Introduction

Linear Algebra

Department of Computer Engineering

Sharif University of Technology

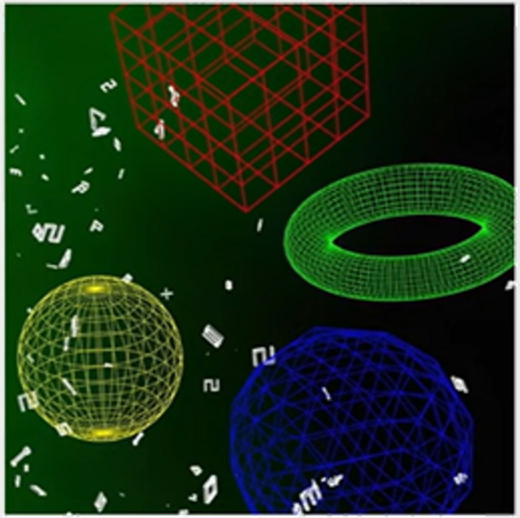
Hamid R. Rabiee rabiee@sharif.edu

Maryam Ramezani maryam.ramezani@sharif.edu

Famous Topics of Linear Algebra in ML

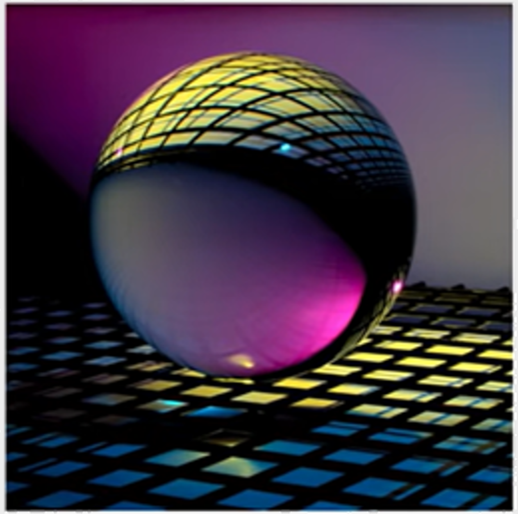
Data Representations

Use basic ideas of linear algebra to represent data in a way that computers can understand: vectors.



Vector Embeddings

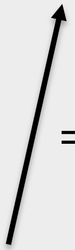
Learn ways to choose these representations wisely via matrix factorizations.



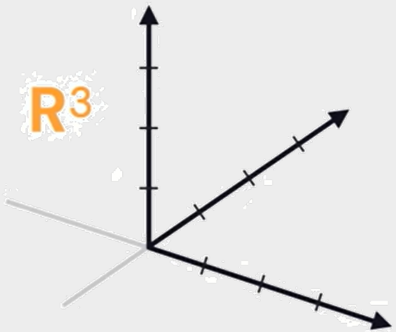
Dimensionality Reduction

Deal with large-dimensional data using linear maps and their eigenvectors and eigenvalues.

- How can we represent data (images, text, user preferences, etc.) in a way that computers can understand?
  - Organize information into a vector!

$V =$  $= \begin{bmatrix} -3 \\ 0.7 \\ 2 \end{bmatrix}$

- A **vector** is a 1-dimensional array of numbers.
  - It has both a magnitude (length) and a direction
- The totality of a vectors with n entries is an **n-dimensional vector space**.

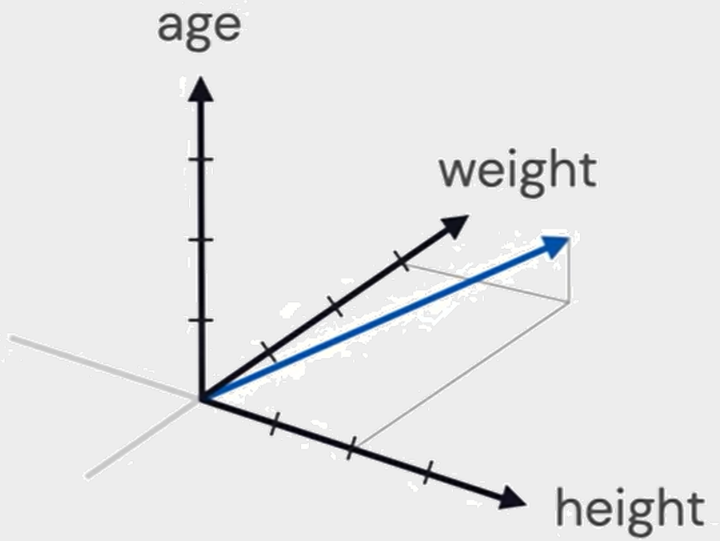


"3-dimensional space" consists of all vectors with 3 entries:

$$\begin{bmatrix} * \\ * \\ * \end{bmatrix}$$

- A feature vector is a vector whose entries represent the "features" of an object.
- The vector space containing them is called feature space.

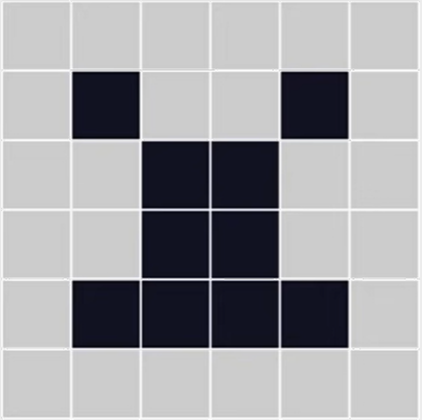
$$P = \begin{array}{c} \begin{bmatrix} 64 \\ 131 \\ 24 \end{bmatrix} \end{array} \begin{array}{l} \text{height} \\ \text{weight} \\ \text{age} \end{array}$$



Data Representation Applications

Image

- ❑ In black and white images, black and white pixels correspond to 0s and 1s.



$$\rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

- ❑ In grayscale pixels are numbers between 0 and 255.

Words and Documents

- Given a collection of documents (e.g. Wikipedia articles), assign to every word a vector whose i^{th} entry is the number of times the word appears in the i^{th} document.
- These vectors can be assemble into a large matrix, useful for latent semantic analytics.

$$\text{dog} = \begin{bmatrix} 0 \\ 7 \\ 0 \\ 0 \\ 51 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \text{Wiki \#1} \\ \text{Wiki \#2} \\ \text{Wiki \#3} \\ \text{Wiki \#4} \\ \text{Wiki \#5} \\ \\ \text{Wiki \#54,000,000} \end{array}$$

Latent Semantic Analysis

- In the sub-field of machine learning for working with text data called natural language processing (NLP), it is common to represent documents as large matrices of word occurrences.

	Quick	Brown	Fox	Jumps	Over	Lazy	Dog
The quick brown fox jumps over the lazy dog	1	1	1	1	1	1	1
If the fox is quick he can jump over the dog.	1	0	1	0	1	0	1
Foxes are quick. Dogs are lazy.	0	1	1	0	0	1	1
Can a fox jump over a dog?	0	0	1	1	1	0	1

- Matrix factorization methods, such as the singular-value decomposition can be applied to this sparse matrix. Documents processed in this way are much easier to compare, query, and use as the basis for a supervised machine learning model.

Yes/No or Ratings

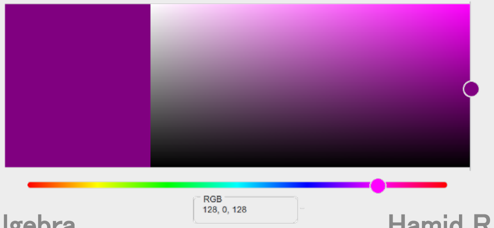
- Given users and items (e.g. movies), vectors can indicate if a user has interacted with the item (yes=1, no=0).
- User's rating a number between 0 and 5.

$$\text{user1} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} \text{No} \\ \text{Yes} \\ \text{No} \\ \text{No} \\ \\ \text{Yes} \\ \text{No} \end{array} \qquad \text{user2} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 3 \\ \vdots \\ 0 \\ 2 \end{bmatrix} \begin{array}{l} ? \\ \text{Love} \\ ? \\ \text{Like} \\ \\ ? \\ \text{Dislike} \end{array}$$

- Sometimes you work with categorical data in machine learning.
- It is common to encode categorical variables to make them easier to work with and learn by some techniques. A popular encoding for categorical variables is the one hot encoding.
- A one hot encoding is:

Example

$$\text{red}=\begin{bmatrix}1\\0\\0\end{bmatrix} \qquad \text{green}=\begin{bmatrix}0\\1\\0\end{bmatrix} \qquad \text{blue}=\begin{bmatrix}0\\0\\1\end{bmatrix}$$



- One-Hot Encodings (standard basis vector)
  - Assign to each word a vector with one 1 and 0s elsewhere.
  - Suppose our language only has four words:

$$\text{apple} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad \text{cat} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad \text{house} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad \text{tiger} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Drawbacks

- Very sparse vectors.
- Are never **similar**!

Dot Product

- The product of numbers is another number.
- The dot product of vectors is not another vector! It is a number!!

$$2 \times 5 = 10$$

Numbers

VS

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 5 \end{bmatrix} = 7$$

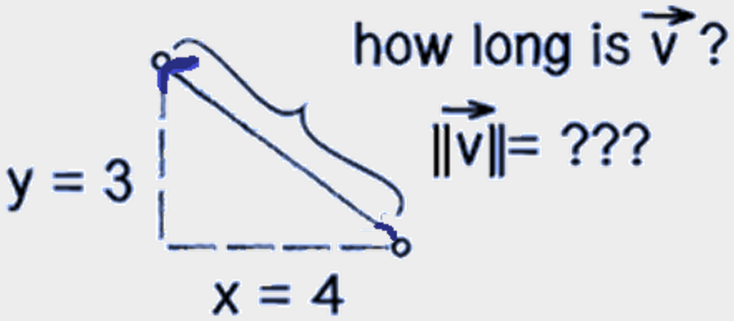
Vectors

A number

$$\begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 7 \\ 2 \\ -1 \end{bmatrix} = (1)(7) + (0)(2) + (3)(-1) = 4$$

Length of vector

- Dot product between a vector and itself: magnitude-squared, the **length** squared, or the squared-norm, of the vector.



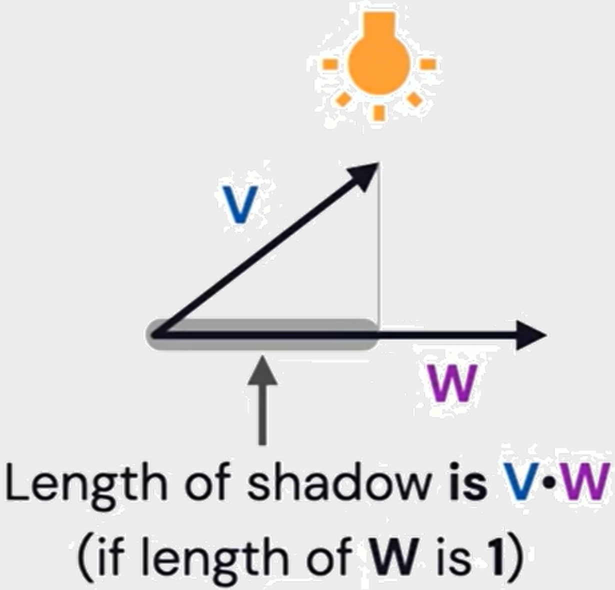
$$\mathrm{v.v} = \begin{bmatrix}4\\3\end{bmatrix}.\begin{bmatrix}4\\3\end{bmatrix}=16+9=25$$

Length(v)=5

$$a^T a = \|a\|^2 = \sum_{i=1}^{n} a_i a_i = \sum_{i=1}^{n} a_i^2$$

Dot Product (Geometric Interpretation and Intuition)

- Represents the length of the "shadow" of one vector along another.
- This indicates how similar the two vectors are.



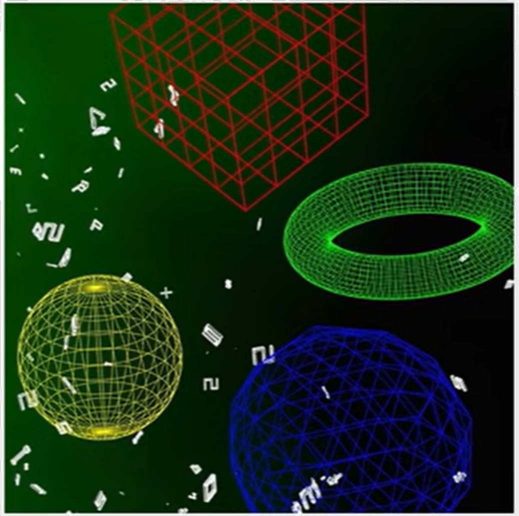
One-Hot Encodings Drawbacks

$$\text{apple} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{cat} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{house} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \text{tiger} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

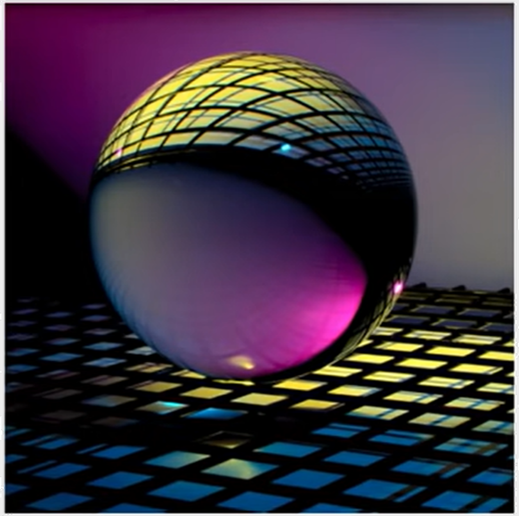
$$\text{apple} \cdot \text{cat} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \text{tiger} \cdot \text{cat}$$

Famous Topics of Linear Algebra in ML

Data Representations
Use basic ideas of linear algebra to represent data in a way that computers can understand: vectors.



Vector Embeddings
Learn ways to choose these representations wisely via matrix factorizations.



Dimensionality Reduction
Deal with large-dimensional data using linear maps and their eigenvectors and eigenvalues.

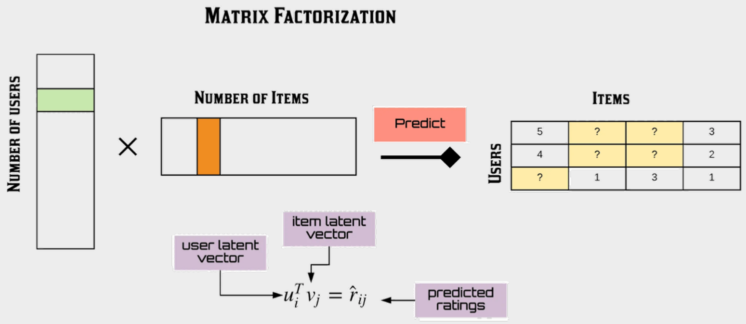
Vector Embeddings (Linear Algebra)

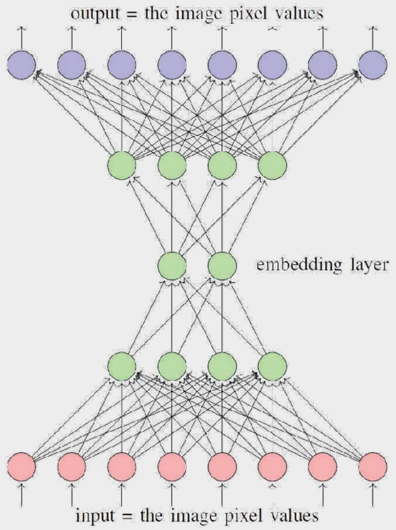
- An embedding of a vector is another vector in a smaller dimensional space.

Replace $\begin{bmatrix} * \\ * \\ * \end{bmatrix}$ with $\begin{bmatrix} * \\ * \end{bmatrix}$

Vector Embeddings (Machine Learning)

- Matrix Factorization
- Neural Networks





What is Matrix?

- A **matrix** is a 2–dimensional array of numbers.
- **Matrix is a linear transformation**
  - It represents a particular process of turning one vector into another: stretching, rotating, scaling or something more complex.

$$\begin{bmatrix} 3 & 0 & 7 \\ 1 & -5 & 9 \\ 2 & 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{input}} = \underbrace{\begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}}_{\text{output}}$$

Image Rotation

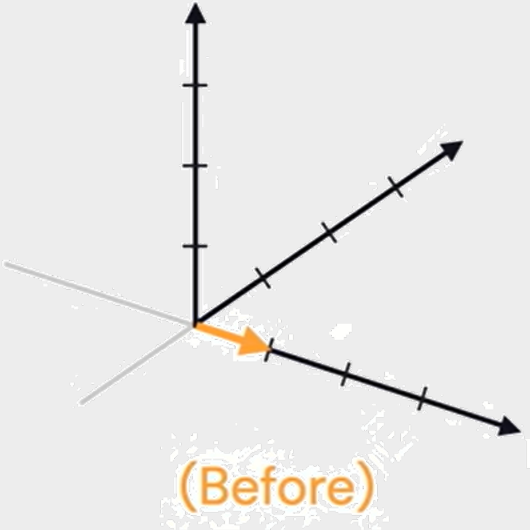
(a) (b)

Image Scaling

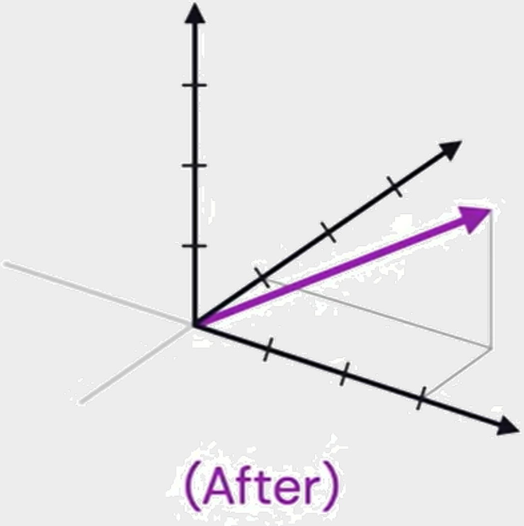
(a) (b)

What is Matrix?

- A **matrix** represents a **transformation** of an entire vector space to another (possibly of different dimensions)



$$\begin{bmatrix} 3 & 0 & 7 \\ 1 & -5 & 9 \\ 2 & 0 & 0 \end{bmatrix}$$



Matrix Factorization

- We can multiply **numbers** and get **number**.
- We can multiply **vectors** by dot product and get **number**.
- We can multiply **matrices** and get a **matrix.**

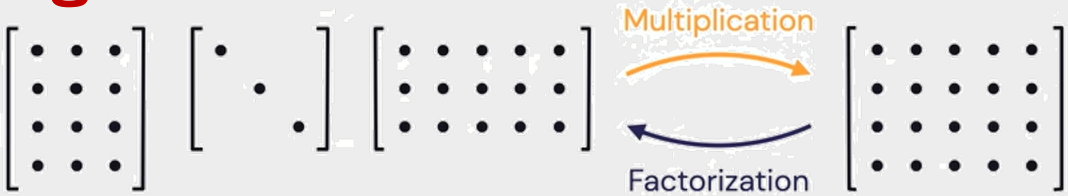
$$\underbrace{\begin{bmatrix} 3 & 0 & 7 \\ 1 & -5 & 9 \\ 2 & 0 & 0 \end{bmatrix}}_{3 \times 3} \underbrace{\begin{bmatrix} 0 & 2 \\ 1 & 0 \\ -1 & 0 \end{bmatrix}}_{3 \times 2} = \underbrace{\begin{bmatrix} -7 & 6 \\ -14 & 2 \\ 0 & 4 \end{bmatrix}}_{3 \times 2}$$

Sizes must match

- Factorization?

$$12 \times 3 \times 15 \underset{\text{Factorization}}{\overset{\text{Multiplication}}{\rightleftarrows}} 540$$

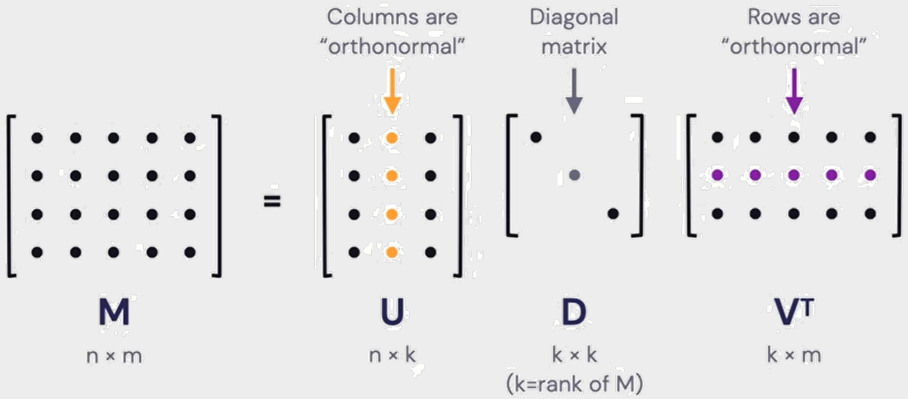
In general factorization is HARD!



- Fundamental Theorem in Linear Algebra:
  - Every matrices can be factored!

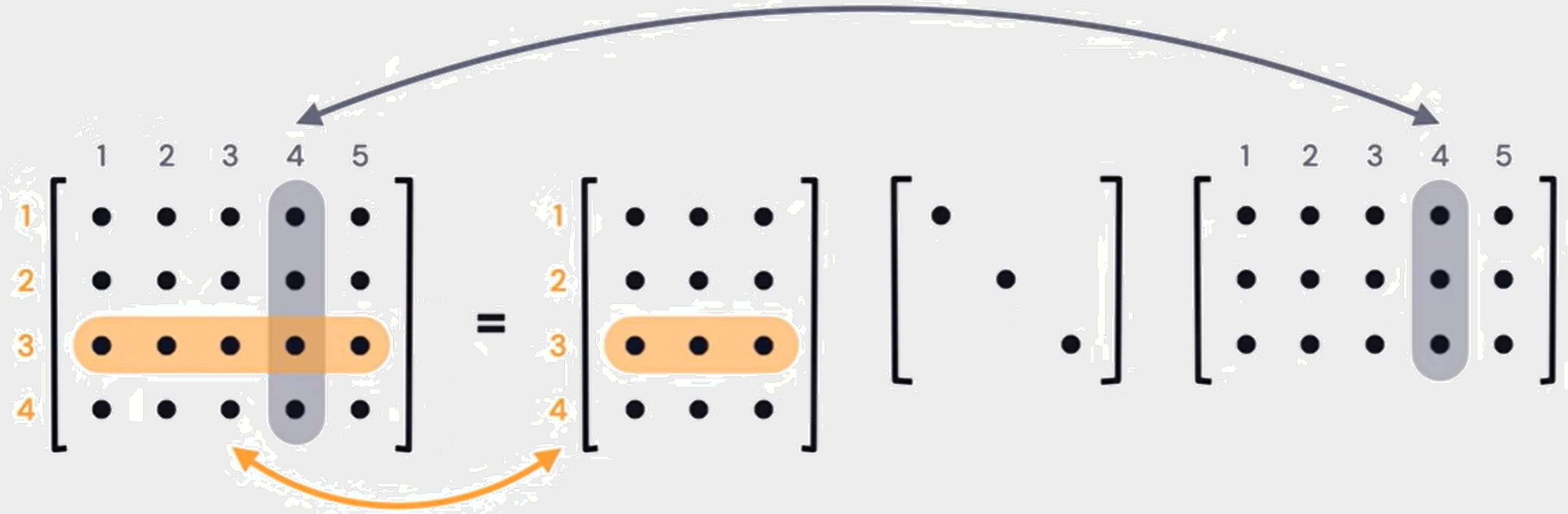
Theorem

Singular Value Decomposition (SVD)
Every $n \times m$ matrix can be written as a product of three smaller matrices as below:



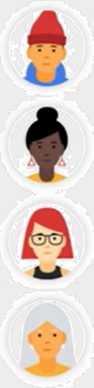
Matrix Factorization (Linear Algebra)

- It has wide use in linear algebra and can be used directly in applications such as feature selection, visualization, noise reduction, and more.
- The columns/rows of the factors are candidates for embeddings.



Vector Embedding Applications

- User – Movie Matrix
  - Checkmarks = watched movie
  - Empty cells = not watched movie

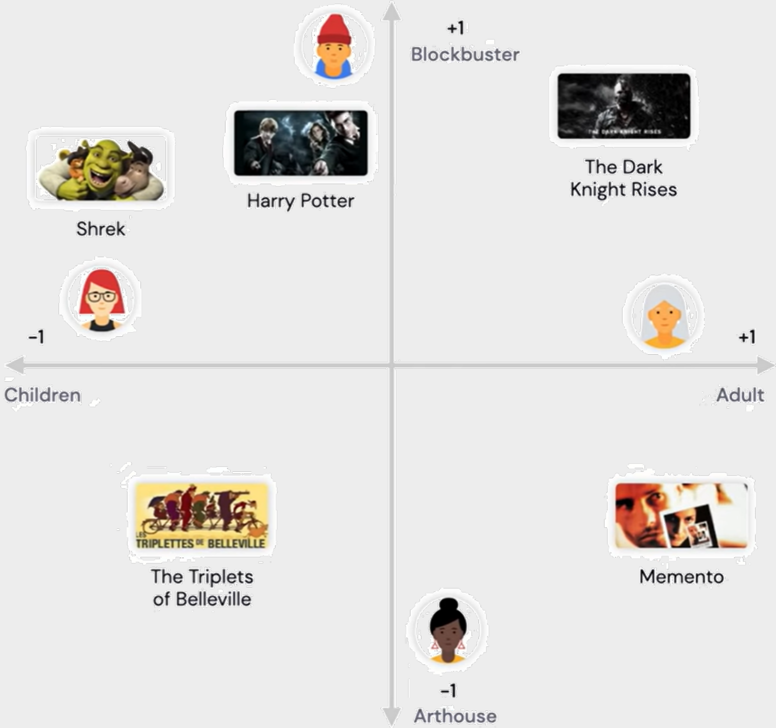


	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

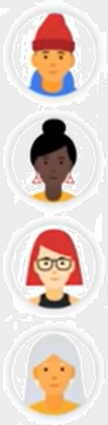
$$\text{User 3} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{Shrek} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- We dont know the features!
  - Example: 2–dimensional "latent" feature space!

- We want to find the new, smaller dimensional vector representations that capture these features.



Recommender Systems

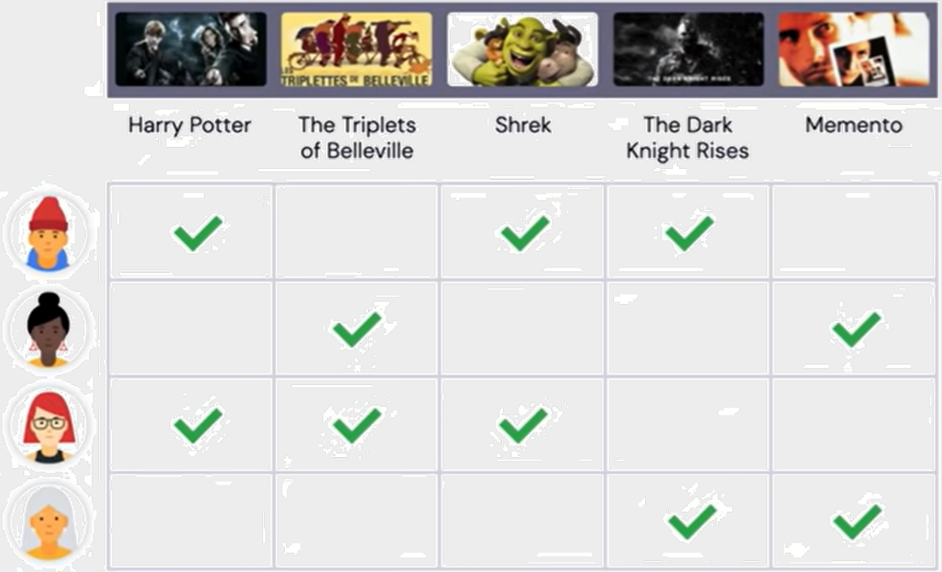


	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

$$\approx \underset{4 \times 2}{\overset{U}{\begin{bmatrix} \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{bmatrix}}} \; \overset{V^T}{\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}}$$

4×5 $\qquad$ 4×2 $\qquad$ 2×5

Recommender Systems



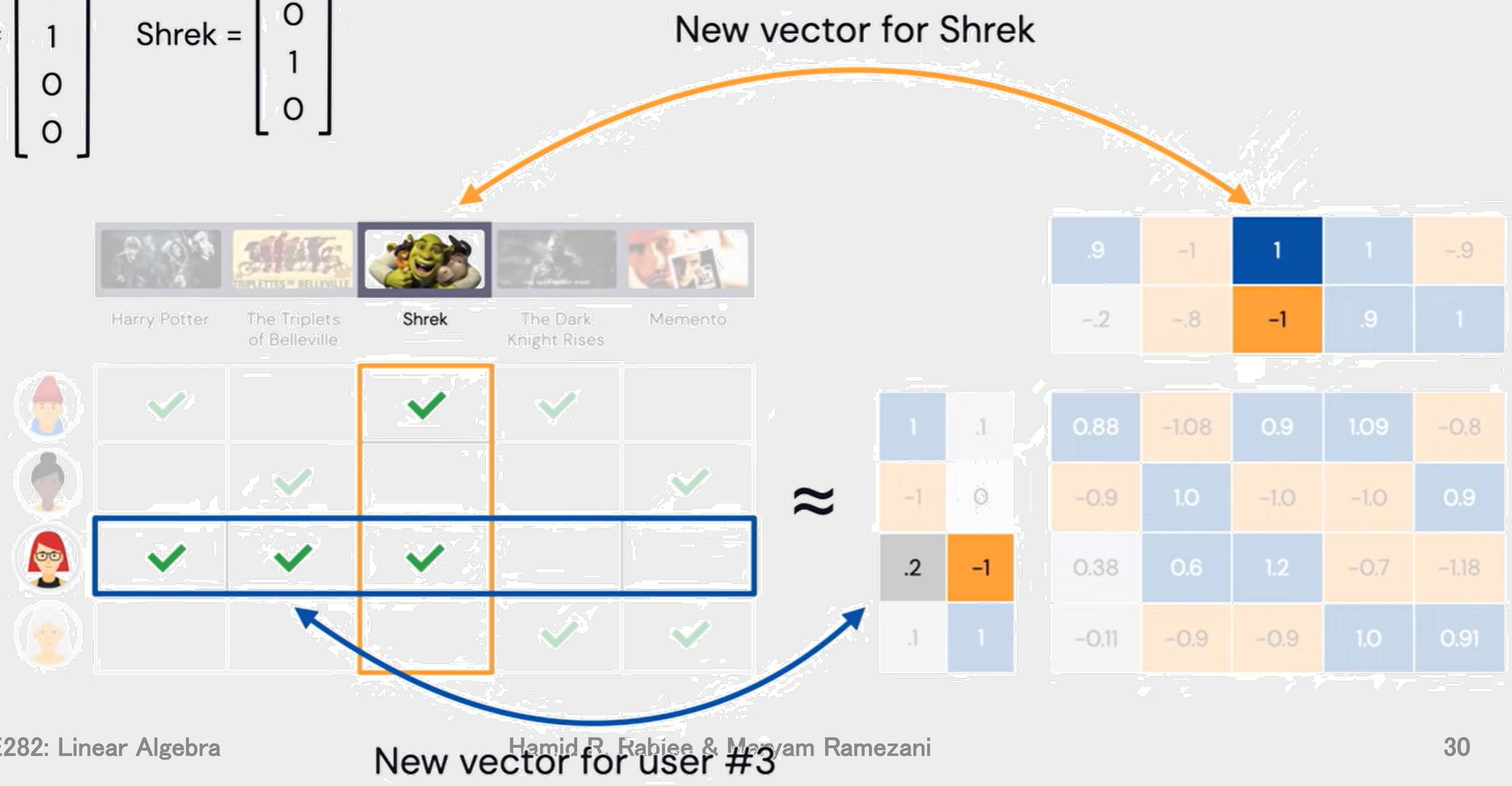
V^T is a feature × **movie** matrix

$$\overbrace{\begin{bmatrix} .9 & -1 & 1 & 1 & -.9 \\ -.2 & -.8 & -1 & .9 & 1 \end{bmatrix}}$$

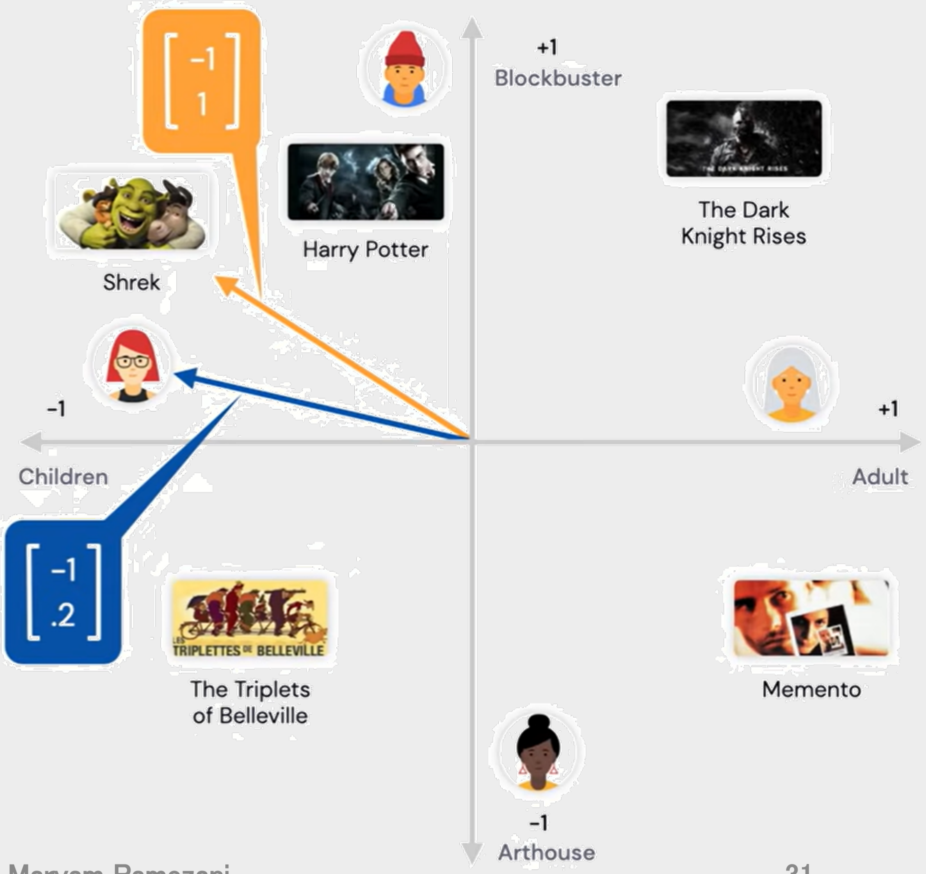
$$\approx \underbrace{\begin{bmatrix} 1 & .1 \\ -1 & 0 \\ .2 & -1 \\ .1 & 1 \end{bmatrix}} \quad \begin{bmatrix} 0.88 & -1.08 & 0.9 & 1.09 & -0.8 \\ -0.9 & 1.0 & -1.0 & -1.0 & 0.9 \\ 0.38 & 0.6 & 1.2 & -0.7 & -1.18 \\ -0.11 & -0.9 & -0.9 & 1.0 & 0.91 \end{bmatrix}$$

U is a **user** × feature matrix

$$\text{User 3} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{Shrek} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



- These two vectors are close!
- The shadow of orange vector onto blue vector is pretty large!

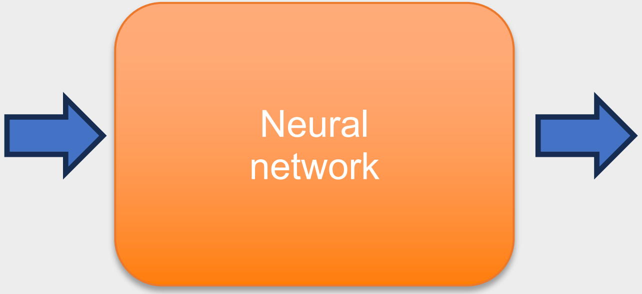


Feed data vector into a Neural network. The output is vector embedding.

Under the hood:
Matrix multiplication *plus* more.

Movies viewed by user

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$



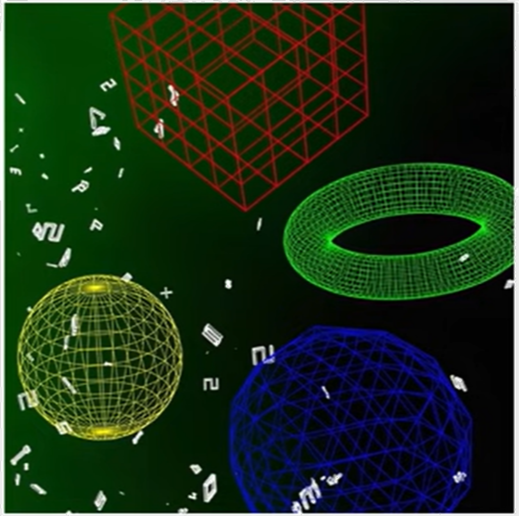
Vector embedding

$$\begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}$$

Famous Topics of Linear Algebra in ML

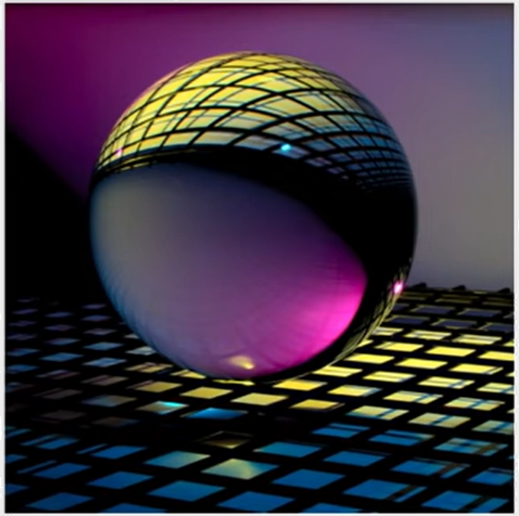
Data Representations

Use basic ideas of linear algebra to represent data in a way that computers can understand: vectors.



Vector Embeddings

Learn ways to choose these representations wisely via matrix factorizations.



Dimensionality Reduction

Deal with large-dimensional data using linear maps and their eigenvectors and eigenvalues.

Goal

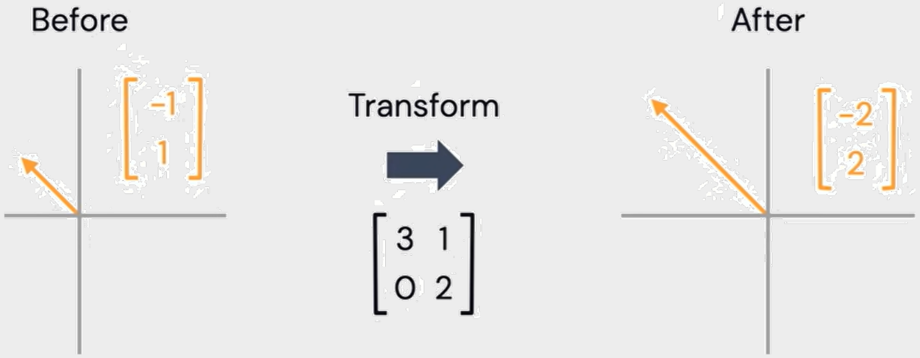
- “Compress” high-dimensional data into a smaller-dimensional, more meaningful subspace.

- This should be done in a way that doesn’t lose too much information.

Data Representations (Linear Algebra)

- Eigenvectors
- Principal components

- Matrix is a **transformation** between vector spaces
- There are some transformations for which some vectors never change direction, but are only scaled.



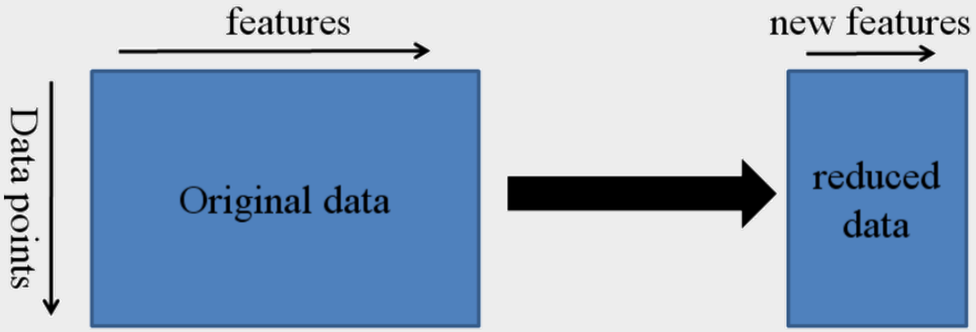
These special vectors are called eigenvectors
The scaling factor is called an eigenvalue.

$$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

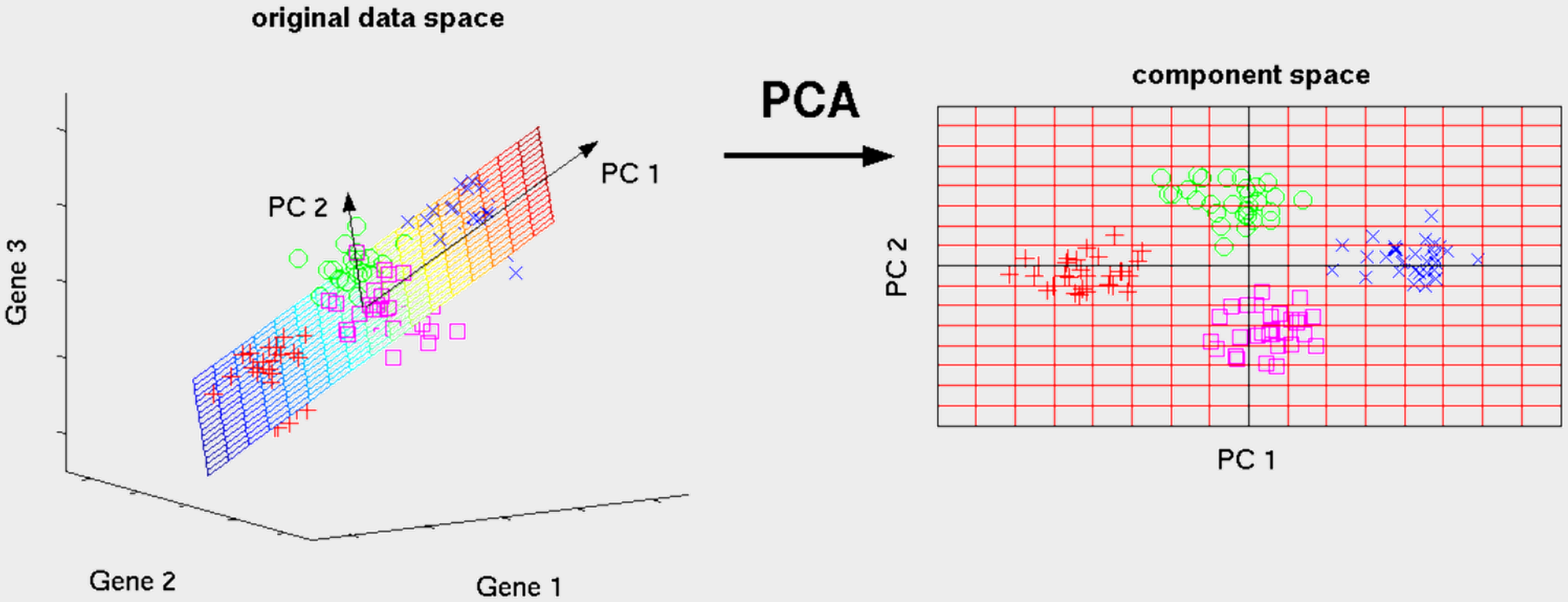
$$M \quad V \quad = \quad 2 \quad V$$

Data Representations (Machine Learning)

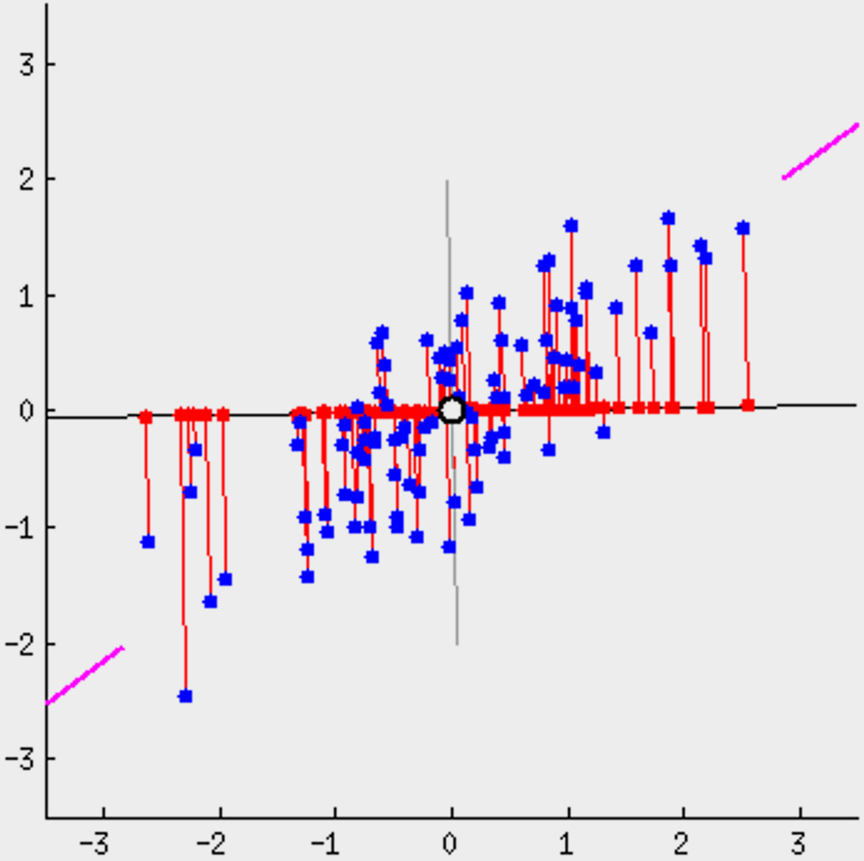
- Principal Component Analysis
  - Often, a dataset has many columns, perhaps tens, hundreds, thousands, or more.
  - Methods for automatically reducing the number of columns of a dataset are called dimensionality reduction, and perhaps the most popular method is called the principal component analysis, or PCA for short
  - The core of the PCA method is a matrix factorization method from linear algebra.



Principal Component Analysis (PCA)



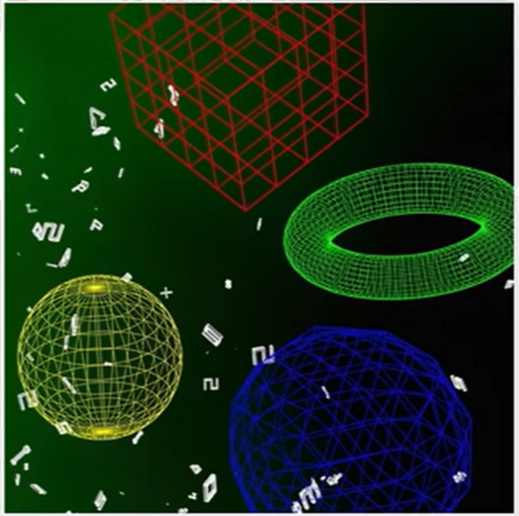
Principal Component Analysis (PCA)



Conclusion

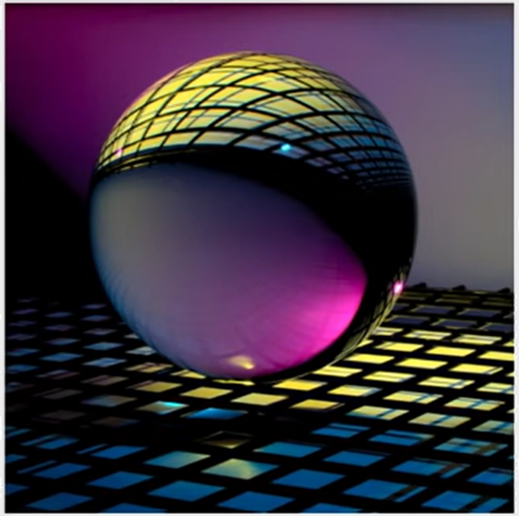
Data Representations

Use basic ideas of linear algebra to represent data in a way that computers can understand: vectors.



Vector Embeddings

Learn ways to choose these representations wisely via matrix factorizations.



Dimensionality Reduction

Deal with large-dimensional data using linear maps and their eigenvectors and eigenvalues.

References

- A friendly introduction to linear algebra for ML (ML Tech Talks) by TensorFlow
- Introduction to Applied Linear Algebra Vectors, Matrices, and Least Squares
- Linear Algebra and its applications
- Linear algebra A Modern Introduction David Poole