Projects 1

Chapter 1 projects

Gas in a Tube

(Shores 62)

Problem Description: You are given a long tube of still dry air in which there are 7 sampling/insertion points equally spaced 1/6 meters apart from each other. The position of each point is measured by setting the leftmost point at 0.0 meters and rightmost at 1.0 meters. Initially, a small amount of a certain gas is inserted in the central insertion point. Subsequently, measurements of the concentration of the gas at each sampling/insertion point are taken at later times in seconds. The results of these measurements, which you may assume are accurate to about 2-3 digits, are specified in Table 1.1. Based on this information, your task is to determine the best estimate you can find for the true value of the diffusion coefficient D of this gas in a motionless air medium. Use this estimate and a marching method to calculate values of the material density function on the interval $[0, 1]$ at times $t = 210$ and $t = 300$ and at the given spatial nodes.

Sec \ Meter	0	1/6	1/3	1/2	2/3	5/6	1
$t = 240$	0.0	0.032	1.23	3.69	1.23	0.032	0.0
$t = 270$	0.0	0.051	1.21	3.48	1.21	0.051	0.0

Table: Concentration data measurements of a gaseous material.

Procedure: You should use equation (1.10) or some variant to move backward and forward in time. These will result in linear systems, which ALAMA calculator or another technology tool can solve. One way to proceed is simply to use trial and error until you think you've hit on a reasonable value of D, that is, the one that gives the best approximation to $t = 180$ from the $t = 360$ values. Do not expect perfect matches - the data is relatively sparse. Then march backwards in time once more to get the initial values at $t = 0$. Finally, march forward in time to compute and plot the resulting approximate density function.

Output: Discus your results and provide a graph of profiles of the material density function at times in the data table along with your computed profiles.

Comments: This project introduces you to a very interesting area of mathematics called "inverse theory." The idea is, rather than proceeding from problem (the governing equations for concentration values) to solution (concentration profiles), you are given the "solution," namely the measured solution values at various points, and are to determine from this information the "problem," i.e., the diffusion coefficient needed to define the governing equations.

Chapter 2 Projects

LU Factorization

We didn't yet get to the discussion of this in class. If you're interested in doing this project, you can look at the second part of the Chapter 2 Lecture 4 slides.

(Shores p. 177) Write a program module that implements Theorem 2.14 using partial pivoting and implicit row exchanges. This means that space is allocated for the $n \times n$ matrix $A = [a[i, j]]$ and an array of row indices, say indx $[i]$. Initially, indx should consist of the integers $1, 2, \ldots, n$. Whenever two rows need to be exchanged, say the first and third, then the indices indx[1] and indx[3] are exchanged. References to array elements throughout the Gaussian elimination process should be indirect: Refer to the $(1, 4)$ th entry of A as the element $a[\text{indx}[1], 4]$. This method of reference has the same effect as physically exchanging rows, but without the work. It also has the appealing feature that we can design the algorithm as though no row exchanges have taken place provided we replace the direct reference $a[i, j]$ by the indirect reference $a[\text{indx}[i], j]$. The module should return the lower/upper matrix in the format of Example 2.70 as well as the permuted array indx $[i]$. Effectively, this index array tells the user what the permutation matrix P is.

Use this module to implement an LU system solver module that uses the LU factorization to solve a general linear system. Also write a module that finds the inverse of an $n \times n$ matrix A by first using the LU factorization module, then making repeated use of the LU system solver to solve $A\mathbf{x}^{(i)} = \mathbf{e}_i$, where $\mathbf{e}_i$ is the i th column of the identity. Then we will have

$$A^{-1} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right]$$

Be sure to document and test your code and report on the results.

Markov Chains

(Shores p. 177)

Refer to Example 2.19 and Section 2.3 for background. Three automobile insurance firms compete for a fixed market of customers. Annual premiums are sold to these customers. Label the companies A, B, and C. You work for Company A, and your team of market analysts

has done a survey that draws the following conclusions: In each of the past three years, the number of A customers switching to B is 20%, and to C is 30%. The number of B customers switching to A is 20%, and to C is 20%. The number of C customers switching to A is 30%, and to B is 10%. Those who do not switch continue to use their current company's insurance for the next year. Model this market as a Markov chain. Display the transition matrix for the model. Illustrate the workings of the model by showing what it would predict as the market shares three years from now if currently A, B, and C owned equal shares of the market.

The next part of your problem is as follows: Your team has tested two advertising campaigns in some smaller test markets and are confident that the first campaign will convince 20% of the B customers who would otherwise stay with B in a given year to switch to A. The second advertising campaign would convince 20% of the C customers who would otherwise stay with C in a given year to switch to A. Both campaigns have about equal costs and would not change other customers' habits. Make a recommendation, based on your experiments with various possible initial state vectors for the market. Will these campaigns actually improve your company's market share? If so, which one do you recommend? Write up your recommendation in the form of a report, with supporting evidence. It's a good idea to hedge on your bets a little by pointing out limitations to your model and claims, so devote a few sentences to those points.

It would be a plus to carry the analysis further (your manager might appreciate that). For instance, you could turn the additional market share from, say B customers, into a variable and plot the long-term gain for your company against this variable. A manager could use this data to decide whether it was worthwhile to attempt gaining more customers from B.

Sports Ranking

(Shores p. 180)

Refer to Example 2.24 and Section 2.3 for background. As a sports analyst you are given the following data about a league of seven teams numbered 1-7, where the pair (j, k) represents a game in which team j defeated team k :

$$E = \{(1, 2), (7, 3), (2, 4), (4, 5), (3, 2), (5, 1), (6, 1), (3, 1), (7, 2), (2, 6),$$
$$(3, 4), (7, 4), (5, 7), (6, 4), (3, 5), (5, 6), (7, 1), (5, 2), (7, 6), (1, 4), (6, 3)\}$$

Based on these data you are to rank the teams. To this end, begin with the simplest method, ranking by win/loss record. Next, treat the data as defining a digraph. Begin this analysis by constructing the adjacency matrix of this digraph and drawing a picture of the digraph either by hand or using some software. Then rank the teams by using the following methods: First use the method of Example 2.26 to find a power ranking of each team. Then use the reverse PageRank idea of Example 2.47 to rank the the teams.

Next, suppose you are given additional information, namely, the game margins (winning score minus losing score) for each game. Following is a list of these margins matching the order of matches in the definition of E :

$$M = \{4, 8, 7, 3, 7, 7, 23, 15, 6, 18, 13, 14, 7, 13, 7, 18, 45, 10, 19, 14, 13\}$$

In order to utilize these data examine your picture of the digraph and label each edge with the margin that matches it in M. You are now dealing with a weighted graph and one can construct a different sort of "adjacency matrix" by entering this margin in the (i, j) th entry according as team i defeated team j by that margin. Use this approach to calculate "power ranking".

IsoRank

Read the discussion of *IsoRank* on pages 172-176.

The notion of embedding one graph into another is a useful idea for some scientific studies. In this report you will test the basic idea of network embedding by using the variant IsoRank of the PageRank technique on three relatively simple examples. This project requires a technology tool for these calculations and the resulting output should be interpreted as in the discussion of the IsoRank technique following Example 2.73 of Section 2.8.
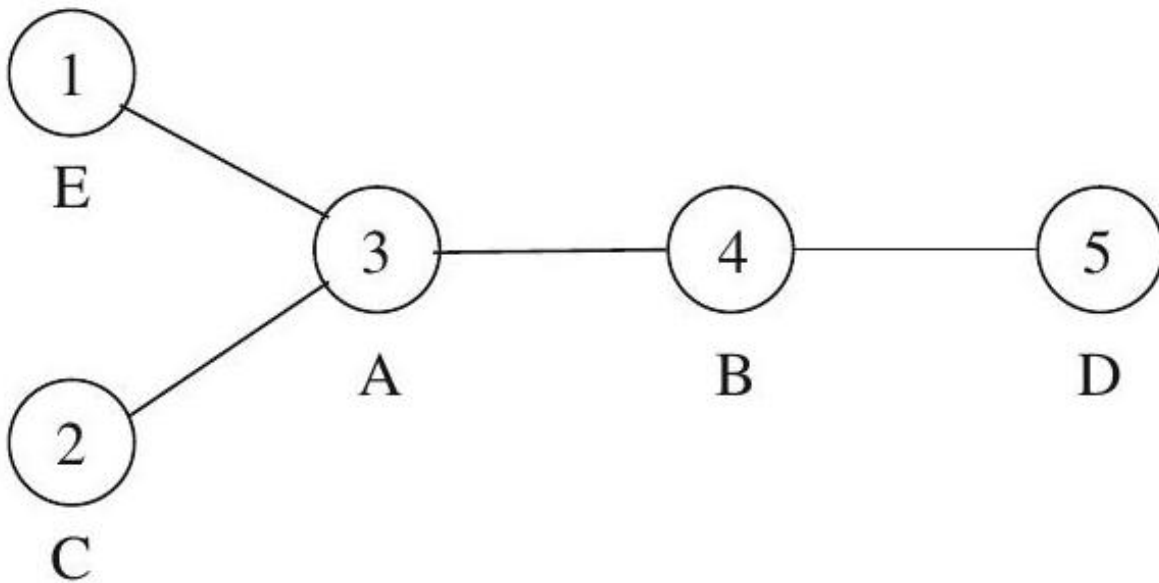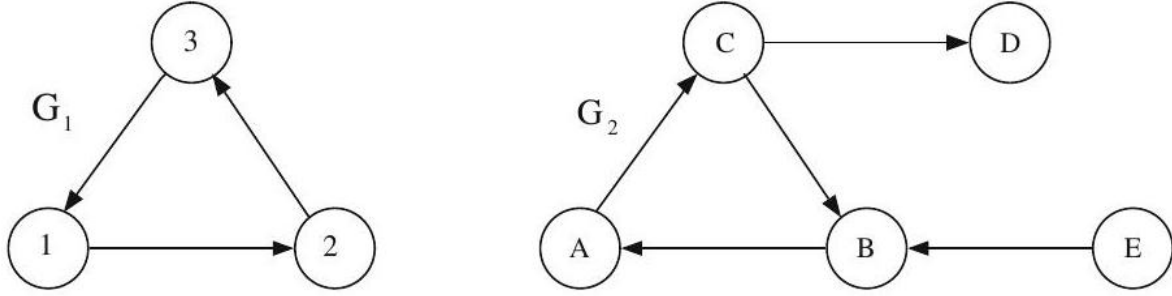

Figure 1: Figure 2.13

Figure 2: Figure 2.14

By an isomorphism of graphs we mean a one-to-one edge preserving map of vertices from one graph onto another. One can think of an isomorphism as simply a relabeling of the vertices of a graph. The first test is to provide an example of how well IsoRank can recognize isomorphisms. Consider the graph of Figure 2.13. Let G_1 be the graph with vertices $1, 2, 3, 4, 5$ in that order and G_2 the same graph with vertices A, B, C, D, E in that order. Apply IsoRank to these two graphs and discuss the validity of your results.

The next embedding test is to remove the edge connecting vertices B and C in Figure 2.12 and use IsoRank with teleportation vector $\mathbf{v} = \mathbf{e}/15$ and teleportation parameter $\alpha = 0.85$ to find the best matchings of the graph G_1 with the resulting graph G_2. List all possible mappings that are calculated.

The last embedding test is to use IsoRank with teleportation vector $\mathbf{v} = \mathbf{e}/15$ and teleportation parameter $\alpha = 0.85$, along with correction vector $\mathbf{u} = \mathbf{e}/5$ for G_2, to find the best matchings of the digraph G_1 with the digraph G_2 in Figure 2.14. List all possible mappings and discuss your calculations.