Review of Matrix Algebra

Ying-Chao Hung

Department of Statistics

National Chengchi University

Email: *hungy@nccu.edu.tw*

Basics

- A data matrix consists of measurement scores for n objects (e.g. individuals) on m items (attributes, variables).
- Usually, a data matrix is written so that the objects form the row and items form the columns.
  An example of a 3×2 matrix:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 5 \\ 4 & 7 \end{pmatrix}$$

- Matrices are usually denoted by capital letters, e.g. A, B, V.
- We denote the (i, j) element of a matrix A by $A(i, j)$, the i^{th} row by $A(i, :)$, the j^{th} column by $A(:, j)$,

Basics

Remark: The number of rows, n, and the number of columns, m, of a matrix define its *order*.

For example, the matrix A in the above has order "3 by 2".

In general, an $n \times m$ matrix A is denoted by $A_{n \times m}$.

If $n = m$, we have a *square* or *quadratic* matrix.

Remark: Matrices where $m = 1$ or $n = 1$ are called *vectors*.

Vectors are usually denoted by small letters, e.g. x, y, z, etc.

An $n \times 1$ vector is called a *column vector* and a $1 \times n$ vector is called a *row vector*.

The matrix A in the above example consists of 3 row vectors and 2 column vectors.

Basics

Transposition: Suppose x is a column vector, by simply writing it as a row vector x' is an operation called *transposition*.

In general, one can form the *transpose* of a matrix A by writing its rows as columns.

Denote the transpose of A by A', for the previous example,

$$A' = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 5 & 7 \end{pmatrix}.$$

Definition: A matrix A is *symmetric* if $A(i, j) = A(j, i)$ for all i, j, or equivalently, $A = A'$.

By definition, symmetric matrices have to be *square* matrices.

Basics

In data analysis, symmetric matrices are commonplace (e.g. correlation matrices, covariance matrices).

Matrix Addition: Addition (or subtraction) is possible only if two matrices A and B have the *same order*.

We write $C = A + B$, which gives that

$$C(i,j) = A(i,j) + B(i,j) \quad \text{for all } i,j.$$

Matrix Multiplication by a Scalar: We can simply multiply a matrix by a scalar k, which is done by multiplying each and every element of the matrix by k. Formally, $B = kA$ has elements given by $B(i,j) = kA(i,j)$ for all i,j.

Basics

Matrix Multiplication: The product of two matrices $A_{n\times k}$ and $B_{k\times m}$ is a matrix C of order $n \times m$ with elements $C(i,j) = \sum_{a=1}^{k} A(i,a)B(a,j)$, $i = 1, ..., n$, $j = 1, ..., m$. Matrix multiplication requires that A has as many columns as B has rows. Hence, if both A and B are square matrices, then both products AB and BA exist and are of the same order. However, it is important to realize that $AB \neq BA$ in general (matrix multiplication **is not commutative**). We therefore use special terminology and speak of *multiplication from the left* and *multiplication from the right*.

Comment: There is another way of multiplying two matrices of order $n \times m$. We get that $C = AoB$ with $C(i,j) = A(i,j)B(i,j)$. The new matrix C is called the *Hadamard* product of A and B. However, this type of product plays a very minor role in most applications of matrix algebra.

Basics

Diagonal and Identity Matrices: A square matrix A with $A(i,i) \neq 0$ and $A(i,j) = 0$ for all $i \neq j$ is called a diagonal matrix. For example

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3.15 & 0 \\ 0 & 0 & -1/125 \end{pmatrix}$$

is a diagonal matrix. A diagonal matrix with $A(i,i) = 1$ for all i is called the *identity* matrix and corresponds to the *neutral element* with respect to matrix multiplication. The identity matrix of order n is denoted by I_n.

Inverse of a Matrix: The inverse of a square matrix $A_{n\times n}$ is another square matrix of order $n \times n$ and denoted by A^{-1} that satisfies

(1) $$A^{-1}A = I_n = AA^{-1}$$

Basics

Rank of a Matrix: The *rank* of a $n \times m$ matrix is the number of *linearly independent* rows or columns of this matrix. Obviously, $r \leq \min(n, m)$. A row (column) is linearly independent if it can NOT be expressed as a *weighted* sum of the remaining rows (columns). A matrix with $r = \min(n, m)$ is called a matrix of *full* rank.

If a square matrix is not of full rank, then it is called *singular*.

Existence of the Inverse of a Matrix: The inverse of a square matrix $A_{n \times n}$ exists if A is a matrix of *full rank*, i.e. $r = n$.

Definition: A $n \times n$ matrix A satisfying $A'A = I_n$ is called *orthonormal*.

But if $A'A = I_n$ this automatically implies that $A' = A^{-1}$ and because A is square we also have that $AA^{-1} = AA' = I_n$. Hence, a square matrix with orthonormal columns has necessarily orthonormal rows.

Basics

Some Basic Useful Properties of Matrices:

- $(AB)C = A(BC)$, associative property
- $(A+B)(C+D) = A(C+D) + B(C+D)$, distribution property
- $(A')' = A$
- $(AB)' = B'A'$
- $(ABC)' = C'B'A'$
- $(A+B)' = A' + B'$
- $(A^{-1})^{-1} = A$
- $(A')^{-1} = (A^{-1})'$
- $(AB)^{-1} = B^{-1}A^{-1}$, provided both A and B are square

Definition: A square matrix $A_{n\times n}$ is called *positive semidefinite* iff for every vector $x \neq 0$ of order n, we have $x'Ax \geq 0$. It is called positive definite if a strict inequality holds.

Scalar Functions of Vectors and Matrices

One can take a matrix or a vector and assign to it, by some rule, a number. There are many such rules, and they are useful for different purposes. Here we discuss some cases that are important in the multivariate analysis context.

Scalar Product of Vectors: Given two real valued column vectors x and y of the same order n, their scalar product is given by

(2) $$< x, y >= x'y = \sum_{i}^{n} x(i)y(i)$$

Of particular importance is the case where $x'y = 0$. Vectors whose scalar product is zero are called *orthogonal*. As an exercise draw on the plane the vectors $x = (-2,\ \sqrt{2})$ and $y = (\sqrt{2},\ 2)$.

Scalar Functions of Vectors and Matrices

Length of a Vector: The *length* of a vector is given by

(3) $$||x|| = \sqrt{x'x} = (\sum_{i=1}^{n} x(i)^2)^{1/2}.$$

The length of a vector is an example of a norm. Norms are used to *normalize* a given vector to unit length. If x is a any real valued vector, then $u = (1/||x||)x$ is a *unit* vector, i.e. $||u|| = 1$.

Remark: All norms satisfy the following properties:

- $||x|| \geq 0$ for $x \neq 0$
- $||x|| = 0$ iff $x = 0$
- $||kx|| = k||x||$, for any scalar k
- $||x + y|| \leq ||x|| + ||y||$ (triangle inequality)

For example in regression problems we minimize $||\epsilon(\beta)||$ with respect to β, where $\epsilon(\beta) = y - X\beta$.

Scalar Functions of Vectors and Matrices

Definition: The *trace* of a *square* matrix A is given by

$$\mathbf{trace}(A) = \sum_{i=1}^{n} A(i,i). \tag{4}$$

Properties of the Trace:

- $\mathbf{trace}(A) = \mathbf{trace}(A')$
- $\mathbf{trace}(A_1 A_2 A_3 \ldots A_n) = \mathbf{trace}(A_2 A_2 \ldots A_n A_1) = \mathbf{trace}(A_3 \ldots A_n A_1 A_2) = \ldots$ (invariance under cyclic permutation)
- $\mathbf{trace}(A+B) = \mathbf{trace}(A) + \mathbf{trace}(B)$ (summation rule)

Frobenius Norm of a Matrix: It is given by

兩個矩陣的相似度

$$||A||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} A(i,j)^2} \tag{5}$$

It is also called the sum of squares norm and represents the natural generalization of the length of a vector to matrices.

Scalar Functions of Vectors and Matrices

A Fact: $||A||_F^2 = \textbf{trace}(A'A) = \textbf{trace}(AA')$.

Let us look at an example.

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 0 \\ 5 & 4 \end{pmatrix}$$

Do the necessary calculations to find that $||A||_F^2 = 55$ and verify the fact.

Eigendecompositions

Now we deal exclusively with **square** matrices.

Definition: For any matrix A (*real or complex*), a number λ (*real or complex*) is an *eigenvalue* if it corresponds to the root of the polynomial equation given by

$$\mathbf{det}(A - \lambda I) = 0. \tag{6}$$

Definition: A nonzero vector x (*real or complex*) is an *eigenvector* of A corresponding to the eigenvalue λ if it satisfies the following equation

$$Ax = \lambda x \tag{7}$$

Remark: If x is an eigenvector of A, so is cx, for any $c \neq 0$. This property follows by plugging in cx in (7).

Eigendecompositions

Remark: (Existence of eigenvalues/eigenvectors)
A matrix $A \in \mathbf{C}^{n \times n}$ (i.e., a matrix with complex entries) has exactly *n* complex eigenvalues. With each eigenvalue corresponds to at least one eigenvector.

To see why this is true, by writing $\det(A - \lambda I_n) = 0$ we get a complex polynomial of degree *n* in λ, which by the fundamental theorem of algebra together with the unique factorization theorem for polynomials, has precisely *n* complex roots.
Moreover, if $\det(A - \lambda I_n) = 0$, then $A - \lambda I_n$ is "singular", which means that "zero" is in the column space of $A - \lambda I_n$ ➔ there exists a column vector

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \text{ such that } \left(A - \lambda I_n\right)\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Thus, at least one eigenvector (x) is associated with a given λ.

Eigendecompositions

Example 1. If A is the identity matrix I_n, then the only eigenvalue is $\lambda = 1$ (with multiplicity n); thus, every nonzero vector in R^n is an eigenvector of A associated with the eigenvalue $\lambda = 1$, i.e., $I_n\mathbf{x} = 1\mathbf{x}$ for any nonzero vector $\mathbf{x}$.

Example 2: Let

$$A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Let us solve (6) again. We have

$$\det(A - \lambda I) = (1-\lambda)^2 + 1 = 0 \Longrightarrow (1-\lambda)^2 = -1. \tag{9}$$

Hence, we have that $\lambda_1 = 1+i$ and $\lambda_2 = 1-i$, with corresponding eigenvectors $x_1 = (1, i)$ and $x_2 = (1, -i)$.

This example shows that real matrices may have *complex* eigenvalues and eigenvectors.

Eigendecompositions

Example 3: Let

$$A = \begin{pmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 3 \end{pmatrix}$$

Let us solve (6) again. We have

$$\det(A - \lambda I) = (2-\lambda)(3-\lambda) - 2 = 0 \Longrightarrow \lambda^2 - 5\lambda + 4 = 0. \tag{10}$$

Hence, we have that $\lambda_1 = 4$ and $\lambda_2 = 1$, with corresponding eigenvectors $x_1 = (1, \sqrt{2})$ and $x_2 = (\sqrt{2}, -1)$.

Observe that in the above example x_1 and x_2 are orthogonal, i.e., $x_1 x_2' = 0$.

Eigendecompositions

If we collect all the eigenvalues of A into a $n \times n$ *diagonal* matrix Λ and all corresponding eigenvectors into a $n \times n$ matrix X (the eigenvectors correspond to the columns of X), (6) can be rewritten as

$$AX = X\Lambda. \tag{11}$$

In addition we require $X'X = I_n$ (standardize them by making them orthonormal).

A few important results for us are given next.

Theorem: Eigenvalues of real symmetric matrices are real and so are their corresponding eigenvectors.

Comment: Due to this Theorem, the fact that we obtained real eigenvalues and eigenvectors in Example 3 should not come as a surprise.

Eigendecompositions

Theorem: A Gramian (i.e symmetric, positive definite) matrix has real *positive* eigenvalues and eigenvectors.

Proof: The fact that they are real follows trivially from the previous Theorem. For being positive, consider any eigenvalue λ and corresponding eigenvector x of A. They obviously satisfy $Ax = \lambda x$; hence, $0 < x'Ax = \lambda x'x$. Since $x'x$ as a sum of squares is positive, so is λ.

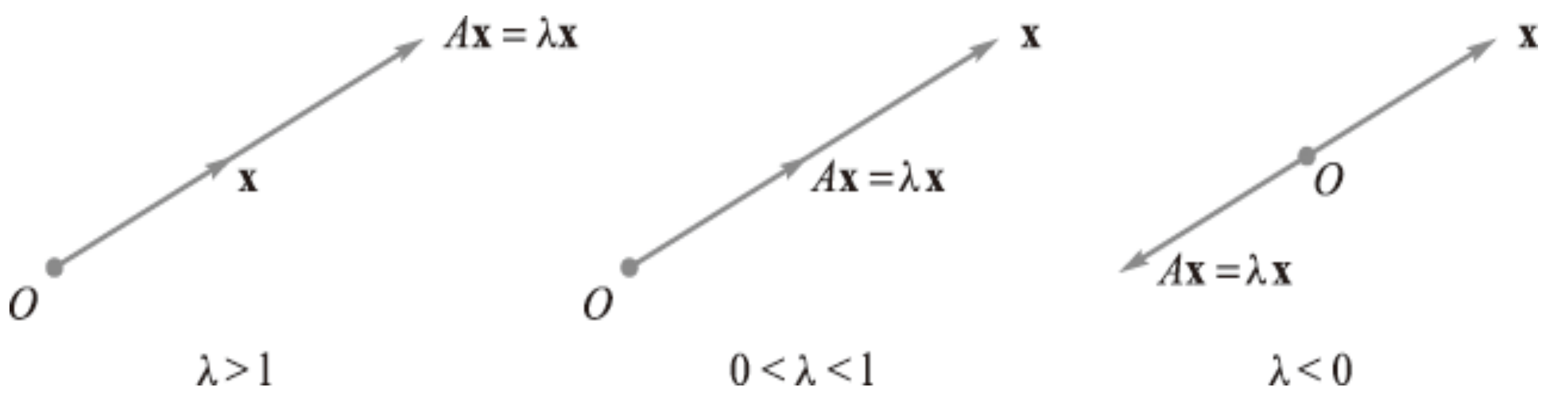
Rayleigh-Ritz Theorem: (Characterization of Eigenvalues). Suppose we want to maximize/minimize the quadratic form $x'Ax$ with respect to x subject to the constraint $x'x = 1$. Then, the maximum eigenvalue achieves the maximum and the minimum eigenvalue the minimum and the solution x is the corresponding eigenvector.

Remark: Suppose we want to calculate $\mathbf{trace}(A)$. Using the eigendecomposition of A we get $\mathbf{trace}(A) = \mathbf{trace}(X\Lambda X') = \mathbf{trace}(\Lambda X'X) = \mathbf{trace}(\Lambda I_n) = \mathbf{trace}(\Lambda) = \sum_{i=1}^{n} \Lambda(i,i)$. Thus, the trace of a matrix A is equal to the *sum* of its eigenvalues.

Eigendecompositions

Remark: Eigenvalues come in handy for computing *inverse* matrices. Let $AX = X\Lambda$ be the eigendecomposition of A, which can be rewritten as $A = X\Lambda X'$. We are looking for a matrix B such that $BA = I_n$ (this fact follows from the definition of the inverse matrix). Notice that A needs to be positive definite in order to have an inverse. So, we want $BX\Lambda X' = I_n$ or $BX\Lambda = X$ or $BX\Lambda\Lambda^{-1} = X\Lambda^{-1}$ or $B = X\Lambda^{-1}X'$. Thus, $A^{-1} = X\Lambda^{-1}X'$.

Geometry: We restrict our attention to real matrices. The real matrix A is among other things a linear transformation of $\mathbb{R}^n$ into $\mathbb{R}^n$. Thus, it associates a vector $y \in \mathbb{R}^n$ with each $x \in \mathbb{R}^n$ by the rule $y = Ax$. If a real vector $x \in \mathbb{R}^n$ is an eigenvector of A, then $Ax = \lambda x$ implies that λ is real. Moreover, the eigen-equation says that A transforms x into a multiple of itself, i.e. x is an *invariant direction* of the transformation.



Eigendecompositions

不太懂

The Generalized Eigenvalue Problem: Suppose that we have a $n \times n$ *symmetric* matrix A and another $n \times n$ *symmetric and positive definite* matrix B.

Then any number λ and any nonzero vector x that satisfy

(12) $$Ax = \lambda Bx$$

defines a *generalized eigenvalue* problem.

Reduction to a Classical Eigenvalue Problem:

Define $y = B^{1/2}x$ (where $B^{1/2} = V\Lambda^{1/2}V'$ since it is symmetric and positive definite).

Then, $x = B^{-1/2}y$ (where $B^{-1/2} = V\Lambda^{-1/2}V'$). So, (12) can be written as

(13) $$Ax = \lambda Bx \Leftrightarrow AB^{-1/2}y = \lambda BB^{-1/2}y \Leftrightarrow B^{-1/2}AB^{-1/2}y = \lambda y$$
$$\Leftrightarrow \bar{A}y = \lambda y,$$

which reduces to a regular eigenvalue problem.

Eigendecompositions

Note that λ in (13) corresponds to the stationary/critical values of the ratio

$$\frac{x^* Ax}{x^* Bx} \quad \text{(called the } \textit{generalized Rayleigh quotient}\text{)}$$

where x^* represents the *conjugate transpose* ($x^* = x'$ if it is real). $\square$

不太懂 **Theorem:** If $A_{n\times n}$ is nonsingular, then $A'A$ is positive definite.

<*Proof*> Since A is nonsingular, $Ax \neq \mathbf{0}$ for any nonzero vector x (i.e., $\mathbf{0}$ does not to the column space).

Thus,

$$x'(A'A)x = (Ax)'(Ax) = \|Ax\|^2 > 0.$$

$\square$

Remark: If there's no collinearity of variables in a data matrix, the covariance (or correlation) matrix are positive definite, thus having positive eigenvalues.

Singular Value Decomposition (SVD)

不太懂

Definition: Suppose that there exist matrices X of order $n \times r$, Σ diagonal and of order $r \times r$ and Y of order $m \times r$, with X and Y orthonormal, such that they satisfy the following two equations:

(14) $$AY = X\Sigma$$

(15) $$A'X = Y\Sigma$$

Then,

(16) $$A = X\Sigma Y'$$

gives the *Singular Value Decomposition* (SVD) of A and $\Sigma = X'AY$ is called the *canonical form* of A. The diagonal elements of Σ are called the *singular values* of A, matrix X contains the *left singular vectors* and matrix Y contains the *right singular vectors* of A, respectively.

Singular Value Decomposition (SVD)

Relation of SVD and Eigendecomposition:
From (16) we get that

$$A'A = A'X\Sigma Y' = Y\Sigma\Sigma Y' = Y\Sigma^2 Y',$$

which implies that the "squared singular value" of A are the "eigenvalues" of $A'A$.
Similarly, we get that

$$AA' = X\Sigma^2 X'.$$

Theorem: (Schmidt-Beltrami-Eckart-Young)
Let $B = \sum_{i=1}^{k} X(:,i)\Sigma(:,i)Y'(:,i)$, that is, form the matrix B by keeping the first k columns of X, Y, as well as the first k singular values. Then B is the *best least squares rank k* approximation of A. In other words, it minimizes

$$trace(A'A - B'B) \text{ over all } B.$$

An Interesting SVD Application: The Orthogonal Procrustes Problem

Main Idea: There are two data sets and we would like by *translating* and *rotating* one of them, to make them as "similar" as possible. For example in the field of morphometrics the data points (called landmarks) describe the essential features of a biological form (e.g. head, face, etc). This is the main idea behind the Orthogonal Procrustes problem.

Notation: Let X and Y be the two $n \times m$ data matrices (see Figures 1 & 2). One of them (say X) is considered to be the *target*, and we are interested in making Y as similar as possible to the target.

Fig. 1

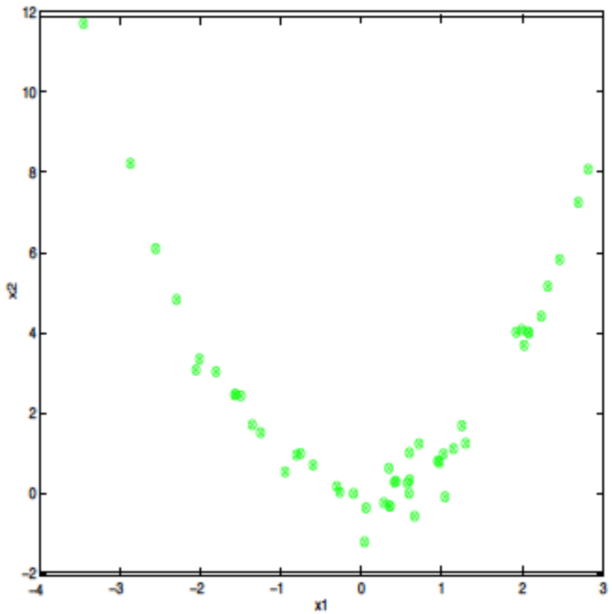
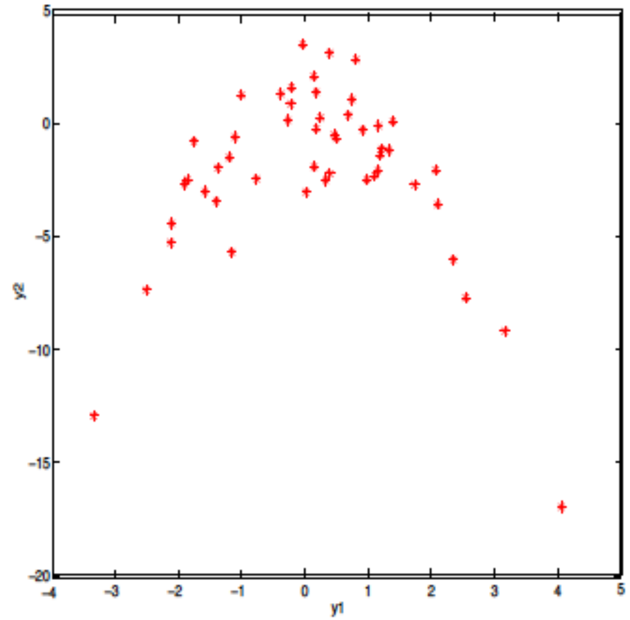


Fig. 2



An Interesting SVD Application: The Orthogonal Procrustes Problem

The Mathematical Problem: In what follows we assume that we have already translated the Y data set so that X and Y have the same "centers." So, we are only interested in rotating Y. We need first to define how we measure similarity between the two data sets. The appropriate measure is given by the Frobenius norm, which is defined for any $n \times m$ matrix as $||A||_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{M} a_{ij}^2\right)^{1/2}$.

Hence, we are interested in finding an orthonormal matrix B (i.e. $B'B = I_m$) that would minimize $||X - YB||_F$. In words, we would like to rotate the Y data set in such a way that would minimize the sum of squared differences between the target data set and the rotated one.

Solution: It is easier to work with $(||X - YB||_F)^2$.

(17)

$$\begin{aligned}(||X - YB||_F)^2 &= \textbf{trace}(X - YB)'(X - YB) = \textbf{trace}(X'X + B'Y'YB - 2B'Y'X) \\ &= \textbf{trace}(X'X) + \textbf{trace}(Y'YBB') - 2\textbf{trace}(B'Y'X) \\ &= \textbf{trace}(X'X) + \textbf{trace}(Y'Y) - 2\textbf{trace}(B'Y'X),\end{aligned}$$

An Interesting SVD Application: The Orthogonal Procrustes Problem

where the last equality follows from the orthonormality constraint on the matrix B.

Eq. (17) shows that in order to minimize $(||X - YB||_F)^2$, it suffices to maximize

$$\mathbf{trace}(B'Y'X).$$

Define $Z = Y'X$ and let us consider the Singular Value Decomposition (SVD) of the matrix Z. It is given by $Z = U\Sigma V'$ with Σ an $m \times m$ diagonal matrix containing the *singular values* and U, V two orthonormal matrices containing the *left* and *right singular vectors* respectively. We then have

$$\begin{aligned} \mathbf{trace}(B'Z) = \mathbf{trace}(B'U\Sigma V') = \mathbf{trace}(V'B'U\Sigma) = \mathbf{trace}(C\Sigma) \\ = \sum_{i=1}^{m} C(i,i)\Sigma(i,i), \end{aligned} \tag{18}$$

where $C = V'B'U$ and where the last equality follows from the definition of the trace.

An Interesting SVD Application: The Orthogonal Procrustes Problem

But due to the fact that U, V and B are orthonormal matrices the following inequality holds:

(19) $$\sum_{i=1}^{m} C(i,i)\Sigma(i,i) \leq \sum_{i=1}^{m} \Sigma(i,i).$$

Notice that by setting $C(i,i) = 1$ for all $i = 1, ..., m$, we achieve the **maximum** upper bound in (19). Set $B = UV'$ and C becomes

(20) $$C = V'B'U = V'(UV')'U = V'VU'U = I_m,$$

as required, where the last equality follows from the fact that U and V are orthonormal matrices.

Hence the optimal B in the Orthogonal Procrustes problem is given by $B = UV'$, the orthogonal polar factor of $Y'X$.

□

An Interesting SVD Application: The Orthogonal Procrustes Problem

For the data sets shown in Figures 1 and 2, the optimal B is given by

$$B = \begin{bmatrix} 0.9989 & 0.0479 \\ 0.0479 & -0.9989 \end{bmatrix}$$

and the resulting picture is given in Figure 3.

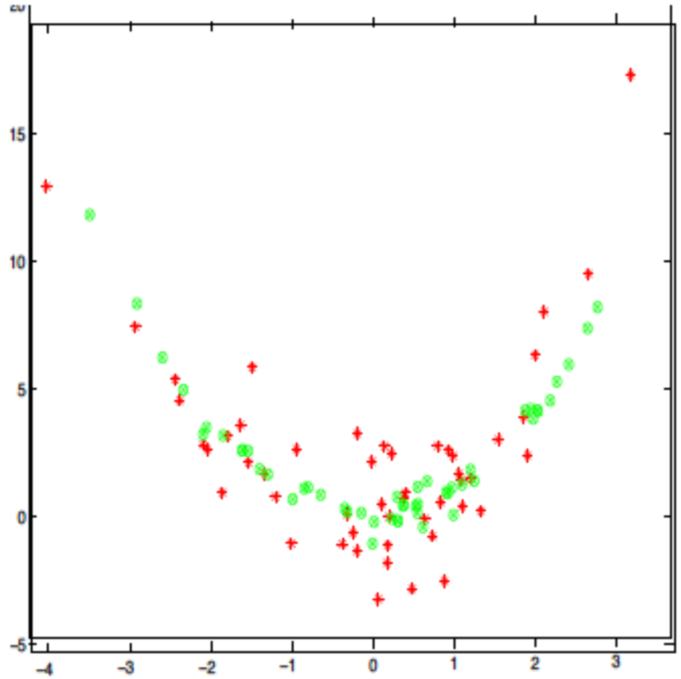


FIGURE 3. Target and rotated data sets shown together

Another Interesting SVD Application: Latent Semantic Indexing

Latent Semantic Indexing

Latent Semantic Indexing (LSI) is an information retrieval technique based on the SVD. Variations of this technique are used by many Web search engines (for more details check the recent book by Berry and Browne, Understanding Search Engines: Mathematical Modeling and Text Retrieval, SIAM Book Series: Software, Environments, and Tools, June 1999).

The main problem in information retrieval is to *match* terms in documents with those appearing in the query. For example, suppose that you are interested in finding which books deal with "multivariate analysis." Then, your query consists of these two terms. In the old days information was retrieved by literal matching, i.e. books whose titles contained these two terms were only returned. The main shortcoming of literal matching is the presence of synonyms. Suppose that your query consisted of the term "car." But documents with the words "vehicle" or "automobile" in their titles would also be relevant to you, as well as documents containing the words "carmakers," "engine," "chassis," etc. Another problem with literal matching is that many words have multiple meanings (polysemy), so terms in a user's query will literally match terms in irrelevant documents. LSI attempts to overcome these difficulties by allowing users to retrieve information on the basis of a conceptual topic or meaning of a document.

Another Interesting SVD Application: Latent Semantic Indexing

The technique assumes that there is some underlying (latent) structure in word usage that is partially obscured by variability in word choice. An SVD is used to estimate the structure in word usage across documents. Retrieval is then performed using the singular values and vectors. To clarify things let us look at an example taken from Berry et al. paper (Berry, M.W., Dumais, S.T, and O'Brien, G.W. (1995), Using Linear Algebra for Intelligent Information Retrieval, SIAM Review). The following table contains a list of titles from books reviewed in SIAM review.

Another Interesting SVD Application: Latent Semantic Indexing

Label	Title
B1	A Course on Integral Equations
B2	Attractors for Semigroups and Evolution Equations
B3	Automatic Differentiation of Algorithms: Theory, Implementation, and Application
B4	Geometrical Aspects of Partial Differential Equations
B5	Ideals, Varieties and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra
B6	Intorduction to Hamiltonian Dynamical Systems and the N-body Problem
B7	Knapsack Problems: ALgorithms and Computer Implementations
B8	Methods of Solving Singular Systems of Ordinary Differential Equations
B9	Nonlinear Systems
B10	Ordinary Differential Equations
B11	Oscillation Theory for Neutral Differential Equations with Delay
B12	Oscillation Theory of Delay Differential Equations
B13	Pseudodifferential Operatos and Nonlinear Partial Differentail Equations
B14	Sinc Methods for Quadrature and Differential Equations
B15	Stability of Stochastic Differential Equations with respect to Semimartingales
B16	The Boundary Integral Approach to Static and Dynamic Contact Problems
B17	The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

Another Interesting SVD Application: Latent Semantic Indexing

The terms we are going to examine are: algorithms, application, delay, differential, equations, implementation, integral, introduction, methods, nonlinear, ordinary, oscillation, partial, problem, systems, theory.

So, we can construct a 16x17 terms by documents 0-1 data matrix (terms corresponds to rows, documents to columns) where A(i,j)=1 indicates that the i^{th} term appears in the title of the j^{th} book and 0 otherwise. The data matrix A is given in Table 1.

Remark: Usually in practice the data matrix A has entries that denote the frequency in which term i occurs in document j.

Another Interesting SVD Application: Latent Semantic Indexing

Terms	Documents																
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
algorithms	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
application	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
delay	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
differential	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
equations	1	1	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
implementation	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
integral	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
introduction	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
methods	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
nonlinear	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
ordinary	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
oscillation	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
partial	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
problem	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
systems	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0
theory	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1

TABLE 1. Data matrix

Another Interesting SVD Application: Latent Semantic Indexing

Now we will make use of the Eckart-Young Theorem to find the best k-rank approximation (in the least squares sense) to the matrix A. For illustrative purposes I will use $k = 2$. The idea is that that the first $k = 2$ *factors* will capture most of the important underlying structure in the association of terms and documents. Intuitively, since $k = 2$ is much smaller than 16, the number of unique terms, minor differences in terminology will be ignored.

Let $A = U\Sigma V'$. In the LSI context U provides information about the terms, while V about the documents. The singular values are given next:
4.5353, 2.9542, 2.6228, 1.9856, 1.8004, 1.7346, 1.6326, 1.2359, 1.0576, 1.0000, 0.8215, 0.6161 0.4718 0.2093 0.0000, 0.0000

It can be seen that the first 3 singular values are much larger than the remaining ones, which implies that the first 3 factors approximate reasonably well the original data. However, I will use only 2 factors to visualize the results.

Another Interesting SVD Application: Latent Semantic Indexing

Recall that $A = U\Sigma V' = (U\Sigma^{1/2})(\Sigma^{1/2}V') = (U\Sigma^{1/2})(V\Sigma^{1/2})' = XY',$

$X = U_{16\times 2}\Sigma_{2\times 2}^{1/2}$: row information (term information)

$Y = V_{17\times 2}\Sigma_{2\times 2}^{1/2}$: column information (document information)

Then, we can plot the terms (red points) on the plane (the first column values in X on the x-axis and the second column values in X on the y-axis) together with the documents (green points) (the first column values in Y on the x-axis and the second column values in Y on the y-axis).

The resulting picture is shown in Figure 4.

Another Interesting SVD Application: Latent Semantic Indexing

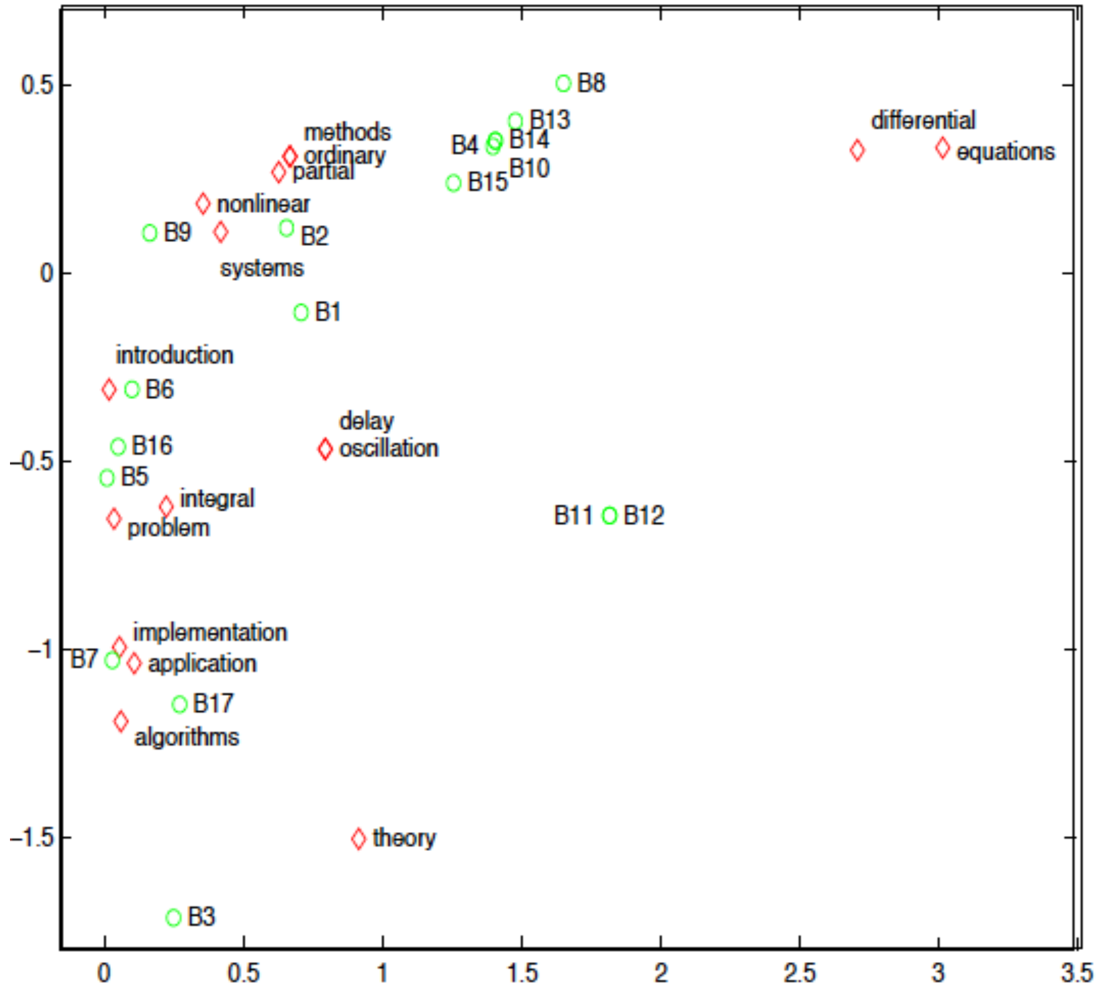


FIGURE 4. Terms (red) and Documents (green)

Another Interesting SVD Application: Latent Semantic Indexing

It can be seen that documents B4, B13, B14 and B15 have similar titles dealing with partial differential equations (they are located between partial, methods, nonlinear, differential, equations), while documents B3, B7, B16 and B17 deal with integrals, problems, implementation and algorithms to a large extent. If I have used $k = 3$ I would have obtained a better and more accurate representation.

Suppose that you issue a query containing the terms (application, theory).
I can present it by the vector a = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1).
Then, its position in the above picture can be found by

$$q = a_{1\times16}U_{16\times2}\Sigma_{2\times2}^{-1/2}.$$

It is plotted as the "blue square" in Figure 5 and indicates that the document best matching this query is B3 followed by B17. In LSI there are techniques of determining which documents to be retrieved that go beyond simple visualization.

Another Interesting SVD Application: Latent Semantic Indexing

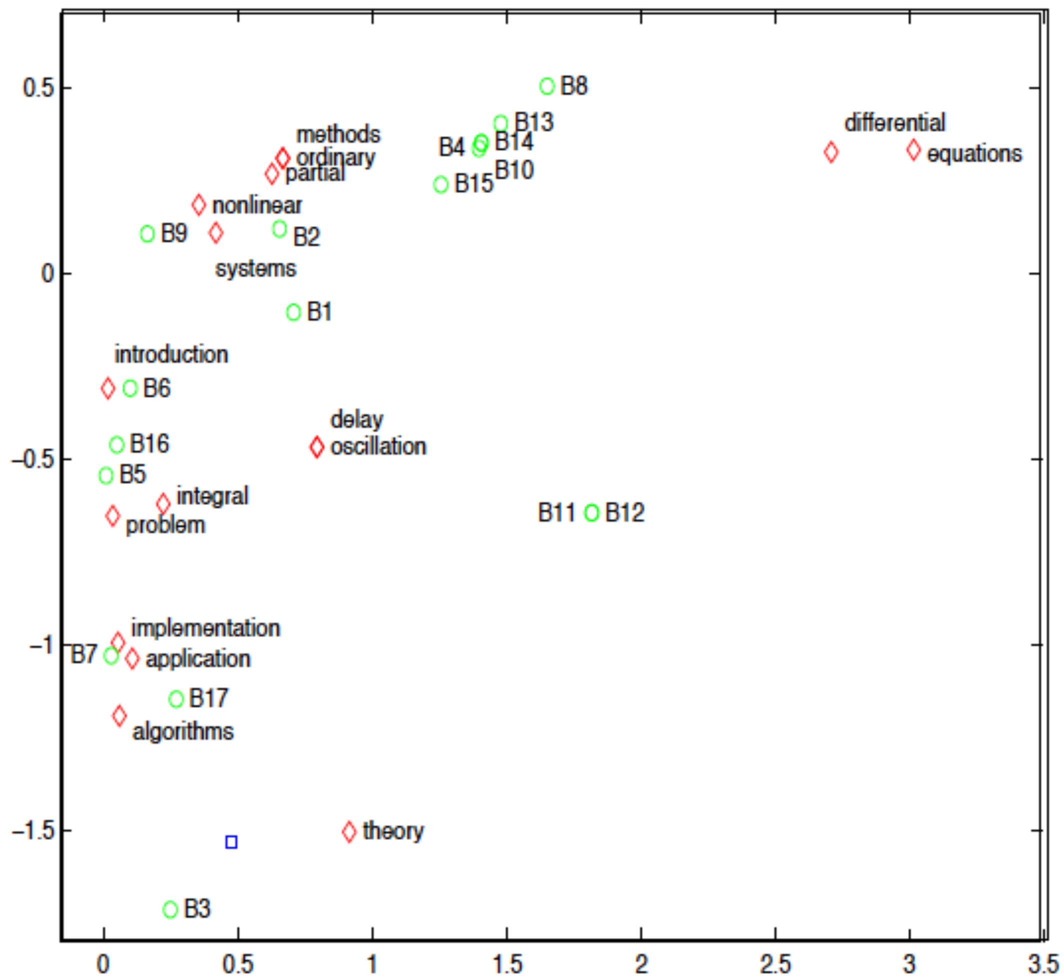


FIGURE 5. Terms (red) and Documents (green) and Query (blue square)

Another Interesting SVD Application: Latent Semantic Indexing

References:

Golub, G. and Van Loan, C. (1998), *Matrix Computations*, (3rd ed), Johns Hopkins University Press

Harville, D. (1997), *Matrix Algebra from a Statisticians Perspective*, Springer

Magnus, J. and Neudecker, H. (1988), *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley

Searle, S. (1982), *Matrix Algebra Useful for Statistics*, Wiley