Singular Value Decomposition

Boston University CS 506 - Lance Galletti

Recall

$$\textbf{n}\text{ data points}\left\{\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nm} \end{pmatrix}}_{\textbf{m}\text{ features}}\right.$$

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

Linear Algebra Review

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

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

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:

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

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

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

has non-zero determinant.

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

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

$$\mathbf{A = UV}$$

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

In Practice

Most datasets are full rank despite containing a lot of redundant 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. $\mathbf{d(A, A^{(k)})}$ 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:

$$\mathbf{A} = \mathbf{U\Sigma V^T}$$

Approximation

Definition:

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

$$\mathbf{A = U\Sigma V^T}$$

where

U is **n x r**

The columns of **U** are orthogonal & unit length ($\mathbf{U^TU = I}$)

V is **m x r**

The columns of **V** are orthogonal & unit length ($\mathbf{V^TV = I}$)

Approximation

Definition:

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

$$\mathbf{A = U\Sigma V^T}$$

where

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

with $\boldsymbol{\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0}$

$\boldsymbol{\sigma_i}$ is the square root of the eigenvalues of $\mathbf{A^TA}$ and are called **singular values**

Approximation

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

$$A = \begin{pmatrix} \boxed{U_1} & U_2 \end{pmatrix} \begin{pmatrix} \boxed{\Sigma_1} & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} \boxed{V_1 \quad} & V_2 \end{pmatrix}$$

$\mathbf{A^{(k)} = U_1 \Sigma_1 V_1^T}$

Where

$\mathbf{U_1}$ is $\mathbf{n \times k}$

$\mathbf{\Sigma_1}$ is $\mathbf{k \times k}$

$\mathbf{V_1}$ is $\mathbf{m \times k}$

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

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

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

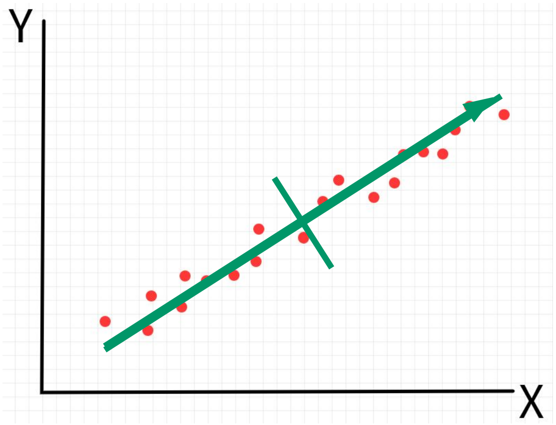
~

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 **ith singular vector** represents the direction of the ith 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

Anomaly Detection

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

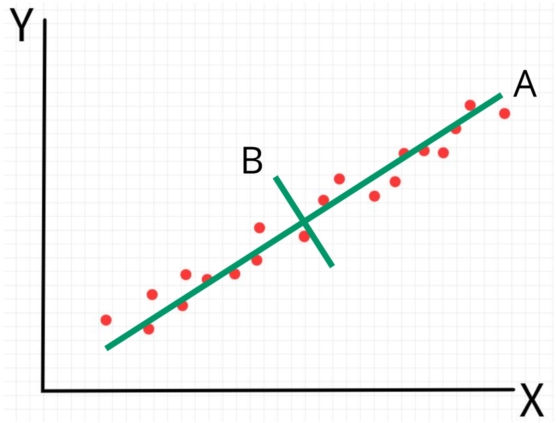
The largest rows of **O** could be considered anomalies

Worksheet a) -> d)

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?

Worksheet e) -> i)

Latent Semantic Analysis

Inputs are documents. Each word is a feature. We can represent each document by:

- The presence of the word (0 / 1)
- Count of the word (0, 1, ...)

Latent Semantic Analysis

	data	information	retrieval	brain	lung
CS-paper-1	1	1	1	0	0
CS-paper-2	2	2	2	0	0
CS-paper-3	1	1	1	0	0
CS-paper-4	5	5	5	0	0
Med-paper-1	0	0	0	2	2
Med-paper-2	0	0	0	3	3
Med-paper-3	0	0	0	1	1

Latent Semantic Analysis

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

Latent Semantic Analysis

CS concept	MD concept
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

Latent Semantic Analysis

CS concept

MD concept

doc-to-concept similarity

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

Latent Semantic Analysis

doc-to-concept
similarity matrix

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

Latent Semantic Analysis

doc-to-concept
similarity matrix

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

x

"strength" of the CS concept

9.64	0
0	5.29

x

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Latent Semantic Analysis

doc-to-concept
similarity matrix

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

x

"strength" of the
each concept

9.64	0
0	5.29

x

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Latent Semantic Analysis

doc-to-concept
similarity matrix

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

x

“strength” of the
each concept

9.64	0
0	5.29

x

term-to-concept similarity

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Latent Semantic Analysis

doc-to-concept similarity matrix

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

x

"strength" of the each concept

9.64	0
0	5.29

x

term-to-concept similarity matrix

0.58	0.58	0.58	0	0
0	0	0	0.71	0.71

Latent Semantic Analysis

We can better represent each document by:

- Frequency of the word (n_i / Σn_i)
- TfiDf

$$\mathrm{tf} \cdot \mathrm{idf}$$

Term frequency in the document

$$\log\left(\frac{\text{number of documents}}{\text{number of documents that contain the term}}\right)$$