Applied Text Analytics &

Natural Language Processing

with Dr. Mahdi Roozbahani
& Wafa Louhichi

SVD and Co-occurrence Matrices

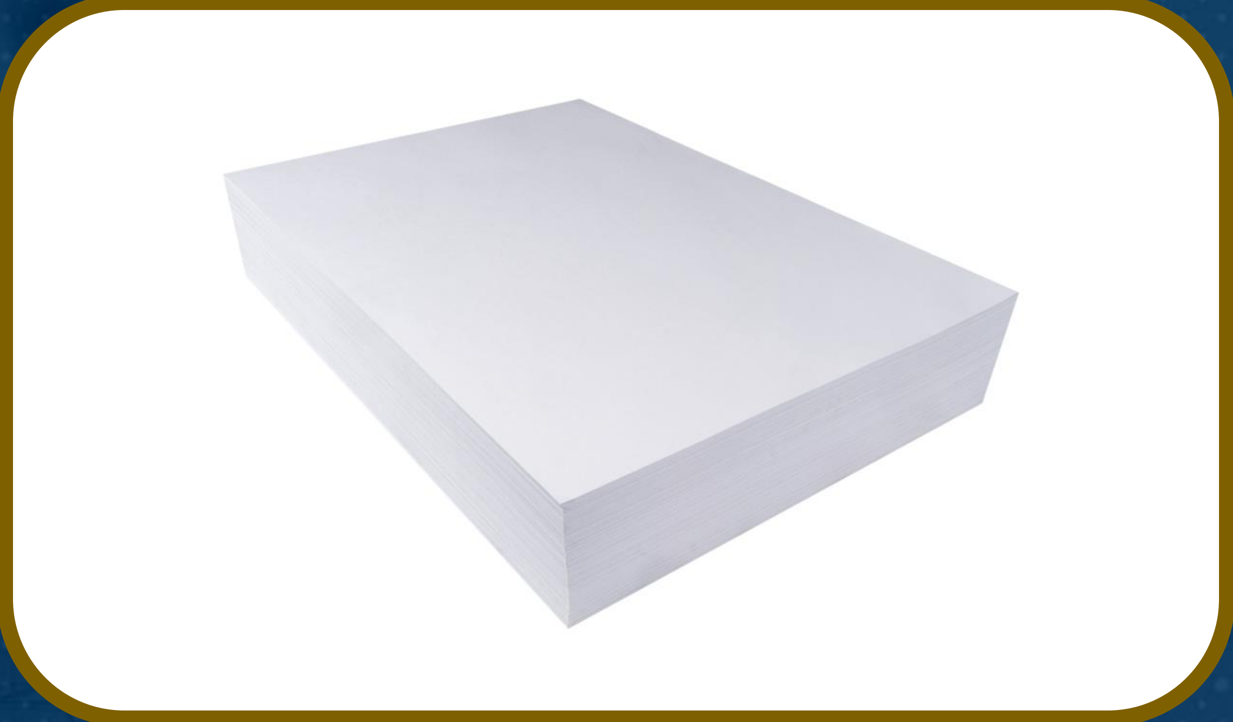
Learning Objectives

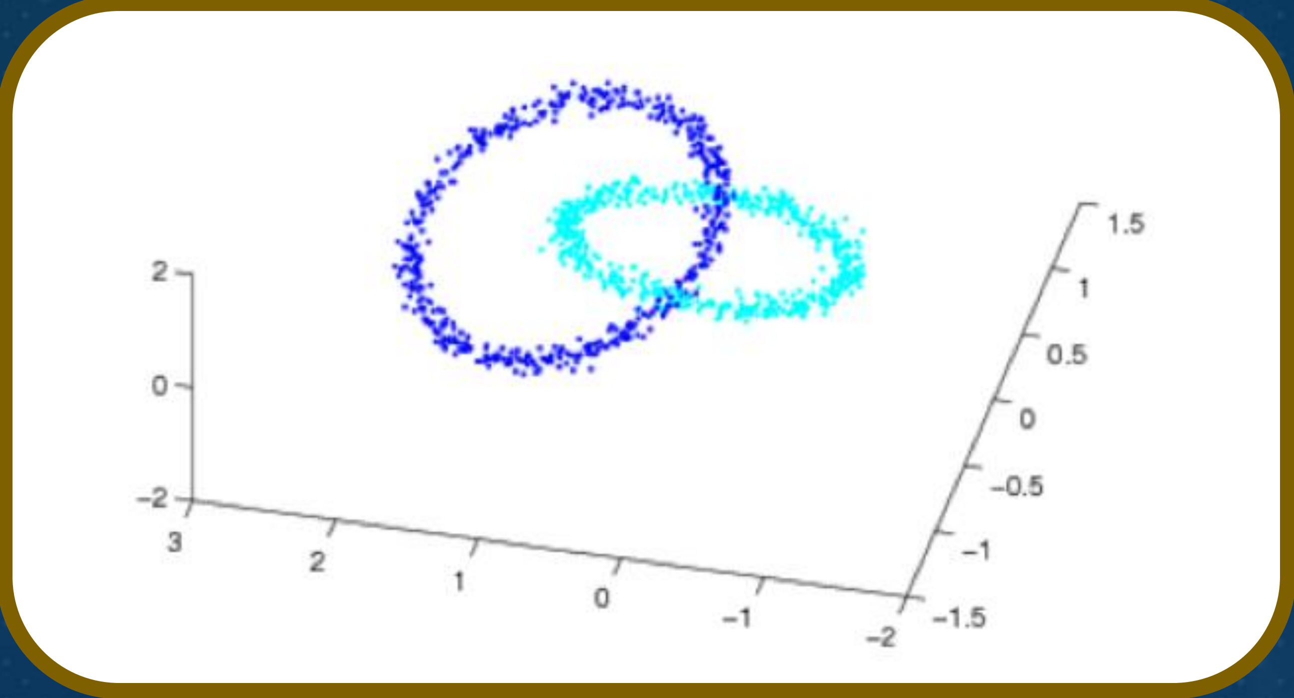
In this course, you will learn

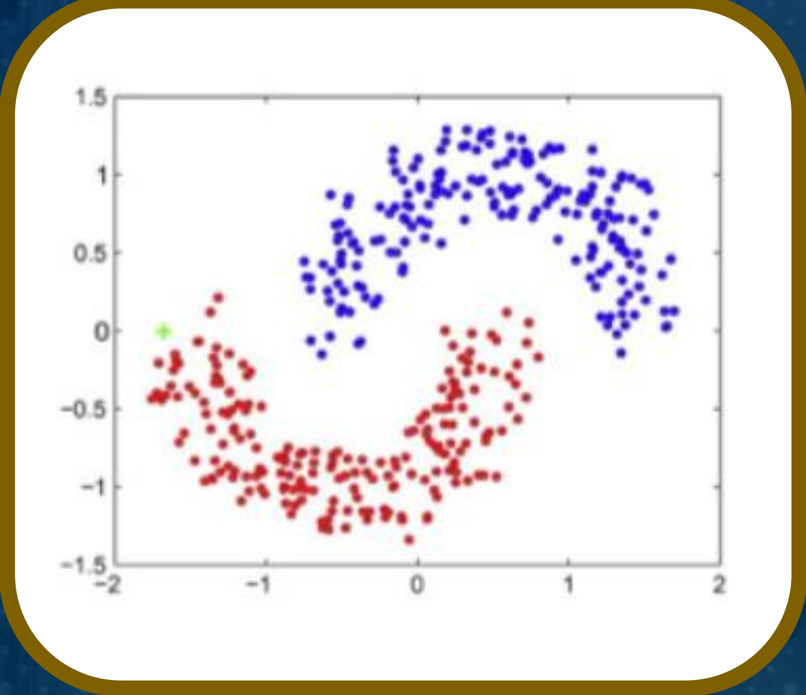
- Dimensionality reduction using Singular Value Decomposition
- Co-occurrence Matrix embeddings

Motivating Example: Dimensionality Reduction for Text

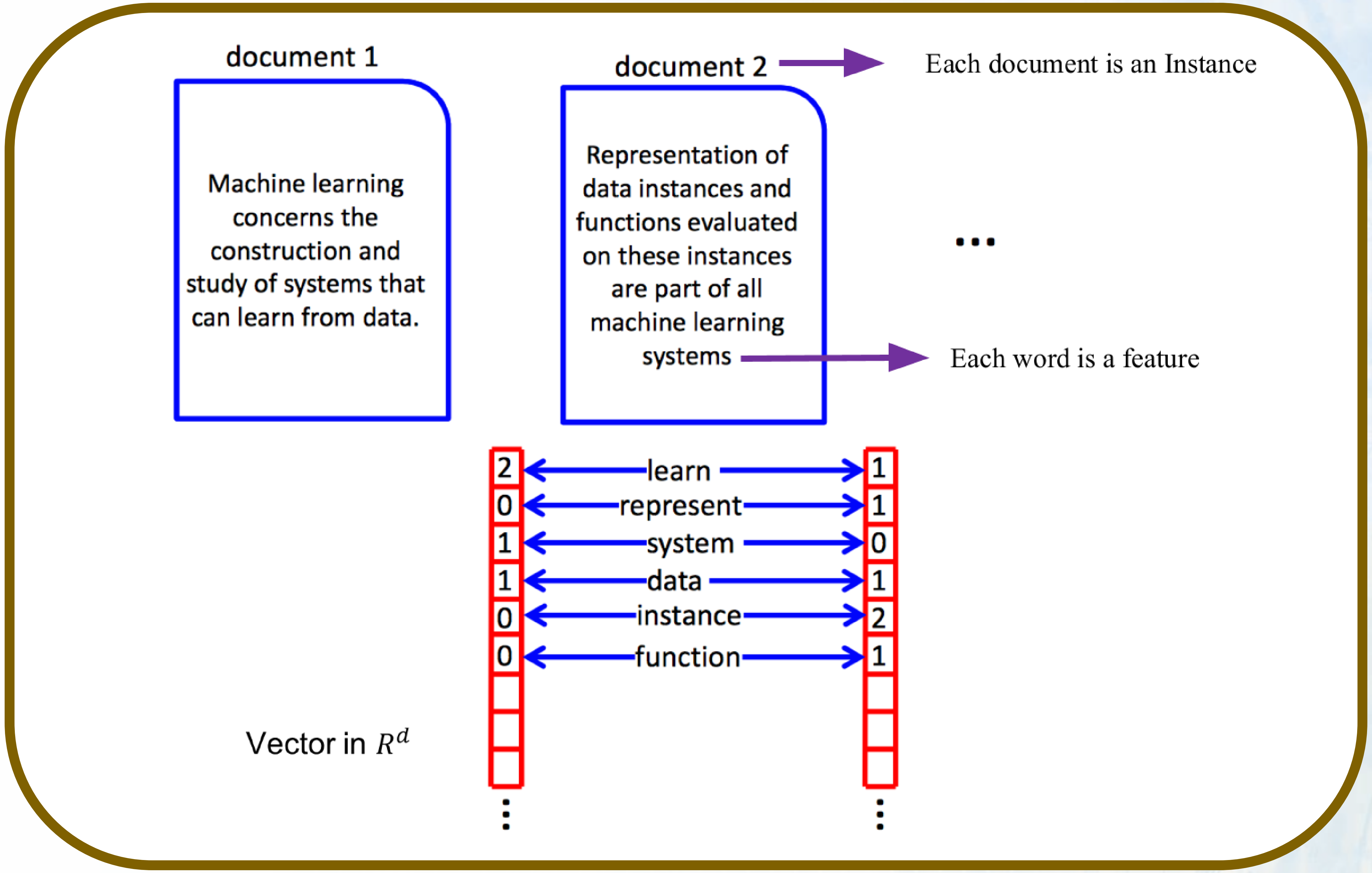
What are the relations between data points?







Motivating Example: Bag of Words Representation



Term-Document Data Matrix – Bag-of-words

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

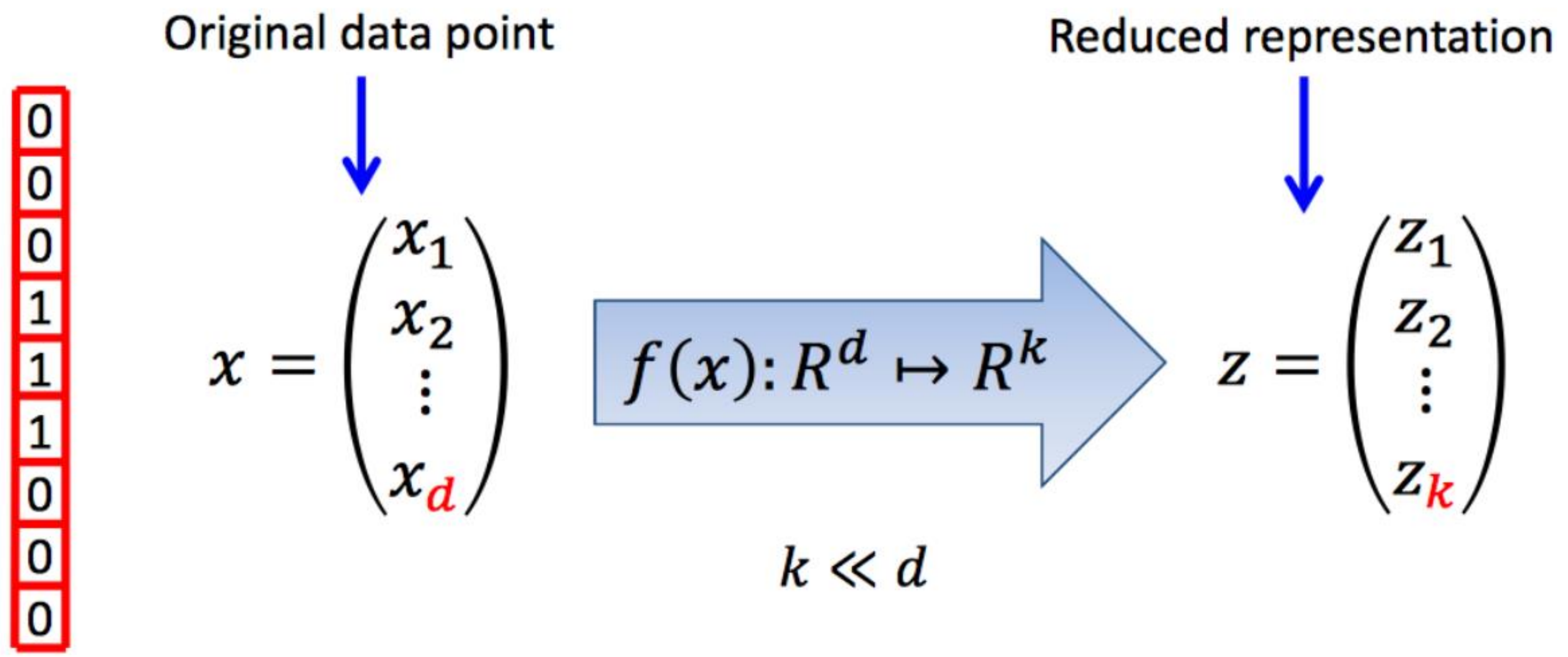
• • • Many more features

Solution:

Dimension Reduction

What is Dimensionality Reduction?

- The process of reducing random variables under consideration
    - One can combine, transform, or select variables
    - One can use linear or non linear operations



Vector in R^d

Intuition

- Approximate a D-dimensional dataset using fewer dimensions
- By first rotating the axes into a new space
- The highest order dimension captures the most variance in the original dataset
- And the next dimension captures the next most variance, etc.

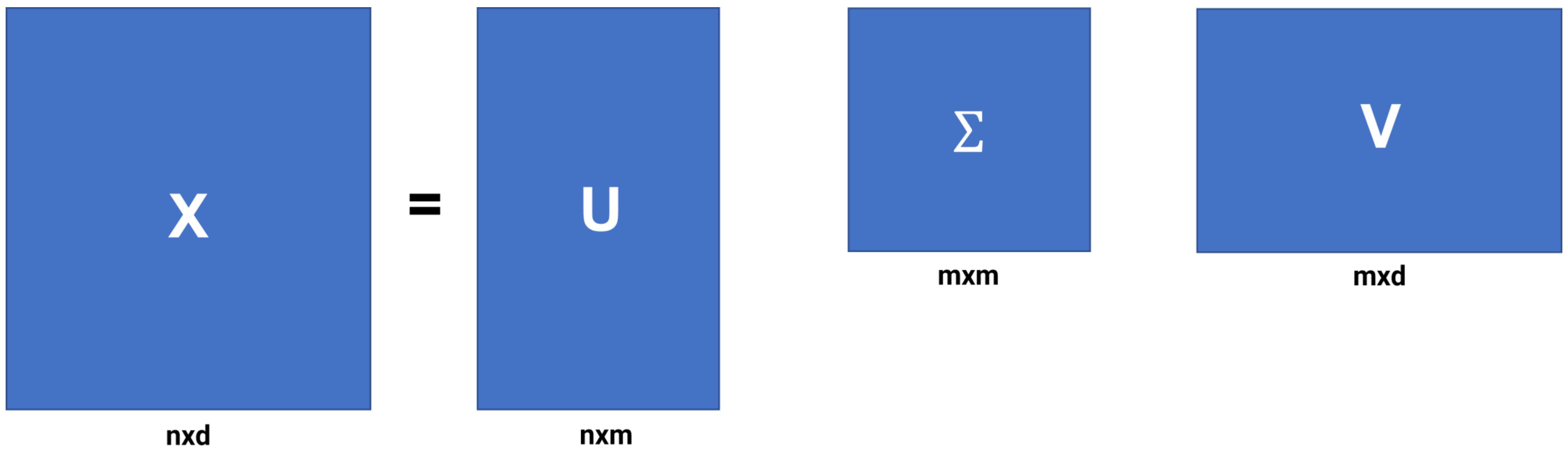
Singular Value Decomposition

- For a Matrix X_{nxd} where n is the number of instances and d is dimension:

$$X = U\Sigma V^{T}$$

Where:

- $U_{nxm} \rightarrow$ unitary matrix $\rightarrow \mathrm{UU^T} = \mathrm{I}$
- $\Sigma_{m\times m} \rightarrow diagonal\ matrix\ of\ singular\ values\ of\ X$
- $V_{m\times d} \rightarrow unitary\ matrix \rightarrow \mathrm{VV^T} = \mathrm{I}$



m columns represent a dimension in a new latent space such that m column vectors are orthogonal to each other and ordered by the amount of variance in the dataset in each dimension. m could at most have **d** dimensions.

Co-Occurrence Matrices

- The meaning of a word is defined by the words in its surroundings
- We define a context window as the number of words appearing around a center word
- We create a co-occurrence matrix as follows:
    - Step 1: Go through each central word - context pair in the corpus (context window length is commonly anything between 1 and 5)
    - Step 2: In each iteration, update in the row of the count matrix corresponding to the central word by adding +1 in the columns corresponding to the context words
    - Step 3: Repeat last 2 steps many times
    - Example: “it was the best of times, it was the worst of times” with a context window =2, the words “was”, “the”, “of” and “times” appear in the context of the central word and central word ”best” and get incremented by +1

Co-Occurrence Matrices

- The meaning of a word is defined by the words in its surroundings
- We define a context window as the number of words appearing around a center word
- We create a co-occurrence matrix as follows:
  - Go through each central word - context pair in the corpus (context window length is commonly anything between 1 and 5)
  - In each iteration, increment in the row the count corresponding to the central word in the columns corresponding to the context words
  - Repeat last 2 steps many times

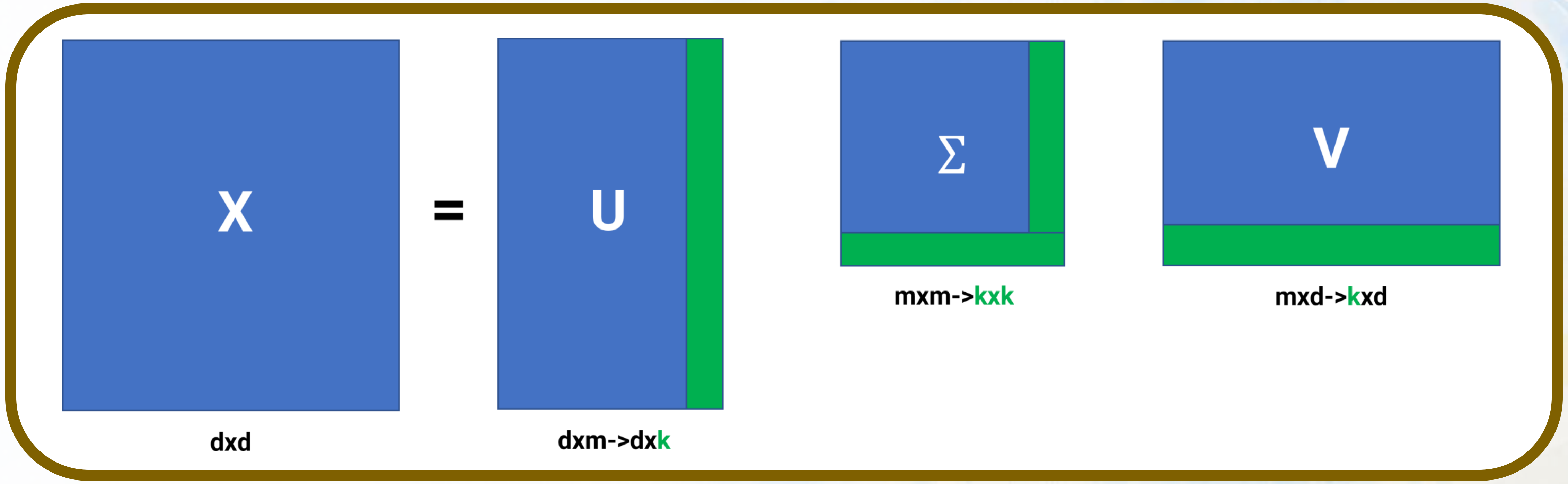
Example corpus: “it was the best of times, it was the worst of times” with a context window =2

Co-occurrence Matrix

	it	was	the	best	of	times	worst
it	0	2	2	0	1	1	0
was	2	0	2	1	0	1	1
the	2	2	0	1	2	0	1
best	0	1	1	0	1	1	0
of	1	0	2	1	0	2	1
times	1	1	0	1	2	0	1
worst	0	1	1	0	1	1	0

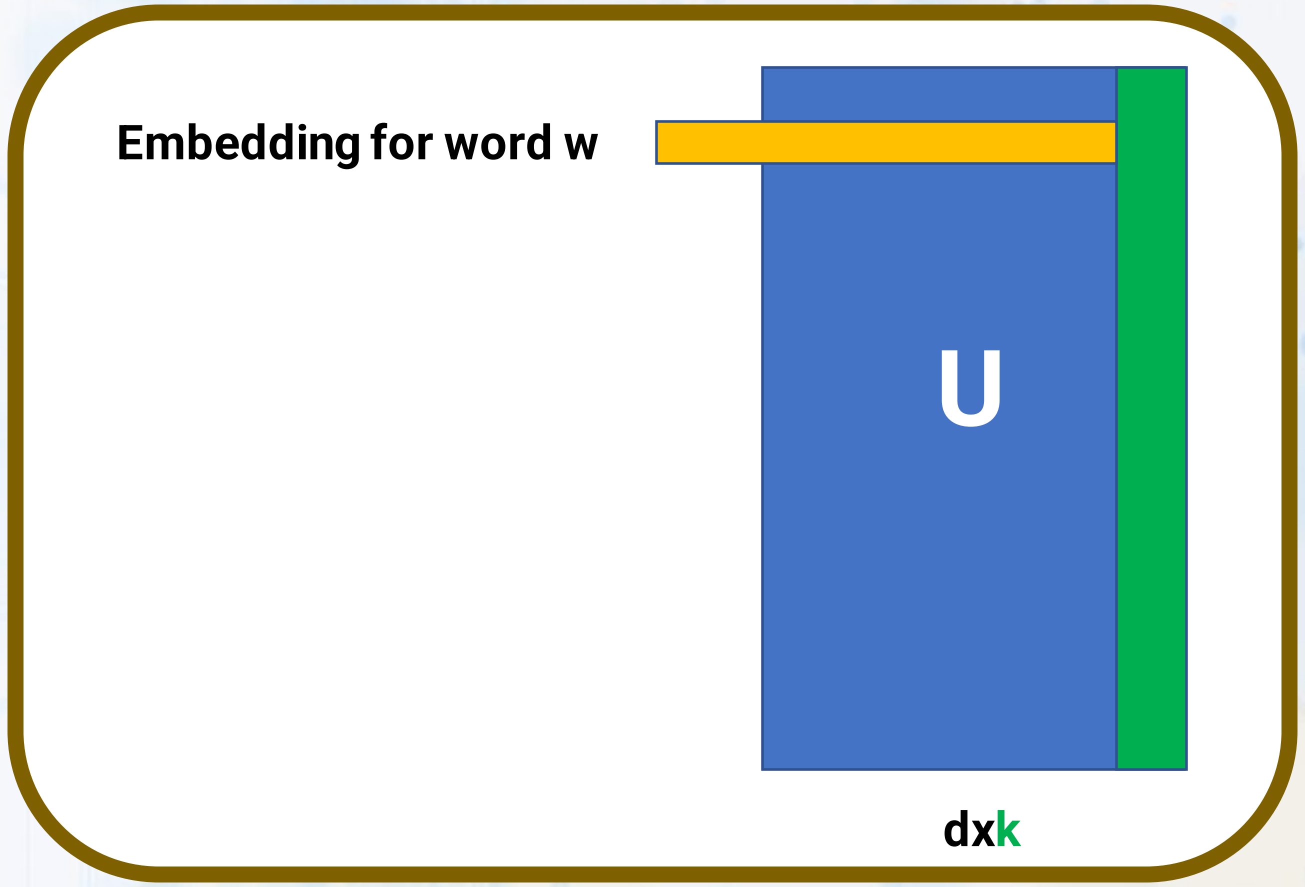
SVD on Co-Occurrence Matrices

- For a corpus with a vocabulary V of size d, the co-occurrence matrix has a size of dxd
- The size of the co-occurrence matrix increases with the vocabulary
- Instead of keeping all dimensions, we can use truncated SVD to keep only the top k singular values, for example 300
- The result is a least-square approximation to the original co-occurrence matrix X



Dense Word Embeddings

- Each row of U is a k-dimensional representation of each word w in the corpus that best preserves the variance
- Generally, we keep the top k dimension which can be ranged from 50 to 500.
- This produces dense vectors for word representations while taking into consideration the word contexts which carry meaning



Advantages of Dense Word Embeddings

- Denoising: low-order dimensions may represent unimportant information
- Truncation may help the models generalize better to unseen data
- Having a smaller number of dimensions may make it easier for classifiers to properly weight the dimensions for the task
- Dense models may do better at capturing higher order co-occurrence
- Dense vectors tend to work better in word similarity
- One word similarity method is cosine similarity between two-word embeddings w and v:

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Summary

- We learned about Singular Value Decomposition
- We learned about co-occurrence matrices, and how to generate dense word embeddings using SVD