Experiment-3
Solving $Ax = b$ by Least-Squares

(Duration: 120 mins)

Author: Oğulcan Erdoğan
ogulcanerdogan@iyte.edu.tr

Purpose: The least-squares solutions have a broad range of applications from data science to physics and more. Recognizing and solving such problems may give deep insight to anyone who wants to use advanced computational techniques. After this experiment, you will understand the basic idea for this method and you will solve a simplified fictional problem.

Introduction

In this experiment, we will consider the Netflix database and we assume that it could be similar to Figure 1. There are two address variables namely x_1 and x_2 which represent the location of each film or TV series digitally. That figure could belong to someone who enjoyed watching comedy films. The blue and red points on the figure are already watched films and they have been labeled according to this user's votes. $+1$ represents the liked films while -1 represents unliked films. Here is the definition of the problem: this system (Netflix) aims to suggest new films for a user based on him/her own votes. This is not tricky since the system has lots of knowledge about the category (sample space) and the labeled films. So what is the system waiting for? The system first tries to find a separation plane between liked or not liked films with respect to user's votes. After finding such a plane, the system uses it to make a prediction which film is the best for this user. For example, which films is the most likely suggested by the system to this user, s_1 or s_2?

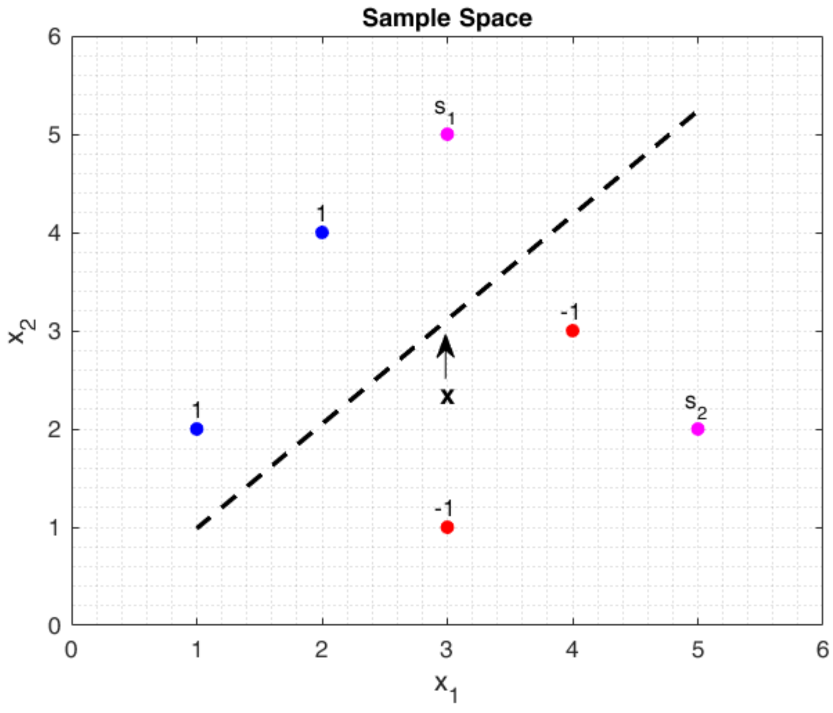


Figure 1: Sample Space of Database

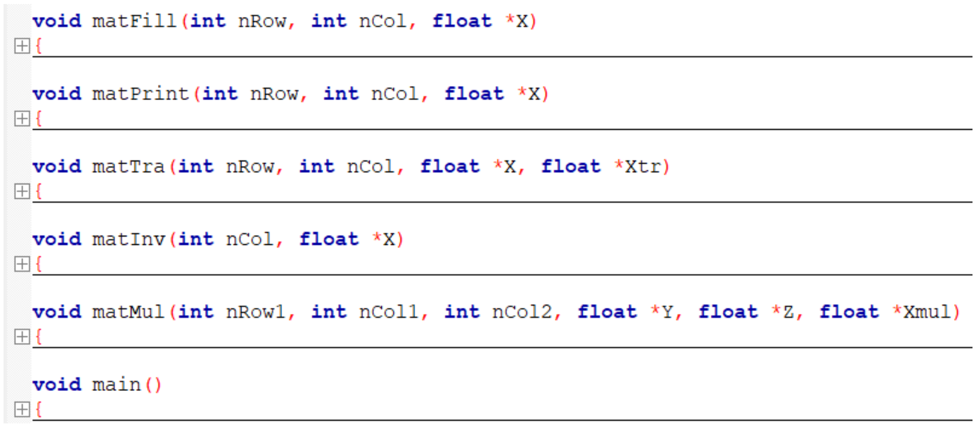
Problem Statement

In this problem, you will find a separation line **x** between blue and red points in the Figure 1. For this operation, you need to first construct a matrix **A** with labeled films based on the location addresses. After that, you will construct another vector **b** which includes $+1$ and -1 corresponding labels. By the way, **x** is an unknown vector represented by $[x_1\, x_2]^T$. Now, you know **A** and **b** and you can solve **Ax=b** to find unknown **x** by the least-squares method.

After finding the separation line **x**, you can define another matrix $\mathbf{A}_{new}$ which consists of location addresses of s_1 and s_2. Then you can obtain the new label vector $\mathbf{b}_{new}$ by solving $\mathbf{b}_{new} = \mathbf{A}_{new}\mathbf{x}$. This new label vector is a suggestion vector including possible user votes for respective films. According to its values, the system can make a good suggestion to the user!

You need to follow the below steps to get full credit! The running program in bash shell should look in Figure 3 (on the last page). An ideal program should:

- Ask to user for the dimensions of matrix **A** (**10pts**)
- Obtain the matrix **A** and label vector **b** from the user (**10pts**)
- Write the required functions to solve $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ where $()^T$ indicates the transpose operation and $()^{-1}$ denotes as an inverse operation. This solution is called the least-squares solution and for further information you can look at (*https://textbooks.math.gatech.edu/ila/least-squares.html*). **Note that:** You need to perform one inverse operation, one transpose operation with storing its values, and three multiplication operations for solving only this equation. It is more desirable to use a function for each operation. ***Thus you are asked to write your codes by similar (not necessarily the same) functions given in Figure 2, otherwise you will get only a 20% grade of this part even when your code works correctly!*** (**Remember the *Pointers*!**) (**50pts**)



```
void matFill(int nRow, int nCol, float *X)
{

void matPrint(int nRow, int nCol, float *X)
{

void matTra(int nRow, int nCol, float *X, float *Xtr)
{

void matInv(int nCol, float *X)
{

void matMul(int nRow1, int nCol1, int nCol2, float *Y, float *Z, float *Xmul)
{

void main()
{
```

Figure 2: Definition of some necessary functions as an example

- Ask to user for the dimensions of new matrix $\mathbf{A}_{new}$ (**10pts**)
- Compute $\mathbf{b}_{new} = \mathbf{A}_{new}\mathbf{x}$ where **x** is your solution. Compare each label value in $\mathbf{b}_{new}$ whether greater than "0" or not. If this specific value is greater than "0", change it to "+1" otherwise change it to "-1" and print the prediction or suggestion results. (**20pts**)

It is expected that your code must give the correct results even if we enter different matrix A and vector b!!

More about Least-Squares

The problem defined here is basically $\mathbf{Ax} = \mathbf{b}$. Suppose the matrix $\mathbf{A}$ has dimensions m by n and the right hand side vector $\mathbf{b}$ has dimension m. Then the solution $\mathbf{x}$, if it exists, has to have dimension n. In fact, there are three different case for such problem:

- If $m<n$, i.e. we have less equations than unknowns, which is called **underdetermined**. In general there were be infinitely many solutions. It's also possible that there are no solutions because the equations do not provide the exact solutions (inconsistent).
- If $m=n$ and $\mathbf{A}$ is invertible, there is one **unique** (exact) solution. In this case, the least-squares solution is reduced to $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.
- If $m>n$, i.e. we have more equations than unknowns, which is called **overdetermined** and we mostly encounter this type of system in many engineering problems! In this case, we can use the least-squares criterion to determine a solution. Instead of demanding that $\mathbf{Ax} = \mathbf{b}$, we look for an $\mathbf{x}$ than makes the difference between $\mathbf{Ax}$ and $\mathbf{b}$ as small as possible such that,

$$\min \, ||\mathbf{Ax} - \mathbf{b}||^2$$

We further expand this minimization problem with the matrix notation,

$$= \min \, (\mathbf{Ax} - \mathbf{b})^T(\mathbf{Ax} - \mathbf{b})$$

$$= \min \, (\mathbf{x}^T\mathbf{A}^T - \mathbf{b}^T)(\mathbf{Ax} - \mathbf{b})$$

$$= \min \, \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}$$

To find the solution $\mathbf{x}$ for this minimization, we should first take the derivative with respect to $\mathbf{x}$ and set it to "0".

$$\frac{\partial}{\partial \mathbf{x}} \, ||\mathbf{Ax} - \mathbf{b}||^2 = 0$$

$$\frac{\partial}{\partial \mathbf{x}} \, (\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}) = 0$$

$$2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = 0$$

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

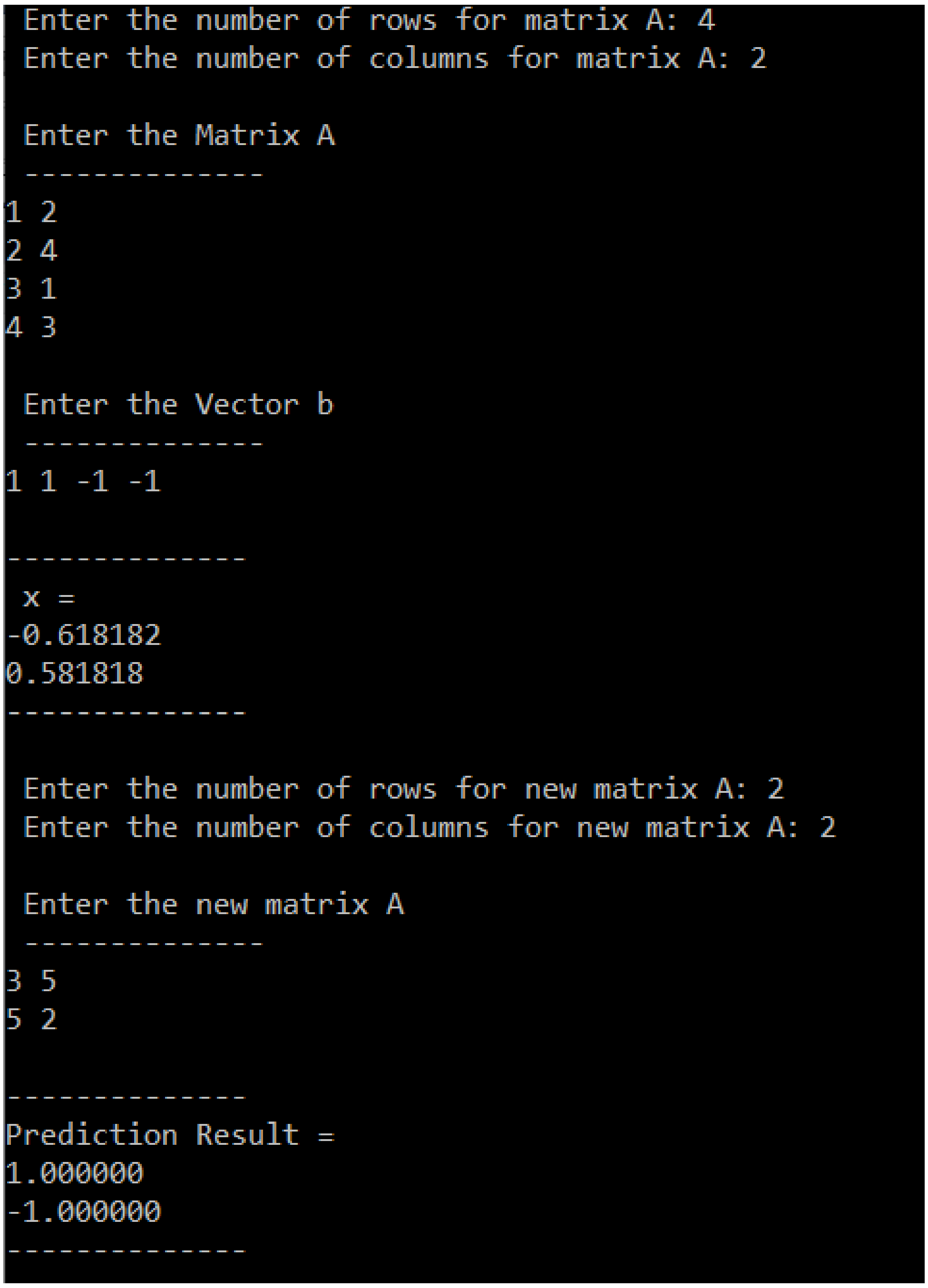
$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

Multiply for both sides with $(\mathbf{A}^T\mathbf{A})^{-1}$,

$$(\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{A})\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

which gives us the final result as a least-squares solution,

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$



```
 Enter the number of rows for matrix A: 4
 Enter the number of columns for matrix A: 2

 Enter the Matrix A
 --------------
1 2
2 4
3 1
4 3

 Enter the Vector b
 --------------
1 1 -1 -1

--------------
 x =
-0.618182
0.581818
--------------

 Enter the number of rows for new matrix A: 2
 Enter the number of columns for new matrix A: 2

 Enter the new matrix A
 --------------
3 5
5 2

--------------
Prediction Result =
1.000000
-1.000000
--------------
```

Figure 3: Snapshot from Bash Shell