Euclidean Distance

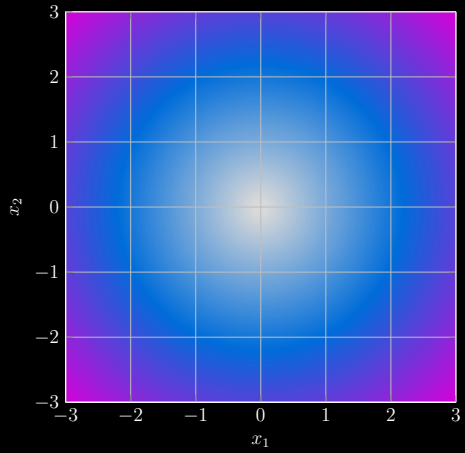
The Euclidean distance of a point $\boldsymbol{x} = (x_1, x_2)^\top \in \mathbb{R}^2$ from $\boldsymbol{\mu} = (\mu_1, \mu_2)^\top \in \mathbb{R}^2$:

$$d = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^\top (\boldsymbol{x} - \boldsymbol{\mu})} \qquad \text{(Square root of the inner product)}$$

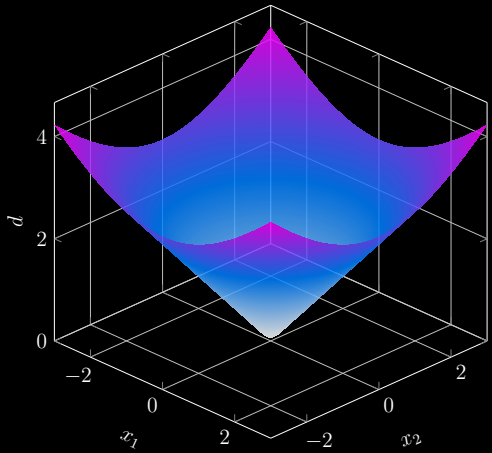
$$= \sqrt{(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2} \qquad \text{(Length of the line segment)}$$

Geometry of Euclidean Distance $d = \sqrt{x_1^2 + x_2^2}$

Contour ($\boldsymbol{\mu} = \mathbf{0}$)



Cone ($\boldsymbol{\mu} = \mathbf{0}$)



Mahalanobis Distance

The Mahalanobis distance of a point $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top \in \mathbb{R}^n$ from a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

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

with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$.

Wikipedia page	Page view (Desktop)	Page view (Mobile)
Machine_learning	88,260	56,966
Algorithm	38,905	36,203
Computer_science	47,723	40,321
Computer_program	13,451	16,587
Reinforcement_learning	30,393	9,841
Statistics	29,673	28,341
Deep_learning	36,741	15,321
Statistical_classification	3,947	1,403
Pattern_recognition	6,119	3,223
Anomaly_detection	7,540	1,514
Supervised_learning	14,120	3,596
Unsupervised_learning	11,693	2,702
Regression_analysis	32,814	16,207
Probability_theory	10,913	8,788
Mathematical_optimization	19,318	6,819
Data_mining	13,168	5,750

The dataset is available at https://arxiv.org/abs/2605.16361

Compute Mean & Covariance

Mean vector: $\boldsymbol{\mu} = \begin{bmatrix} 2.5 \times 10^4 \\ 1.6 \times 10^4 \end{bmatrix}$ (Desktop) (Mobile)

Covariance matrix: $\boldsymbol{\Sigma} = \begin{bmatrix} 4.6 \times 10^8 & 3.2 \times 10^8 \\ 3.2 \times 10^8 & 2.7 \times 10^8 \end{bmatrix}$

Covariance Matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$

Eigenvalue decomposition:

$$\boldsymbol{\Sigma} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top$$

with eigenvalues:

$$\lambda_1 = 6.98 \times 10^8$$
$$\lambda_2 = 0.28 \times 10^8$$

and eigenvectors:

$$\boldsymbol{u}_1 = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \quad \boldsymbol{u}_2 = \begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix}$$

Covariance Matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$

Eigenvalue decomposition:

$$\boldsymbol{\Sigma} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top$$

with eigenvalues:

$$\lambda_1 = 6.98 \times 10^8$$
$$\lambda_2 = 0.28 \times 10^8$$

and eigenvectors:

$$\boldsymbol{u}_1 = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \quad \boldsymbol{u}_2 = \begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix}$$

```python
import numpy as np

S = np.array([[4.57602874e+08,
    3.21703384e+08], [3.21703384e
    +08, 2.68455504e+08]])
eig_val, eig_vec = np.linalg.eig(S)

print('lambda_1 = ', eig_val[0])
print('lambda_2 = ', eig_val[1])
print('u_1 = ', eig_vec[:, 0])
print('u_2 = ', eig_vec[:, 1])
```

Contours of squared Mahalanobis distance are ellipses.

$$
\begin{aligned}
d^2 =& (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \\
=& (\boldsymbol{x} - \boldsymbol{\mu})^\top \left(\lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top \right)^{-1} (\boldsymbol{x} - \boldsymbol{\mu})
\end{aligned}
$$

Contours of squared Mahalanobis distance are ellipses.

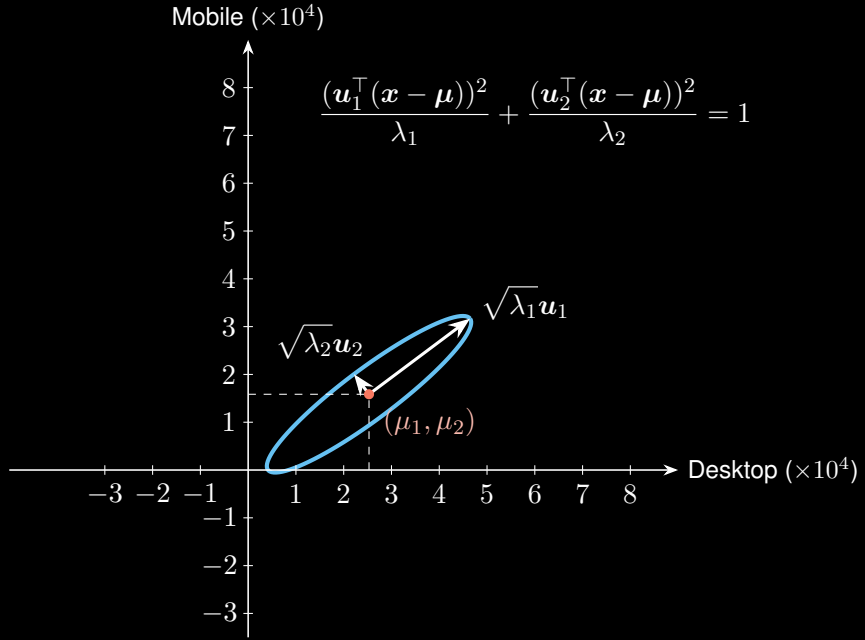
$$
\begin{aligned}
d^2 =& (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \\
=& (\boldsymbol{x}-\boldsymbol{\mu})^\top \left(\lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top\right)^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \\
=& (\boldsymbol{x}-\boldsymbol{\mu})^\top \left(\frac{1}{\lambda_1} \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \frac{1}{\lambda_2} \boldsymbol{u}_2 \boldsymbol{u}_2^\top\right) (\boldsymbol{x}-\boldsymbol{\mu}) = c
\end{aligned}
$$

Contours of squared Mahalanobis distance are ellipses.

$$
\begin{aligned}
d^2 =& (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \\
=& (\boldsymbol{x}-\boldsymbol{\mu})^\top \left(\lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top\right)^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \\
=& (\boldsymbol{x}-\boldsymbol{\mu})^\top \left(\frac{1}{\lambda_1} \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \frac{1}{\lambda_2} \boldsymbol{u}_2 \boldsymbol{u}_2^\top\right) (\boldsymbol{x}-\boldsymbol{\mu}) = c
\end{aligned}
$$

$$
\Rightarrow \qquad \frac{(\boldsymbol{u}_1^\top (\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_1} + \frac{(\boldsymbol{u}_2^\top (\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_2} = d^2 \qquad \text{(ellipse)}
$$

with eigenvalues λ_1, λ_2 and eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2$.

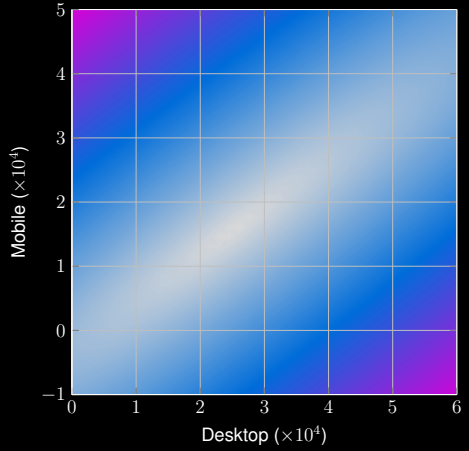

Mobile ($\times 10^4$)
$\frac{(\boldsymbol{u}_1^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_1} + \frac{(\boldsymbol{u}_2^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_2} = 1$
$\sqrt{\lambda_1}\boldsymbol{u}_1$
$\sqrt{\lambda_2}\boldsymbol{u}_2$
(μ_1, μ_2)
Desktop ($\times 10^4$)
8
7
6
5
4
3
2
1
−3 −2 −1
1 2 3 4 5 6 7 8
−1
−2
−3

Geometry of Mahalanobis Distance

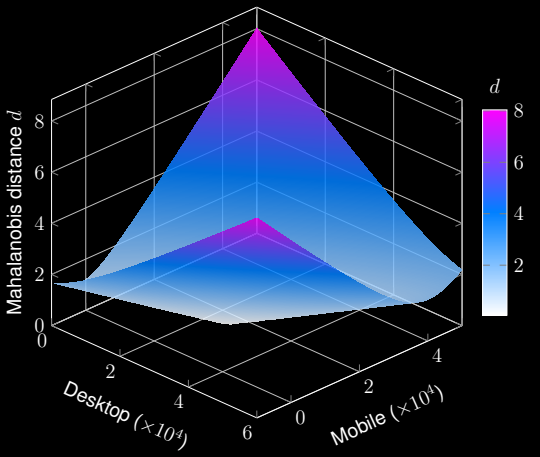
$$d = \sqrt{\frac{(\boldsymbol{u}_1^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_1} + \frac{(\boldsymbol{u}_2^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_2}}$$ (Elliptical cone)

Geometry of Mahalanobis Distance

Contour

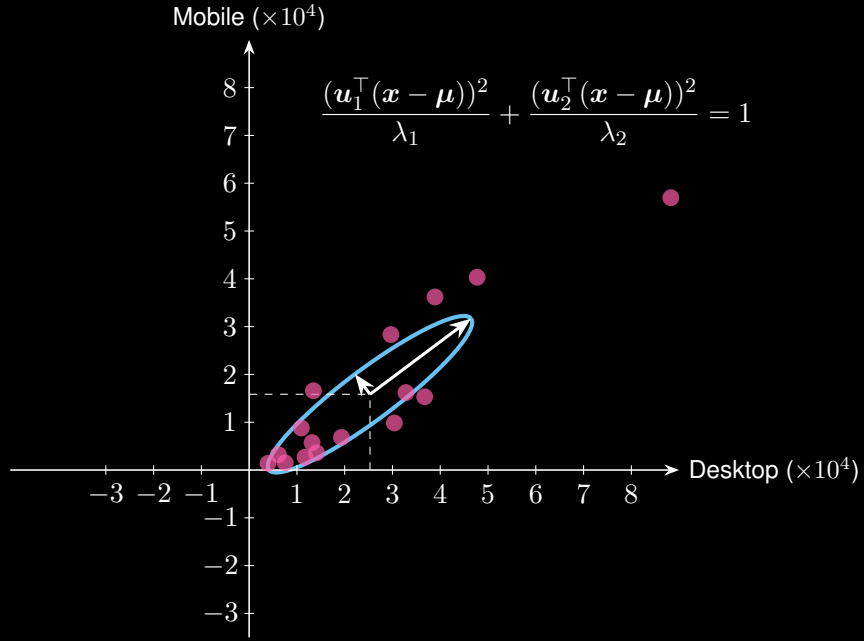


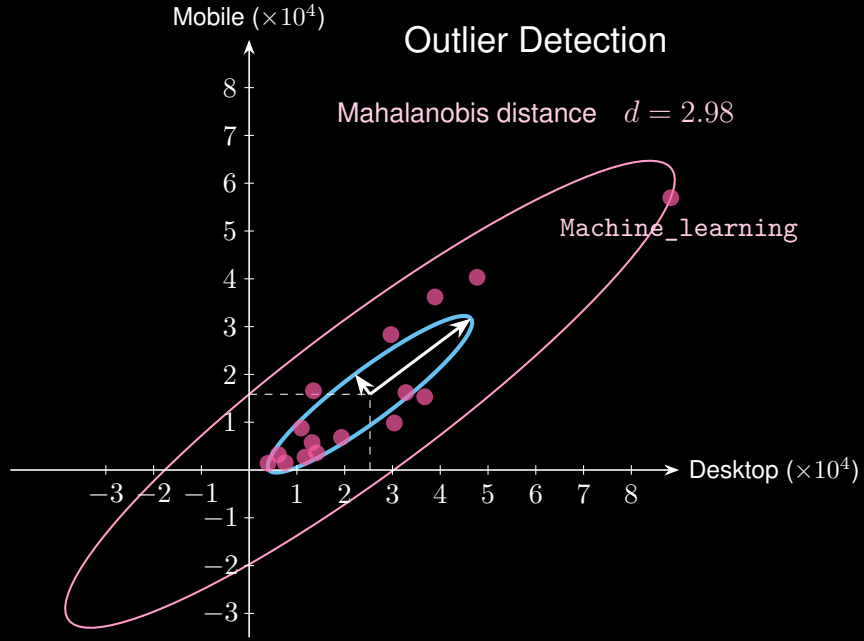
Elliptical cone

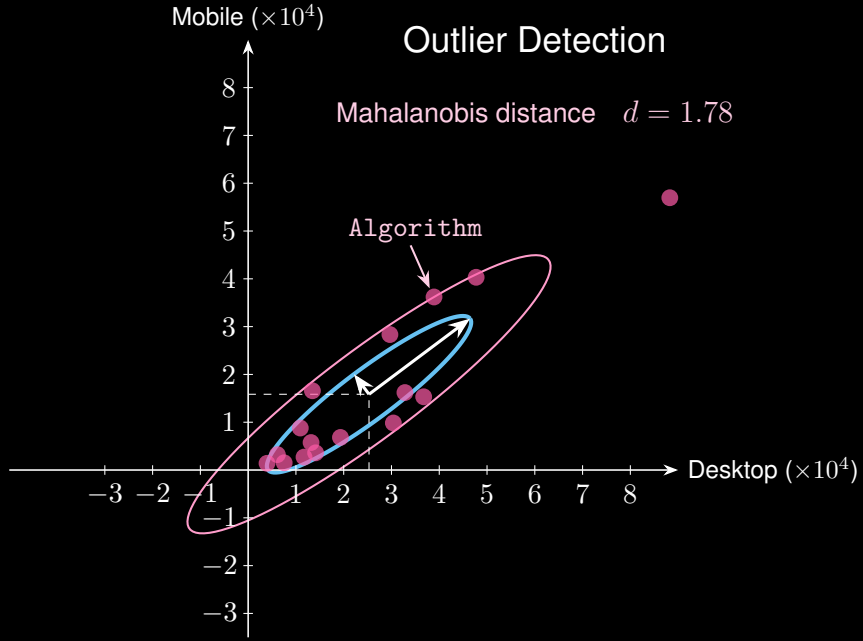


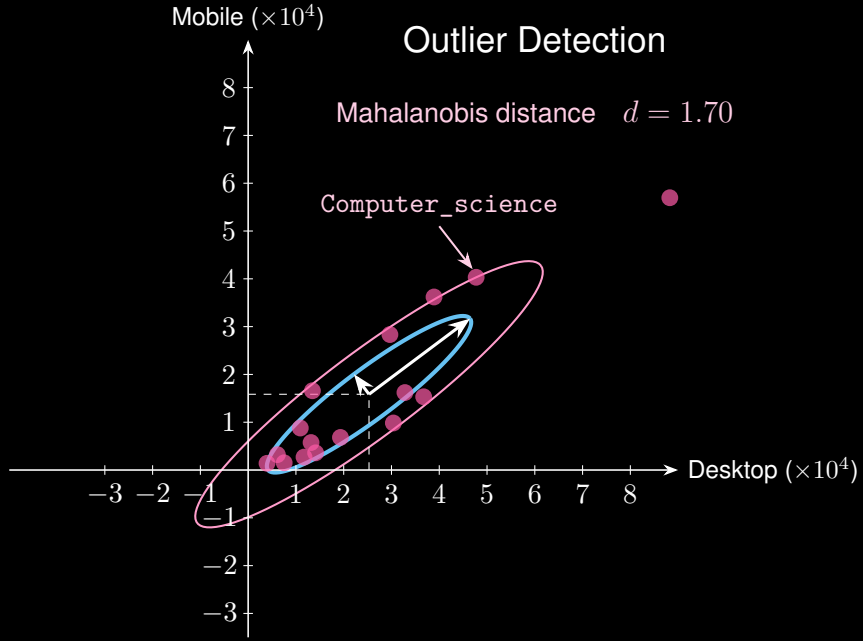
Wikipedia page	Page view (Desktop)	Page view (Mobile)
Machine_learning	88,260	56,966
Algorithm	38,905	36,203
Computer_science	47,723	40,321
Computer_program	13,451	16,587
Reinforcement_learning	30,393	9,841
Statistics	29,673	28,341
Deep_learning	36,741	15,321
Statistical_classification	3,947	1,403
Pattern_recognition	6,119	3,223
Anomaly_detection	7,540	1,514
Supervised_learning	14,120	3,596
Unsupervised_learning	11,693	2,702
Regression_analysis	32,814	16,207
Probability_theory	10,913	8,788
Mathematical_optimization	19,318	6,819
Data_mining	13,168	5,750

The dataset is available at https://arxiv.org/abs/2605.16361


Mobile ($\times 10^4$)
$\frac{(\boldsymbol{u}_1^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_1} + \frac{(\boldsymbol{u}_2^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2}{\lambda_2} = 1$
Desktop ($\times 10^4$)
8
7
6
5
4
3
2
1
−1
−2
−3
−3 −2 −1
1 2 3 4 5 6 7 8


Outlier Detection
Mahalanobis distance $d = 2.98$
Machine_learning
Mobile ($\times 10^4$)
Desktop ($\times 10^4$)
8
7
6
5
4
3
2
1
−3
−2
−1
1
2
3
4
5
6
7
8
−1
−2
−3


Outlier Detection
Mahalanobis distance $d = 1.78$
Algorithm
Mobile ($\times 10^4$)
Desktop ($\times 10^4$)
8
7
6
5
4
3
2
1
−1
−2
−3
−3
−2
−1
1
2
3
4
5
6
7
8


Outlier Detection
Mahalanobis distance $d = 1.70$
Computer_science
Mobile ($\times 10^4$)
Desktop ($\times 10^4$)
8
7
6
5
4
3
2
1
−1
−2
−3
−3
−2
−1
1
2
3
4
5
6
7
8

Wikipedia page	Desktop	Mobile	Mahalanobis distance d
Machine_learning	88,260	56,966	2.982756
Algorithm	38,905	36,203	1.776715
Computer_science	47,723	40,321	1.700476
Computer_program	13,451	16,587	1.500239
Reinforcement_learning	30,393	9,841	1.493654
Statistics	29,673	28,341	1.462395
Deep_learning	36,741	15,321	1.422524
Statistical_classification	3,947	1,403	1.001899
Pattern_recognition	6,119	3,223	0.906244
Anomaly_detection	7,540	1,514	0.877565
Supervised_learning	14,120	3,596	0.854178
Unsupervised_learning	11,693	2,702	0.841361
Regression_analysis	32,814	16,207	0.834884
Probability_theory	10,913	8,788	0.820096
Mathematical_optimization	19,318	6,819	0.792922
Data_mining	13,168	5,750	0.616373

Thanks for your attention!

About me:

- Homepage: https://xinychen.github.io
- How to reach me: chenxy346@gmail.com