Chapter 1

SVD, PCA & Pre-processing

Part 3: Interpreting and computing the SVD

max planck institut
informatik

Interpreting SVD

Skillicorn chapter 3.2

Factor interpretation

- Let $\boldsymbol{A}$ be objects-by-attributes and $\boldsymbol{U\Sigma V}^T$ its SVD
  - If two columns have similar values in a row of $\boldsymbol{V}^T$, these attributes are similar (have strong correlation)
  - If two rows have similar values in a column of $\boldsymbol{U}$, these objects are similar

Example

- Data: people's ratings on different wines
- Scatterplot of first two LSV
  - SVD doesn't know what the data is
- Conclusion: winelovers like red and white alike, others are more biased

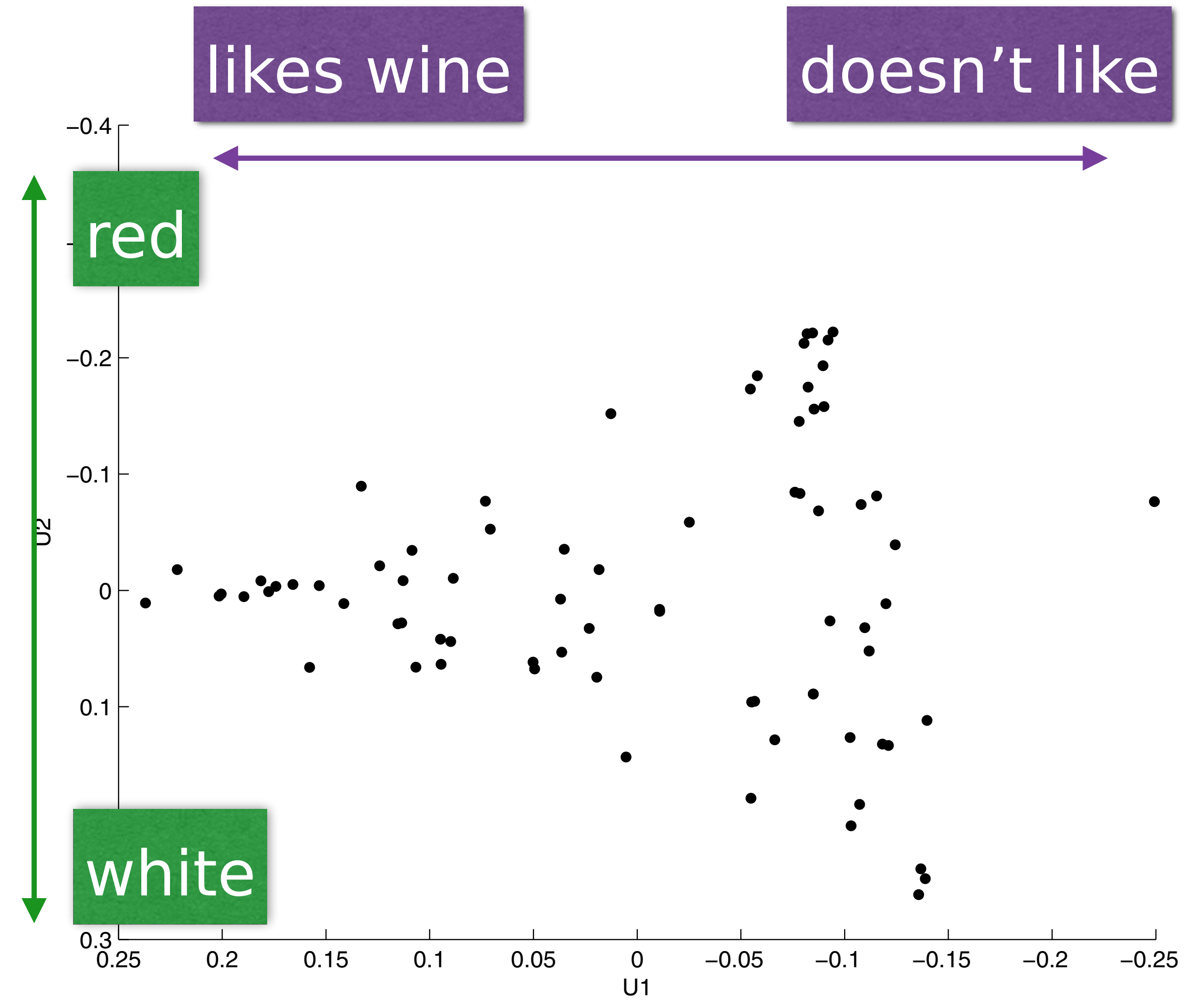
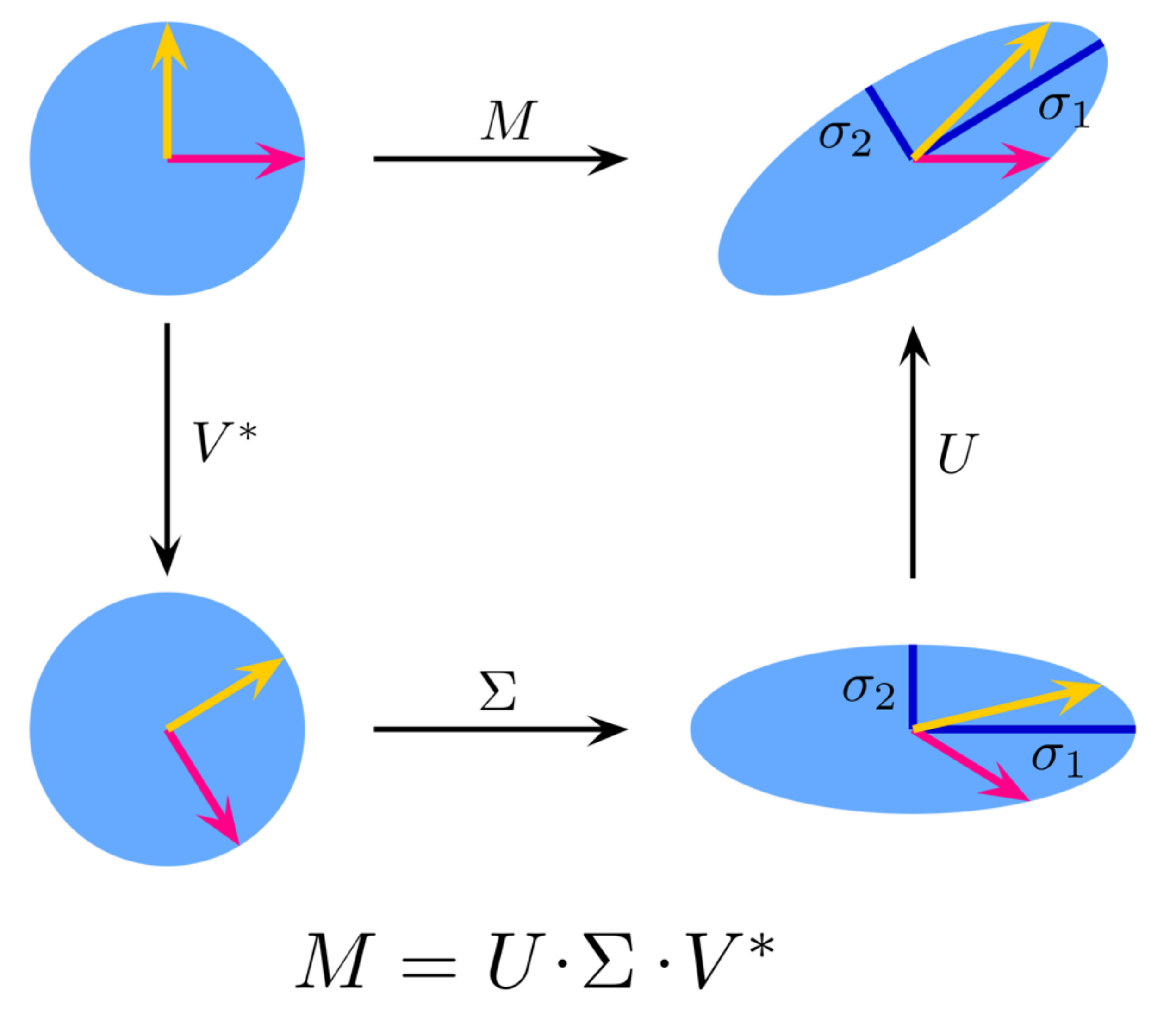


Figure 3.2. *The first two factors for a dataset ranking wines.*

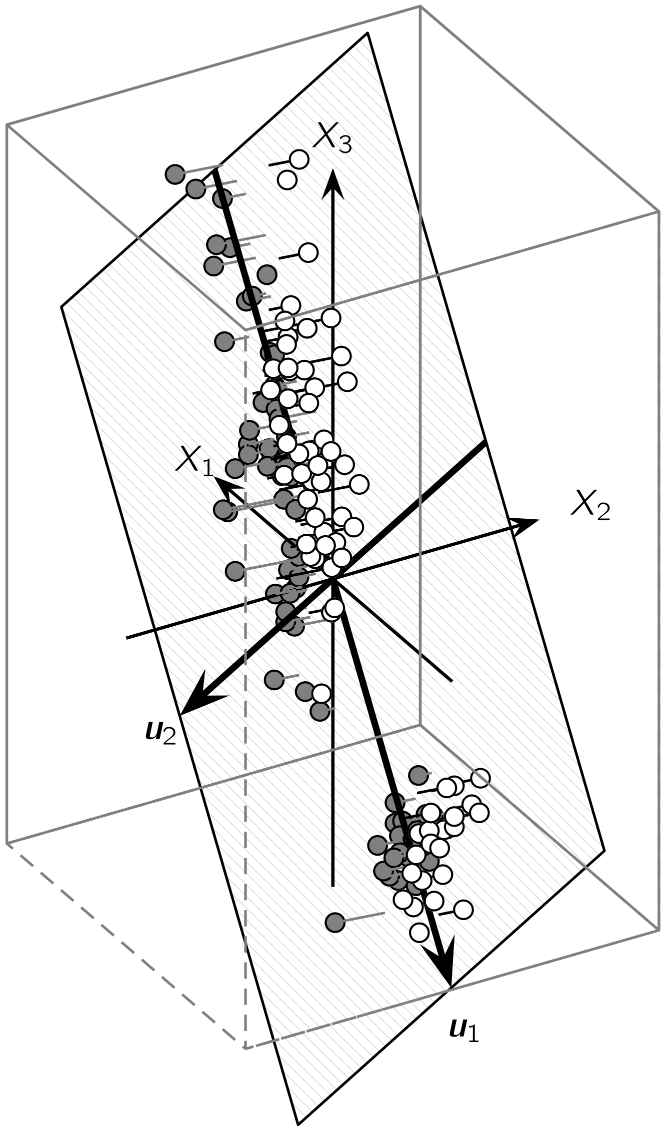
Geometric interpretation

- Let $\boldsymbol{M} = \boldsymbol{U\Sigma V}^T$
- Any linear mapping $\boldsymbol{y}=\boldsymbol{Mx}$ can be expressed as a rotation, stretching, and rotation operation
  - $\boldsymbol{y}_1 = \boldsymbol{V}^T\boldsymbol{x}$ is the first rotation
  - $\boldsymbol{y}_2 = \boldsymbol{\Sigma y}_1$ is the stretching
  - $\boldsymbol{y} = \boldsymbol{U y}_2$ is the final rotation



Direction of largest variances

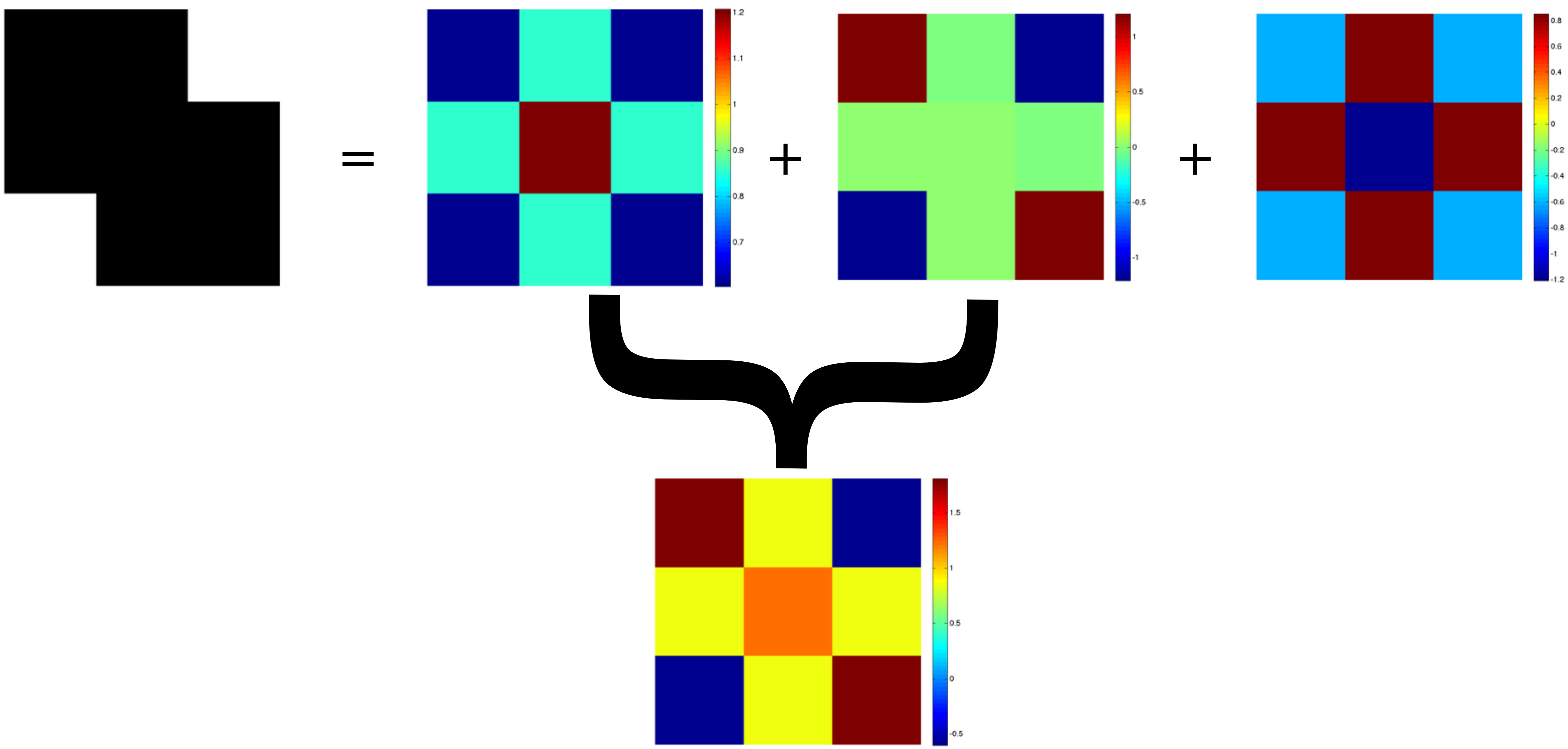
- The singular vectors give the directions of the largest variances
  - First singular vector points to the direction of the largest variance
  - Second to the second-largest
    - Spans a hyperplane with the first
- The projection distance to these hyperplanes is minimal over all hyperplanes (Eckart–Young)



Component interpretation

- We can write $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T = \sum_i \boldsymbol{A}_i$
- This explains the data as a sum of rank-1 layers
  - First layer explains the most, the second updates that, the third updates that, …
- Each individual layer don’t have to be very intuitive

Example

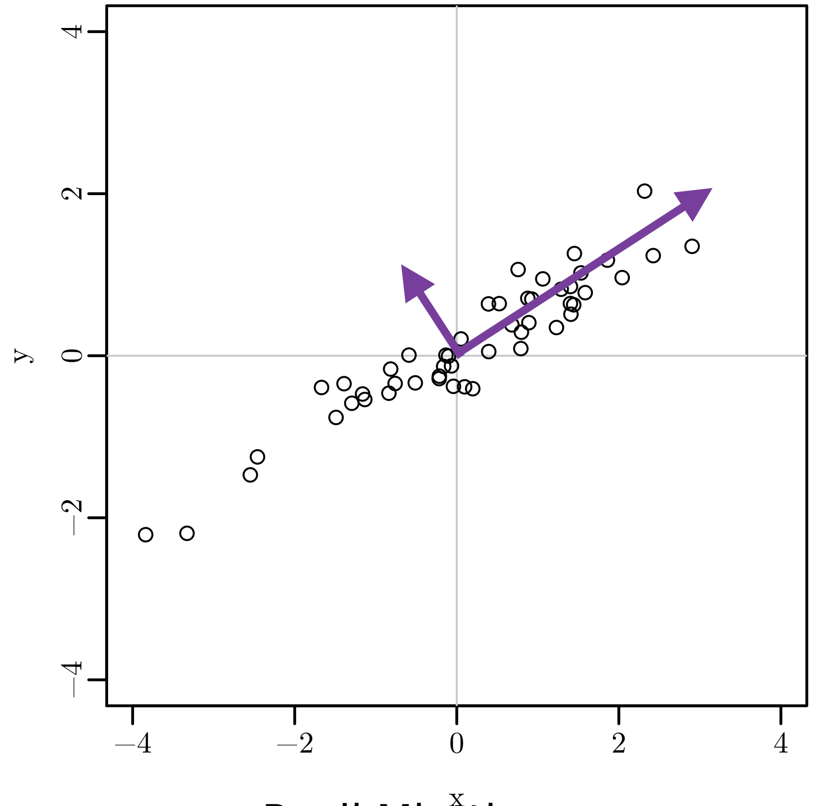


Applications of SVD

Skillicorn chapter 3.5; Leskovec et al. chapter 11.3

Removing noise

- SVD is often used as a pre-processing step to remove noise from the data
  - The rank-*k* truncated SVD with proper *k*



$\sigma_1 = 11.73$

$\sigma_2 = 1.71$

Removing dimensions

- SVD can be used to project the data to smaller-dimensional subspace
  - Original dimensions can have complex correlations
  - Subsequent analysis is faster
  - Points seem close to each other in high-dimensional space

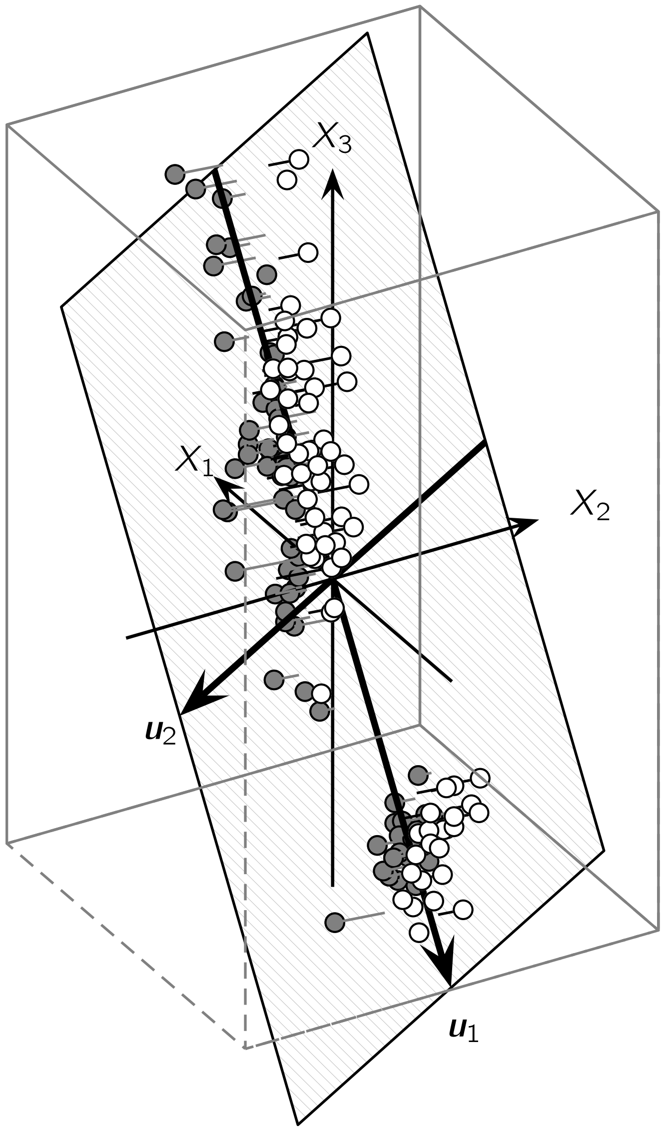
Curse of dimensionality

Karhunen–Loève transform

- The **Karhunen–Loève transform** (KLT) works as follows:
  - Normalize $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ to z-scores
  - Compute the SVD $\boldsymbol{U\Sigma V}^T = \boldsymbol{A}$
  - Project $\boldsymbol{A} \mapsto \boldsymbol{AV}_k \in \mathbb{R}^{n \times k}$
    - $\boldsymbol{V}_k$ = top-k right singular vectors
- A.k.a. the **principal component analysis** (PCA)

More on KLT

- The columns of $\boldsymbol{V}_k$ show the main directions of variance in columns
- The data is expressed in a new coordinate system
- The average projection distance is minimized



Visualization

Scatter plots

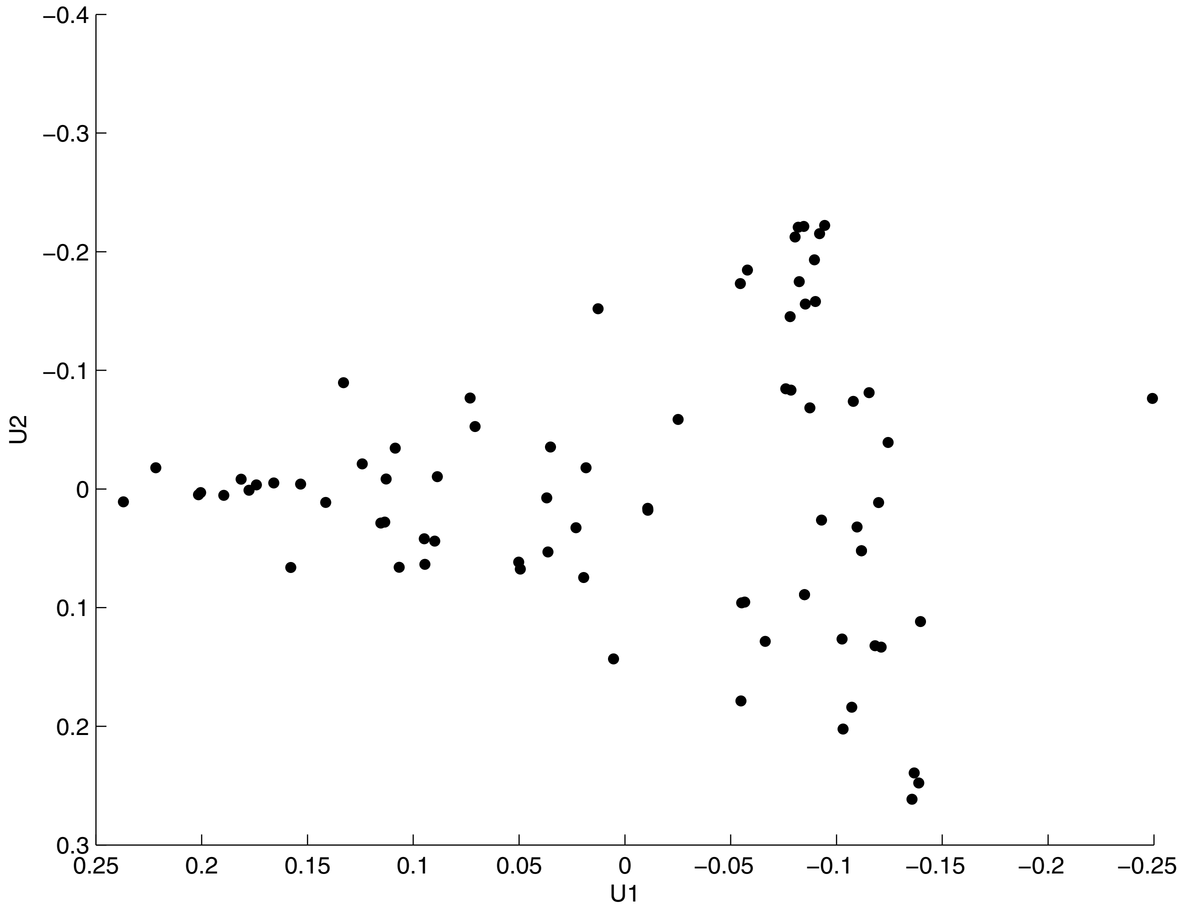
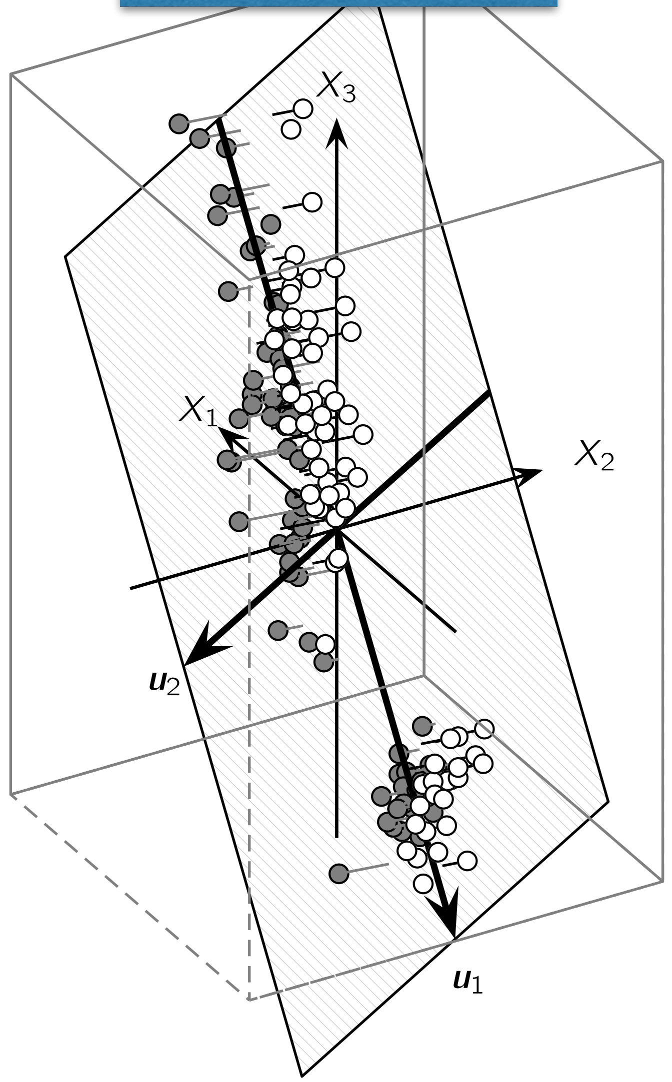


Figure 3.2. *The first two factors for a dataset ranking wines.*

2D or 3D KLT



Latent Semantic Analysis & Indexing

- **Latent semantic analysis** (LSA) is a **latent topic model**
  - Documents-by-terms matrix $\boldsymbol{A}$
    - Typically normalized (e.g. tf/idf)
- Goal is to find the “topics” doing SVD
  - $\boldsymbol{U}$ associates documents to topics
  - $\boldsymbol{V}$ associates topics to terms
- Queries can be answered by projecting the query vector $\boldsymbol{q}$ to $\boldsymbol{q}' = \boldsymbol{q}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}$ and returning rows of $\boldsymbol{U}$ that are similar to $\boldsymbol{q}'$

And many more…

- Determining the rank, finding the least-squares solution, recommending the movies, ordering results of queries, …

Computing the SVD

Golub & Van Loan chapters 5.1, 5.4.8, and 8.6

Very general idea

- SVD is unique
  - If $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal s.t. $\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V} = \boldsymbol{\Sigma}$, then $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{A}$
- Idea: find orthogonal $\boldsymbol{U}$ and $\boldsymbol{V}$ s.t. $\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}$ is as desired
  - Iterative process: find orthogonal $\boldsymbol{U}_1$, $\boldsymbol{U}_2$, … and set $\boldsymbol{U} = \boldsymbol{U}_1\boldsymbol{U}_2\boldsymbol{U}_3\cdots$
    - Still orthogonal

First attempt

- Recall: $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{AA}^T$ and σ_i^2 are the associated eigenvalues
- Idea: Compute the eigenvectors and values of $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$ to get the SVD of $\boldsymbol{A}$
  - Not the most optimal idea because it requires $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$
- We need a way to build orthogonal matrices that make matrices more diagonal

Rotations and reflections

2D **rotation**

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Rotates counterclockwise through an angle θ

2D **reflection**

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}$$

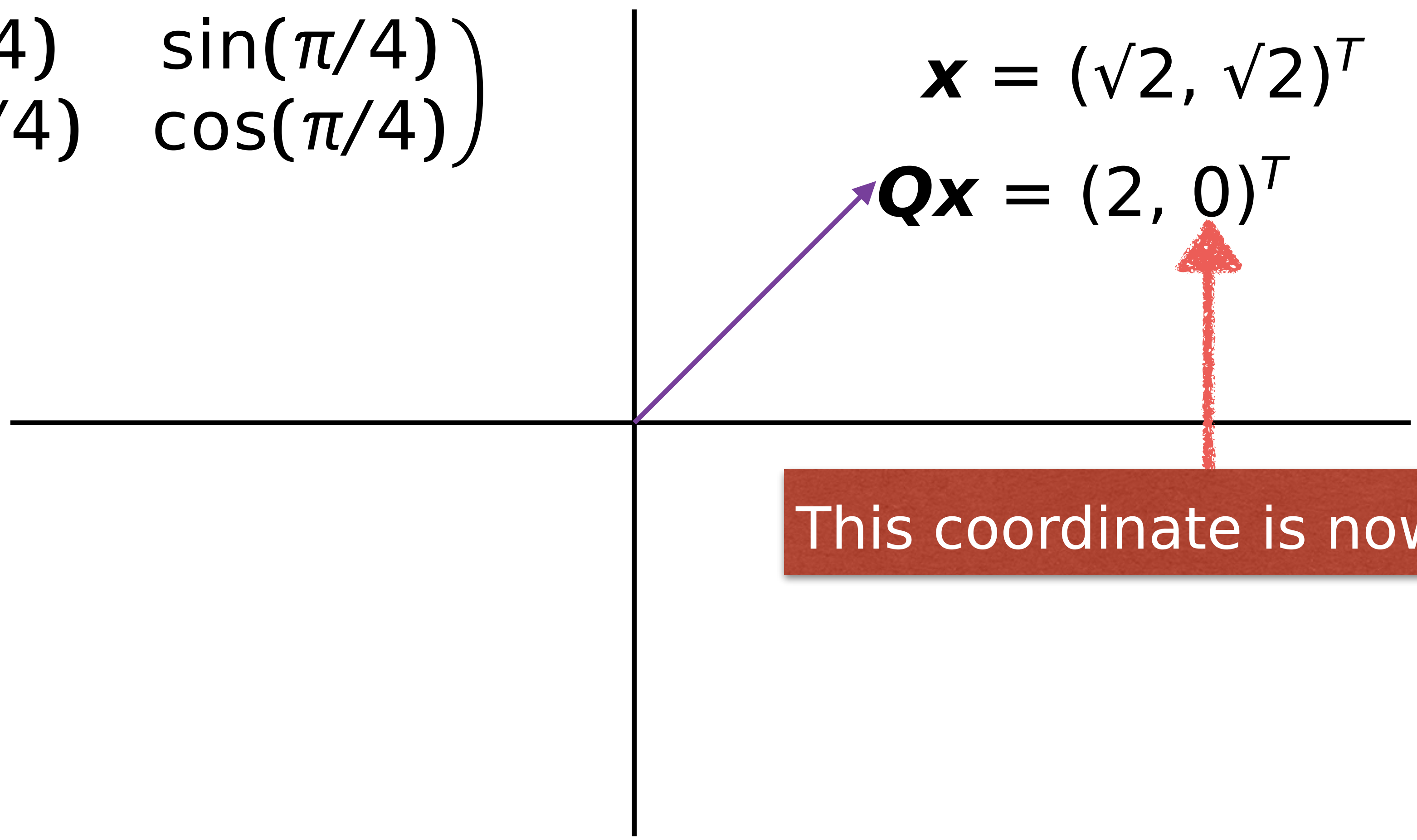
Reflects across the line spanned by $(\cos(\theta/2), \sin(\theta/2))^T$

Example

$$\boldsymbol{Q} = \begin{pmatrix} \cos(\pi/4) & \sin(\pi/4) \\ -\sin(\pi/4) & \cos(\pi/4) \end{pmatrix}$$

$\boldsymbol{x} = (\sqrt{2}, \sqrt{2})^T$

$\boldsymbol{Qx} = (2, 0)^T$



This coordinate is now 0!

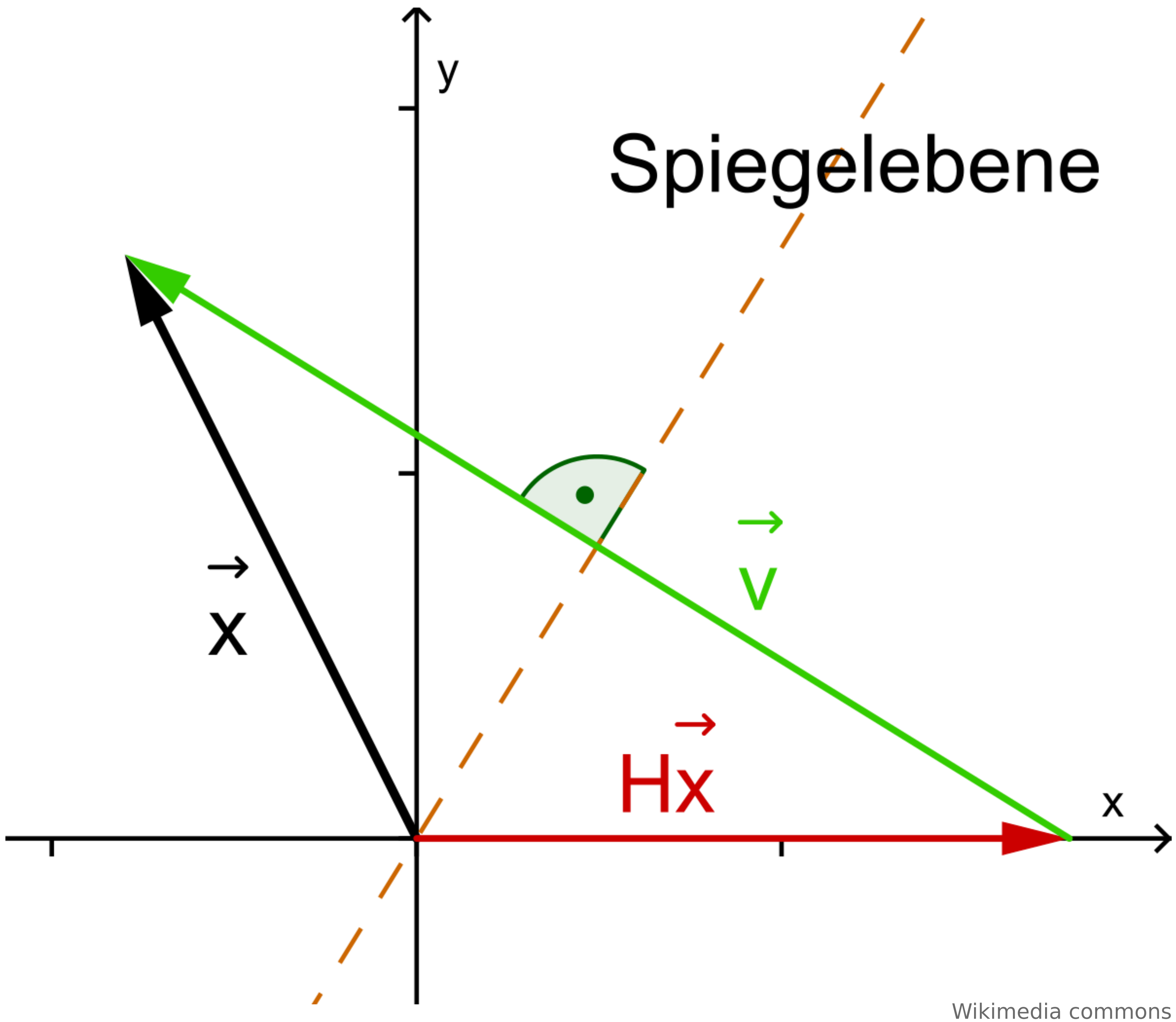
Householder reflections

- A **Householder reflection** is n-by-n matrix

$$\boldsymbol{P} = \boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^T \quad \text{where} \quad \beta = \frac{2}{\boldsymbol{v}^T\boldsymbol{v}}$$

- If we set $\boldsymbol{v} = \boldsymbol{x} - ||\boldsymbol{x}||_2\boldsymbol{e}_1$, then $\boldsymbol{P}\boldsymbol{x} = ||\boldsymbol{x}||_2\boldsymbol{e}_1$
  - $\boldsymbol{e}_1 = (1, 0, 0, \ldots, 0)^T$
- Note: $\boldsymbol{PA} = \boldsymbol{A} - (\beta\boldsymbol{v})(\boldsymbol{v}^T\boldsymbol{A})$ where $\beta = 2/(\boldsymbol{v}^T\boldsymbol{v})$
  - We never have to compute matrix $\boldsymbol{P}$

Example



Wikimedia commons

Almost there: bidiagonalization

- Given n-by-m ($n \geq m$) $\boldsymbol{A}$, we can **bidiagonalize** it with Householder transformations
  - Fix $\boldsymbol{A}[1{:}n,1]$, $\boldsymbol{A}[1,2{:}m]$, $\boldsymbol{A}[2{:}n,2]$, $\boldsymbol{A}[2,3{:}m]$, A[3:n,3], A[3,4:m]…
- The results has non-zeros in main diagonal and the one above it

Example

$$
\boldsymbol{U}_4^T\boldsymbol{U}_3^T\boldsymbol{U}_2^T\boldsymbol{U}_1^T\boldsymbol{A}\boldsymbol{V}_1\boldsymbol{V}_2 = \begin{pmatrix} * & * & 0 & 0 \\ 0 & * & * & 0 \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \end{pmatrix}
$$

Givens rotations

- Householder is too crude to give identity
- **Givens rotations** are rank-2 corrections to the identity of form

$$
\boldsymbol{G}(i,k,\theta) = \begin{array}{cc}
\begin{pmatrix}
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & & \vdots & & \vdots \\
0 & \cdots & \cos(\theta) & \cdots & \sin(\theta) & \cdots & 0 \\
\vdots & & \vdots & \ddots & \vdots & & \vdots \\
0 & \cdots & -\sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\
\vdots & & \vdots & & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}
&
\begin{array}{c}
\\ \\ i \\ \\ k \\ \\ \\
\end{array}
\\
\begin{array}{ccccccc}
\quad & \quad & i & \qquad & k & \quad & \quad
\end{array}
&
\end{array}
$$

Applying Givens

- Set θ s.t.

  $\cos(\theta) = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}$ and $\sin(\theta) = \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}$

- Now

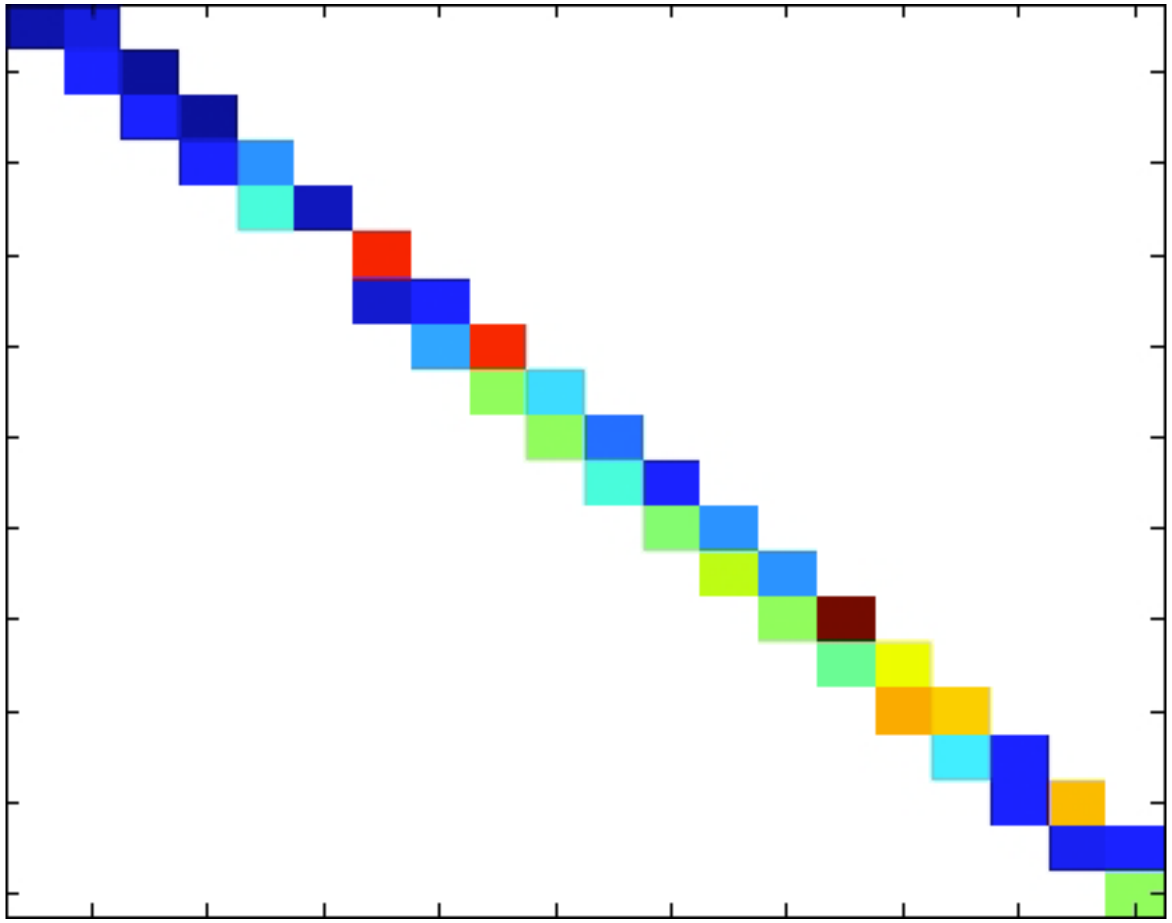
$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}^T \begin{pmatrix} x_i \\ x_k \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

- N.B. $\boldsymbol{G}(i, k, \theta)^T\boldsymbol{A}$ only affects to the 2 rows $\boldsymbol{A}[\mathrm{c}(i, k),]$
  - Also, no inverse trig. operations are needed

Givens in SVD

- We use Givens transformations to erase the superdiagonal
  - Consider principal 2-by-2 submatrices $\boldsymbol{A}[k{:}k+1,k{:}k+1]$
  - Rotations can introduce unwanted non-zeros to $\boldsymbol{A}[k+2,k]$ (or $\boldsymbol{A}[k,k+2]$)
    - Fix them in the next sub-matrix

Example



Putting it all together

1. Compute the bidiagonal matrix $\boldsymbol{B}$ from $\boldsymbol{A}$ using Householder transformations
2. Apply the Givens rotations to $\boldsymbol{B}$ until it is fully diagonal
3. Collect the required results

Time complexity

Output	Time
$\boldsymbol{\Sigma}$	$4nm^2 - 4m^3/3$
$\boldsymbol{\Sigma}, \boldsymbol{V}$	$4nm^2 + 8m^3$
$\boldsymbol{\Sigma}, \boldsymbol{U}$	$4n^2m - 8nm^2$
$\boldsymbol{\Sigma}, \boldsymbol{U}_1$	$14nm^2 - 2m^3$
$\boldsymbol{\Sigma}, \boldsymbol{U}, \boldsymbol{V}$	$4n^2m + 8nm^2 + 9m^3$
$\boldsymbol{\Sigma}, \boldsymbol{U}_1, \boldsymbol{V}$	$14nm^2 + 8m^3$

Summary of computing SVD

- Rotations and reflections allow us to selectively zero elements of a matrix with orthogonal transformations
  - Used in many, many decompositions
- Fast and accurate results require careful implementations
- Other techniques are faster for truncated SVD in large, sparse matrices

Summary of SVD

- Truly the workhorse of numerical linear algebra
  - Many useful theoretical properties
    - Rank-revealing, pseudo-inverses, scalar norm computation, …
  - Reasonably easy to compute
- But it also has some major shortcomings in data analysis… *stay tuned!*