This session will be taught in Hybrid format

When joining on Zoom

- When your microphone or camera are switched on you will be able to be heard in the teaching room and may be seen on the screen in the room.
- If the screen is recorded, you may be captured on the recording if you choose to contribute and have your camera switched on.
- Chat messages to everyone and privately to the host may be visible on screen in the room.

When joining in the Teaching room

- Academic teaching staff should let you know when Zoom is connected and you may see the Zoom meeting on the screen.
- Once connected to Zoom, microphones may capture sound from the whole room. Your voice may be shared to Zoom when speaking at a normal talking volume.
- Once connected to Zoom, cameras will capture video images from the room. These may be shared to Zoom. The camera capture will be a wide shot of the whole room or block of seating. It will not capture close-up images of individual students.
- If the session is recorded, video from the room will be captured on the recording. Your voice will be captured on the recording if you choose to contribute.
- If you connect to the Zoom meeting from within the room (for example to use the chat), please keep your audio on mute or use headphones to avoid feedback issues with the room audio.

The recordings may be stored in the cloud ad any personal information within the recording will be processed in accordance to the General Data Protection Regulations 2018.
A substantial contribution is considered to be anything more than merely answering questions or participating in a group discussion. Where you make a substantial contribution to the delivery of the recorded events, a signed consent form will be obtained prior to the recording being made available for viewing. The Consent Form will address your personal informationand any copyright or other intellectual property in the recording.

Dimensional analysis by co-occurrence

LSA

Felipe Orihuela-Espina

28-Oct-2024

Table of Contents

Section 1

Latent Semantic Analysis (LSA) - Problem statement

Latent Semantic Analysis

LSA credentials

- First formulated in 1990 by Deerwster et al, but maths (SVD) were already very well known.
- Although originally formulated in the domain of natural language processing for stusying relations between documents, because it is only the SVD of the co-occurrence matrix its applicability can be generalized.
- Relaxes the PCA requirement of a square matrix (covariance) substituting it for another of co-occurrence of the events. This is convenient for many applications, for instance in information retrieval is referred to as latent semantic indexing (LSI).

Latent Semantic Analysis

In PCA, one departs from a set of random (column) variables X defined on an ambient (explicit) space of explicit dimensionality n.

- The rows represent the point cloud of m observations.
- If there ever was an intended "semantic" associated to these variables, this is now lost in time (at least to me).

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Latent Semantic Analysis

. . . and you explore the co-variances of these variables, which capture how much the different variables (or explicit dimensions or features) are related to the others in terms of their phase oscillations around the central tendency.

In the covariance matrix, both rows and columns represent the same type of entity (covariances of variables). This is referred to as one-mode factor analysis.

- The covariance matrix is *always* a square matrix and hence the eigendecomposition is suitable.

$$A = cov(X) = \begin{pmatrix} \sigma^2_{X_1} & \sigma_{X_1,X_2} & \cdots & \sigma_{X_1,X_n} \\ \sigma_{X_2,X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2,X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_n,X_1} & \sigma_{X_n,X_2} & \cdots & \sigma^2_{X_n} \end{pmatrix}$$

Latent Semantic Analysis

When all variables are unimodal and more or less centered, the co-variance is an excellent estimator of the relations.

But for many distributions, e.g. multi-modal, the central tendency is not that relevant...

Another way at looking at the relation between these variables is looking at the co-occurrence of events.

Latent Semantic Analysis

Analogously, in LSA, one departs from a set of random (column) variables X defined on an ambient (explicit) space of explicit dimensionality n.

- The intended "semantic" associated to these variables were a collection of different documents and the sampling space from where the observations are retrieve are the words or terms contained in these documents.
- The rows represent the point cloud of frequencies of m observations (terms).

$$t_i^T \rightarrow \begin{pmatrix} & & d_j \downarrow & \\ x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Latent Semantic Analysis

A column in this matrix will be a vector corresponding to a document (random variable) expressing its relation to each term (event)

$$d_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{bmatrix}$$

A row in this matrix will be a vector corresponding to a term (event) expressing its relation to each document (random variable):

$$t_i^T = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{in} \end{bmatrix}$$

Latent Semantic Analysis

In other words, X can be seen as an occurrence matrix describing how frequent different terms (events) happen in each documents (random variable).

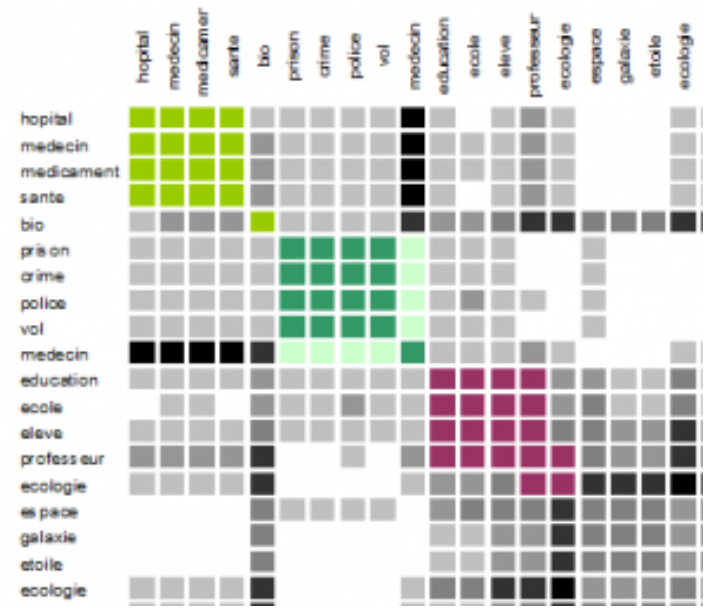


Figure: Ocurrence matrix. Figures from: [https://www.xlstat.com/en/solutions/features/latent-sementic-analysis-lsa]

Latent Semantic Analysis

LSA attempts to solve the problem of constructing "a *semantic* space wherein terms and documents that are closely associated are place near one another." [2].

Moreover, "Singular value decomposition allows the arrangement of this [semantic] space" reflecting the major associative patterns. "The result can be represented geometrically by a spatial configuration in which the dot product[1] or cosine between vectors representing two documents corresponds to their estimated similarity." [2]

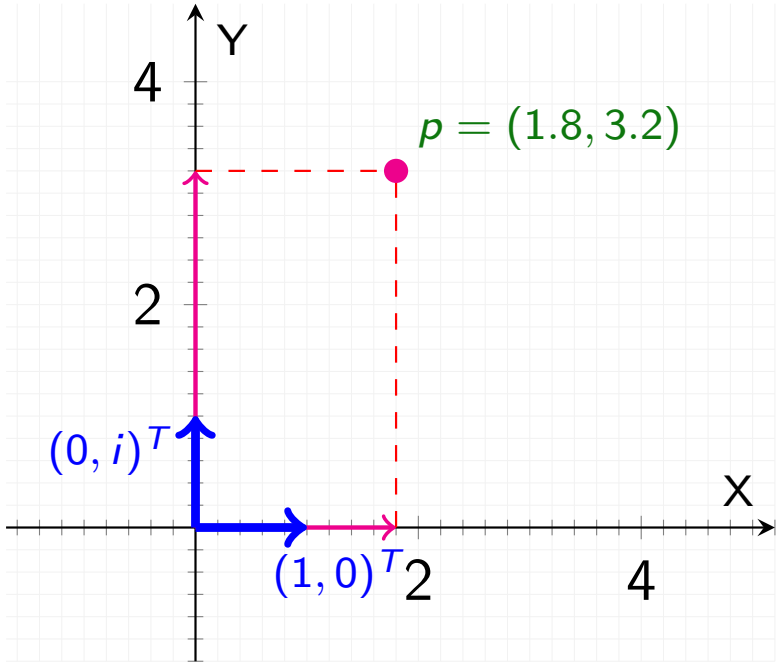
[1] Dot product: $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = \|a\| \, \|b\| \, cos(\theta)$

Subsection 1

What we already know from this module

Coordinate basis

The Argand basis of the Complex plane $\mathbb{C}$



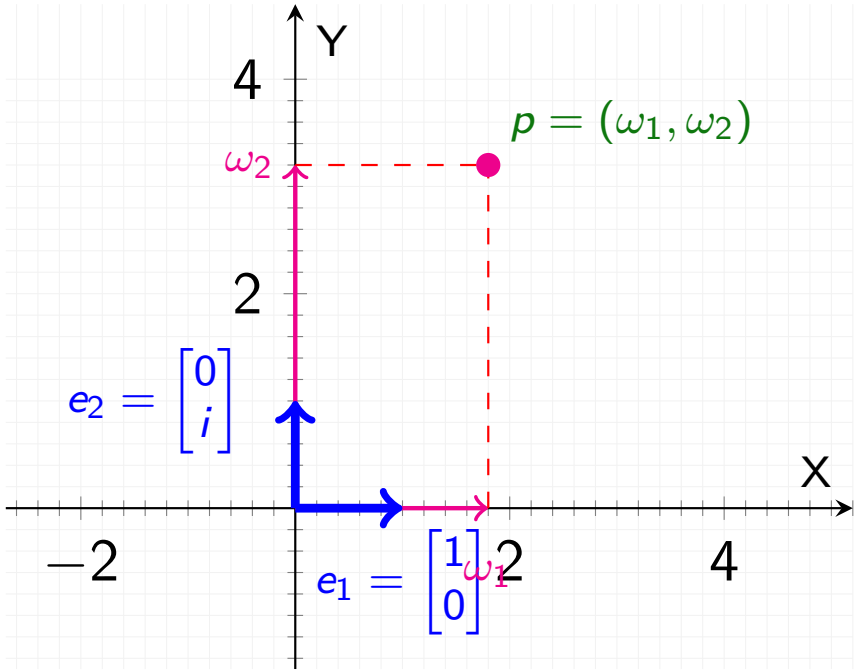
$$p = 1.8 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3.2 \cdot \begin{bmatrix} 0 \\ i \end{bmatrix}$$

$$= \begin{bmatrix} 1.8 & 3.2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = 1.8 + 3.2i$$

Coordinate basis

The Argand basis of the Complex plane $\mathbb{C}$

Let's move from arithmetics to algebra;



$$p = \omega_1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \omega_2 \cdot \begin{bmatrix} 0 \\ i \end{bmatrix}$$
$$= \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$
$$= \begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix} \cdot \begin{bmatrix} e_1 & e_2 \end{bmatrix}$$
$$= \omega_1 \cdot e_1 + \omega_2 \cdot e_2$$
$$= \sum_i \omega_i \cdot e_i$$

$$\boxed{p = \sum_i \omega_i \cdot e_i}$$

Eigendescomposición

Let a matrix A be symmetrical (and therefore square). The eigendecomposition is a factorization such that:

$$AP = DP$$

where:

- D is a diagonal matrix of non-singular values (scaling matrix - of the rotated system), and
- P is a square matrix (rotation or change of basis matrix).
- . . . and invertible; in fact the true factorization is $A = PDP^{-1}$.

Eigendescomposición

The eigendecomposition must satisfy:

$$
\begin{aligned}
AP &= A[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n] \\
&= [A\mathbf{p}_1, A\mathbf{p}_2, \ldots, A\mathbf{p}_n] \\
&\overset{A\mathbf{p}_i = \lambda_i \mathbf{p}_i}{=} [\lambda_1 \mathbf{p}_1, \lambda_2 \mathbf{p}_2, \ldots, \lambda_n \mathbf{p}_n] \\
&= \begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1n} \\ p_{21} & p_{22} & \ldots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \ldots & p_{nn} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix} \\
&= PD
\end{aligned}
$$

In magenta, the key step restricts the factorization. Note that it is a "simple" system of equations that "must be solved"; $A\mathbf{p}_i = \lambda_i \mathbf{p}_i$

Section 2

Complex conjugates and the conjugate transpose matrix

Complex conjugates

The complex conjugate of a complex number $z \in \mathbb{C}$, denoted z^*[2] is the number with an equal real part and an imaginary part equal in magnitude but opposite in sign.

Example

Let $z = a + bi$. Then the complex conjugate of z is $z^* = a - bi$.

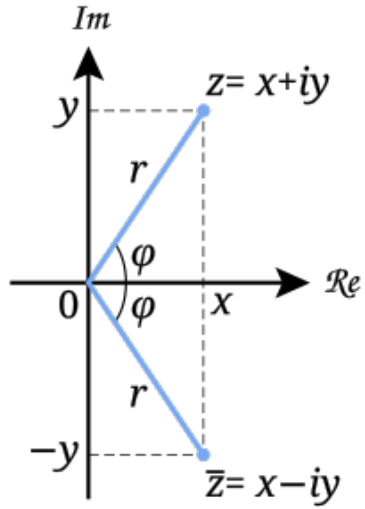


Figure: Complex conjugate. Figures from: [Wikipedia:Complex_conjugate]

[2] . . . or also sometimes $\bar{z}$, but we shall avoid this notation here as we are already using $\bar{\bullet}$ to denote the average.

Complex conjugates

Properties:

- The product of a complex number and its conjugate is a real number: $a^2 + b^2$
- For any two complex numbers, conjugation is distributive over addition, subtraction, multiplication and division;
  - $(z + w)^* = z^* + w^*$
  - $(z - w)^* = z^* - w^*$
  - $(zw)^* = z^* w^*$
  - $\left(\frac{z}{w}\right)^* = \frac{z^*}{w^*} \qquad w \neq 0$
- Conjugation does not change the modulus of a complex number; $|z^*| = |z|$
- $(z^*)^* = z$
- Conjugation is commutative for power $(z^n)^* = (z^*)^n$

Complex conjugates

Why are complex conjugates important?

There are many reasons why complex conjugates are important, here there is a non-exhaustive few;

- They permit de-rotation (reversing a rotation in the Argand plane).
  - Example: Multiplying complex vector$z$ by the complex vector $1+i$ will rotate z by $45°$. To reverse this rotation, we multiply by the complex conjugate of $1+i$, namely, $1-i$.
- They can (substantially) help to simplify expressions e.g. rationalization operations.
- Finding roots of polynomial expressions, e.g. it can be proven that complex roots always appear in pairs.
- Calculating Fourier's DFT and its inverse.
- They naturally appear in quantum physics.
- etc

Conjugate transpose matrix

The conjugate transpose matrix, also known as the Hermitian transpose, of an $m \times n$ complex matrix A is an $n \times m$ matrix obtained by transposing A and applying complex conjugation to each entry.

$$A_{m\times n} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{pmatrix} \quad \Rightarrow \quad A^*_{n\times m} = \begin{pmatrix} z^*_{11} & z^*_{21} & \cdots & z^*_{m1} \\ z^*_{12} & z^*_{22} & \cdots & z^*_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ z^*_{1n} & z^*_{2n} & \cdots & z^*_{mn} \end{pmatrix}$$

Conjugate transpose matrix

An invertible complex square matrix A is unitary if its matrix inverse A^{-1} equals its conjugate transpose A^*, $A^{-1} = A^*$, and hence:

$$A^*A = AA^* = I$$

In other words, the unitary matrix is the complex analogue of an orthogonal matrix for real numbers.

If in addition, its determinant equals 1, then it is called special unitary.

Section 3

Matrix inversion

Subsection 1

Inverse matrix

Inverse matrix

The inverse matrix or reciprocal matrix is like the inverse of a number but for matrices;

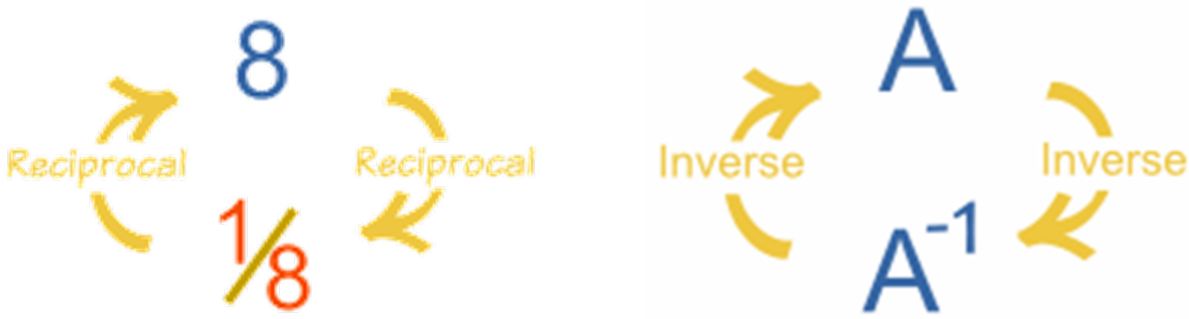


Figure: Inverse of a matrix. Figures from: [http://www.mathsisfun.com/algebra/matrix-inverse.html]

Inverse matrix

Definition

The inverse matrix or reciprocal matrix A^{-1} of a matrix A is one that helds;

$$AA^{-1} = A^{-1}A = I$$

- Note that not only is it required that $AA^{-1} = I$ and that $A^{-1}A = I$ separately, it is also required that $AA^{-1} = A^{-1}A$.
- This is "exploited" for the proof that only square matrices have inverses.

Inverse matrix

Why is the inverse matrix so important?

- Note that we have defined matrix product, but not division.
- This is not an oversight; matrix division is not defined .
- So to "divide" matrices, what we really do is multiply by the inverse .

Example

To calculate X given $XA = B$ we cannot calculate $X = B/A$. Instead we multiply by the inverse on both sides:

$$XAA^{-1} = BA^{-1} \Rightarrow XI = X = BA^{-1}$$

Inverse matrix

Theorem

Only a square matrix can have an inverse.

Proof.

Suppose the matrix A is of order $n \times m$ and is not necessarily square, that is, it could be that $n \neq m$. Remember that for a matrix A to have as its inverse A^{-1} with order $m \times n$, it must be fulfilled that $AA^{-1} = A^{-1}A = I$.

- In the first part of the equality it is required that A^{-1} has a size $m \times n$; $AA^{-1} = I_n$.
- In the second part of the equality it is required that A^{-1} has a size $m \times n$; $A^{-1}A = I_m$.
- In addition it must be fulfilled that $AA^{-1} = A^{-1}A$; and therefore, it is required that $I_n = I_m$, which can only occur if $n = m$.

□

Watch out!

The converse is not true. Just because it is square does not mean it always has an inverse. Its determinant could be 0.

Inverse matrix

Theorem

The inverse of a square matrix, if it exists, is unique

Proof.

Suppose B and C are inverses of A. Then it holds that:

- $AB = BA = I$
- $AC = CA = I$
- Let's just multiply in one of the cases by "the other inverse":

$$(BA)C = IC = C$$

- Relying on the associative, we could solve differently:

$$(BA)C = B(AC) = BI = B$$

- Since $C = (BA)C = (BA)C = B$; The only way this can be true is that $B = C$ and therefore implies that the inverse is unique.

Inverse matrix

Remark

- These two previous theorems about the inverse show the importance of square matrices.
- For non-square matrices, there are generalizations called pseudo-inverse or generalized inverse [1]. Perhaps the best known is the Moore-Penrose pseudo-inverse [3]. But we will not see them here.

Subsection 2

Matrix inversion

Matrix inversion

The calculation of the inverse matrix is not trivial

- . . . although there are simple cases; for example, to obtain the inverse of a diagonal matrix it is enough to invert the elements of the main diagonal.

Let's look at the "classic" method of the adjoint.

Matrix inversion

Cofactors

- In the computation of the determinant of the matrix A, we call cofactor of a_{ij} in $det(A)$ the set of terms that have as a common element a_{ij} and in which a_{ij} itself has been taken as a common factor.
- The cofactor is often denoted by A_{ij}.

- In other words, they are the terms resulting from their minor affected by the corresponding sign; in fact, we can rewrite the determinant as:

$$det(A) = \sum_{i=1}^{n} a_{i1} A_{i1}$$

- Note that the sign of the term in the determinant calculation is "absorbed" by the cofactor; in other words, *the cofactor includes the sign*.

Cofactors

Example

Let A be the matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

The cofactors of the elements a_{ij} will be[a]:

$$A_{11} = a_{22}a_{33} - a_{23}a_{32} \quad A_{12} = a_{23}a_{31} - a_{21}a_{33} \quad A_{13} = a_{21}a_{32} - a_{22}a_{31}$$
$$A_{21} = a_{13}a_{32} - a_{12}a_{33} \quad A_{22} = a_{11}a_{33} - a_{13}a_{31} \quad A_{23} = a_{12}a_{31} - a_{11}a_{32}$$
$$A_{31} = a_{12}a_{23} - a_{13}a_{22} \quad A_{32} = a_{13}a_{21} - a_{11}a_{23} \quad A_{33} = a_{11}a_{22} - a_{12}a_{21}$$

[a] The "negative" ones (those whose original sign in the calculation of the determinant is negative) are highlighted in blue. Note that here they are already "flipped" (i.e.

Matrix inversion

Cofactors

The result of replacing each element by its cofactor is called the cofactor matrix.

- Formal: Let $A = [a_{ij}]$ be a matrix. The matrix $cof(A) = [A_{ij}]$ is called the cofactor matrix, and is denoted by $cof(A)$, where A_{ij} is the cofactor of a_{ij} in $det(A)$.

Cofactors

Example

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \Rightarrow \quad cof(A) = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

Matrix inversion

Adjoint matrix:

The adjoint matrix is the transposed cofactor matrix.

Definition

Let $A = [a_{ij}]$ be a matrix. The matrix $adj(A) = cof(A)^T$ is called an **adjoint matrix**, and is denoted by $adj(A)$.

By extension, $det(adj(A))$ is called adjoint determinant.

Matrix inversion

Adjoint matrix:

Example

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \Rightarrow \quad cof(A) = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

$$adj(A) = cof(A)^T = \begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{pmatrix}$$

Matrix inversion

Adjoint matrix:

Note that, the sum of the products of the elements of a line of A and the cofactors of the corresponding elements of a different parallel line (whether rows or columns) is always 0 [[?], pg 71].
It follows;

$$A \cdot adj(A) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix} = \begin{pmatrix} det(A) & 0 & \dots & 0 \\ 0 & det(A) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & det(A) \end{pmatrix}$$

. . . and in the same manner:

$$adj(A) \cdot A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} det(A) & 0 & \dots & 0 \\ 0 & det(A) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & det(A) \end{pmatrix}$$

. . . or what is the same:

$$A \cdot adj(A) = adj(A) \cdot A = det(A) \cdot I$$

Matrix inversion

Recall that the inverse matrix of a matrix A is the one that satisfies that $AA^{-1} = I$

And we already know that $adj(A) \cdot A = det(A) \cdot I$.

Multiplying by A^{-1} on both sides:

- $adj(A) \cdot A \cdot A^{-1} = det(A) \cdot I \cdot A^{-1}$
- $\Rightarrow adj(A) \cdot I = det(A) \cdot I \cdot A^{-1}$
- $\Rightarrow adj(A) = det(A) \cdot A^{-1}$

Hence:

$$A^{-1} = \frac{adj(A)}{det(A)}$$

- Now it is obvious why the matrix must be non-singular to have an inverse.

Matrix inversion

Summary of the adjoint method for calculating the inverse:

1. Calculate the $det(A)$
2. Calculate $cof(A)$
3. Obtain the transpose of $cof(A)$; $adj(A)$
4. Divide by the $det(A)$; $\frac{adj(A)}{det(A)}$
   - Remember that $det(A)$ is a scalar, not a matrix, so this division operation is per element.

Matrix inversion

Remark

This adjoint method is simple, but computationally expensive and very prone to rounding errors

Of course there are other methods to calculate the inverse. For example:

- By Gauss-Jordan elimination/reduction to transform $[A|I]$ in $[I|A^{-1}]$.
  - http://www.mathwords.com/i/inverse_of_a_matrix.htm
  - http://www.purplemath.com/modules/mtrxinvr.htm
- By partitioning [[**?**], pg 132-136]

Each method has its advantages and disadvantages, but we will not go into them here.

Matrix inversion

Example

Let A be the matrix

$$A = \begin{pmatrix} 8 & 4 & 2 \\ 3 & 0 & 1 \\ 2 & -1 & 3 \end{pmatrix}$$

1. Calculate the $det(A)$

$$det(A) = a_{11}A_{11} + a_{21}A_{21} + a_{31}A_{31} = 8 \cdot 1 + 3 \cdot -14 + 2 \cdot 4 = 8 - 42 + 8 = -26$$

Matrix inversion

Example

2. We calculate $cof(A)$ [a]

$$A = \begin{pmatrix} (0\cdot 3)-(-1\cdot 1) & (1\cdot 2)-(3\cdot 3) & (3\cdot -1)-(0\cdot 2) \\ (2\cdot -1)-(4\cdot 3) & (8\cdot 3)-(2\cdot 2) & (4\cdot 2)-(8\cdot -1) \\ (4\cdot 1)-(2\cdot 0) & (2\cdot 3)-(8\cdot 1) & (8\cdot 0)-(4\cdot 3) \end{pmatrix} = \begin{pmatrix} 1 & -7 & \\ -14 & 20 & \\ 4 & -2 & \end{pmatrix}$$

3. Obtain the transpose of $of(A)$; $adj(A)$

$$adj(A) = cof(A)^T = \begin{pmatrix} 1 & -14 & 4 \\ -7 & 20 & -2 \\ -3 & 16 & -12 \end{pmatrix}$$

[a] The "negatives" are highlighted in blue (those whose original sign in the calculation of the determinant is negative). Here they are already "flipped" (multiplied by the negative sign).

Matrix inversion

Example

4. Divide by the $det(A)$; $adj(A)/det(A)$

$$A^{-1} = \frac{adj(A)}{det(A)} = \begin{pmatrix} \frac{1}{-26} & \frac{-14}{-26} & \frac{4}{-26} \\ \frac{-7}{-26} & \frac{20}{-26} & \frac{-2}{-26} \\ \frac{-3}{-26} & \frac{16}{-26} & \frac{-12}{-26} \end{pmatrix}$$

Does anyone dare to check that $AA^{-1} = I$?

Matrix inversion

Theorem

A square matrix A has an inverse if and only if its determinant is not equal to 0.

Proof.

See the method of calculating the inverse using the adjoint, and remember that the inverse, if it exists, it is unique.

- For a more formal demonstration see:
  - Lipschutz, S. "Invertible Matrices." Schaum's Outline of Theory and Problems of Linear Algebra, 2nd ed. New York: McGraw-Hill, pp. 44-45, 1991

□

Section 4

Singular Value Decomposition (SVD)

Singular Value Decomposition

Singular Value Decomposition (SVD) is the *generalization* of the eigendecomposition for non-square matrices.

Analogous to the eigendecomposition, Singular Value Decomposition (SVD) is a matrix factorization technique i.e. it decomposes a matrix into a product of other matrices.

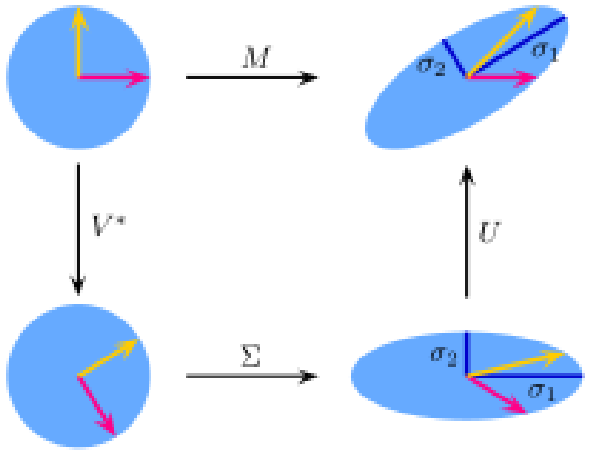
$$A = U\Sigma V^* \overset{If\ a_{ij}\in\mathbb{R}}{\equiv} U\Sigma V^T$$

where:

- U is an $m \times m$ complex unitary matrix,
- Σ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal, and
- V^* is the conjugate transpose of V, and in turn an V is an $n \times n$ complex unitary matrix.

Singular Value Decomposition

The new product is also analogous to the eigendecomposition producing a rotation, scaling and de-rotation effects.



$$M = U \cdot \Sigma \cdot V^*$$

Figure: Singular Value Decomposition. Illustration of matrix transformations. Figure from [Wikipedia:Singular_value_decomposition]

Singular Value Decomposition

It may not be obvious why the conjugate transpose appears when we want to carry out a rotation.

To fully understand this, it is convenient to recall that a complex number $z = a + bi$ can be usefully represented by 2×2 real matrices;

$$z = a + bi \equiv \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

But this is precisely the pattern of a rotation matrix! Indeed, multiplying by a complex number equates to a rotation.

Singular Value Decomposition

Further, matrices U and V are orthonogonal (unitary to be precise), i.e. multiplying by their transpose yields the identity matrix.

- In other words, the new coordinate basis it produces are orthogonal vectors.

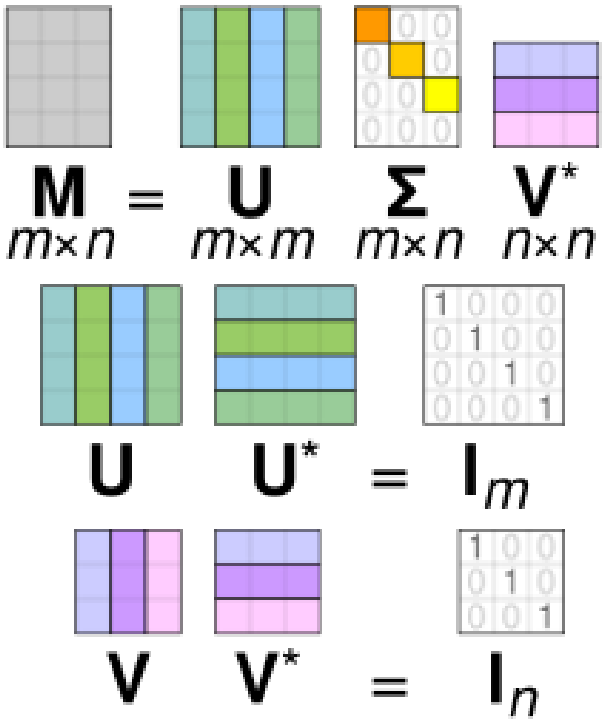


Figure: Singular Value Decomposition. Matrices U and V are orthonogonal. Figure from [Wikipedia:Singular_value_decomposition]

Singular Value Decomposition

With the above notions, we can see how the eigendecomposition;

$$A\mathbf{p}_i = \lambda \mathbf{p}_i$$

. . . is a specific case of SVD;

$$A\mathbf{v}_i = \sigma \mathbf{u}_i$$

. . . but SVD does not need A to be squared.

Singular Value Decomposition

Specifically, the following two relations hold:

$$AA^* = U(\Sigma\Sigma^*)U^*$$

$$A^*A = V(\Sigma^*\Sigma)V^*$$

. . . that's the eigendecomposition!

Singular Value Decomposition

Analogous to the eigenvalues, the diagonal entries $\sigma_i = \Sigma_{ii}$ of Σ are uniquely determined by A and are known as the singular values of A.

Singular Value Decomposition

Finally, how do we solve SVD?

Nowadays, there are a large number of algorithms each with advantages and limitations;

- Lanczos algorithm an iterative method to find the m "most useful" eigenvalues and eigenvectors of an Hermitian matrix, where m is often but not necessarily much smaller than n.
  - Efficient but numerically instable. There are now variants that can make it stable.
- One-sided Jacobi algorithm an iterative algorithm where matrix A is iteratively transformed into a matrix with orthogonal columns; $A_{new} \leftarrow A_{old} J(p, q, \theta)$ with $J(p, q, \theta)$ a rotation matrix (the Jacobi rotation)
- Two-sided Jacobi algorithm an iterative algorithm that generalizes of the Jacobi eigenvalue algorithm where by $A_{new} \leftarrow J^T G(p, q, \theta) A_{old} J$ with $G(p, q, \theta)$ the Givens rotation matrix J the Jacobi rotation matrix.
- etc (truly many others!)

Singular Value Decomposition

Finally, how do we solve SVD?

Unfortunately, most algorithms to solve SVD are "expensive" computationally for didactic purposes.

In principle, for the specific case of A being of order 2×2 there exist an **analytical solution**! [https://en.wikipedia.org/wiki/Singular_value_decomposition], but alas! this is based on Pauli's matrices which we haven't studied, ergo not suitable either for didactic purposes here.

Singular Value Decomposition

Finally, how do we solve SVD?

Besides, for small toy cases, there are few other ad-hoc solutions. Here is a couple of examples;

- https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf
- https://medium.com/intuition/singular-value-decomposition-svd-working-example-c2b61356

Singular Value Decomposition

SVD vs eigendecomposition

Let be the eigendecomposition $A = PDP^{-1}$ and the SVD $A = U\Sigma V^*$;

- The eigendecomposition only exists for square matricies. SVD always exists.
- The eigenvectors are not necessarily orthogonal (although they are for PCA), whereas the singular vectors are always orthogonal.
- In SVD, U and V are not necessarily the inverse of the other.
- Eigenvalues λ can be complex and/or negative numbers, whereas singular values σ are all real and non-negative.

Section 5

Latent Semantic Analysis - The solution

Latent Semantic Analysis

- The original paper Deerwester et al [[2], pg 395] discusses the possibility of using the eigendecomposition but discards it because the entry matrix to the eigendecomposition are the same type of entity e.g. documents (variables) vs documents (variables) i.e. one-mode factor analysis.
- Instead, Deerwester et al [[2], pg 395] wanted to carry out a two-mode factor analysis between terms $\times$ documents. But this means, that the matrix entering the analysis, we no longer square necessarily; hence SVD. LSA groups both documents that contain similar words, as well as words that occur in a similar set of documents.

Latent Semantic Analysis

After the construction of the occurrence matrix, LSA finds a low-rank approximation to the term-document matrix.

Some of the original motivations for searching for a low-rank approximation were:

- Merging the dimensions associated with terms that have similar meanings i.e. tackling the polysemy.
- Back in the 1990s, approximating helps to save precious computational power; the original term-document matrix was presumed too large for the computing resources at the time.
- With a sufficiently large vocabulary (i.e. a large number of events), the occurrence matrix was likely to be sparse.

Latent Semantic Analysis

$$
\begin{array}{ccccc}
 & & X & & \\
 & & (\mathbf{d}_j) & & \\
 & & \downarrow & & \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix} \\
 & & U & & \Sigma \quad V^T \\
 & & & & (\hat{\mathbf{d}}_j) \\
 & & & & \downarrow
\end{array}
$$

Figure: Latent Semantic Analysis in a nutshell. Figure from [Wikipedia:Latent_semantic_analysis]

Latent Semantic Analysis

The new dimensions do *not* intend to relate to any comprehensible concepts. They are only meant to be a lower-dimensional approximation of the higher-dimensional space.

. . . yet this new space has many applications! Just to mention a few;

- Variable clustering and classification
- Information retrieval
- Feature engineering and specifically, feature generation
- Variable matching (e.g. supporting searches)

Thanks

Thank you! Questions?

References I


Adi Ben-Israel and Thomas NE Greville.
Generalized inverses: theory and applications.
Springer Science & Business Media, 2006.


Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman.
Indexing by latent semantic analysis.
Journal of the American society for information science, 41(6):391–407, 1990.


Roger Penrose.
A generalized inverse for matrices.
In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955.