dimensional random variable. Stack them into a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ (rows = observations, columns = features). First **centre** the data by subtracting the column means so every feature has mean zero. Then the **sample covariance matrix** is:

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{d \times d}.$$

$\mathbf{\Sigma}$ is always symmetric and positive semi-definite. Entry (i, j) is the covariance between feature i and feature j — how much they vary together. Diagonal entries are variances.

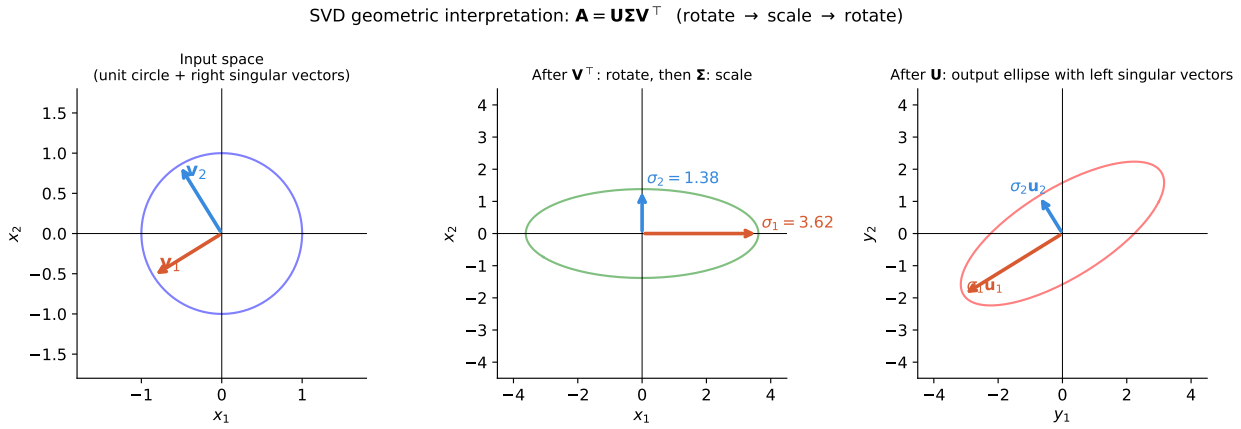

Figure 1: SVD geometric interpretation. **Left:** Unit circle in input space, with right singular vectors $\mathbf{v}_1$ (orange) and $\mathbf{v}_2$ (blue) shown as the natural input axes. **Middle:** After $\mathbf{V}^\top$ rotates (aligning with the natural axes) and $\mathbf{\Sigma}$ scales, the circle becomes an axis-aligned ellipse with semi-axes σ_1 and σ_2. **Right:** After $\mathbf{U}$ rotates the result into output space, the ellipse is oriented along the left singular vectors $\sigma_i \mathbf{u}_i$.

3.2 SVD of X Gives the Covariance Eigenvectors Directly

Compute the SVD of the centred data matrix: $\mathbf{X} = \mathbf{USV}^\top$.

Now look at $\mathbf{X}^\top\mathbf{X}$:

$$\mathbf{X}^\top\mathbf{X} = (\mathbf{USV}^\top)^\top(\mathbf{USV}^\top) = \mathbf{VS}^\top\mathbf{U}^\top\mathbf{USV}^\top = \mathbf{VS}^2\mathbf{V}^\top.$$

(using $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and $\mathbf{S}^\top\mathbf{S} = \mathbf{S}^2$ since $\mathbf{S}$ is diagonal with non-negative entries.)

Therefore:

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X} = \mathbf{V}\underbrace{\frac{\mathbf{S}^2}{n-1}}_{\mathbf{\Lambda}}\mathbf{V}^\top,$$

SVD–Covariance Connection

The **right singular vectors** $\mathbf{v}_i$ (columns of $\mathbf{V}$ from SVD of $\mathbf{X}$) are the **eigenvectors** of the covariance matrix $\mathbf{\Sigma}$.
The **eigenvalues** of $\mathbf{\Sigma}$ are $\lambda_i = \sigma_i^2/(n-1)$, where σ_i are the singular values of $\mathbf{X}$.
This means: **you never need to explicitly form $\mathbf{\Sigma}$ to find its eigenvectors — just SVD the data matrix directly.** This is numerically more stable and what PCA implementations use in practice.

3.3 Geometric Meaning of Eigenvectors

The eigenvectors of $\mathbf{\Sigma}$ (the columns of $\mathbf{V}$) are the **principal directions** of the data cloud — the directions along which the data varies the most ($\mathbf{v}_1$, corresponding to largest λ_1), second-most ($\mathbf{v}_2$), and so on. They are orthogonal to each other. Collectively they define a new coordinate system aligned with the natural axes of the data ellipsoid.

The corresponding eigenvalues λ_i measure the **variance** along each principal direction.

4 Mahalanobis Distance: Definition and Derivation

4.1 The Problem with Euclidean Distance

Euclidean distance treats all directions equally. But in a correlated, anisotropic data distribution, not all directions are equal — the data is naturally spread out more in some directions than others. A point that is "far" from the mean in a direction where the data has high variance is actually quite typical. A point that is equally far in a direction where the data is tightly clustered is much more unusual.

Euclidean distance cannot distinguish these two situations. Mahalanobis distance was designed specifically to fix this.

4.2 Definition

Mahalanobis Distance

For a point $\mathbf{x}$ relative to a distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the Mahalanobis distance is:

$$d_M(\mathbf{x}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}.$$

Setting $\mathbf{e} = \mathbf{x} - \boldsymbol{\mu}$ (the deviation from the mean):

$$d_M^2 = \mathbf{e}^\top \boldsymbol{\Sigma}^{-1}\mathbf{e} = \|\mathbf{e}\|_{\boldsymbol{\Sigma}}^2.$$

This is the **weighted norm** notation used throughout this document.

When $\boldsymbol{\Sigma} = \mathbf{I}$ (identity — all directions equally uncertain): $d_M = \|\mathbf{e}\|$ = plain Euclidean distance. The Mahalanobis distance is a strict generalisation.

4.3 Deriving Mahalanobis Distance from SVD

Here is where SVD and Mahalanobis distance meet explicitly. Start from the eigendecomposition of the covariance matrix (which we know comes from SVD of the data matrix):

$$\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top, \qquad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d).$$

The inverse is:

$$\boldsymbol{\Sigma}^{-1} = (\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top)^{-1} = \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top, \qquad \boldsymbol{\Lambda}^{-1} = \operatorname{diag}(1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_d).$$

(For a symmetric matrix, $(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top)^{-1} = \mathbf{V}^{-\top}\boldsymbol{\Lambda}^{-1}\mathbf{V}^{-1} = \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top$ since $\mathbf{V}$ is orthogonal, so $\mathbf{V}^{-1} = \mathbf{V}^\top$.)

Substitute into d_M^2:

$$\begin{aligned} d_M^2 &= \mathbf{e}^\top \boldsymbol{\Sigma}^{-1}\mathbf{e} \\ &= \mathbf{e}^\top \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top\mathbf{e} \\ &= (\mathbf{V}^\top\mathbf{e})^\top \boldsymbol{\Lambda}^{-1}(\mathbf{V}^\top\mathbf{e}). \end{aligned}$$

Now define $\tilde{\mathbf{e}} = \mathbf{V}^\top\mathbf{e}$. This is just $\mathbf{e}$ expressed in the eigenvector coordinate system — the rotation step from SVD. Then:

$$d_M^2 = \tilde{\mathbf{e}}^\top \mathbf{\Lambda}^{-1} \tilde{\mathbf{e}} = \sum_{i=1}^{d} \frac{\tilde{e}_i^2}{\lambda_i} = \sum_{i=1}^{d} \left(\frac{\tilde{e}_i}{\sqrt{\lambda_i}} \right)^2 .$$

The SVD Interpretation of Mahalanobis Distance

Each term $\tilde{e}_i^2/\lambda_i = (\tilde{e}_i/\sqrt{\lambda_i})^2$ is the squared deviation in the i-th eigenvector direction, normalised by the standard deviation $\sqrt{\lambda_i}$ along that direction.
In other words: **how many standard deviations away is the point along each principal axis?** The Mahalanobis distance is the Euclidean length of this vector of standard-deviation-counts.
Written compactly using the SVD factors:

$$d_M^2 = \|\mathbf{\Lambda}^{-1/2} \mathbf{V}^\top \mathbf{e}\|^2, \qquad \mathbf{\Lambda}^{-1/2} = \mathrm{diag}(1/\sqrt{\lambda_1}, \ldots, 1/\sqrt{\lambda_d}).$$

The two operations are:

1. $\mathbf{V}^\top$: **rotate** the error vector into the eigenvector basis (the SVD rotation step)
2. $\mathbf{\Lambda}^{-1/2}$: **scale** each component by $1/\sqrt{\lambda_i}$ (the SVD scaling step, inverted)

Then take the Euclidean norm. Mahalanobis distance is Euclidean distance after undoing the rotation and scaling that the covariance matrix encodes.

5 Numerical Example: Mahalanobis Distance in 2D

5.1 Setup

Consider a 2D Gaussian distribution with mean $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} 4.0 & 2.8 \\ 2.8 & 2.5 \end{bmatrix}.$$

The off-diagonal entry 2.8 indicates strong positive correlation: when x_1 is large, x_2 tends to also be large.

5.2 Step 1 — Eigendecomposition of $\mathbf{\Sigma}$

Find eigenvalues via $\det(\mathbf{\Sigma} - \lambda \mathbf{I}) = 0$:

$$(4 - \lambda)(2.5 - \lambda) - (2.8)^2 = 0 \implies \lambda^2 - 6.5\lambda + 2.16 = 0.$$

$$\lambda = \frac{6.5 \pm \sqrt{42.25 - 8.64}}{2} = \frac{6.5 \pm \sqrt{33.61}}{2} = \frac{6.5 \pm 5.798}{2}.$$

$\lambda_1 \approx 6.149$ (large: high variance direction), $\lambda_2 \approx 0.351$ (small: low variance direction).

Standard deviations: $\sqrt{\lambda_1} \approx 2.480$, $\sqrt{\lambda_2} \approx 0.592$.
Eigenvectors (solving $(\mathbf{\Sigma} - \lambda_i \mathbf{I})\mathbf{v} = \mathbf{0}$ and normalising):

$$\mathbf{v}_1 \approx \begin{bmatrix} 0.830 \\ 0.558 \end{bmatrix}, \qquad \mathbf{v}_2 \approx \begin{bmatrix} -0.558 \\ 0.830 \end{bmatrix}.$$

SVD Connection

These eigenvectors are exactly the right singular vectors $\mathbf{v}_i$ you would get by computing SVD of the centred data matrix $\mathbf{X}$ and looking at the columns of $\mathbf{V}$. The eigenvalues λ_i equal $\sigma_i^2/(n-1)$ where σ_i are the singular values of $\mathbf{X}$.

5.3 Step 2 — Two Query Points

Point	Coordinates	Euclidean dist. from origin	Mahalanobis dist.
$\mathbf{p}_A$	$(2.2,\ 1.5)$	$\sqrt{2.2^2+1.5^2}=\sqrt{7.09}\approx 2.66$	(computed below)
$\mathbf{p}_B$	$(-0.3,\ 2.0)$	$\sqrt{0.09+4.00}=\sqrt{4.09}\approx 2.02$	(computed below)

Point $\mathbf{p}_A$ is further from the origin by Euclidean distance. But is it more "unusual" relative to this distribution?

5.4 Step 3 — Rotate into Eigenvector Basis

Compute $\tilde{\mathbf{e}} = \mathbf{V}^\top \mathbf{e}$ (rotation step — corresponds to $\mathbf{V}^\top$ in SVD):

$$\mathbf{V}^\top = \begin{bmatrix} 0.830 & 0.558 \\ -0.558 & 0.830 \end{bmatrix}.$$

For point $\mathbf{p}_A = (2.2, 1.5)$:

$$\tilde{\mathbf{e}}_A = \begin{bmatrix} 0.830(2.2)+0.558(1.5) \\ -0.558(2.2)+0.830(1.5) \end{bmatrix} = \begin{bmatrix} 1.826+0.837 \\ -1.228+1.245 \end{bmatrix} = \begin{bmatrix} 2.663 \\ 0.017 \end{bmatrix}.$$

For point $\mathbf{p}_B = (-0.3, 2.0)$:

$$\tilde{\mathbf{e}}_B = \begin{bmatrix} 0.830(-0.3)+0.558(2.0) \\ -0.558(-0.3)+0.830(2.0) \end{bmatrix} = \begin{bmatrix} -0.249+1.116 \\ 0.167+1.660 \end{bmatrix} = \begin{bmatrix} 0.867 \\ 1.827 \end{bmatrix}.$$

Reading: $\tilde{e}_{A,1} = 2.663$ means point A is 2.663 units along the major principal axis ($\mathbf{v}_1$, the direction of maximum variance). $\tilde{e}_{A,2} = 0.017$ means it is almost exactly on the major axis. Point B has a large component (1.827) along the minor axis ($\mathbf{v}_2$, low variance direction).

5.5 Step 4 — Scale by $1/\sqrt{\lambda_i}$

Divide each rotated component by the standard deviation along that axis (corresponds to $\mathbf{\Lambda}^{-1/2}$ in SVD):

For $\mathbf{p}_A$:

$$\frac{\tilde{e}_{A,1}}{\sqrt{\lambda_1}} = \frac{2.663}{2.480} \approx 1.074, \qquad \frac{\tilde{e}_{A,2}}{\sqrt{\lambda_2}} = \frac{0.017}{0.592} \approx 0.029.$$

$$d_M^2(\mathbf{p}_A) = 1.074^2 + 0.029^2 \approx 1.154 + 0.001 = 1.155, \qquad d_M(\mathbf{p}_A) \approx \mathbf{1.074}.$$

For $\mathbf{p}_B$:

$$\frac{\tilde{e}_{B,1}}{\sqrt{\lambda_1}} = \frac{0.867}{2.480} \approx 0.350, \qquad \frac{\tilde{e}_{B,2}}{\sqrt{\lambda_2}} = \frac{1.827}{0.592} \approx 3.086.$$

$$d_M^2(\mathbf{p}_B) = 0.350^2 + 3.086^2 \approx 0.122 + 9.524 = 9.646, \qquad d_M(\mathbf{p}_B) \approx \mathbf{3.106}.$$

5.6 Comparison and Interpretation

Point	Euclidean dist.	Mahalanobis dist.	More unusual?	Why
$\mathbf{p}_A$ $(2.2, 1.5)$	2.66 (larger)	1.07 (smaller)	No	Along major axis, inside ellipse
$\mathbf{p}_B$ $(-0.3, 2.0)$	2.02 (smaller)	3.11 (larger!)	Yes	Against minor axis, outside ellipse

Key insight: Point A is further from the origin in Euclidean distance, but it lies *along the direction where the data naturally spreads* (the major axis, $\lambda_1 = 6.149$ is large). Being 2.66 units in that direction corresponds to only $2.663/2.480 = 1.07$ standard deviations — comfortably typical.

Point B is closer to the origin in Euclidean distance, but it goes *against the minor axis* ($\lambda_2 = 0.351$ is small — data barely spreads there). Being 1.827 units in the minor direction corresponds to $1.827/0.592 = 3.09$ standard deviations — an unusual position.

Euclidean distance called A more unusual. Mahalanobis distance correctly identifies B as more unusual.

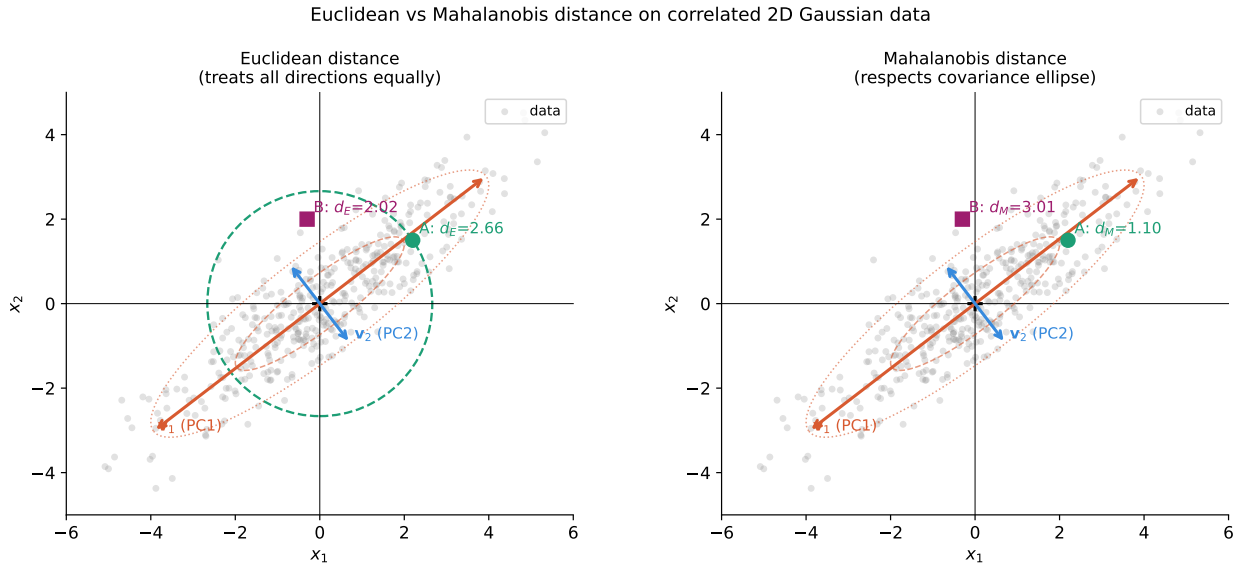

Figure 2: Euclidean vs Mahalanobis distance on data sampled from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with the covariance from our example. Orange and blue arrows are the eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ scaled by 2 standard deviations. Dashed and dotted ellipses are 1 and 2 standard deviation Mahalanobis contours. Green dot $= \mathbf{p}_A$, purple square $= \mathbf{p}_B$. **Left:** Euclidean distance — the dashed circle shows equal Euclidean distance, making A look further. **Right:** Mahalanobis distance — B (outside the 2σ ellipse) is correctly identified as much further.

6 The Whitening Transform

6.1 Making the Ellipse a Circle

The two-step process (rotate by $\mathbf{V}^\top$, then scale by $\mathbf{\Lambda}^{-1/2}$) that computes Mahalanobis distance is called **whitening** or **sphering** the data. The combined transform is:

$$\mathbf{W} = \mathbf{\Lambda}^{-1/2}\mathbf{V}^\top = \mathbf{\Sigma}^{-1/2}, \qquad \tilde{\mathbf{x}} = \mathbf{W}\mathbf{x}.$$

The whitened data $\tilde{\mathbf{x}}$ has covariance matrix $\mathbf{I}$ — it has been transformed into an isotropic (spherically symmetric) distribution. In this whitened space, Euclidean distance *equals* Mahalanobis distance in the original space.

Why "Whitening"?

"White noise" is noise that has equal power (variance) in all directions — it looks like a sphere in data space. Whitening transforms correlated, anisotropic data into data that looks like white noise. The name comes from the analogy with white light, which contains all frequencies equally.

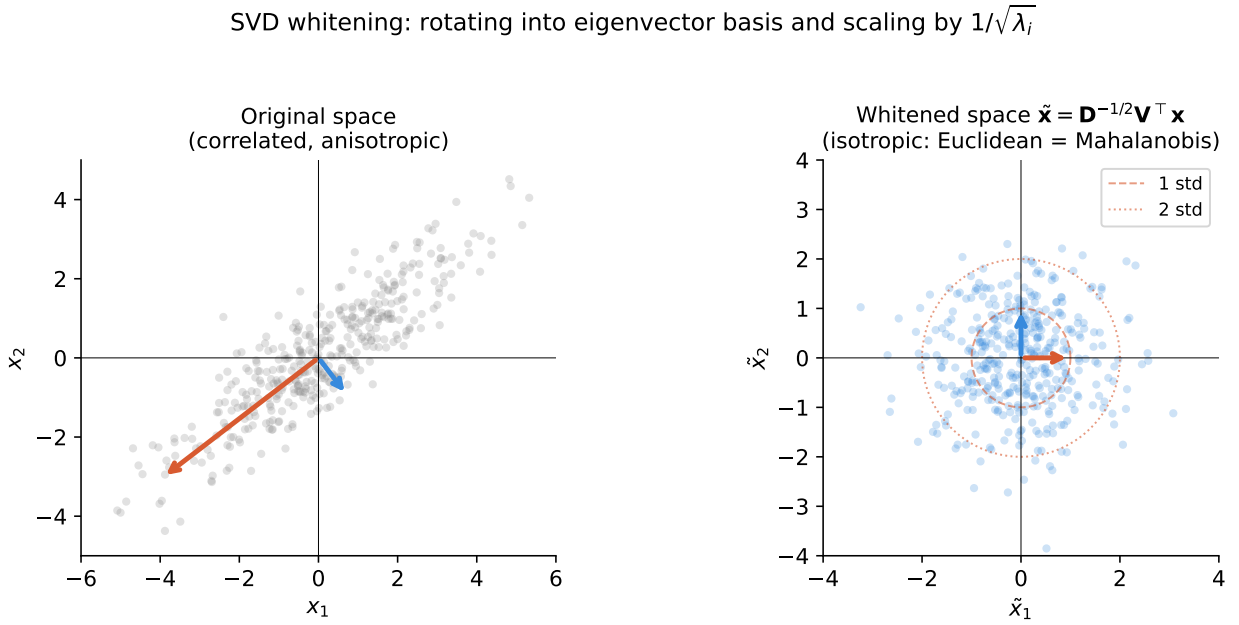

Figure 3: **Left:** Original correlated data (grey) with principal component arrows. The data ellipse is tilted and anisotropic. **Right:** After the whitening transform $\mathbf{W} = \mathbf{\Lambda}^{-1/2}\mathbf{V}^\top$ (rotate into eigenvector basis, scale by $1/\sqrt{\lambda_i}$), the data becomes isotropic — the ellipse becomes a circle. In this whitened space, standard Euclidean distance equals the original Mahalanobis distance.

6.2 Connection Back to SVD

The whitening transform $\mathbf{W} = \mathbf{\Lambda}^{-1/2}\mathbf{V}^\top$ is exactly the composition of the two SVD steps we have been tracking throughout:

1. $\mathbf{V}^\top$: rotate into the right singular vector basis — same as the $\mathbf{V}^\top$ factor in SVD of the data matrix.
2. $\mathbf{\Lambda}^{-1/2}$: scale by the reciprocal square roots of the eigenvalues — the *inverse* of the scaling by singular values that SVD performs.

Mahalanobis distance is, therefore, literally the Euclidean distance after you **undo** the transformation that the data covariance encodes — and SVD is the tool that tells you exactly how to undo it.

7 Summary: SVD and Mahalanobis Distance in One Picture

The Complete Chain

$$\underbrace{\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top}_{\text{SVD of data matrix}} \implies \underbrace{\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top}_{\text{eigendecomposition of covariance, same } \mathbf{V}} \implies \underbrace{d_M^2 = \|\boldsymbol{\Lambda}^{-1/2}\mathbf{V}^\top \mathbf{e}\|^2}_{\text{Mahalanobis = Euclidean after rotate + scale}}$$

SVD concept	Role in Mahalanobis distance	Meaning
Right singular vectors $\mathbf{v}_i$	Eigenvectors of $\boldsymbol{\Sigma}$	Principal directions of data
Singular values σ_i	$\lambda_i = \sigma_i^2/(n-1)$	Variances along each axis
$\mathbf{V}^\top$ (rotation)	$\tilde{\mathbf{e}} = \mathbf{V}^\top \mathbf{e}$	Express error in principal axes
$\boldsymbol{\Sigma}^{-1}$ scaling	$1/\lambda_i$ weighting	Normalise by each axis's variance
Whitening $\boldsymbol{\Lambda}^{-1/2}\mathbf{V}^\top$	Converts to isotropic space	Euclidean = Mahalanobis after this

The one-sentence summary: SVD decomposes a covariance matrix into principal directions ($\mathbf{V}$) and principal variances ($\boldsymbol{\Lambda}$); Mahalanobis distance rotates into those directions and normalises by those variances — and in that transformed space, it reduces to plain Euclidean distance.