A FIRST COURSE

IN

LINEAR ALGEBRA

A FIRST COURSE

IN

LINEAR ALGEBRA

MAT2040 Notebook

Prof. Tom Luo

The Chinese University of Hongkong, Shenzhen

Prof. Ruoyu Sun

University of Illinois Urbana-Champaign

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Survey Methodology / Robert M. Groves . . . [et al.].
p. cm.—(Wiley series in survey methodology)
"Wiley-Interscience."
Includes bibliographical references and index.
ISBN 0-471-48348-6 (pbk.)
1. Surveys—Methodology. 2. Social sciences—Research—Statistical methods. I. Groves, Robert M. II. Series.

HA31.2.S873 2004
001.4'33—dc22 2004044064
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Contents

Contributors

Zhi-quan Luo, Shenzhen Research Institute of Big Data, Lecturer

Ruoyu Sun, Industrial and Enterprise Systems Engineering, Lecturer

Jie Wang, The Chinese University of Hongkong, Shenzhen, Typer

Foreword

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Preface

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

place
date

Acknowledgments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

I. R. S.

Acronyms

ASTA	Arrivals See Time Averages
BHCA	Busy Hour Call Attempts
BR	Bandwidth Reservation
b.u.	bandwidth unit(s)
CAC	Call / Connection Admission Control
CBP	Call Blocking Probability(-ies)
CCS	Centum Call Seconds
CDTM	Connection Dependent Threshold Model
CS	Complete Sharing
DiffServ	Differentiated Services
EMLM	Erlang Multirate Loss Model
erl	The Erlang unit of traffic-load
FIFO	First in - First out
GB	Global balance
GoS	Grade of Service
ICT	Information and Communication Technology
IntServ	Integrated Services
IP	Internet Protocol
ITU-T	International Telecommunication Unit – Standardization sector
LB	Local balance
LHS	Left hand side

LIFO	Last in - First out
MMPP	Markov Modulated Poisson Process
MPLS	Multiple Protocol Labeling Switching
MRM	Multi-Retry Model
MTM	Multi-Threshold Model
PASTA	Poisson Arrivals See Time Averages
PDF	Probability Distribution Function
pdf	probability density function
PFS	Product Form Solution
QoS	Quality of Service
r.v.	random variable(s)
RED	random early detection
RHS	Right hand side
RLA	Reduced Load Approximation
SIRO	service in random order
SRM	Single-Retry Model
STM	Single-Threshold Model
TCP	Transport Control Protocol
TH	Threshold(s)
UDP	User Datagram Protocol

1.3. Friday

1.3.1. Matrix Multiplication

1.3.1.1. How to compute matrix multiplication quickly?

Given $m \times n$ matrix $\boldsymbol{A}$ and $n \times k$ matrix $\boldsymbol{B}$, then the result of $\boldsymbol{AB}$ should be a $m \times k$ matrix.

Let's show a specific example:

■ **Example 1.8** Given 4×3 matrix $\boldsymbol{A}$ and 3×2 matrix $\boldsymbol{B}$, then the result of $\boldsymbol{AB}$ should be a 4×2 matrix:

$$\boldsymbol{AB} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}_{4\times 2}.$$

- The (i,j)th entry of the result should be the **inner product** between the ith row of $\boldsymbol{A}$ and the jth column of $\boldsymbol{B}$.

  SInce the result has 4×2 entries, we have to process such progress 4×2 times to obtain the final result.

- But we can try a more effecient method. We can calculate the *entire row* of the result more easily.

  – For example, note that

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}.$$

The first row of the result is the linear combination of the row of matrix

B, and the coefficients are entries of the first row of matrix A:

$$\begin{bmatrix}1\end{bmatrix}\times\begin{bmatrix}1&1\end{bmatrix}+\begin{bmatrix}2\end{bmatrix}\times\begin{bmatrix}1&1\end{bmatrix}+\begin{bmatrix}3\end{bmatrix}\times\begin{bmatrix}1&1\end{bmatrix}=\begin{bmatrix}6&1\end{bmatrix}.$$

– On the other hand, we can also calculate the *entire column* of the result quickly:

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\\10&11&12\end{bmatrix}\begin{bmatrix}1&1\\1&0\\1&0\end{bmatrix}=\begin{bmatrix}6&\times\\15&\times\\24&\times\\33&\times\end{bmatrix}.$$

The first column of the result is the linear combination of the column of matrix A, and the coefficients are entries of the first column of matrix B:

$$\begin{bmatrix}1\\4\\7\\10\end{bmatrix}\begin{bmatrix}1\end{bmatrix}+\begin{bmatrix}2\\5\\8\\11\end{bmatrix}\begin{bmatrix}1\end{bmatrix}+\begin{bmatrix}3\\6\\9\\12\end{bmatrix}\begin{bmatrix}1\end{bmatrix}=\begin{bmatrix}6\\15\\24\\33\end{bmatrix}.$$

You can do the remaining calculation by yourself, and the final result is given by:

$$\boldsymbol{AB}=\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\\10&11&12\end{bmatrix}_{4\times3}\times\begin{bmatrix}1&1\\1&0\\1&0\end{bmatrix}_{3\times2}=\begin{bmatrix}6&1\\15&4\\24&7\\33&10\end{bmatrix}_{4\times2}.$$

1.3.2. Elementary Matrix

So let's review the concept for elementary matrix by an example:

R In this course you can think there is **only one** type of elementary matrix. This may contradict what you see in the textbook.

Definition 1.11 [Elementary Matrix] An elementary matrix $\boldsymbol{E}_{ij}$ is a matrix that its *diagonal entries* are all 1 and the (i,j)th column is a *scalar*, and the remaining entries are all *zero*. ■

For example, the matrix $\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix}$ is elementary matrix.

■ **Example 1.9** Given vector $\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^{\mathrm{T}}$ and elementary matrix $\boldsymbol{E}_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -l_{31} & 0 & 1 \end{bmatrix}$, the effct of *postmultiplying* $\boldsymbol{E}_{31}$ for $\boldsymbol{b}$ has the same effect of doing row operation:

$$\boldsymbol{E}_{31}\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 - l_{31}b_1 \end{bmatrix}$$

Let's do more practice. Given matrix $\boldsymbol{E}_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, we can calculate the result of $\boldsymbol{E}_{21} \times (\boldsymbol{E}_{31}\boldsymbol{b})$ and $\boldsymbol{E}_{21}\boldsymbol{E}_{31}$:

$$\boldsymbol{E}_{21} \times (\boldsymbol{E}_{31}\boldsymbol{b}) = \begin{bmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 - l_{31}b_1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - l_{21}b_1 \\ b_3 - l_{31}b_1 \end{bmatrix}$$

$$\boldsymbol{E}_{21}\boldsymbol{E}_{31} = \begin{bmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -l_{31} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ -l_{31} & 0 & 1 \end{bmatrix}$$

Additionally, we can use matrix multiplication to derive the result of $(\boldsymbol{E}_{21}\boldsymbol{E}_{31}) \times \boldsymbol{b}$:

$$(\boldsymbol{E}_{21}\boldsymbol{E}_{31}) \times \boldsymbol{b} = \begin{bmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ -l_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - l_{21}b_1 \\ b_3 - l_{31}b_1 \end{bmatrix}$$

Amazingly, we find that the result of $\boldsymbol{E}_{21} \times (\boldsymbol{E}_{31}\boldsymbol{b})$ is actually the same as $(\boldsymbol{E}_{21}\boldsymbol{E}_{31}) \times \boldsymbol{b}$, which is one of the properties of matrix. ■

1.3.3. Properties of Matrix

Operations on matrix has the following properties:

1. $\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC}$.
2. $\boldsymbol{AB} \neq \boldsymbol{BA}$, i.e., $\boldsymbol{AB}$ doesn't *necessarily* equal to $\boldsymbol{BA}$.

   R In some special cases, $\boldsymbol{AB}$ may equal to $\boldsymbol{BA}$. For example, for elementary matrix, we have $\boldsymbol{E}_{21}\boldsymbol{E}_{31} = \boldsymbol{E}_{31}\boldsymbol{E}_{21}$, this means the order of row operation can be changed sometimes.

   However, for most cases the equality is not satisfied. given row vector $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$ and column vector $\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, the result of $\boldsymbol{ab}$ and $\boldsymbol{ba}$ is given by:

$$\boldsymbol{ab} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1b_1 + a_2b_2 + a_3b_3$$

$$\boldsymbol{ba} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} b_1a_1 & b_1a_2 & b_1a_3 \\ b_2a_1 & b_2a_2 & b_2a_3 \\ b_3a_1 & b_3a_2 & b_3a_3 \end{pmatrix}.$$

3. **Block Multiplication**. We use an example to show the process of block multiplicaion:

■ **Example 1.10** Given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we want to compute $\boldsymbol{C} := \boldsymbol{A} \times \boldsymbol{B}$, which can be done by **block multiplication**. We can partition $\boldsymbol{A}$ and $\boldsymbol{B}$ with appropriate sizes. For example,

$$\boldsymbol{A} = \left[\begin{array}{cc|c} 4 & 0 & 4 \\ 6 & 6 & 8 \\ \hline -9 & 5 & -8 \end{array}\right] = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix}, \quad \boldsymbol{B} = \left[\begin{array}{cc|c} 8 & -3 & -7 \\ 3 & -7 & -4 \\ \hline 4 & -4 & 1 \end{array}\right] = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{B}_3 & \boldsymbol{B}_4 \end{bmatrix}.$$

Then $\boldsymbol{A}$ and $\boldsymbol{B}$ could be considered as 2×2 **block matrices**. As a result, $\boldsymbol{C}$ have 2×2 blocks:

$$\boldsymbol{AB} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \boldsymbol{B}_3 & \boldsymbol{B}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B}_1 + \boldsymbol{A}_2\boldsymbol{B}_3 & \boldsymbol{A}_1\boldsymbol{B}_2 + \boldsymbol{A}_2\boldsymbol{B}_4 \\ \boldsymbol{A}_3\boldsymbol{B}_1 + \boldsymbol{A}_4\boldsymbol{B}_3 & \boldsymbol{A}_3\boldsymbol{B}_2 + \boldsymbol{A}_4\boldsymbol{B}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \\ \boldsymbol{C}_3 & \boldsymbol{C}_4 \end{bmatrix}$$

As a result, there is an effective way to calculate $\boldsymbol{C}_1$, that is the block multiplication method shown below:

$$\boldsymbol{C}_1 = \boldsymbol{A}_1\boldsymbol{B}_1 + \boldsymbol{A}_2\boldsymbol{B}_3 = \begin{bmatrix} 4 & 0 \\ 6 & 6 \end{bmatrix} \begin{bmatrix} 8 & -3 \\ 3 & -7 \end{bmatrix} + \begin{bmatrix} 4 \\ -8 \end{bmatrix} \begin{bmatrix} 4 & -4 \end{bmatrix} = \begin{bmatrix} 48 & -28 \\ 34 & -28 \end{bmatrix}.$$

You can do the remaining calculation to get result of $\boldsymbol{AB}$:

$$\boldsymbol{AB} = \boldsymbol{C} = \left[\begin{array}{cc|c} 48 & -28 & -24 \\ 34 & -28 & -74 \\ \hline -89 & 24 & 35 \end{array}\right].$$

■

There are also two useful ways to compute $\boldsymbol{AB}$:

- If $\boldsymbol{B}$ has k columns, we can partition $\boldsymbol{B}$ into k blocks to compute $\boldsymbol{AB}$:

$$\boldsymbol{AB} = \boldsymbol{A} \times \left[\begin{array}{c|c|c|c} B_1 & B_2 & \dots & B_k \end{array}\right] = \left[\begin{array}{c|c|c|c} \boldsymbol{A}B_1 & \boldsymbol{A}B_2 & \dots & \boldsymbol{A}B_k \end{array}\right].$$

- If $\boldsymbol{A}$ has m rows, we can partition $\boldsymbol{A}$ into m blocks to compute $\boldsymbol{AB}$:

$$\boldsymbol{AB} = \begin{bmatrix} \boldsymbol{A}_1 \\ \hline \boldsymbol{A}_2 \\ \hline \dots \\ \hline \boldsymbol{A}_m \end{bmatrix} \times \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B} \\ \hline \boldsymbol{A}_2\boldsymbol{B} \\ \hline \dots \\ \hline \boldsymbol{A}_m\boldsymbol{B} \end{bmatrix}$$

1.3.4. Permutation Matrix

Note that there also exists one kind of matrix $\boldsymbol{P}$ such that postmultiplying $\boldsymbol{P}$ for arbitararily matrix $\boldsymbol{A}$ has the same effect of interchanging two rows of $\boldsymbol{A}$.

For example, if $\boldsymbol{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then by postmultiplying $\boldsymbol{P}$ for $\boldsymbol{A}$ we obtain:

$$\boldsymbol{PA} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}.$$

This progress has the same effect of interchanging the first row and the second row of $\boldsymbol{A}$.

This kind of matrix is called **permutaion matrix**:

Definition 1.12 [Permutation Matrix] $\boldsymbol{P}$ is a **permutation matrix** if postmultiplying $\boldsymbol{P}$ for matrix $\boldsymbol{A}$ has the same effect of interchanging rows of matrix $\boldsymbol{A}$. ■

Definition 1.13 [Row Exchange Matrix] $\boldsymbol{P}$ is a **row exchange matrix** if postmultiplying $\boldsymbol{P}$ for matrix $\boldsymbol{A}$ has the same effect of interchanging only two rows of matrix $\boldsymbol{A}$.

We use the notation $\boldsymbol{P}_{ij}$ to denote a matrix that has the effect of exchanging row i and row j of $\boldsymbol{A}$. ■

The way to obtain $\boldsymbol{P}_{ij}$ is simple. After an identity matrix's ith and jth row being exchanged, we could obtain the row exchange matrix $\boldsymbol{P}_{ij}$.

Let's raise some examples to show what is row exchange matrix:

■ **Example 1.11** P_{23} has the effect of exchanging 2th row and 3th row of arbitarary matrix. It is converted from an identity matrix:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xRightarrow{\text{Interchange row 2 and 3}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = P_{23}.$$

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xRightarrow{\text{Interchange row 2 and 3}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = P_{23}.$$

Postmultiplying by P_{23} exchanges row 2 and row 3 of any matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 6 & 7 \\ 15 & 4 \\ 24 & 3 \end{bmatrix} = \begin{bmatrix} 6 & 7 \\ 24 & 3 \\ 15 & 4 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 24 \\ 15 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \\ 24 \\ 4 \end{bmatrix}$$

■

R You may be confused about the concept between *permutation matrix* and *row exchange matrix.* The row exchange matrix is a special case of permutation matrix, but permutation matrix could exchange several rows. For example, row 1,2,3,4 could be changed into row 4,3,2,1.

Before talking about the properties of permutation matrix, let's introduce the definition for nonsingular and inverse matrix:

Definition 1.14 [Nonsigular matrix] Let A be an $n \times n$ matrix, the following statements are equivalent:

1. A is **nonsingular** or **invertible**.
2. There exists a matrix B such that $AB = BA = I$. And the matrix B is said to be

the **inverse** of $\boldsymbol{A}$, and we can write $\boldsymbol{B} = \boldsymbol{A}^{-1}$.

3. After multiplying finite numbers of **elementary matrix**, $\boldsymbol{A}$ can be converted to identity matrix $\boldsymbol{I}$.
4. The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a unique solution.

If matrix $\boldsymbol{A}$ is **not** nonsingular, this matrix is called **singular**.

We are interested in the inverse of permutation matrix.

Proposition 1.3

1. For a permutation matrix $\boldsymbol{P}$, it can always be decomposed into finite multiplications of row exchange matrices $\boldsymbol{P}_{ij}$:

$$\boldsymbol{P} = \boldsymbol{P}_{i_1 j_1} \boldsymbol{P}_{i_2 j_2} \dots \boldsymbol{P}_{i_n j_n}$$

2. The inverse of a row exchange matrix is actually equal to itself:

$$\boldsymbol{P}_{ij} \boldsymbol{P}_{ij} = \boldsymbol{I} \Longleftrightarrow \boldsymbol{P}_{ij}^{-1} = \boldsymbol{P}_{ij}$$

3. For a permutation matrix written as $\boldsymbol{P} = \boldsymbol{P}_{i_1 j_1} \boldsymbol{P}_{i_2 j_2} \dots \boldsymbol{P}_{i_n j_n}$, its inverse matrix is given by:

$$\boldsymbol{P}^{-1} = \boldsymbol{P}_{i_n j_n}^{-1} \boldsymbol{P}_{i_{n-1} j_{n-1}}^{-1} \dots \boldsymbol{P}_{i_1 j_1}^{-1} = \boldsymbol{P}_{i_n j_n} \boldsymbol{P}_{i_{n-1} j_{n-1}} \dots \boldsymbol{P}_{i_1 j_1}$$

4. For a $n \times n$ permutation matrix $\boldsymbol{P}$ and a $n \times n$ matrix $\boldsymbol{A}$ given by:

$$\boldsymbol{P} = \left[\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & & & \\ \vdots & & \boldsymbol{P}_{(n-1)\times(n-1)} & \\ 0 & & & \end{array}\right] \qquad \boldsymbol{A} = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{A}_{(n-1)\times(n-1)} & \\ 0 & & & \end{array}\right]$$

the multiplication result $\boldsymbol{P}\boldsymbol{A}$ has the form:

$$\boldsymbol{P}\boldsymbol{A} = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{P}_{(n-1)\times(n-1)} \boldsymbol{A}_{(n-1)\times(n-1)} & \\ 0 & & & \end{array}\right]$$

Proofoutline.

- For proposition 2, it is because that if we exchange two rows of any matrix $\boldsymbol{A}$, and then we exchange the same rows again, the effect is cancelled out!
- For proposition 3, it is because that we just need to do the reverse order of our process in order to obtain the inverse matrix.

■

1.3.5. LU decomposition

After learning matrix multiplication, we should be familiar some basic results of matrix multiplication:

1. **Product of upper triangular matries is also an upper triangular matrix.**

$$
\begin{bmatrix}
\times & \times & \times & \times & \times \\
 & \times & \times & \times & \times \\
 & & \times & \times & \times \\
 & \mathbf{0} & & \times & \times \\
 & & & & \times
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
 & \times & \times & \times & \times \\
 & & \times & \times & \times \\
 & \mathbf{0} & & \times & \times \\
 & & & & \times
\end{bmatrix}
=
\begin{bmatrix}
\times & \times & \times & \times & \times \\
 & \times & \times & \times & \times \\
 & & \times & \times & \times \\
 & \mathbf{0} & & \times & \times \\
 & & & & \times
\end{bmatrix}.
$$

2. **Product of diagonal matrices is also a diagonal matrix.**

$$
\begin{bmatrix}
d_1 & & & \\
 & d_2 & & \mathbf{0} \\
 & & \ddots & \\
 & \mathbf{0} & & \ddots & \\
 & & & & d_n
\end{bmatrix}
\begin{bmatrix}
e_1 & & & \\
 & e_2 & & \mathbf{0} \\
 & & \ddots & \\
 & \mathbf{0} & & \ddots & \\
 & & & & e_n
\end{bmatrix}
=
\begin{bmatrix}
d_1e_1 & & & \\
 & d_2e_2 & & \mathbf{0} \\
 & & \ddots & \\
 & \mathbf{0} & & \ddots & \\
 & & & & d_ne_n
\end{bmatrix}.
$$

Just like permutation matrix, there are also some intersting properties of elementary matrix:

Proposition 1.4

The inverse of an elementary matrix is also an elementary matrix.

■ **Example 1.12** $E_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is an elementary matrix, the result of postmultiplying E_{21} for identity matrix is given by:

$$E_{21}I = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which has the same effect of adding $(-2)\times$ row 1 to row 2 of I. How to get the identity matrix again? We just need to add $2\times$ row 1 to row 2 of I, which could be viewed as postmultiply another elementary matrix for I:

$$\overline{E_{21}}(E_{21}I) = \overline{E_{21}}E_{21} = \overline{E_{21}} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I.$$

Hence, $\overline{E_{21}}$ is the inverse matrix of E_{21}, which is also an elementary matrix. ■

The elementary matrix $E_{ij}(i < j)$ is a lower triangular matrix; and $E_{ij}(i > j)$ is an upper triangular matrix. Let's look at an example:

■ **Example 1.13** Let's try Gaussian Elimination for a matrix that is nonsingular. Here we use elementary matrix to describle row operation above the arrow (without row exchange):

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \xrightarrow{E_{21}} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ -2 & 7 & 2 \end{bmatrix} \xrightarrow{E_{31}} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 8 & 3 \end{bmatrix} \xrightarrow{E_{32}} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = U$$

In this process we have

$$E_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad E_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Finally we convert A into an upper triangular matrix U. Let's do the reverse of this process to find some interesting results:

$$\begin{aligned} E_{32}E_{31}E_{21}A &= U \\ \implies E_{32}^{-1}E_{32}E_{31}E_{21}A = E_{32}^{-1}U &\implies E_{31}E_{21}A = E_{32}^{-1}U \\ \cdots \implies A = E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}U &:= LU, \end{aligned}$$

where $L = E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}$, which is lower triangular matrix.

Hence, we successfully decompose matrix A into the multiplication of a lower triangular matrix L and a upper triangular matrix U. ■

Actually, any nonsingular matrix *without row exchanges, i.e., does not require the row exchange during the Gaussian Elimination*, could be decomposed as the multiplication of a lower triangular matrix with a upper triangular matrix U, which is called **LU decomposition**.

1.3.5.1. One Square System = Two Triangular Systems

When considering the *nonsingular* case without row exchanges, recall what we have done before this lecture:

we are working on A and b in **one** equation $Ax = b$.

To somplify computation, we aim to deal with A and b in **separate** equations. The LU decomposition can help us do that:

1. **Decomposition:** By Gaussian elimination on matrix A, we can decompose A into matrix multiplications: $A = LU$.
2. **Solve:** forward elimination on b using L, then back substitution for x using U.

R

The detail of Solve process.

(a) First, we apply forward elimination on $\boldsymbol{b}$. In other words, we are actually solving $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$.

(b) After getting $\boldsymbol{y}$, we then do back substitution for $\boldsymbol{x}$. In other words, we are actually solving $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$.

One square system = Two triangular systems. During this process, the original system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is converted into two triangular systems:

Forward and Backward Solve $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$ and then solve $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$.

There is nothing new about those steps. This is exactly what we have done all the time. We are really solving the triangular system $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$ as elimination went forward. Then we use back substitution to produce $\boldsymbol{x}$. An example shows what we actually did:

■ **Example 1.14** Forward elimination on $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ will result in equation $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Longleftrightarrow \begin{cases} u + 2v = 5 \\ 4u + 9v = 21 \end{cases} \text{forward elimination implies} \begin{cases} u + 2v = 5 \\ v = 1 \end{cases} \Longleftrightarrow \boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}.$$

We could express such process into matrix form:

LU Decomposition. : We could decompose $\boldsymbol{A}$ into product of $\boldsymbol{L}$ and $\boldsymbol{U}$:

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

$\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$. In this system of equation, in oder to solve $\boldsymbol{y}$, we only need to multiply the inverse of $\boldsymbol{L}$ both sides:

$$\begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \times \boldsymbol{y} = \begin{bmatrix} 5 \\ 21 \end{bmatrix} \implies \boldsymbol{y} = \boldsymbol{L}^{-1} \begin{bmatrix} 5 \\ 21 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 21 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

$\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$. In this system of equation, in oder to solve $\boldsymbol{x}$, we only need to multiply the inverse of $\boldsymbol{U}$ both sides:

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \times \boldsymbol{x} = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \implies \boldsymbol{x} = \boldsymbol{U}^{-1} \begin{bmatrix} 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Both **Forward** and **Back substitution** has $O(n^2)$ time complexity. ■

1.3.6. LDU decomposition

The aim of LDU decomposition is to let the diagonal entries of $\boldsymbol{U}$ and $\boldsymbol{L}$ to be **one**.

Suppose we have decomposed $\boldsymbol{A}$ into $\boldsymbol{L}\boldsymbol{U}$, where the upper triangular matrix $\boldsymbol{U}$ is given by:

$$\begin{bmatrix} d_1 & \times & \times & \times & \times \\ & d_2 & \times & \times & \times \\ & & d_3 & \times & \times \\ & 0 & & d_4 & \times \\ & & & & d_5 \end{bmatrix}$$

If we want to set its diagonal entries of $\boldsymbol{U}$ to be all **one**, we just need to multiply a matrix $\boldsymbol{D}^{-1}$ that is given by:

$$\boldsymbol{D}^{-1} := \begin{bmatrix} d_1^{-1} & & & & \\ & d_2^{-1} & & 0 & \\ & & d_3^{-1} & & \\ & 0 & & d_4^{-1} & \\ & & & & d_5^{-1} \end{bmatrix} \implies \boldsymbol{D}^{-1}\boldsymbol{U} = \begin{bmatrix} 1 & \times & \times & \times & \times \\ & 1 & \times & \times & \times \\ & & 1 & \times & \times \\ & 0 & & 1 & \times \\ & & & & 1 \end{bmatrix}.$$

We can convert LU decomposition into LDU decomposition by simply adding the multiplying factor $\boldsymbol{D}\boldsymbol{D}^{-1}$:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{D}^{-1}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{D}(\boldsymbol{D}^{-1}\boldsymbol{U}) = \boldsymbol{L}\boldsymbol{D}\hat{\boldsymbol{U}},$$

where $\hat{\boldsymbol{U}} = \boldsymbol{D}^{-1}\boldsymbol{U}$ is also an upper triangular matix.

Here $\boldsymbol{D}$ is the inverse matrix of $\boldsymbol{D}^{-1}$:

$$\boldsymbol{D} = \begin{bmatrix} d_1 & & & & \\ & d_2 & & 0 & \\ & & d_3 & & \\ & 0 & & d_4 & \\ & & & & d_5 \end{bmatrix}$$

Note that the *diagonal* entries of $\boldsymbol{D}$ are all **pivots values** of $\boldsymbol{U}$.

Similarly, we can also proceed this step again to let *diagonal* entries of $\boldsymbol{L}$ to be **one**.

Definition 1.15 [LDU Decomposition] In conclusion, we decompose matrix $\boldsymbol{A}$ into the form:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{U}$$

where: $\boldsymbol{L}$ is lower triangular matrix with unit entries in diagonal

$\boldsymbol{D}$ is diagonal matrix

$\boldsymbol{U}$ is upper triangular matrix with unit entries in diagonal

This decomposition is called **LDU decomposition**. ■

Here is a property of LDU decomposition, the proof of which is omitted.

Proposition 1.5 **LDU decomposition is unique to any matrix**. Let $\boldsymbol{L}, \boldsymbol{L}_1$ denote a lower triangular matrix, $\boldsymbol{D}, \boldsymbol{D}_1$ diagonal, and $\boldsymbol{U}, \boldsymbol{U}_1$ upper triangular.

If $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{U}$, and also, $\boldsymbol{A} = \boldsymbol{L}_1\boldsymbol{D}_1\boldsymbol{U}_1$, then we have $\boldsymbol{L} = \boldsymbol{L}_1, \boldsymbol{D} = \boldsymbol{D}_1, \boldsymbol{U} = \boldsymbol{U}_1$.

1.3.7. LU Decomposition with row exchanges

How can we handle row exchange in our $\boldsymbol{LU}$ decomposition?

Assume we are going to do Gaussian Elimination with matrix $\boldsymbol{A}$ with row exchange.

- At first We can postmultiply some elementary matrices $\boldsymbol{E}$ to get $\boldsymbol{EEEA}$.
- Sometimes we need to multiply by $\boldsymbol{P}_{ij}$ to do *row exchange* to continue Gaussian Elimination.
- So we may end our elimination with something like $\boldsymbol{PEEEEPEEEPEEEEA}$.
- If we can get all the elementary matrix $\boldsymbol{L}$ together, we could convert them into one single $\boldsymbol{L}$ that has the same effect as before.
- The key problem is that how can we get all the row exchange matrix $\boldsymbol{P}$ out from the elementary matrices?

Theorem 1.1 If $\boldsymbol{A}$ is *nonsingular*, then there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{PA} = \boldsymbol{LU}$.

The proof is omitted.

R For the nonsingular matrix $\boldsymbol{A}$ **without** row exchange, we can always decompose it as $\boldsymbol{A} = \boldsymbol{LU}$; but for the row exchange case, we have to postmultiply a specific permutation matrix to obtain such LU decomposition.