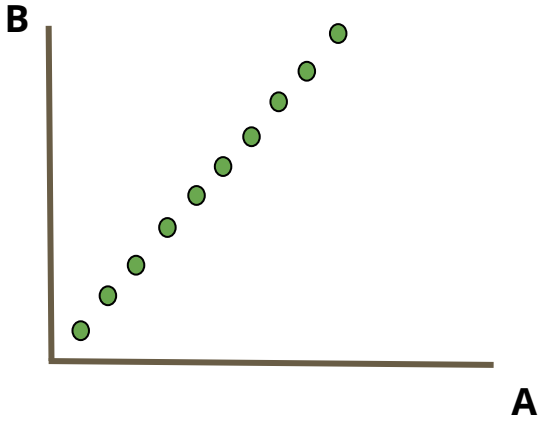
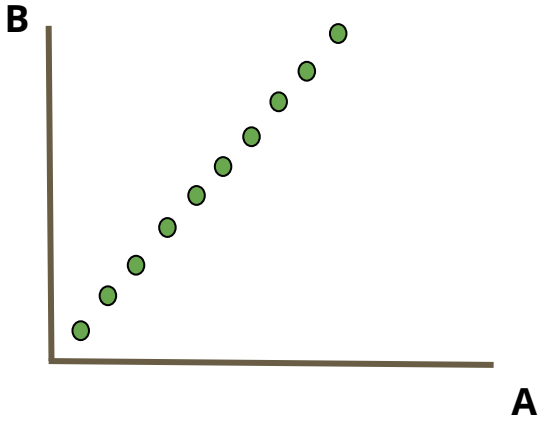
Singular Value Decomposition

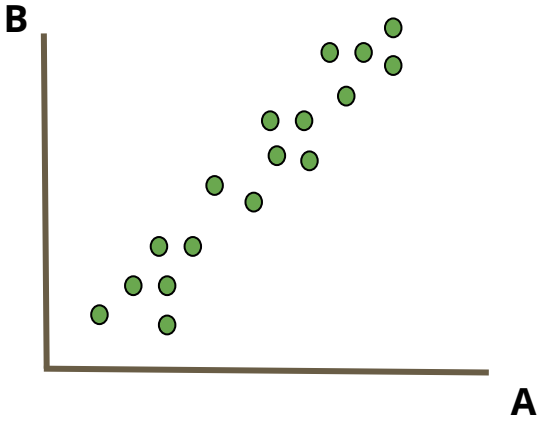
Boston University CS 506 - Lance Galletti

Characteristics of a dataset to look for

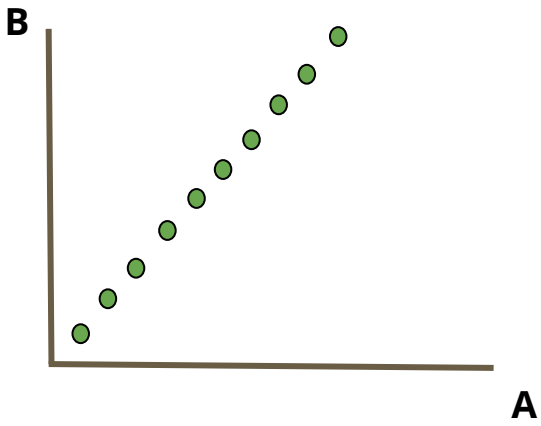


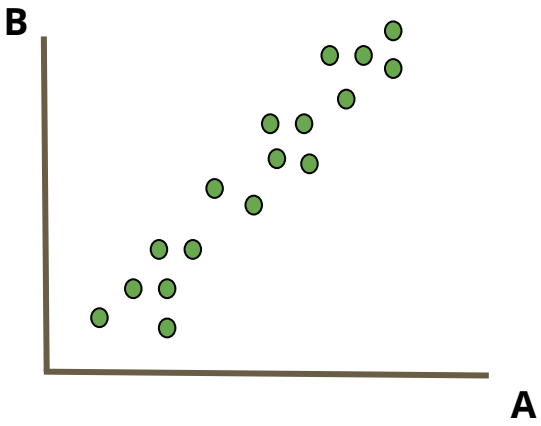
Characteristics of a dataset to look for

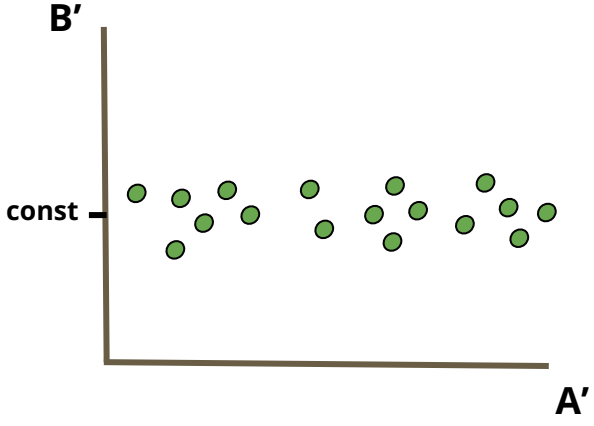


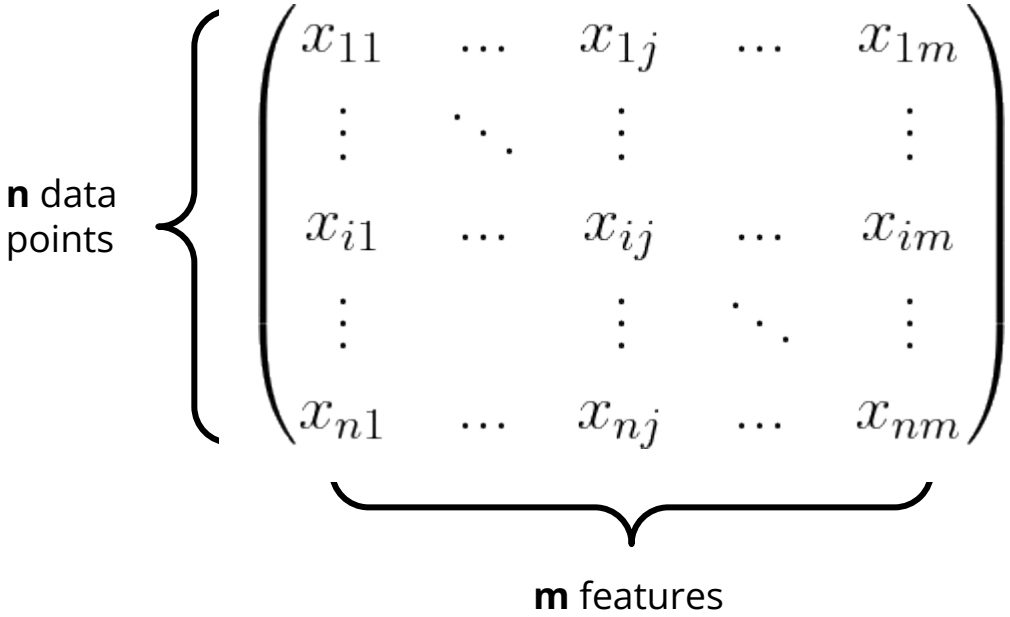


Characteristics of a dataset to look for





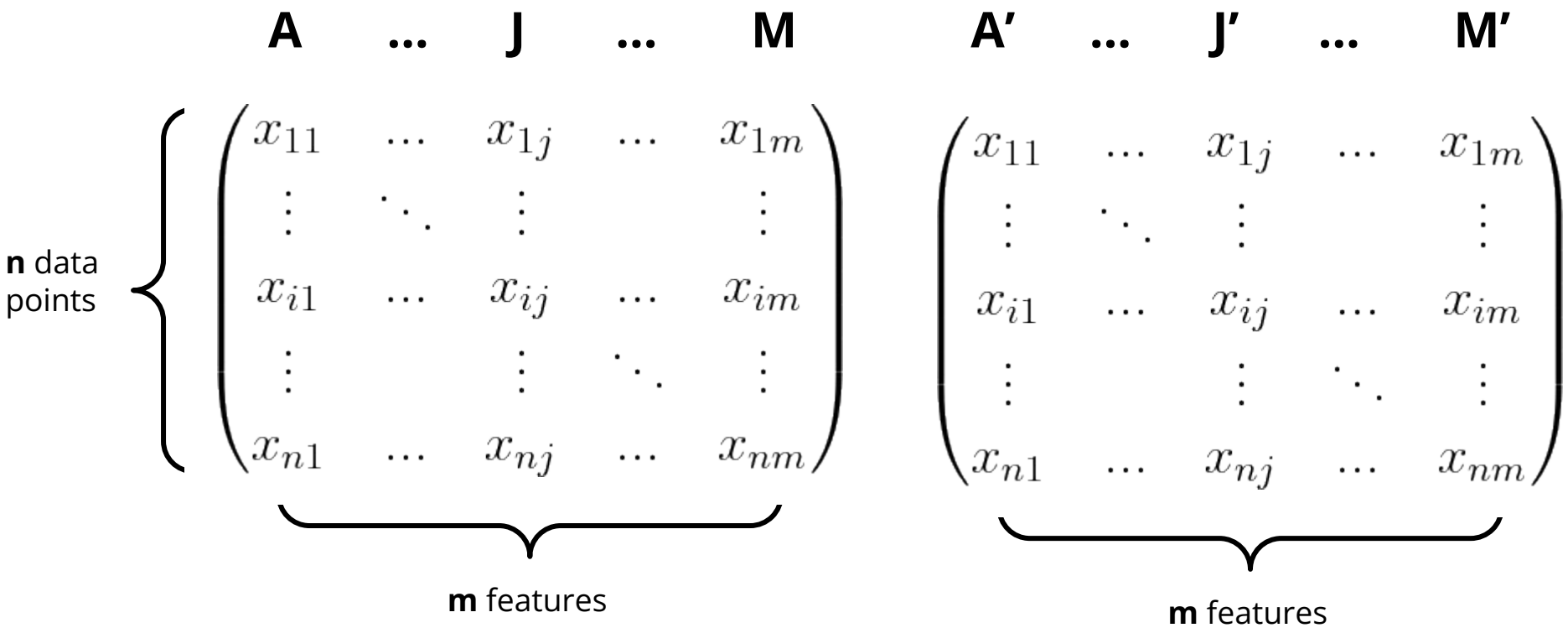



n data points
$\begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nm} \end{pmatrix}$
m features

Goal

Examine this matrix and uncover its linear algebraic properties to:

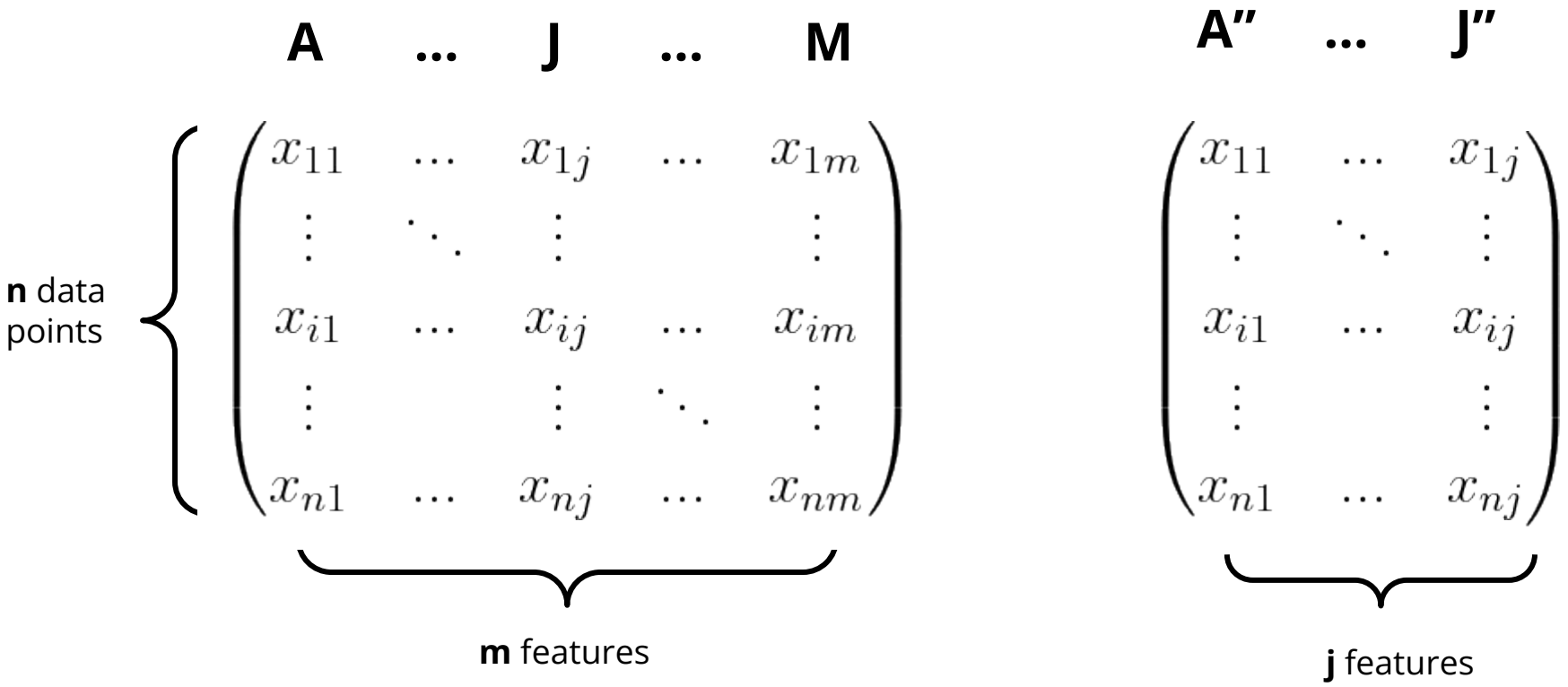
1. Approximate A with a smaller matrix B that is easier to store but contains similar information as A


A … J … M
A' … J' … M'
n data points
m features
m features

Goal

Examine this matrix and uncover its linear algebraic properties to:

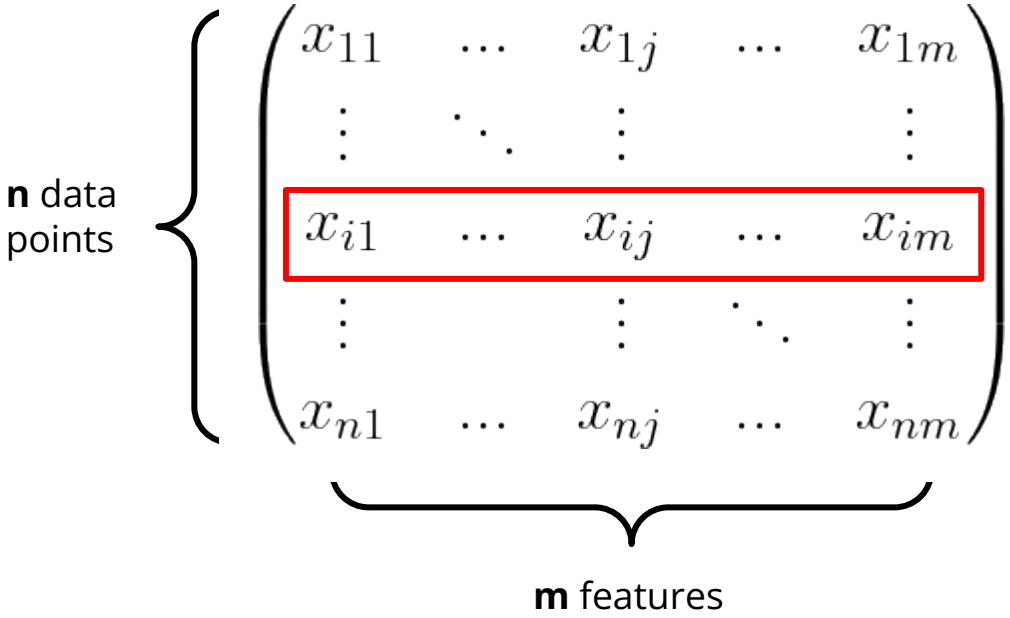
1. Approximate A with a smaller matrix B that is easier to store but contains similar information as A
2. Dimensionality Reduction / Feature Extraction


A
...
J
...
M
n data points
m features
A”
...
J”
j features

Goal

Examine this matrix and uncover its linear algebraic properties to:

1. Approximate A with a smaller matrix B that is easier to store but contains similar information as A
2. Dimensionality Reduction / Feature Extraction
3. Anomaly Detection & Denoising


$\begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nm} \end{pmatrix}$
n data points
m features

Linear Algebra Review

Definition: The vectors in a set $\mathbf{V = \{ \vec{v}_1, \ldots, \vec{v}_n \}}$ are **linearly independent** if

$$\mathbf{a_1\vec{v}_1 + \ldots + a_n\vec{v}_n = \vec{0}}$$

can only be satisfied by $\mathbf{a_i = 0}$

Note: this means no vector in that set can be expressed as a **linear combination** of other vectors in the set.

Linear Algebra Review

Definition:

The **determinant** of a square matrix A is a scalar value that encodes properties about the **linear mapping** described by A.

2x2:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

det(A) = ad - bc

Linear Algebra Review

Definition:

The **determinant** of a square matrix A is a scalar value that encodes properties about the **linear mapping** described by A.

3x3:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad det(A) = a \cdot det\begin{pmatrix} e & f \\ h & i \end{pmatrix} - b \cdot det\begin{pmatrix} d & f \\ g & i \end{pmatrix} + c \cdot det\begin{pmatrix} d & e \\ g & h \end{pmatrix}$$

Linear Algebra Review

Definition:

The **determinant** of a square matrix A is a scalar value that encodes properties about the **linear mapping** described by A.

n X n:

Can recursively compute it. How?

Linear Algebra Review

Property:

n vectors **{** $\vec{v}_1$, ..., $\vec{v}_n$ **}** in an n-dimensional space are **linearly independent** iff the matrix **A**:

$$\mathbf{A} = [\vec{v}_1, \ldots, \vec{v}_n] \quad (n \times n)$$

has non-zero determinant.

Q: Can **m > n** vectors in an **n**-dimensional space be linearly independent?

No

Linear Algebra Review

Definition:

The **rank** of a matrix **A** is the dimension of the vector space spanned by its column space. This is equivalent to the maximal number of linearly independent columns / rows of **A**.

Definition:

A matrix **A** is **full-rank** iff **rank(A) = min(m, n)**

Note: Get the rank of a matrix through the **Gram-Schmidt process**

Matrix Factorization

Any matrix **A** of rank **k** can be factored as

$$A = UV$$

where

U is **n x k**
V is **k x m**

Matrix Factorization

To store an **n x m** matrix **A** requires storing **m · n** values.

However, if the rank of the matrix of **A** is **k**, since **A** can be factored as

$$A = UV$$

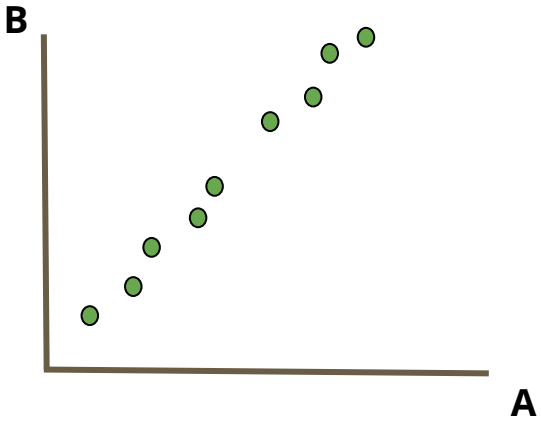
which requires storing **k(m + n)** values.

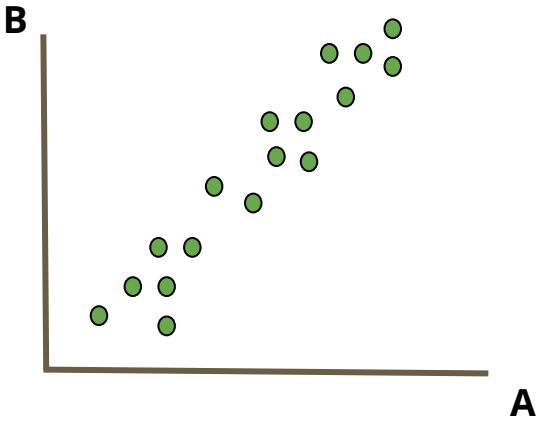
In Practice

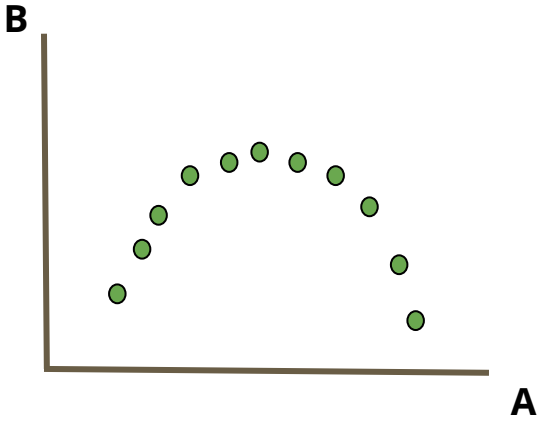
Most datasets are full rank despite containing a lot of redundant /similar information...

In Practice

Most datasets are full rank despite containing a lot of redundant /similar information...







In Practice

Most datasets are full rank despite containing a lot of redundant /similar information...

But we might be able to approximate the dataset with a lower rank one that contains similar information.

Approximation

Goal:

Approximate **A** with $\mathbf{A^{(k)}}$ (low-rank matrix) such that

1. **d(A, $A^{(k)}$)** is small
2. **k** is small compared to **m** & **n**

Frobenius Distance

$$d_F(A, B) = \|A - B\|_F = \sqrt{\sum_{i,j}(a_{ij} - b_{ij})^2}$$

i.e. the pairwise sum of squares difference in values of A and B

Approximation

Definition:

When **k < rank(A)**, the **rank-k approximation** of **A** (in the least squares sense) is

$$A^{(k)} = \underset{\{B|rank(B)=k\}}{\arg\min} \; d_F(A, B)$$

Matrix Factorization Improved

Not only can we factorize a matrix **A** of rank **k** as **A = UV**. But we can factorize **A** using a process called Singular Value Decomposition where:

$$A = U\Sigma V^T$$

Approximation

Definition:

The **Singular Value Decomposition** of a rank-r matrix A has the form

$$A = U\Sigma V^T$$

where

U is **n x r**
The **columns** of **U** are orthogonal & unit length ($U^TU = I$)

V is **m x r**
The **columns** of **V** are orthogonal & unit length ($V^TV = I$)

Approximation

Definition:

The **Singular Value Decomposition** of a rank-r matrix A has the form

$$A = U\Sigma V^T$$

where

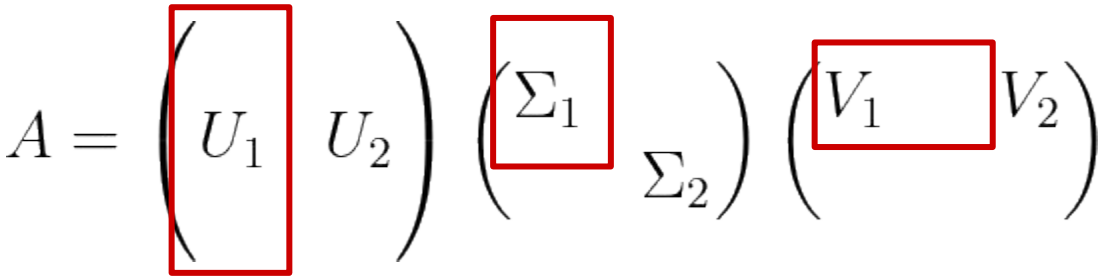
$$\Sigma = \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{pmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$

σ_i is the square root of the eigenvalues of A^TA and are called **singular values**

Approximation

Find $\mathbf{A}^{(k)}$ by decomposing **A**:



$\mathbf{A}^{(k)} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T$

Where

$\mathbf{U}_1$ is **n x k**
$\mathbf{\Sigma}_1$ is **k x k**
$\mathbf{V}_1$ is **m x k**

Approximation

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
0	0	0	2	2
0	0	0	3	3
0	0	0	1	1

=

0.18	0
0.36	0
0.18	0
0.90	0
0	0.53
0	0.80
0	0.27

x

9.64	0
0	5.29

x

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Approximation

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
0	0	0	2	2
0	0	0	3	3
0	0	0	1	1

~

0.18	0
0.36	0
0.18	0
0.90	0
0	0.53
0	0.80
0	0.27

x

9.64	0
0	0

x

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Approximation

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
0	0	0	2	2
0	0	0	3	3
0	0	0	1	1

~

0.18	0
0.36	0
0.18	0
0.90	0
0	0.53
0	0.80
0	0.27

x

9.64	0
0	0

x

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Approximation

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
0	0	0	2	2
0	0	0	3	3
0	0	0	1	1

~

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Approximation

The **i^{th} singular vector** represents the direction of the i^{th} most variance.

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ & & \ddots \\ 0 & & \sigma_r \end{pmatrix}$$

Singular Values express the importance / significance of a singular vector

Approximation

Property:

$$d_F(A, A^{(k)})^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Note: the larger **k** is, the smaller the distance.

Approximation

To find the right **k** you can:

1. Look at the singular value plot to find the elbow point
2. Look at the residual error of choosing different **k**

Related to Principal Component Analysis (PCA)

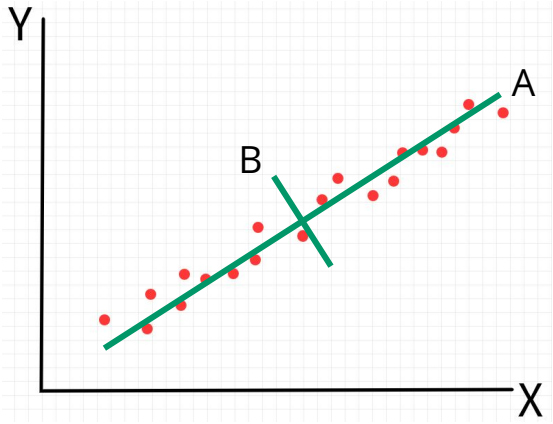
SVD and PCA are related

See demo

Dimensionality Reduction

Idea: project the data onto a subspace generated from a subset of singular vectors / principal components.

We want to project onto the components that capture most of the variance / information in the data.



Which principal component should we project on?

Anomaly Detection

Define $\mathbf{O = A - A^{(k)}}$

The largest rows of **O** could be considered anomalies