Applied Linear Algebra for Business, Economics and Finance

Nathaniel Karst
Division of Mathematics and Science
Babson College

January 22, 2013

Contents

0.1

Chapter 1

An introduction to linear systems

1.1 Linear systems and their solutions

You probably encountered the idea of a line quite a while ago in your mathematical career. You might remember something like

$$y = \frac{2}{3}x + 4.$$

We used words like "slope" and "y-intercept" to glean information about how these functions behaved. And since you're interested in the applications of mathematics to business, you probably used linear functions like the one above to model things like total cost, total revenue, supply, demand, population or any number of other quantities. Let's remind ourselves how these models worked by developing one from scratch.

Imagine you have a business selling t-shirts. You're a data-driven business person, and your records tell you that when you sold each t-shirt for \$20, you sold 400 shirts a month, but when you increased the price to \$25 per shirt, you only sold 300 units. The line connecting these points is

$$q = -20p + 800,$$

where q is the demand in units of t-shirts, and p is the price in units of dollars. Let's start by making a slight but important change to the way we've been thinking about lines. We can rearrange the equation above to read

$$q + 20p = 800.$$

Here we call q and p *variables* or *unknowns* exactly because we don't yet know exactly what they are. The numbers in front of the variables are known as *coefficients*. And for one more bit of important terminology, we say that 800 is a *linear combination* of q and p. Notice that by rearranging the equation, we haven't fundamentally changed anything, we've just put the variables q and p on equal footing instead of thinking of q as a function of p.

Any pair (q, p) that satisfies the equation above is called a *solution* to the equation. For instance, (20,400) and (25,300) are both solutions to our featured equation. You can probably convince yourself that there are many, many more solutions.

Reading question 1: What are some other solutions to $q + 20p = 800$? Try drawing these points in the q-p plane. How many solutions are there?

We can make things more interesting by adding another equation into the mix. Your supplier isn't willing to just give the shirts away. Even more, she'd like to sell you more shirts at a higher price. The past two deals you've made with her have been for 50 t-shirts for \$15 each and 100 t-shirts for \$20 each. If you assume your supplier's willingness to sell you t-shirts grows linearly with the price, we could write a linear model of her behavior as

$$q = 10p - 100.$$

We'd like to find a point where we're selling just as many t-shirts as our supplier is willing to provide, that is, where the supply equals the demand. So let's consider the system of linear equations (or *linear system*) defining the supply and demand together.

$$\begin{aligned} q + 20p &= 800 \\ q - 10p &= -100. \end{aligned}$$

So our question about supply equalling demand becomes: does there exist at least one pair (p, q) that satisfies both equations simultaneously? To get a qualitative handle on the situation, let's try graphing both lines. It's a simple but profound idea that the points lying on the graphical line we associate with either of linear equations above are exactly the solutions to the linear equation in question. It stands to reason then that if we plot the two lines and they have an intersection, then the (p, q) pair representing the intersection simultaneously satisfy both equations.

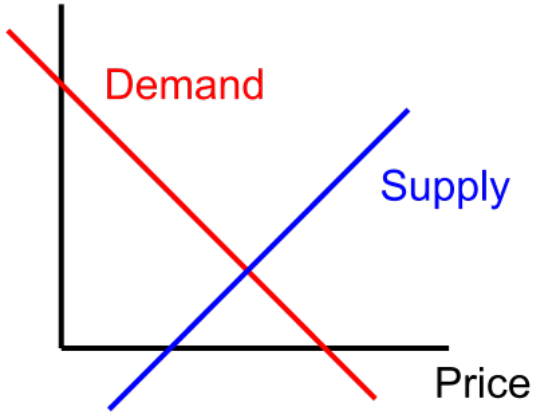

Figure 1.1: Our linear supply and demand models have a unique equilibrium price.

This type of reasoning leads us to the first hint of a generalizable result: we know from the old days of a pencil and a ruler that two lines either have no intersection (if they are parallel), infinitely many intersection points (if the two lines lie directly on one another) and exactly one intersection point in all other cases. We will see that this type of reasoning generalizes to much more broad circumstances.

When it comes to solving systems of linear equations, the words "elimination" and "substitution" might ring a bell. (We're going to focus on elimination here, but substitution is just as valid.) The elimination technique involves scaling one of the equations by multiplying both sides by a carefully chosen number

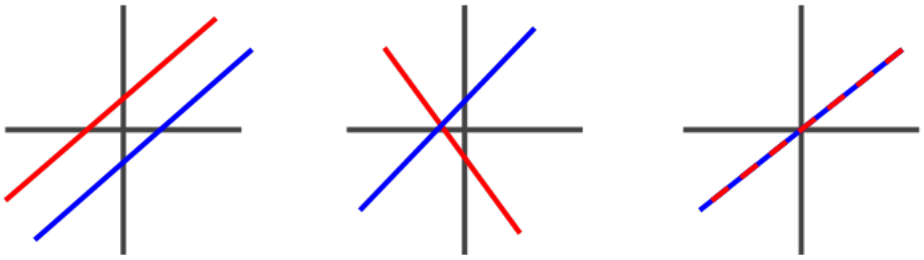

Figure 1.2: A system of two linear equations can have no solutions (parallel lines), a unique solution (nonidentical nonparallel lines) or infinitely many solutions (identical lines).

and adding two equations so that at least one variable is no longer present in the sum. For instance, imagine that I scale the second equation in our example by a factor of -1, so that the system now reads

$$\begin{aligned} q + 20p &= 800 \\ -q + 10p &= 100. \end{aligned}$$

At this point, you might be thinking that something is a little fishy. After all, maybe I've changed the solutions of this system in what I've just done. But notice that the solutions to $q - 10p = -100$ are the same as the solutions to $-(q - 10p) = -(-100)$. So everything really is OK.

I can now add the equations together and replace the second equation with this new sum so that the linear system now reads

$$\begin{aligned} q + 20p &= 800 \\ 30p &= 900. \end{aligned}$$

We've *eliminated* the unknown q from the second equation. This allows us to solve for $p = 30$. Let's take the time to be really precise about what we've discovered here: for there to be a solution (p, q) to this system of linear equations, it must be the case that $p = 30$. We don't know what the value or values of q which would lead to a solution, but we definitely have pegged down the value of p.

To finish off this example, we can first scale the second equation by -20, add the two equations, and replace the first equation with the new sum so that the system now takes the form

$$\begin{aligned} q &= 200 \\ p &= 30. \end{aligned}$$

So $p = 30$ and $q = 200$ is the *unique* solution to this system of linear equations. In terms of our original model, we should sell each t-shirt in our inventory for

$30 dollars in order to match supply to demand. When we do this, we'll sell exactly $200.

1.1.1 From systems to matrices

It may seem sort of weird that as we solved for x and y, we never really *did* anything with them. Sure, we manipulated their coefficients by scaling equations, and adding two equations together, but to be honest, the unknowns x and y were really just taking up space. We can leverage this fact to represent the system of linear equations we've been working on in a compact form:

$$\begin{array}{ll} q + 20p & = 800 \\ q - 10p & = -100 \end{array} \quad \leftrightarrow \quad \left[\begin{array}{cc|c} 1 & 20 & 800 \\ 1 & -10 & -100 \end{array}\right]$$

We call the object on the right side the *augmented matrix* of the linear system. In general, an $r \times c$ matrix is just a rectangular array of numbers having r rows and c columns. Nothing too special there. But why the "augmented" part? Well, we could also think about what a matrix containing only the coefficients of the unknowns would tell us about the system. (We'd call this, not shockingly, the *coefficient matrix*). So the augmented matrix of the system is just the coefficient matrix of the system with an additional column representing the right side of the linear system tacked on. It's important to remember that rows represent equations and columns represent unknowns.

The good news is that all the same calculations we performed to solve linear systems when the unknowns x and y were present still work in the new matrix context. But instead of adding and subtracting equations of the linear system, we add and subtract rows of the matrix.

$$\left[\begin{array}{cc|c} 1 & 20 & 800 \\ 1 & -10 & -100 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 20 & 800 \\ -1 & 10 & 100 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 20 & 800 \\ 0 & 30 & 900 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 20 & 800 \\ 0 & 1 & 30 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 0 & 200 \\ 0 & 1 & 30 \end{array}\right]$$

It's a useful exercise to translate an augmented matrix back into a linear system. Often seeing the linear system makes a result make more sense that just staring at a matrix. Let's take the last matrix for an example. Remember that the first column represents the coefficient of the unknown q and the second column represents the coefficient of the unknown p. Reading off the first row gives $1q + 0p = 200$, and reading off the second row gives $0q + 1p = 30$. These are exactly the values that we found using the full equations.

1.1.2 Reduced row echelon form

These days no one solves systems of linear equations by hand; we use a computer. Both elimination and substitution involve quite a few simple calculations. These are time-consuming to do by hand, and it's easy to make minor arithmetic mistakes that ultimately invalidate any subsequent results. Fortunately for us,

computers are both faster and more accurate than we are. The formal procedure used to solve systems of linear equations is known as *Gaussian elimination* in honor of Carl Gauss, a famous mathematician living in the 1800s, though "his" method had been used by Chinese mathematicians over 1000 years earlier. (Yes, that's *three* zeros after the one.)

The industry tool of choice for type of thing is Matlab. Defining a matrix in Matlab is pretty easy. Let's use our running example as our first try. Before we discuss the exact syntax, let's just see it.

```
EDU>> M = [1 20 800; -1 10 100]

M =

     1    20   800
    -1    10   100
```

Square brackets start and stop the matrix, entries are listed across the row, and a semicolon denotes a new row. Here we've set the matrix to the new variable **M**, but we could've called it almost anything or even nothing at all. We could also put commas between the entries of the matrix without changing the output

So now that we have our matrix **M**, how do we find the solutions of the system of liner equations it represents? Well, it's good to know that the final form of the matrix we're looking for is call *reduced row echelon form* (for reasons we'll talk about in a minute). This gives us a convenient way to remember the command: take the first letter from each word and you get `rref`. Let's try taking `rref(M)` in Matlab.

```
EDU>> rref(M)

ans =

     1     0   200
     0     1    30
```

So `rref` takes a matrix as an input and returns to us a matrix as an output. Let's convert the output matrix back into the notation of a linear system.

$$rref(M) = \left[\begin{array}{cc|c} 1 & 0 & 200 \\ 0 & 1 & 30 \end{array}\right] \leftrightarrow \begin{array}{rl} 1q + 0p & = 200 \\ 0q + 1p & = 30 \end{array}$$

So the matrix that `rref(M)` spits out represents the solution we found earlier! This brings us to the first big idea of the course: **If you want to solve a linear system, no matter what kind, no matter where you see it, seriously... any linear system, just `rref` its augmented matrix**.

1.1.3 Free variables

You may have notice in the previous two reading questions that all of the rows contain a 1 which has only 0 entries to its left. These 1 entries are known as *pivots*, and every row will either have a pivot or contain only 0 entries.

But why are pivots important? Well, let's see what happens in an example in which there aren't as many pivots as we might expect. Let's consider the system

$$\begin{aligned} 2x_1 + 3x_2 &= 6 \\ 4x_1 + 6x_2 &= 12. \end{aligned}$$

The corresponding reduced row echelon form is

```
EDU>> rref([2 3 6; 4 6 12])

ans =

    1.0000    1.5000    3.0000
         0         0         0
```

But, wait. Something's strange here. To get a better handle on what's going on here, let's convert this matrix back into a system of linear equations.

$$\left[\begin{array}{cc|c} 1 & 1.5 & 3 \\ 0 & 0 & 0 \end{array}\right] \leftrightarrow \begin{aligned} x_1 + 1.5x_2 &= 3 \\ 0 &= 0 \end{aligned}$$

The second equation of the system is satisfied for any choice of x_1 and x_2, so that's not much help. For the first equation there are an infinite number of solutions, namely all the points that lie on the line $x_1 + 1.5x_2 = 3$ in the x_1–x_2 plane. In other words, this system does not have a unique solution. The really interesting fact is that we could infer this from the reduced row echelon form of the matrix. A column (except the rightmost column) without a pivot in the reduced row echelon form of an augmented matrix represents a *free variable*, and any system with a free variable has infinitely many solutions.

1.1.4 Inconsistencies

From our general discussion earlier, we know that a system of linear equations has either infinitely many solutions, a unique solution, or no solution. We've dealt with the first two, so it shouldn't surprise us that we have one more case to deal with. Let's consider the linear system

$$\begin{aligned} 0.04x_1 + 0.81x_2 &= 0.65 \\ 0.07x_1 + 0.81x_2 &= 0.51 \\ 0.52x_1 + 0.72x_2 &= 0.97 \end{aligned} \quad \leftrightarrow \quad \left[\begin{array}{cc|c} 0.04 & 0.81 & 0.65 \\ 0.07 & 0.81 & 0.51 \\ 0.52 & 0.72 & 0.97 \end{array}\right].$$

A quick Matlab computation gives us the reduced row echelon form of the system

```
EDU>> rref([0.04 0.81 0.65; 0.07 0.81 0.51; 0.52 0.72 0.97])

ans =

     1     0     0
     0     1     0
     0     0     1
```

Let's convert the reduced form back into a system of equations to see what's going on. (Hopefully this methodology is starting to feel natural by now; we'll soon stop pointing out this conversion explicitly.)

$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \leftrightarrow \begin{array}{rl} x_1 & = 0 \\ x_2 & = 0 \\ 0 & = 1 \end{array} .$$

At first glance, this is probably the weirdest of all the cases. Every column contains a pivot, so there are definitely no free variables. So what's this linear system trying to tell us? Well, the last equation, namely $0 = 1$, is *never* true, no matter *how* we pick x_1 and x_2. In other words, this linear system has no solutions! And again the positioning of the pivots was key in alerting us to the fact. If the reduced row echelon form of the augmented matrix of a linear system has a row in which the pivot occurs in the rightmost column, then the system has no solutions.

1.1.5 Generalization

Any system of r linear equations in c unknowns can be written as

$$\begin{array}{rl} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1c}x_c = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2c}x_c = & b_2 \\ \vdots & \vdots \\ a_{r1}x_1 + a_{r2}x_2 + \ldots + a_{rc}x_c = & b_r \end{array} \quad \leftrightarrow \quad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \ldots & a_{1c} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2c} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{r1} & a_{r2} & \ldots & a_{rc} & b_r \end{array}\right]$$

We've learned that computing the reduced row echelon form of the augmented matrix on the right side will tell us how many solutions this system of linear equations will have. If every column but the rightmost column in the output of `rref` has a pivot, then the system has a unique solution. If some column other than the rightmost column has no pivot, then the system has a free variable and has infinitely many solutions. If the rightmost column has a pivot, then the system has no solution.

1.2 Matrix arithmetic

Now that we have a good handle on the solutions to linear systems, let's take some time to develop a sense of how to manipulate linear systems. Let's start with the most fundamental concept: matrix-vector multiplication. In some sense, this idea is just a repackaging of a linear system into a more compact form. Rather than discuss all the details right away, let's first see an example.

$$\begin{array}{rl} 0.7x_1 + 0.23x_2 + 0.11x_3 &= 0.27 \\ 0.57x_1 + 0.77x_2 + 0.76x_3 &= 0.92 \\ 0.42x_1 + 0.03x_2 + 0.98x_3 &= 0.74 \end{array} \quad \leftrightarrow \quad \begin{bmatrix} 0.7 & 0.23 & 0.11 \\ 0.57 & 0.77 & 0.76 \\ 0.42 & 0.03 & 0.98 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.27 \\ 0.92 \\ 0.74 \end{bmatrix}.$$

So what happened in the conversion here? Well, the right side of the linear system got packaged up into a 3×1 matrix. We call a $r \times 1$ matrix a *column vector* (because it only has one column); similarly, a $1 \times c$ matrix is a *row vector*. We also grouped the variables x_1, x_2 and x_3 into a column vector. The coefficients of the linear system were placed into a matrix in much the same way we formed the augmented matrix, with the coefficient of the j^{th} unknown in the i^{th} equation being placed in the (i, j) position in the matrix.

Here's a completely natural series of questions: Why in the world would we want to do this? Aren't we making things more complicated? What benefit are we going to get out of all of this? To convince ourselves that this is a really, really useful abstraction to make, let's see the concept in action.

1.2.1 Three Industries

Let's try a slightly more interesting application.[1] Suppose that a model of an the manufacturing sector of an economy consists of just three industries: coal, electricity and steel production. Define p_c, p_e and p_s to be the total annual output of the coal, electricity and steel industries, respectively. In this model, we'll assume that all output is consumed. A natural question that should come up any time you're dealing with a real world application is "what are the units?" And it's a great question here, too. For our purposes, we'll think of our annual production in terms of their currency value; let's use dollars in this example. But as you might suspect, these industries don't operate in isolation. The coal industry buys steel to build new mines, the steel industry buys electricity to power its plants, and so on. We can capture these interrelated rates of consumption in Table 1.2.1. Here the intersection of row i and column j is the fraction of industry i's output that was sold to industry j. So, for instance, the electricity buys 72% of the coal industry's annual output (as measured in dollars).

One question we might ask about a system like this is how much each industry should produce so that every industry exactly breaks even. Remember that the break-even point is where revenue equals cost. Let's start with cost. How

[1] Adapted from Lay's *Linear Algebra and Its Applications*, third edition, section 1.6

	Coal	Electricity	Steel
Coal	0	0.56	0.43
Electricity	0.72	0.11	0.40
Steel	0.28	0.33	0.17

Table 1.1: Consumptions (in percent) in our manufacturing sector economic model. The intersection of row i and column j is the fraction of industry i's output that was sold to industry j. Notice that this implies that the column should each sum to 1.

much does the coal industry spend every every? Well, it busy 0% of its own output, 56% of the electricity industries output, and 43% of the steel industries output. Using symbols instead of words, this sentence reads

$$0p_c + 0.56p_e + 0.43p_s.$$

We could perform the same conversion for the other two industries. Let's think about forming a *cost vector* that we'll call $\mathbf{c}$. (A variable in bold font is a vector.) We typically think about vectors in terms of their entries, also called components, which we order from top to bottom so that the top entry is the first component and the bottom entry is the last component. Here, the components of our cost vector will be, in order, the cost incurred by the coal, electricity and steel industries, respectively.

$$\begin{aligned}
\mathbf{c} = \begin{bmatrix} \text{cost of coal industry} \\ \text{cost of electricity industry} \\ \text{cost of steel industry} \end{bmatrix} &= \begin{bmatrix} 0.00p_c + 0.56p_e + 0.43p_s \\ 0.72p_c + 0.11p_e + 0.40p_s \\ 0.28p_c + 0.33p_e + 0.17p_s \end{bmatrix} \\
&= \begin{bmatrix} 0.00 & 0.56 & 0.43 \\ 0.72 & 0.11 & 0.40 \\ 0.28 & 0.33 & 0.17 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} \\
&= C\mathbf{p},
\end{aligned}$$

where C is the matrix and $\mathbf{p}$ is the vector of annual outputs. To see why this matrix-vector product expression is powerful, we need to discuss the revenue vector. By the definition of the problem statement, the total annual revenue of the coal industry is just p_c, and similarly for the remaining two industries. Let's think about defining a revenue vector $\mathbf{r}$ in a similar way to the cost vector, so that the first component of $\mathbf{r}$ is the revenue of the coal industry, the second of the electricity industry, and the third of the steel industry. Replacing the words

with symbols,

$$\begin{aligned}\mathbf{r} = \begin{bmatrix} \text{revenue of coal industry} \\ \text{revenue of electricity industry} \\ \text{revenue of steel industry} \end{bmatrix} &= \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} \\ &= R\mathbf{p}.\end{aligned}$$

When all industries break-even simultaneously, we have

$$\mathbf{c} = \begin{bmatrix} \text{cost of coal industry} \\ \text{cost of electricity industry} \\ \text{cost of steel industry} \end{bmatrix} = \begin{bmatrix} \text{revenue of coal industry} \\ \text{revenue of electricity industry} \\ \text{revenue of steel industry} \end{bmatrix} = \mathbf{r}.$$

But how do we actually *solve* for this case? How can we determine what value(s) or p_c, p_e and p_s lead to all industries breaking even? Well, we have matrix-vector product expressions for both $\mathbf{r}$ and $\mathbf{c}$, both of which contain the variables p_c, p_e and p_s that we're after. Maybe substituting these matrix-vector products is a good place to start.

$$\begin{bmatrix} 0.00 & 0.56 & 0.43 \\ 0.72 & 0.11 & 0.40 \\ 0.28 & 0.33 & 0.17 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix}$$
$$C\mathbf{p} = R\mathbf{p}.$$

For this to be a useful, we need to note that matrix-vector multiplication *distributes*. You might be a little rusty on these old terms, and for good reason; we take these properties for granted all the time. Distributivity is the property that says, for instance, that $3x - 5x = (3 - 5)x$. We say that the multiplication by x is *distributed* across the terms. So what does that have to do with our situation here? Well, moving both matrix-vector products to the left side of the equation gives

$$\begin{bmatrix} 0.00 & 0.56 & 0.43 \\ 0.72 & 0.11 & 0.40 \\ 0.28 & 0.33 & 0.17 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$
$$C\mathbf{p} - R\mathbf{p} = \mathbf{0}$$

This is equivalent to saying that at the break even point, the cost minus the revenue is equal to zero for each industry independently. But, since matrix-

vector multiplication distributes, we can write

$$\begin{aligned} C\mathbf{p} - R\mathbf{p} &= (C - R)\mathbf{p} \\ &= \left(\begin{bmatrix} 0.00 & 0.56 & 0.43 \\ 0.72 & 0.11 & 0.40 \\ 0.28 & 0.33 & 0.17 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix}. \end{aligned}$$

But this brings to another new topic: how do we add and subtract matrices? Well, if you had to define how to add or subtract two matrices of the same size, how would you do it? Seriously, take a second and think about it. Got an idea? Good. If you thought, "I'd line up the matrices and add or subtract the equivalent positions in each matrix", you're on the right track. If you thought something else, let me know; I'd love to hear your idea.

So let's see how to do this matrix subtraction.

$$\begin{aligned} \left(\begin{bmatrix} 0.00 & 0.56 & 0.43 \\ 0.72 & 0.11 & 0.40 \\ 0.28 & 0.33 & 0.17 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} &= \begin{bmatrix} 0.00 - 1 & 0.56 & 0.43 \\ 0.72 & 0.11 - 1 & 0.40 \\ 0.28 & 0.33 & 0.17 - 1 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} \\ &= \begin{bmatrix} -1 & 0.56 & 0.43 \\ 0.72 & -0.89 & 0.40 \\ 0.28 & 0.33 & -0.83 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix}. \end{aligned}$$

But remember, the reason we started down this trail was to figure out the solutions to $C\mathbf{p} = R\mathbf{p}$. Now we know that these solutions are the same as the solutions to $(C - R)\mathbf{p} = \mathbf{0}$. We can easily do this computation in Matlab.

```
EDU>> C = [0.00 0.56 0.43; 0.72 0.11 0.40; 0.28 0.33 0.17]

C =

         0    0.5600    0.4300
    0.7200    0.1100    0.4000
    0.2800    0.3300    0.1700

EDU>> R = [1 0 0; 0 1 0; 0 0 1]

R =

     1     0     0
     0     1     0
     0     0     1
```

Now that we have the matrices C and R in Matlab, we can confirm that we didn't make any arithmetic mistakes earlier in our subtraction $C - R$.

```
EDU>> C-R

ans =

   -1.0000    0.5600    0.4300
    0.7200   -0.8900    0.4000
    0.2800    0.3300   -0.8300
```

But how do we make the augmented matrix of the linear system $(C - R)\mathbf{p} = \mathbf{0}$. We know from the previous section that we want the matrix $(C - R)$ the the column vector $\mathbf{0}$ append on the right side. On straightforward option is to type this all in manually.

```
EDU>> M = [-1.00 0.56 0.43 0.00; 0.72 -0.89 0.48 0.00; 0.28 0.33 -0.83 0.00]

M =

   -1.0000    0.5600    0.4300         0
    0.7200   -0.8900    0.4800         0
    0.2800    0.3300   -0.8300         0
```

But as you might imagine, this can be really cumbersome for large matrices. A much faster and more direct way is to make a new matrix with $C - R$ placed next to $\mathbf{0}$.

```
EDU>> M = [C-R [0;0;0]]

M =

   -1.0000    0.5600    0.4300         0
    0.7200   -0.8900    0.4000         0
    0.2800    0.3300   -0.8300         0
```

Regardless of how we computed the matrix M, we know what we have to do to find the solutions to the linear system: `rref` it!

```
EDU>> rref(M)

ans =

    1.0000         0   -1.2463         0
         0    1.0000   -1.4577         0
         0         0         0         0
```

So what does this mean? Well, the third column does not have a pivot, so the linear system has a free variable, and so there are infinitely many solutions to this system of equations. But let's try to be more specific. We can convert the

reduced row echelon form of the augmented matrix back into a system of linear equations.

$$\left[\begin{array}{ccc|c} 1 & 0 & -1.2463 & 0 \\ 0 & 1 & -1.4577 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \leftrightarrow \begin{bmatrix} 1 & 0 & -1.2463 \\ 0 & 1 & -1.4577 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_c \\ p_e \\ p_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} p_c - 1.2463p_s &= 0 \\ \leftrightarrow \quad p_e - 1.4577p_s &= 0 \\ 0 &= 0 \end{aligned}$$

The value of p_s is not constrained by any of the equations from the reduced row echelon form, and so it is a free variable. Only once we have chosen a particular value for p_s can the values of p_c and p_e be computed. In general, though, `rref(M)` tells us some interesting things about the nature of the solutions to this system. For instance, regardless of the annual output from the steel industry, for all industries to break even, the coal industry's output must be 124.63% that of the steel industry. Similarly, the electricity industry's annual output must be 145.77% of the steel industries output. Even if there is not a unique solution to a system of linear equations, we can often still tell quite a bit about what the solution space looks like.

The key idea here is that knowing the rules of matrix arithmetic allowed us to solve a problem that at first seemed intractable. We had to leverage several different properties, including matrix addition and distributivity, in order to arrive at system which is both equivalent to the original and solvable. In short: while pushing symbols around can seem boring at best and a huge pain at worst, it is sometimes a very useful technique for making headway on a problem. And in order to know that what we're during is legal in the linear algebraic world, we have to know the rules.

1.2.2 Matrix arithmetic rules

We encountered some matrix arithmetic rules over the course of the preceding example. Let's go through and more systematically deal with the topic. I know it may seem silly to be pointing out simple things like this, but before too long we'll see examples where properties that we normally assume to be true do not in fact hold.

Throughout this section, we'll use the matrices M and N defined below for our calculations. This choice is arbitrary, as seen in their random constructions; any matrices would do.

```
EDU>> M = round(5*rand(3,3))

M =

     2     4     1
```

```
     5     3     2
     1     2     2

EDU>> N = round(5*rand(3,3))

N =

     2     4     1
     1     2     1
     3     0     1
```

A useful Matlab command will appear throughout. We can grab matrix entries using the row and column notation we've been developing. For instance `M(2,3)` returns the entry in row 2 and column 3.

```
EDU>> M(2,3)

ans =

     2
```

We can also grab an entire row or column at once by using the `:` symbol which you can read as "all." For instance, the command `M(:,1)` will return all row entries that are in column 1.

```
EDU>> M(:,1)

ans =

     2
     5
     1
```

Similarly, the command `M(2,:)` will return all column entries that are in row 2.

```
EDU>> M(2,:)

ans =

     5     3     2
```

Notice that this command returns a *row* vector.

Scalar multiplication

In the context of linear algebra, a scalar is just a "normal" number, like 2, -7, π, *etc.* If k is any scalar and M is any matrix (including vectors), then scalar-matrix multiplication is done component-wise so that if m_{ij} is the entry in row i and column j of M, then the entry in row i and column j of kM is just km_{ij}. Notice this means that $kM = Mk$.

```
EDU>> -2*M

ans =

    -4    -8    -2
   -10    -6    -4
    -2    -4    -4

EDU>> M*-2

ans =

    -4    -8    -2
   -10    -6    -4
    -2    -4    -4
```

Matrix addition

As we discussed in the example, matrix-matrix addition is component-wise, so that if a_{ij} and b_{ij} are the entries in row i and column j of matrices A and B respectively, then the entry in row i and column j in their sum $A + B$ is just $a_{ij} + b_{ij}$.

```
EDU>> M + N

ans =

     4     8     2
     6     5     3
     4     2     3
```

Matrix-vector multiplication

Suppose we have a $r \times c$ matrix M with rows $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_r$ and columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_c$. If $\mathbf{x}$ is a $c \times 1$ column vector, the matrix-vector multiplication $M\mathbf{x}$ can be thought of in two ways. In the first, we think of $M\mathbf{x}$ as a linear combination of the columns of M with the coefficients being the corresponding entries of $\mathbf{x}$.

$$M\mathbf{x} = M \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_c \end{bmatrix} = x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + \ldots + x_c\mathbf{c}_c.$$

```
EDU>> M*[1;2;3]

ans =
```

```
    13
    17
    11

EDU>> 1*M(:,1) + 2*M(:,2) + 3*M(:,3)

ans =

    13
    17
    11
```

In the second, we think of $M\mathbf{x}$ as a vector whose components are the dot product of $\mathbf{x}$ with the corresponding row of M.

$$M\mathbf{x} = \begin{bmatrix} \mathbf{x} \circ \mathbf{r}_1 \\ \mathbf{x} \circ \mathbf{r}_2 \\ \vdots \\ \mathbf{x} \circ \mathbf{r}_r \end{bmatrix}.$$

For a refresher, the dot product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is just the sum of the products of the components $\sum_i x_i y_i$. You might have seen a similar concept in the formula for correlation in an introductory statistics course. In Matlab, we accomplish this using the `dot` command. To learn about this function, enter `doc dot` or `help dot` in the command line.

```
EDU>> [dot([1;2;3],M(1,:)); dot([1;2;3],M(2,:)); dot([1;2;3],M(3,:))]

ans =

    13
    17
    11
```

Notice that both definitions imply that if M is $r \times c$, then $\mathbf{x}$ must be $c \times 1$ for the computation $M\mathbf{x}$ to make sense. Matlab will always let you know if you make a dimension mistake. It's nothing to worry about, and here's what the error will look like.

```
EDU>> rand(4,3) * rand(2,1)
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

We'll see over and over again that being able to think in terms of both of these definitions will make problems much more approachable. Some problems are easier to think about in terms of a linear combination of the columns of a matrix, and others are easier to think about in terms of dot products of the rows.

Matrix-matrix multiplication

Multiplying two matrices is a lot like matrix-vector multiplication. Roughly speaking, "stack up" a bunch of matrix-vector multiplications in order to make the product matrix. More formally, let A be a $r \times c$ matrix and let B be a $c \times r$ matrix with columns $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r$. Then the product AB can be defined in terms of r distinct matrix-vector products between A and the columns of B.

$$AB = \begin{bmatrix} | & | & & | \\ A\mathbf{b}_1 & A\mathbf{b}_2 & \cdots & A\mathbf{b}_r \\ | & | & & | \end{bmatrix}.$$

In our running example, we have

```
EDU>> M*N

ans =

    11    16     7
    19    26    10
    10     8     5

EDU>> [M*N(:,1) M*N(:,2) M*N(:,3)]

ans =

    11    16     7
    19    26    10
    10     8     5
```

There are several really important things to note here. First, if A is $r \times c$ then for AB to make sense, B *must* be $c \times r$. (A quick way to remember this is to think of the product as $(r \times c) \times (c \times r)$ and make sure that the "inner" and "outer" dimensions match.) Second, even if $r = c$ so that both AB and BA make sense, it is *almost never* the case that $AB = BA$. For instance,

```
EDU>> M*N

ans =

    11    16     7
    19    26    10
    10     8     5

EDU>> N*M

ans =
```

```
    25     22     12
    13     12      7
     7     14      5
```

This is probably completely outside of your experience in arithmetic. After all, for real or complex number a and b, we always have $ab = ba$. But this is not the case in the linear algebraic universe. We'll see later that both AB and BA have interesting conceptual interpretations, even if they aren't the same.

The identity matrix

In the real numbers, there is a very special number x such that any other number y multiplied with x returns $xy = yx = y$. This special number, called the *multiplicative identity*, is $x = 1$, and the preceding sentence is just a precise way to say "any number multiplied by 1 is itself." There is a similar concept in linear algebra. Here we have the *identity matrix*, a square matrix with 1 along the main diagonal (*i.e.*, entries at row i and column i) and zero everywhere else. When used as an identity, we will always call this matrix I.

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}.$$

If an identity matrix has n rows (and so also n columns), we call it *the identity matrix of order n*. To make an identity matrix of order n in Matlab, we use the command `eye(n)`.

```
EDU>> M*eye(3)

ans =

     2     4     1
     5     3     2
     1     2     2

EDU>> eye(3)*M

ans =

     2     4     1
     5     3     2
     1     2     2
```

Powers of matrices

We can take powers of a square matrix A, and they're defined in much the same way that powers of a real number are defined. For instance $A^2 = AA$, $A^3 = A^2(A) = AAA$. In general, we can define powers recursively using $A^n = A^{n-1}A$. Remember, it only makes to talk about powers of a matrix if the matrix is square.

All the rest

All the rest of the properties that we've used for years in the real number context also hold in the linear algebraic world. These include distributivity of matrix-vector multiplication, so that $(A + B)\mathbf{x} = A\mathbf{x} + B\mathbf{x}$ and $A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y}$. By the relationship between matrix-vector multiplication and matrix-matrix multiplication, this also means that if A,B, and C are matrices, then $(A+B)C = AC + BC$. Addition of matrices commutes, so that $A + B = B + A$ and both addition and multiplication are associative so that $(A + B) + C = A + (B + C)$ and $(AB)C = A(BC)$. These are properties we've taken for granted for a long time. But the most important thing to remember is that matrix multiplication does not commute; generally speaking $AB \neq BA$.

Chapter 2

The matrix inverse

2.1 Leontief Input-Output Model

We've already seen examples of supply networks in which product A is both used to make product B and is sold itself directly to the consumer. We can formalize these notions by calling the former *intermediate demand* and the latter *final demand*. Let's investigate systems featuring intermediate and final demands in a slightly different context than supply chain networks.

Imagine we have two industries, manufacturing and services, and suppose that in order to make 1 unit of output, the manufacturing sector must consume 0.4 units of its own output, and 0.2 units of service industry output. (Here "units of output" could be measured in whatever way we want so long as the measurement method is consistent across the industries.) Similarly, suppose that in order to make 1 unit of output, the services industry must consume 0.7 units of the manufacturing industries output, and 0.1 units of its own output. Let x_1, x_2 be the number of units produced by the manufacturing and services industries, respectively. Then the intermediate demand necessary to create these units out production is

$$\begin{bmatrix} 0.4 & 0.7 \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Suppose now that we introduce a final demand vector $\mathbf{d}$ representing the number of units of production of both the manufacturing and services industry that are demanded not by other industries, but by the public at large. Ideally, the total number of units produced by both industries must be equal to the sum of the intermediate and final demands; this is just the familiar idea of supply equaling demand in equilibrium. Mathematically, we can express this idea as follows

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0.7 \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$\mathbf{x} = C\mathbf{x} + \mathbf{d}$$

Suppose that we have somehow measured of estimated the demands contained in $\mathbf{d}$, and we're trying to determine our ideal production levels $\mathbf{x}$. Rearranging the equation a little bit gives

$$\begin{aligned} \mathbf{x} &= C\mathbf{x} + \mathbf{d} \\ (I - C)\mathbf{x} &= \mathbf{d}. \end{aligned}$$

This type of linear system is called a *Leontief input-output model*.

In the past, we've solved systems of linear equations like this input-output model using a `rref` in Matlab. We know how to interpret the results of this computation in terms of unique solutions, free variables and all the rest. But `rref` isn't an ideal solution generally for a couple of reasons. First, an individual `rref` computation doesn't help us do another `rref` computation more efficiently;

for a different demand vector $\mathbf{d}'$, we would simply repeat the entire process. Second, `rref` is almost useless in a theoretic context; sometimes, like it or not, pushing symbols around leads to serious discoveries about fundamental properties of a given system. Here's my claim: it would be great if we could find a matrix A such that

$$\begin{aligned} A(I-C)\mathbf{x} &= A\mathbf{d} \\ I\mathbf{x} = \mathbf{x} &= A\mathbf{d}. \end{aligned}$$

If you think about it, this is exactly what's happening every time you solve for x in an equation involving only real numbers. For instance when solving the equation $7x = 14$, you find a number a such that $a \cdot 7x = 1x = x$, namely $a = 1/7$. While it's true that such an inverse element a will exist over the real numbers, it's not the case that an inverse element will always exist when we're dealing with matrices. If the inverse of a matrix C exists, we denote it C^{-1}. Just as if real numbers, the matrix C^{-1} satisfies $C^{-1}C = I = CC^{-1}$.

Example 1: Consider the matrices

$$C = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$$
$$D = \begin{bmatrix} 2 & -3 \\ -3 & 5 \end{bmatrix}$$

Then $D = C^{-1}$.

Reading question 2: Verify the previous claim.

2.1.1 The matrix inverse

Before we dive into computing the inverse of a matrix, let's gather up some preliminary results. For now, let's consider an arbitrary $r \times c$ matrix C. Generally speaking, the matrix C takes a vector of length c and produces a vector of length r. Another way of saying this is that the domain of C is $\mathbb{R}^c$ and the range is $\mathbb{R}^r$. Remember, for an inverse of C to exist, we need a one-to-one and onto correspondence between the elements of the domain and the elements of the range. If $r > c$, then there are more elements in the range than in the domain. But then it's impossible that every element in the range is of the form $C\mathbf{x}$ for some $\mathbf{x}$, exactly because there are more elements in the range than there are $\mathbf{x}$'s in the domain. Another way of saying this is that C is not an onto mapping in this case. On the other hand, if $c > r$, then there are more elements in the domain than there are elements in the range, and so some element $\mathbf{y}$ in the range such that $C\mathbf{x}_1 = \mathbf{y}$ and $C\mathbf{x}_2 = \mathbf{y}$. Another way of saying this it that

C is not a one-to-one mapping in this case. Since neither $r > c$ nor $c > r$ are suitable situations, for C to have any chance of having an inverse, it must be the case that $r = c$ so that C is a square matrix.

But even if C is square, we can still run into problems. For a ridiculous case, imagine that C is a square matrix containing all zeros. Then C maps every input vector $\mathbf{x}$ to the output vector $\mathbf{y} = \mathbf{0}$. So clearly C is not a one-to-one mapping in this case. We've seen that in general if the columns of C are not linearly independent, then `rref(C)` contains a free variable, and so there are some $\mathbf{y}$ in the range of C which have infinitely many solutions $\mathbf{x}$ satisfying $C\mathbf{x} = \mathbf{y}$. Hence, if the columns of C are linearly dependent, then C is not an one-to-one mapping, and hence there can be no inverse for C.

So suppose that the columns of C are linearly independent. In this case, `rref(C)` can have no free variables, and since C is square, this implies that `rref(C)` is an identity matrix. Therefore, for every $\mathbf{y}$ in the range of C, there is a unique solution $\mathbf{x}$ such that $C\mathbf{x} = \mathbf{y}$. This is exactly what we want in order for the inverse of C to exist!

Reading question 3: Prove that if C is square and `rref(C)` has no free variable, then `rref(C)` is an identity matrix.

Reading question 4: Prove that if `rref(C)` is the identity matrix, then the equation $C\mathbf{x} = \mathbf{y}$ has a unique solution for every $\mathbf{y}$ in the range of C.

So we can add the existence of a matrix inverse to our list of equivalent conditions concerning linear independence of the columns of a matrix. (Here we assume that A is square.)

- The equation $A\mathbf{x} = \mathbf{y}$ has a unique solution $\mathbf{x}$ for every $\mathbf{y}$.
- The equation $A\mathbf{x} = \mathbf{0}$ has only the trivial solution $\mathbf{x} = \mathbf{0}$.
- The columns of a A are linearly independent.
- `rref(A)` contains no free variables.
- `rank(A)` $= c$, the number of columns of A.
- The inverse A^{-1} exists.

2.1.2 Building intuition

Before we get into the nitty gritty, let's try to form a hypothesis about what a matrix inverse of a 2×2 matrix A will look like. In Matlab, try

```
EDU>> A = [5 3; 3 2]

A =
```

```
     5     3
     3     2

EDU>> A^(-1)
```

What do you see? How are the entries of A^{-1} related to the entries of A?

Next try

```
EDU>> A = [3 1; 5 2]

A =

     3     1
     5     2

EDU>> A^(-1)
```

Does your hypothesis from the first example fit the second example?

One last example:

```
EDU>> A = [4 7; 2 4]

A =

     4     7
     2     4

EDU>> A^(-1)
```

How is your hypothesis holding up now? If it failed, can you revise it to make sense of all three examples?

2.1.3 A first matrix inverse

So far we know that a matrix inverse of A exists if and only if A is square and has linearly independent columns. But this doesn't help much with the actual computation. Consider the general 2×2 matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Let's make two observations:

$$A \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}$$
$$A \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix}$$

We know A^{-1} must exist, and so applying A^{-1} to both sides of both equations is totally legal.

$$A^{-1}A\begin{bmatrix}1\\0\end{bmatrix} == A^{-1}\begin{bmatrix}a\\c\end{bmatrix}$$
$$A^{-1}A\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} = A^{-1}\begin{bmatrix}b\\d\end{bmatrix}$$

To find out what A^{-1} actually is, let's first define its components and then solve for them using the relations we just found. Let

$$A^{-1} = \begin{bmatrix}e & f\\g & h\end{bmatrix}.$$

Remember, we're given A, so we know exactly the values of a, b, c and d. We don't know e, f, g or h; these are variables. Our relations from above become

$$A^{-1}\begin{bmatrix}a\\c\end{bmatrix} = \begin{bmatrix}e & f\\g & h\end{bmatrix}\begin{bmatrix}a\\c\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}$$
$$\begin{bmatrix}ae+cf\\ag+ch\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}$$

and

$$A^{-1}\begin{bmatrix}a\\c\end{bmatrix} = \begin{bmatrix}e & f\\g & h\end{bmatrix}\begin{bmatrix}b\\d\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$
$$\begin{bmatrix}be+df\\bg+dh\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$

We have 4 equations in the variables e, f, g and h. We can gather these up in to a single system of linear equations.

$$\begin{bmatrix}ae+cf\\ag+ch\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}$$
$$\begin{bmatrix}be+df\\bg+dh\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$
$$\Rightarrow \begin{bmatrix}a & c & 0 & 0\\0 & 0 & a & c\\b & d & 0 & 0\\0 & 0 & b & d\end{bmatrix}\begin{bmatrix}e\\f\\g\\h\end{bmatrix} = \begin{bmatrix}1\\0\\0\\1\end{bmatrix}.$$

We can use a tool like WolframAlpha to solve this system. We find

$$\begin{aligned} e &= \frac{d}{ad-bc} \\ f &= \frac{-b}{ad-bc} \\ g &= \frac{-c}{ad-bc} \\ h &= \frac{a}{ad-bc}. \end{aligned}$$

But remember, the whole point of this exercise was to find the components of the inverse matrix A^{-1}. We can now just read them off.

$$\begin{aligned} A^{-1} &= \begin{bmatrix} e & f \\ g & h \end{bmatrix} \\ &= \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \end{aligned}$$

Reading question 5: Show that we've found is accurate by computing AA^{-1} and $A^{-1}A$.

The scalar outside the matrix portion of A^{-1} is really interesting. We say something similar when we considered eigenvalues and eigenvectors. In particular, we showed (in our current notation) that if $ad - bc = 0$, then the columns of A are scalar multiples of each other, and hence linearly dependent.

Reading question 6: Remind yourself why the preceding statements are true.

Notice that the same type of reasoning used in computing the inverse of a given 2×2 matrix could be used to explicitly construct the inverse of any $n \times n$ matrix!

Reading question 7: Find the inverse of the arbitrary 3×3 matrix

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

2.1.4 Computing production

Now that we have the inverse of an arbitrary 2×2 matrix, computing the total production necessary to satisfy the demand in a Leontief input-output model should be a piece of cake. For instance, imagine that the total demand is 200

units from the manufacturing industry and 100 units from the services industry. Then our matrix equation looks like

$$\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}0.4 & 0.7\\ 0.2 & 0.1\end{bmatrix}\right)\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}200\\ 100\end{bmatrix}$$
$$A\mathbf{x} = \mathbf{d}$$

So we first need to compute A^{-1} and then multiply by $\mathbf{d}$. Since we've already done the tough stuff, we can literally just substitute values in order for find the inverse.

$$\begin{aligned} A^{-1} &= \frac{1}{(0.6)(0.9)-(0.7)(0.2)}\begin{bmatrix}0.9 & -0.7\\ -0.2 & 0.6\end{bmatrix}\\ &= 2.5\begin{bmatrix}0.9 & -0.7\\ -0.2 & 0.6\end{bmatrix}\\ &= \begin{bmatrix}2.25 & 1.75\\ 0.5 & 1.5\end{bmatrix}\end{aligned}$$

Not so bad, right? But a lot of times, things don't work out so neatly. In those cases, it's usually better just to use Matlab. We can compute the inverse of A using Matlab in a couple different ways.

```
EDU>> A = eye(2) - [0.4 0.7; 0.2 0.1]

A =

    0.6000   -0.7000
   -0.2000    0.9000

EDU>> inv(A)

ans =

    2.2500    1.7500
    0.5000    1.5000

EDU>> A^(-1)

ans =

    2.2500    1.7500
    0.5000    1.5000
```

All three methods agree, so we know we're doing everything correctly. So to compute the number of units needed to satisfy demand $[200, 100]^T$, we just need to multiply.

```
EDU>> d = [200;100]

d =

   200
   100

EDU>> inv(A) * d

ans =

  625.0000
  250.0000
```

So the manufacturing industry and service industry must produce 625 units and 100 units, respectively, in order to satisfy both intermediate and final demand exactly. This type of operation comes up so frequently that Matlab has provided an even easier way to compute $\mathbf{x} = A^{-1}\mathbf{d}$.

```
EDU>> A \ d

ans =

   625
   250
```

Notice that this is a backslash, not forward slash as we typically use in division!

2.2 Parameterized Leontief Input-Output

The Leontief input-output model that we considered previously took the form

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0.7 \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$\mathbf{x} = C\mathbf{x} + \mathbf{d},$$

where $\mathbf{x}$ represented the number of units produced of each good, and $\mathbf{d}$ represented the final demand. Just so we have some terminology to throw around, remember that we called $C\mathbf{x}$ the *intermediate demand*.

We reframed the idea of finding a specific production level $\mathbf{x}$ that would perfectly satisfy demand as a matrix inverse problem, in particular the problem of finding the inverse of the matrix $I - C$.

Reading question 8: Remind yourself why we care about the inverse of the matrix $I - C$.

Just as in our investigations of eigenvalues and eigenvectors, it is often a productive exercise to think of one of the entries of C as a tunable parameter. This serves two functions: first, it let's us represent uncertainty as to the exact values of the matrix C. Remember, the (i, j) entry of C represents the number of units of product i that are necessary to produce 1 unit of product j.

Reading question 9: Remind yourself why the preceding statement is true.

We could easily imagine that this value fluctuates or is not known definitively, and so having some control over the value of the (i, j) entry of C could give us some information about how the system behaves as various intermediates demands change. Another reason to introduce tunable parameters is more mathematical: generalizing the matrix C will help us build intuition about when and how inverses exist or fail to exist.

Let's consider an input-output model where the number of units of product 1 necessary to make 1 unit of product 2 is represented by the variable k. Since an industry should be adding value by creating a new product, it's safe to assume that $k < 1$, and since it doesn't make sense to have negative values of production, we can also feel good about bounding $k \geq 0$. Our model now takes the form

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.4 & k \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$\mathbf{x} = C_k\mathbf{x} + \mathbf{d},$$

Here's a natural question: for what values of k does there exist a unique production level that satisfies final demand exactly? Mathematically, we're asking ourselves the following question: for what values of k does an inverse of $I - C_k$ exist.

We have many different and equivalent ways that characterize whether the inverse of a particular matrix exists. In some ways, the only real mathematics involved in the process of determining whether an inverse exists or not is having some insight into which formulation will give use the easiest route to determining the existence of the inverse. For instance, we could try to prove that the determinant of the matrix is nonzero. Or, alternatively and equivalently, we could try to prove that the columns of the matrix are linearly independent. Or we could try to prove that the kernel of the matrix contains only the zero vector. All of these (and more) and perfectly valid, but one or more will usually prove to be easier than the others.

Since computing the determinant of a 2×2 matrix is easy, let's start there. Remember, we're trying to investigate the matrix $I - C_k$, not just C_k itself. So the question becomes this: for what values of k does the following statement hold:

$$\det\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}0.4 & k\\ 0.2 & 0.1\end{bmatrix}\right) \neq 0$$

Simplifying the matrices first and then taking the determinant gives

$$\det(I - C_k) = (0.6)(0.9) - (0.2)k.$$

Reading question 10: Verify the previous claim.

Notice that the determinant is linear in k, and hence there will be one and only one value of k for which $\det(I - C_k)$ equals any given value, and in particular only one value of k for which the determinant is zero. A little bit of mathematical elbow grease will give us our answer.

$$\begin{aligned} 0 &= .54 - .2k \\ k &= 2.7. \end{aligned}$$

Bringing this result back to the context of the problem, there will exist a unique production level which uniquely satisfies *any* final demand vector $\mathbf{d}$ so long as $k \neq 2.7$. This is a really good thing, because we've already decided that for the model to make any kind of physical sense, it must be the case that $k \in [0, 1)$.

Reading question 11: Repeat the following procedure assuming that the (2,1) entry of C is unknown.

It's not much of a stretch to imagine that a different entry in the matrix C is unknown. For instance, let's imagine that entry (2,2) of C is unknown. In terms of our model, this means that we're unsure how many units of product 2 will be recycled to make a single unit of product 2. Our model now takes the form

$$\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}0.4 & 0.7\\ 0.2 & k\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}d_1\\ d_2\end{bmatrix}$$
$$\mathbf{x} = B_k\mathbf{x} + \mathbf{d},$$

(Note: I'm using B_k here just to clearly denote that the matrix we're considering here is not exactly the same as the C_k considered before.) Again, let's determine when the inverse of $I - B_k$ by using determinants. Here the determinant condition for invertability takes the form

$$\det\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}0.4 & 0.7\\ 0.2 & k\end{bmatrix}\right) \neq 0$$

Simplifying and performing the determinant computation gives

$$\det(I - B_k) = (0.6)(1 - k) - (0.7)(0.2)$$

Again, the determinant is linear in k, and so there is one and only one value of k for which $I - B_k$ is not invertible:

$$\begin{aligned} 0 &= (0.6)(1 - k) - (0.7)(0.2)\\ k &= 1 - \frac{0.14}{0.6}\\ &\approx 0.77 \end{aligned}$$

Reading question 12: Verify the previous computation.

Here, the computed value of k certainly does lie within the bounds we set up earlier. In other words, this value of k could conceivably come up in the real world. But what would it mean if it would? Remember, systems of linear equations either have a unique solution, no solution or infinitely many solutions. We've just proved that if k in this situation has a particular value, then it is definitely the case that the a unique solution to the Leontief input-output model does not exist. Therefore, we can conclude that there are either infinitely many production levels that perfectly satisfy a given final demand, or no production level that perfectly satisfies a given final demand. Which situation actually occurs completely depends on the given final demand.

Now, for the grain of salt. We've shown that a matrix inverse of $I - B_k$ fails to exist if and only if k is one particular value. In the real world, such a narrow window of badness is very rarely realized. That being said, sometimes very weird things can happen if a matrix is "close" to be non-invertible. In other words, we could see strange behavior of our production levels if k is very near the point at which a matrix inverse fails to exist. The point here is that we as mathematically oriented business people need to be aware of the failures of our models and do our best to mitigate those failures.

2.2.1 A larger example

How about a more difficult example? Consider a 3 industry Leontief input-output model of the form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.2 & 0.4 \\ 0.3 & 0.5 & 0.4 \\ 0.3 & 0.2 & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$
$$\mathbf{x} = C_k\mathbf{x} + \mathbf{d}.$$

As with the previous example, there are a lot of different ways that we could determine when $I - C_k$. It's tough to say which is going to be the easiest, but let's try to start with analyzing the determinant condition. But we've never talked about how to take the determinant of a 3×3 matrix by hand! Have no fear! WolframAlpha eats problems like this for breakfast. Let's try entering the following statement into WolframAlpha:

```
det IdentityMatirx[3]-{{0.3, 0.2, 0.4},{0.3, 0.5, 0.4},{0.3, 0.2, k}}
```

Now, we could've used the statement

```
det {{1-0.3, -0.2, -0.4},{-0.3, 1-0.5,-0.4},{-0.3, -0.2, 1-k}}
```

You might think one or the other is a little better. Regardless of which one we end up choosing, the result is

$$\det(I - C_k) = 0.126 - 0.29k.$$

Then the determinant condition of invertibility tells us that $I - C_k$ is non-invertible if and only if

$$0 = 0.126 - 0.29k$$
$$k \approx 0.434.$$

2.3 Leontief Price Equation

The Leontief input-output model we've been investigating also has another interesting application. To give this application justice, we need a simple but powerful operation: the *matrix transpose*. Given a matrix A, the matrix transpose A^T is formed by taking the first column of A as the first row of A^T, the second column of A as the second row of A^T and so on. Here're some examples:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}, \quad I^T = I$$

The transpose has all sorts of interesting properties. Let's start with some simpler ones. For instance, $(A^T)^T = A$. If A and B are matrices (or vectors) of the same size, then $(A+B)^T = A^T + B^T$. Scalar multiplication also works well with the transpose, in the sense that $(cA)^T = cA^T$ for any scalar c.

Reading question 13: Verify that the preceding properties of the transpose hold.

There are some more complicated properties that are also immensely useful. For instance, $\det(A^T) = \det(A)$.

Reading question 14: Verify the preceding statement for the arbitrary 2×2 matrix.

But why should we care about a property like this? Well, we know that an inverse of A exists if and only if $\det(A) \neq 0$. If we believe the property above, then the matrix A has an inverse if and only if the matrix A^T has an inverse. Another interesting property is that for suitably sized matrices A and B, we have $(AB)^T = B^T A^T$.

Reading question 15: Verify that the preceding statement is true for an arbitrary 2×2 matrix A and an arbitrary 2×3 matrix B.

Using the previous two properties, we can actually determine the exact form of the inverse of A^T provided that it exists. To see how this works, let's first assume that a matrix A has an inverse A^{-1}. From the determinant property of the matrix transpose, we can claim there exists a matrix B such that $BA^T = A^TB = I$, namely that B is in the inverse of A^T. Then

$$\begin{aligned} BA^T &= I \\ (BA^T) &= I^T \\ AB^T &= I \\ B^T &= A^{-1}I \\ B &= (A^{-1})^T \end{aligned}$$

In other words, the inverse of the transpose is the transpose of the inverse of A. In symbols, $(A^T)^{-1} = (A^{-1})^T$. At this point, I have a feeling that you're wondering why this is important. So let's get on with the application.

Let's imagine that industry i charges p_i dollars (or other unit of currency) for each unit of its output. Considering a Leontief system with 2 different industries, we can bundle these prices into a *price vector* $\mathbf{p}$, where the i^{th} component of p is p_i, the price charged by industry i for 1 unit of its output. For a concrete example, let's go back to the manufacturing and services example that we investigated previously. Remember, this model took the form

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0.4 & 0.7 \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \\ \mathbf{x} &= C\mathbf{x} + \mathbf{d} \end{aligned}$$

Imagine now that industry 1 (manufacturing) charges \$200 per unit of its output, and industry 2 (services) charges \$100 per unit of its output, so that our price vector has the form $\mathbf{p} = [200, 100]^T$. Since industry 1 requires 0.4 units of output from industry 1 and 0.2 units of output from industry 2, industry 1 incurs a total cost of $(0.4)(200) + (0.2)(100)$ in order to make one unit of output. We can write this in symbols, too.

$$\begin{aligned} (0.4)(200) + (0.2)(100) &= \begin{bmatrix} 0.4 & 0.2 \end{bmatrix} \begin{bmatrix} 200 \\ 100 \end{bmatrix} \\ &= \mathbf{c}_1^T \mathbf{p}, \end{aligned}$$

where here $\mathbf{c}_1$ is the first column of the consumption matrix C. Similarly, we could write the cost incurred to make a single unit of output from industry 2 as

$$\begin{aligned} (0.7)(200) + (0.1)(100) &= \begin{bmatrix} 0.7 & 0.1 \end{bmatrix} \begin{bmatrix} 200 \\ 100 \end{bmatrix} \\ &= \mathbf{c}_2^T \mathbf{p} \end{aligned}$$

In fact, we can bundle up these costs into a single matrix-vector quantity using the matrix-vector quantity $C^T\mathbf{p}$.

Reading question 16: Run through the matrix arithmetic to prove that the preceding statement is true.

Now, cost is only part of any economic equation. What we're really interested in is what price each industry needs to charge in order to cover depreciation, the wages of its employees, *etc.* and also make a fixed amount of profit. We'll be measuring all of these quantities per unit of output. (This is important, because it allows us to make fair comparisons.) All together, the wages, depreciation, profit, *etc.* can be summed into a single number: the *added value* v_i of industry i. Bundling up the added values of every industry, we can form an added value vector $\mathbf{v}$. Then given a consumption matrix C and a value added vector $\mathbf{v}$, we would like to find at price vector $\mathbf{p}$ such that

$$\mathbf{p} = C^T\mathbf{p} + \mathbf{v}.$$

How does this equation mean? Well, here the prices of 1 unit of output from each industry have been set so that every industry exactly covers both its costs incurred from buying other industries' goods, represented by $C^T\mathbf{p}$, and its added value. But how can we find such a special price vector? We can follow a similar path as we did in the original Leontief model.

$$(I - C^T)\mathbf{p} = \mathbf{v}.$$

If the matrix $(I - C^T)$ is invertible, then

$$\mathbf{p} = (I - C^T)^{-1}\mathbf{v}.$$

Let's revisit our two industry example, but this time, rather than using a fixed price vector, let's imagine having a fixed value added vector $\mathbf{v} = [50, 25]^T$.

Reading question 17: Describe in words what $\mathbf{v}$ means in terms of quantities from our two industry model.

Then our price equation takes the form

$$\begin{aligned}
\mathbf{p} &= C^T\mathbf{p} + \mathbf{v} \\
\mathbf{p} &= \begin{bmatrix} 0.4 & 0.2 \\ 0.7 & 0.1 \end{bmatrix}\mathbf{p} + \begin{bmatrix} 50 \\ 25 \end{bmatrix} \\
\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.4 & 0.2 \\ 0.7 & 0.1 \end{bmatrix}\right)\mathbf{p} &= \begin{bmatrix} 50 \\ 25 \end{bmatrix}
\end{aligned}$$

Just as a refresher, let's remind ourselves how to solve this system of linear equations the old way, that is, the way we solved systems of linear equations before we knew about the concept of inversion. Short story: `rref` the augmented matrix.

```
EDU>> rref([.6 -.2 50; -.7 .9 25])

ans =

     1     0   125
     0     1   125
```

But if we had a bigger matrix, this might be a little bit of a pain. If we're sure that $I - C^T$ has an inverse, we can compute the solution price vector $\mathbf{p}$ using matrix inversion. But before we do, we need to learn how to take the transpose of a matrix in Matlab. Fortunately for everyone involved, this is pretty easy

```
EDU>> C = [0.4 0.7; 0.2 0.1]

C =

    0.4000    0.7000
    0.2000    0.1000

EDU>> C'

ans =

    0.4000    0.2000
    0.7000    0.1000
```

The symbol to do the transpose, in case it's unclear, is the single quote. You can also type `help transpose` in Matlab to learn more about the syntax.

Now we're ready to actually do our matrix inversion. Remember that there are two ways to do this in Matlab. The first way looks a lot like what we would write analytically:

```
EDU>> inv(eye(2) - C') * [50;25]

ans =

   125
   125
```

The second looks a little bit different, and more like division notation than inverse notation.

```
EDU>> (eye(2) - C') \ [50;25]

ans =

  125.0000
  125.0000
```

Regardless of how you get to this point, we've concluded that for the manufacturing and service industries to have added values of \$50 and \$25 per unit of output, respectively, then both should charge \$125 dollars per unit of output.

2.4 Sensitivity in Leontief Models

We've seen that the entries of the consumption matrix C of a Leontief input-output have an useful physical interpretation: the (i, j) entry of C represents the number of units of output from industry i necessary for industry j to create 1 unit of output. Let's try to develop similar intuition about what the entries of the matrix $(I - C^T)^{-1}$ and $(I - C)^{-1}$.

The method through which we'll try to gain this intuition is the same one we've used previously: in order to understand the conceptual meaning of the entries of a matrix A, pick a test vector $\mathbf{x}$ which has some known meaning, and interpret the results of $A\mathbf{x}$ in terms of the model under consideration. Probably the best way to get a handle on this strategy is to see an implementation.

2.4.1 Price Equation

Remember that the Leontief price equation is given by

$$\mathbf{p} = C^T\mathbf{p} + \mathbf{v},$$

where $\mathbf{p}$ is the *price vector*, C is the *consumption matrix* and $\mathbf{v}$ is the value added vector. All quantities are measured per unit of output of each industry. If the matrix $I - C^T$ is invertible, then we can uniquely determine appropriate prices for any value added vector $\mathbf{v}$ via

$$\mathbf{p} = (I - C^T)^{-1}\mathbf{v}.$$

Remember that in the particular example we had been dealing with, we had

$$\begin{aligned}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} &= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.4 & 0.2 \\ 0.7 & 0.1 \end{bmatrix} \right)^{-1} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\ &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}\end{aligned}$$

Reading question 18: Verify that the matrix in the previous equation is actually invertible.

Let's imagine that the value added per unit in industry 1 is $v_1 = 17$ and the value added per unit in industry 2 is $v_2 = 24$. What are the appropriate prices given all of our assumptions? Matlab can crunch this type of thing, no problem:

```
EDU>> C = [0.4 0.7; 0.2 0.1]

C =

    0.4000    0.7000
    0.2000    0.1000
```

```
EDU>> (eye(2) - C') \ [17;24]

ans =

   50.2500
   65.7500
```

So for all industries to be simultaneously satisfied, industry 1 should sell 1 unit of its output for \$50.25 and industry 2 should sell 1 unit of its output for \$65.75. (Side note: If you're more comfortable simplifying the matrix expression before you put it into Matlab, that's totally fine. Myself, I'm pretty bad at arithmetic, so when there's a chance, I let Matlab do it for me.)

Now, what happens to the price solution if the value added per unit of industry 1 changes slightly from $v_1 = 17$ to $v_1 = 18$ while the value added per unit in industry 2 remains constant at $v_2 = 24$?

```
EDU>> (eye(2) - C') \ [18;24]

ans =

   52.5000
   67.5000
```

So for all industries to be simultaneously satisfied, industry 1 should sell 1 unit of its output for \$52.25, and industry 2 should sell 1 unit of its output for \$67.50. Combining the previous results, we can write an expression for the *change* in prices when the value added of industry 1 is increased by 1.

$$\begin{aligned}\Delta\mathbf{p}_1 &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 18 \\ 24 \end{bmatrix} - \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 17 \\ 24 \end{bmatrix} \\ &= \begin{bmatrix} 52.50 \\ 67.50 \end{bmatrix} - \begin{bmatrix} 50.25 \\ 65.75 \end{bmatrix} \\ &= \begin{bmatrix} 2.25 \\ 1.75 \end{bmatrix}\end{aligned}$$

We can think of this vector as the sensitivity of the prices of each industry to changes in the value added per unit of industry 1. The larger the absolute values of the entries of this vector $\Delta\mathbf{p}_1$, the bigger the changes in price when industry 1 changes its value added just a little bit.

The really surprising thing is the this price change information is encoded in the matrix $(I - C^T)^{-1}$! To see this, let's start with the identical first line

above and follow a different mathematical course:

$$\begin{aligned}\Delta\mathbf{p}_1 &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 18 \\ 24 \end{bmatrix} - \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 17 \\ 24 \end{bmatrix} \\ &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \left(\begin{bmatrix} 18 \\ 24 \end{bmatrix} - \begin{bmatrix} 17 \\ 24 \end{bmatrix} \right) \\ &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}.\end{aligned}$$

But for *any* matrix A, the matrix multiplication $A[1, 0, \ldots, 0]^T$ returns the first column of A. And so $\Delta\mathbf{p}_1$ is the first column of the matrix $(I - C^T)^{-1}$.

Reading question 19: Convince yourself that for any $r \times c$ matrix A, the matrix multiplication $A[0, \ldots, 0, 1, 0, \ldots, 0]^T$, where the single 1 is located in the j^{th}component, $1 \leq j \leq c$, returns the j^{th} column of A.

Simply computing the inverse of $I - C^T$ shows us that our analysis is correct.

```
EDU>> inv(eye(2) - C')

ans =

    2.2500    0.5000
    1.7500    1.5000
```

This is a good place to note that only the *difference* of the value added vectors mattered in the preceding computations; their absolute levels make absolutely no impact.

Reading question 20: Repeat the preceding steps using the value added vectors $\mathbf{v} = [1, 2000]^T$ and $\mathbf{v}' = [1, 2001]^T$ in order to show that the second column of $(I - C^T)^{-1}$ represents the sensitivity of prices to changes in the value added per unit of industry 2.

2.4.2 Production Equation

Remember that the Leontief production equation is given by

$$\mathbf{x} = C\mathbf{x} + \mathbf{d},$$

where $\mathbf{x}$ is the *production vector*, C is the *consumption matrix*, and $\mathbf{d}$ is the *final demand vector*. If the matrix $I - C$ is invertible, then the appropriate production for a given final demand can be computed directly via

$$\mathbf{x} = (I - C)^{-1}\mathbf{d}.$$

In the particular case we had been considering, this read

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.4 & 0.7 \\ 0.2 & 0.1 \end{bmatrix} \right)^{-1} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}.$$

The appropriate production level for the final demand vector $\mathbf{d} = [100, 200]^T$ can be computed in Matlab.

```
EDU>> C = [0.4 0.7; 0.2 0.1]

C =

    0.4000    0.7000
    0.2000    0.1000

EDU>> (eye(2) - C) \ [100;200]

ans =

   575
   350
```

In words, to satisfy final demand of 100 units from industry 1 and 200 units from industry 2, industry 1 should manufacture 575 total units, and industry 2 should manufacture 350 total units. But how does the production solution change if the final demand from industry 2 changes by 1 unit so that $\mathbf{d} = [100, 201]^T$. Well,

```
EDU>> (eye(2) - C) \ [100;201]

ans =

  576.7500
  351.5000
```

In words, we need 576.75 total units from industry 1 and 351.5 total units from industry 2. Combining the two preceding results, we can write a numerical form of the change in production associated with an increase in final demand from

industry 2 of 1 unit:

$$\begin{aligned}\Delta\mathbf{x}_2 &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 201 \end{bmatrix} - \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \end{bmatrix} \\ &= \begin{bmatrix} 576.75 \\ 351.5 \end{bmatrix} - \begin{bmatrix} 575 \\ 350 \end{bmatrix} \\ &= \begin{bmatrix} 1.75 \\ 1.5 \end{bmatrix}.\end{aligned}$$

We can think of $\Delta\mathbf{x}_2$ as the sensitivity of production to changes in final demand of output from industry 2. But this sensitivity information is contained within the inverse of $I - C$. We can see this by developing the original expression for $\Delta\mathbf{x}_2$ along a different track.

$$\begin{aligned}\Delta\mathbf{x}_2 &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 201 \end{bmatrix} - \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \end{bmatrix} \\ &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \left(\begin{bmatrix} 100 \\ 201 \end{bmatrix} - \begin{bmatrix} 100 \\ 200 \end{bmatrix} \right) \\ &= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.9 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.\end{aligned}$$

But for *any* matrix A, the matrix multiplication $A[0, 0, \ldots, 1]^T$ returns the last column of A. So $\Delta\mathbf{x}_2$, the change in production associated with an increase of 1 unit of final demand of product 2, is the second column of the matrix $(I - C)^{-1}$.

A simple Matlab computation of $(I - C)^{-1}$ confirms that the two interpretations are identical.

```
EDU>> inv(eye(2) - C)

ans =

    2.2500    1.7500
    0.5000    1.5000
```

Reading question 21: Repeat the preceding steps using the final demand vectors $\mathbf{d} = [75, 1]^T$ and $\mathbf{d} = [76, 1]^T$ in order to show that the first column of $(I - C)^{-1}$ represents the sensitivity of production to changes in the final demand of product 1.

Chapter 3

Discrete-time models

3.1 Converting Customers

Imagine that we model our business based on two types of customers: one-time customers and repeat customers. Say that the number of one-time and repeat customers in month t are o_t and r_t, respectively. These populations are disjoint, so that every customer is either a one-time or a repeat, but no customer is both. Naturally, a one-time customer can become a repeat customer. From data you've gathered, you know that each month 40% of your one-time customers remain one-time customers. You also know that each month around 10% of your repeat customers refer a new customer. Using these two ideas, we can write an expression for the number of one-time customers we expect to have in month $t+1$ in terms of the number of one-time and repeat customers in month t.

$$o_{t+1} = 0.4o_t + 0.1r_t.$$

You also know that there's a 95% chance that repeat customers continue to buy your goods. (A common metric that I've heard is that if a repeat customer has not bought something from you in 3 months then they are removed from the repeat customer group.) Since 40% of your one-time customers remain one-time customers, we can claim that the other 60% have become repeat customers. We can combine these two facts to write an expression for the number of repeat customers we expect to have in month $t+1$ in terms o the number of one-time and repeat customers in month t.

$$r_{t+1} = 0.6o_t + 0.95r_t.$$

We can combine the two previous equations into a single matrix-vector equation which describes how our customer populations change from month to month.

$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.1 \\ 0.6 & 0.95 \end{bmatrix} \begin{bmatrix} o_t \\ r_t \end{bmatrix}$$

$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \mathbf{C} \begin{bmatrix} o_t \\ r_t \end{bmatrix}$$

3.1.1 One-time or repeat?

Here's the question we're going to set out to answer. In the long term, what will the ratio of one-time to repeat customers be for the business modeled by the matrix $\mathbf{C}$? This is a serious question that a lot of industries have to deal with. High end industries in particular often decide that repeat customers are the segment on which they want to focus. After all, the pool containing their potential clientele is small, so it makes sense to work hard to keep any customers you have. Cheap products often rely on the fact that they will have a large number of constantly changing one-time customers, known as *customer churn*, to support their business. Whatever your strategy, it's important make sure that you know what you're getting yourself into. And eigenvectors and eigenvalues can help us make data-based predictions.

An eigenvalue, eigenvector pair λ, $\mathbf{v}$ of a matrix $\mathbf{A}$ satisfy the equation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}.$$

Why in the world would we care about such a thing? Well, imagine that we've found an eigenvalue, eigenvector pair $\lambda, \mathbf{v}$ of our customer transition matrix $\mathbf{C}$. If we start with $\mathbf{v}$ customers in month t, then we have $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$ customers in month $t+1$. There are two things going on here. One, the *total* number of customers has grown by a factor of λ. In other words, λ is the *growth rate* of our customer base, provided that our initial population was described by an eigenvector. Second, if we started with o_t one-time customers and r_t repeat customers in month t, then in month $t+1$ we have $o_{t+1} = \lambda o_t$ one-time customers and $r_{t+1} = \lambda r_t$ repeat customers. Both populations have grown (or shrank) by our growth factor λ, but they have remained *constant* relative to each other. For instance, if one-time customers represented 67% of our total customer base in month t, then they continue to represent 67% of our customer base in time $t+1$. We'll see later that we can say interesting things about the long term behavior of the system even if our initial condition does not line on the span of an eigenvector.

Now, there are a lot of different ways to get the eigenvalues and eigenvectors of a matrix. Probably the easiest is to have Matlab do it for us. First we need to enter the matrix:

```
EDU>> C = [0.4 0.1; 0.6 0.95]

C =

    0.4000    0.1000
    0.6000    0.9500
```

Now, to be honest, I never remember exactly how to use the `eig` function in Matlab; I have to check every time:

```
EDU>> help eig
 EIG    Eigenvalues and eigenvectors.
    E = EIG(X) is a vector containing the eigenvalues of a square
    matrix X.

    [V,D] = EIG(X) produces a diagonal matrix D of eigenvalues and a
    full matrix V whose columns are the corresponding eigenvectors so
    that X*V = V*D.
```

So if we simply type `eig(C)`, Matlab thinks that we just want the eigenvalues:

```
EDU>> eig(C)

ans =
```

```
    0.3067
    1.0433
```

That's pretty handy. We know that the eigenvalues of our matrix are $\lambda_1 = 0.3067$ and $\lambda_2 = 1.0433$. So we know our growth rates, but we don't know the customer bases to which they correspond. To get these, we need to give Matlab a little more guidance:

```
EDU>> [V,D] = eig(C)

V =

   -0.7313   -0.1536
    0.6821   -0.9881


D =

    0.3067         0
         0    1.0433
```

Notice that the main diagonal of $\mathbf{D}$ contains exactly the eigenvalues that we found using our first guess at `eig(C)`. The columns of the matrix $\mathbf{V}$ are the corresponding eigenvectors. For instance, the first column of $\mathbf{V}$ is the eigenvector $\mathbf{v}_1$ corresponding to the eigenvalue $\lambda_1 = 0.3067$. How could we check this? Well, we know that

$$\mathbf{C}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1.$$

In Matlab notation, the left side of the equation becomes

```
C*[-.7313; .6821]

ans =

   -0.2243
    0.2092
```

and the right side becomes

```
EDU>> 0.3067 * [-0.7313; 0.6821]

ans =

   -0.2243
    0.2092
```

They're equal! We've actually seen eigenvalues and eigenvectors in action numerically! Think about what just happened: somehow, magically, the matrix-vector multiplication $\mathbf{C}\mathbf{v}_1$ acted to just shrink both components of $\mathbf{v}_1$ to 30.67% of their original value.

Before we go on, we should clear up a few things related to these eigenvectors. First, you might think that we shouldn't have a fractional number of customers. And you'd certainly be right. Is there any way we can turn these into some that actually represent realistic numbers of people? Sure! Let's think about the vector $\mathbf{w} = 10,000\mathbf{v}_1$. Let's convince ourselves that $\mathbf{w}$ is an eigenvector of $\mathbf{C}$ with eigenvector λ_1:

$$\begin{aligned}\mathbf{C}\mathbf{w} &= 10,000(\mathbf{C}\mathbf{v}_1) \\ &= 10,000\lambda_1\mathbf{v}_1 \\ &= \lambda_1\mathbf{w}.\end{aligned}$$

So we've shown that $\mathbf{w}$ is an eigenvector of $\mathbf{C}$ with eigenvalue λ_1. And you might have already guessed that there's nothing special about the factor of 10,000 outside; we can scale an eigenvector by *any* scalar and still have an eigenvector.

You might also be worried about the fact the first eigenvector predicts that the number of one-time customers will be negative. This is a correct interpretation of the eigenvector's meaning, but we'll see in a little while a context in which this idea makes more sense. We'll table the idea for now.

3.1.2 Simulation

But let's get back to our original question: after a large number of months, what is the ratio of one-time customers to repeat customers? A natural place to start is to simulate the system over time and plot the results. The following code computes the customer populations over time and plots the results.

```
%% define our customer transition matrix
C = [0.4 0.1; 0.6 0.95];

%% set our maximum time frame
t = 50;

%% we'll store the populations as the columns of the matrix X
% if you're not familiar with the function zeros(r,c), check out the
% documentation using the command 'doc zeros'
X = zeros(2,t+1);

%% set the initial populations
% remember X(:,1) means 'column 1, all rows of matrix X'
X(:,1) = [200;100];
```

```
%% iterate over time
for j = 1:t
    X(:,j+1) = C * X(:,j);
end

%% plot the results
plot(X(1,:),X(2,:),'b.');
xlabel('One-time customers');
ylabel('Repeat customers');
title('Customer populations over time');
```

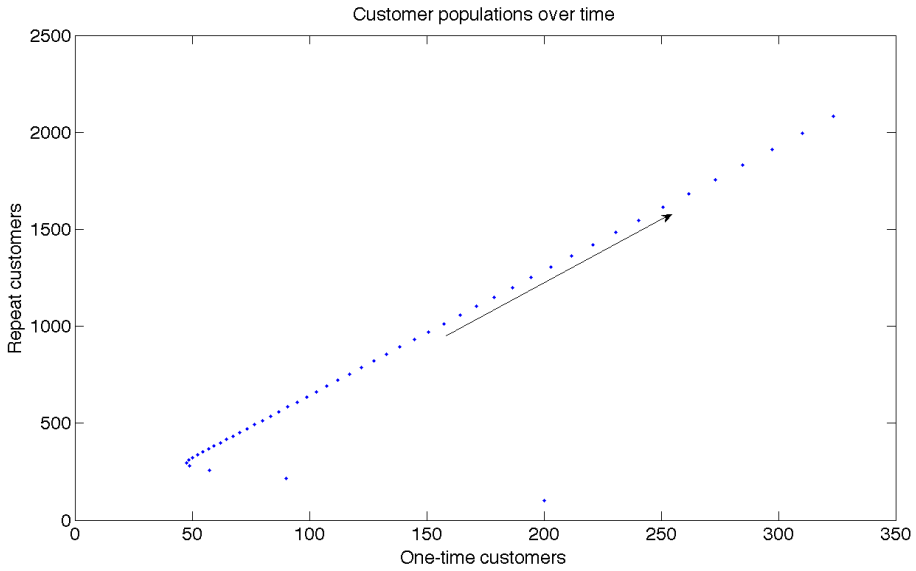

Figure 3.1: Customers over time with time progressing in the same direction as the black arrow.

Figure 2.1 shows the result of executing the code exactly as it's featured here. How do we interpret these results? Well, most pertinent to our question of what will be the ratio of one-time customers to repeat customers, it seems that as time goes on, the customer populations seem to begin to lie on a line which pass through the origin. This is an interesting idea, because all points on a line passing through the origin represent a constant ratio of one-time customers to repeat customers.

The only real question left is this: what is the ratio that we're observing? One easy way to get this value is to print the population at $t = 50$ from the code above. This can be easily accomplished by adding the line `X(:,50)` after the completion of the `for` loop in our code. If we run the code again and the

look in our main Matlab window, we should see something like

```
ans =

   1.0e+03 *

    0.3101
    1.9948
```

Remember that `1.0e03` means, to normal people, 1000, so the model predicts 310.1 one-time customers and 1,994.8 repeat customers in month 50. Not too shabby considering we started with a total of 300 customers in month 1. But getting back to the point, what is our ratio of one-time customers to repeat customers? Well, we can do some simple division to find out

$$\begin{aligned} r &= \frac{310.1}{1,994.8} \\ &= .1555. \end{aligned}$$

In words, we have 15.55% percent as many one-time customers as we have repeat customers. This seems to be implying that our business gets customers and keeps them. Another interesting metric is the percent of our customer base that is composed of one-time customers. A simple calculation shows that only 13.45% of our customers in month 50 are one-time customers.

Completely out of the blue, let me make an observation. Consider the eigenvalue $\mathbf{v}_2$ of $\mathbf{C}$, and let's compute the ratio of the first component (the number of one-time customers) to the second component (the number of repeat-customers):

$$\begin{aligned} \mathbf{v}_2 &= \begin{bmatrix} -0.1536 \\ -0.9881 \end{bmatrix} \\ \Rightarrow r &= \frac{-0.1536}{-0.9881} \\ &= 0.1554. \end{aligned}$$

This is conspicuously close to the ratio we found from our simulation. In other words, the data point representing month is very, very close to a scaled version of the eigenvector $\mathbf{v}_2$. Figure 2.2 shows this graphically.

But we can clearly see from Figure 2.2 that we don't start near the span of the eigenvector. So what's going on here? Let's trying investigating this phenomenon algebraically.

Remember that our initial population vector was

$$\mathbf{x}_1 = \begin{bmatrix} 200 \\ 100 \end{bmatrix}$$

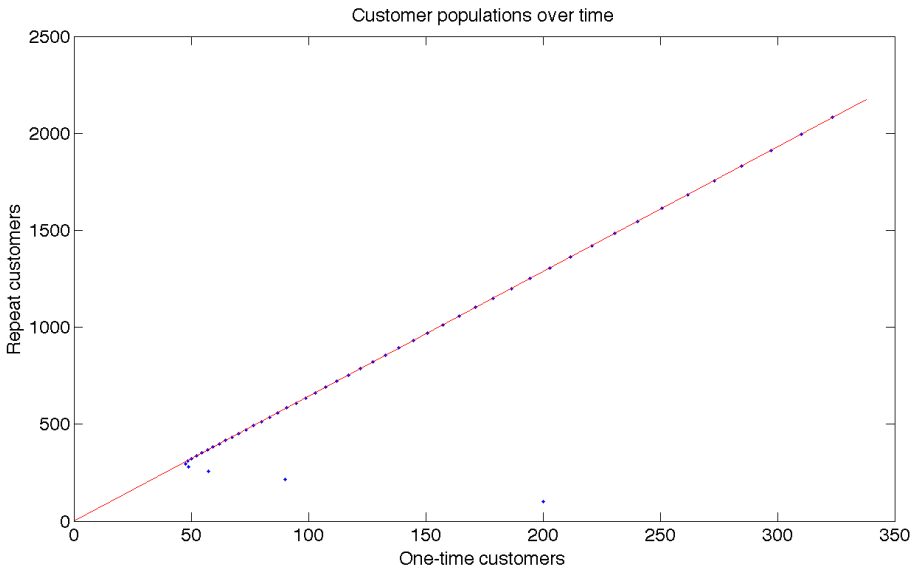

Figure 3.2: Customer populations over time. Red line indicates the span of the eigenvector associated with eigenvalue $\lambda_2 = 1.0236$.

We can represent this as a linear combination of our eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$:

$$\begin{aligned}\mathbf{x}_1 &= c_1\mathbf{v}_1 + c_2\mathbf{v}_2 \\ &= c_1\begin{bmatrix}-0.7313\\0.6821\end{bmatrix} + c_2\begin{bmatrix}-0.1536\\-0.9881\end{bmatrix} \\ &= -220.2946\begin{bmatrix}-0.7313\\0.6821\end{bmatrix} - 253.2649\begin{bmatrix}-0.1536\\-0.9881\end{bmatrix}\end{aligned}$$

Since we're sure that we can represent $\mathbf{x}_1$ as a linear combination of the eigenvectors, let's switch back to symbolic notation to make things a little cleaner. Keep in mind that we could go back to numerical notation any time we wanted.

Now, think about what happens we compute $\mathbf{x}_2$. We have

$$\begin{aligned}\mathbf{x}_2 &= \mathbf{C}\mathbf{x}_1 \\ &= \mathbf{C}(c_1\mathbf{v}_1 + c_2\mathbf{v}_2) \\ &= c_1\mathbf{C}\mathbf{v}_1 + c_2\mathbf{C}\mathbf{v}_2 \\ &= c_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2.\end{aligned}$$

All we've done here is use the fact that $\mathbf{v}_1$ and $\mathbf{v}_2$ are eigenvectors with associated eigenvalues λ_1 and λ_2, respectively. But what have we done in qualitative

terms? We've scaled each eigenvector component independently by its associated growth factor. So the $\mathbf{v}_1$ component of $\mathbf{x}_1$ was shrunk to 30.67% of its original length, and the $\mathbf{v}_2$ component of $\mathbf{x}_1$ was stretched to 104.33% of its original length. What happens if we move from the populations in month 2 to the populations in month 3?

$$\begin{aligned}\mathbf{x}_3 &= \mathbf{C}\mathbf{x}_2 \\ &= \mathbf{C}(c_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2) \\ &= c_1\lambda_1\mathbf{C}\mathbf{v}_1 + c_2\lambda_2\mathbf{C}\mathbf{v}_2 \\ &= c_1\lambda_1^2\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2.\end{aligned}$$

The independent stretching and shrinking have happened again! Now the $\mathbf{v}_1$ component of $\mathbf{x}_3$ is 9.41% of its original length (in $\mathbf{x}_0$) and the $\mathbf{v}_2$ component of $\mathbf{x}_3$ is 108.85% of its original length.

We can generalize this idea to deal with month t:

$$\mathbf{x}_t = c_1\lambda_1^t\mathbf{v}_1 + c_2\lambda_2^t\mathbf{v}_2.$$

Notice that in our case, $\lambda_1 = .3067$, for for large t, the $\mathbf{v}_1$ component of $\mathbf{x}_t$ is nearly nonexistent, because $(.3067)^t \approx 0$ for large t. On the other hand, the $\mathbf{v}_2$ component continues to grow and grow as time passes, exactly because $\lambda_2 = 1.0433$ so that λ_2^t grows without bound as t marches along. What we are observing in the "hook" shape of Figure 2.2 is exactly this phenomenon: the $\mathbf{v}_1$ component of $\mathbf{x}_t$ *shrinks* to zero exponentially while the $\mathbf{v}_2$ component of $\mathbf{x}_t$ *grows* exponentially.

3.1.3 Dominate eigenvectors

Imagine now that we have a much more detailed model of our customer base. After all, breaking our customers into only two groups doesn't give us much of a chance to market to specific groups. Suppose that we are modeling our customer base with n different types of customer, with number of customers of type i during month t contained in a vector $\mathbf{x}_t$. The probability that a customer of type i becomes a customer of type j over the course of a month is contained in the (i, j) entry of a matrix $\mathbf{C}$. (This is exactly how things worked in our two customer model.) Just as we have been doing all along, we can forecast the number of users of all n types in month $t+1$ by using the populations in month t:

$$\mathbf{x}_{t+1} = \mathbf{C}\mathbf{x}_t.$$

Suppose that $\mathbf{C}$ has n distinct eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ with associated eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$. Furthermore, let's assume that the n eigenvectors span $\mathbb{R}^n$, so that any initial condition $\mathbf{x}_1$ can be written as

$$\mathbf{x}_1 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n.$$

We can ask ourselves the same question we did in the two customer type case: in the long term, what does the distribution of customer types look like? Let's first look at the populations in month 2:

$$\mathbf{x}_2 = \mathbf{C}(c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n)$$
$$\mathbf{x}_2 = c_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2 + \ldots + c_n\lambda_n\mathbf{v}_n$$

Just as in the two customer type case, by apply $\mathbf{C}$, and hence update our populations from month 1 to month 2, we have effectively scaled the $\mathbf{v}_i$ component of $\mathbf{x}_t$ by λ_i for each $i = 1, 2, \ldots, n$ in order to form $\mathbf{x}_{t+1}$. Imagine we keep doing this for to compute the populations in month t:

$$\mathbf{x}_t = \mathbf{C}\mathbf{x}_{t-1}$$
$$\mathbf{x}_t = c_1\lambda_1^t\mathbf{v}_1 + c_2\lambda_2^t\mathbf{v}_2 + \ldots + c_n\lambda_n^t\mathbf{v}_n$$

Here's the crucial idea: for large t, λ_1^t is much, much bigger than λ_i^t for any $i = 2, 3, \ldots, n$. So only the first term in the summation describing $\mathbf{x}_t$ actually matters. Mathematically, we can say

$$\mathbf{x}_t \approx c_1\lambda_1^t\mathbf{v}_1.$$

So regardless of how big and/or complicated our matrix $\mathbf{C}$ is, if we can represent any initial condition $\mathbf{x}_1$ as a linear combination of n distinct eigenvectors of $\mathbf{C}$, then the long term distribution of customer types is very close to the distribution described by the eigenvector corresponding to the eigenvalue with largest absolute value! For reasons that should now be clear, this eigenvector is called the *dominant eigenvector*.

3.2 Critical Parameters

Using the concept of dominant eigenvectors, we can make forecasts about the long term distribution of one-time and repeat customers. But how do the parameters of our model affect the dominant eigenvector? In the following sections, we'll step through many of the underlying assumptions of our two customer. By considering these rates related to these assumptions as tunable parameters rather than constants, we'l be able to investigate the affect of changing the underlying assumptions of our two customer model on the long term behavior of the system.

3.2.1 Referral Rate

Imagine that our business launches a campaign which incentivizes repeat customers referring new one-time customers. (We've all seen this type of thing: "Refer a friend, get a month of free cable!") Instead of thinking of our referral rate as being a constant 10% as we did in the previous sections, let's instead think of the referral rate as a tunable parameter r so that our system takes the form

$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \begin{bmatrix} 0.4 & r \\ 0.6 & 0.95 \end{bmatrix} \begin{bmatrix} o_t \\ r_t \end{bmatrix}$$
$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \mathbf{C}_r \begin{bmatrix} o_t \\ r_t \end{bmatrix}.$$

Here's a natural question: what is the minimal referral rate that leads to our business remaining solvent in the long term? Like so many other examples, the first thing we need to do here is to turn this qualitative question into a quantitative one. The idea of dominant eigenvectors and their associated eigenvalues allows us to do this.

We learned in the previous sections that an eigenvalue λ corresponds to the growth rate of a particular eigenvector. If $|\lambda| > 1$, then the associated eigenvector component of our initial conniption grows exponentially over time, and if $|\lambda| < 1$, then the associated eigenvector component of our initial condition shrinks exponentially over time. So in order for our business to not shrink over time, it must be the case that the dominant eigenvector λ must satisfy $|\lambda| \geq 1$. Notice that the limiting case here is $|\lambda| = 1$.

The concept of characteristic equations may have started to seem a little useless once we found out that we can compute the eigenvalues of a matrix using Matlab. But when we allow one of the entries of a matrix be a parameter instead of a fixed value, Matlab can no longer help us. We have to turn back to the characteristic equation.

Take for instance our matrix $\mathbf{C}_r$. The characteristic equation of this matrix

is

$$\begin{aligned}(0.4-\lambda)(0.95-\lambda)-0.6r&=0\\ \lambda^2-(0.4+0.95)\lambda+((0.4)(0.95)-0.6r)&=0\\ \lambda^2-1.35\lambda+(0.38-0.6r)&=0.\end{aligned}$$

Reading question 22: Verify that the preceding statement is true.

Rearranging the equation, we can think of the referral rate as a function of the eigenvalue λ.

$$r=\frac{1}{0.6}(\lambda^2-1.35\lambda+0.38)$$

Notice that for a given λ, there is exactly one corresponding referral rate r. So to find the referral rate r which leads to $\lambda=1$, we just need evaluate the characteristic equation at $\lambda=1$.

$$\begin{aligned}r&=\frac{1}{0.6}(1-1.35+0.38)\\ &=0.05.\end{aligned}$$

So every month we need a minimal referral rate of one new one-time per 20 repeat customers in order for our business to say afloat.

There's an altogether different method we could have used in order to find the value of the referral rate r which would lead to an eigenvalue of $\lambda=1$, and using this method will help us hone some of our skills concerning free variables and linear dependence. Just like the characteristic equation, these ideas keep coming back in different guises in many different linear algebraic problems.

In order for $\lambda=1$ to be an eigenvalue of $\mathbf{C}_r$, it must be the case that the matrix equation

$$(\mathbf{C}_r-\lambda\mathbf{I}_2)\mathbf{v}=\mathbf{0}.$$

has some nontrivial solution $\mathbf{v}$, and this solution $\mathbf{v}$ is exactly the eigenvector associated with eigenvalue $\lambda=1$. We know from our previous work that for there to be a nontrivial solution to the matrix equation $\mathbf{Ax}=\mathbf{0}$, it must be the case that the columns of $\mathbf{A}$ are linearly dependent. Let's take a took at our matrix $\mathbf{C}_r-\lambda\mathbf{I}_2$ when $\lambda=1$.

$$\begin{aligned}\mathbf{C}_r-\mathbf{I}_2&=\begin{bmatrix}0.4-1 & r\\ 0.6 & 0.95-1\end{bmatrix}\\ &=\begin{bmatrix}-0.6 & r\\ 0.6 & -0.05\end{bmatrix}.\end{aligned}$$

Remember that a collection of vectors $\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_n$ are linearly dependent if there exists a nontrivial linear combination of these vectors which equals the

zero vector. In symbols, there must exist $x_1, x_2, \ldots, x_n$, not all zero, such that $x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + \ldots + x_n\mathbf{c}_n = \mathbf{0}$. What does this mean in the case at hand? Well, if $\mathbf{c}_1$ and $\mathbf{c}_2$ are the first and second columns of $\mathbf{C}_r - \mathbf{I}_2$, respectively, then

$$\begin{aligned} x_1\mathbf{c}_1 + x_2\mathbf{c}_2 &= \mathbf{0} \\ -\frac{x_1}{x_2}\mathbf{c}_1 &= \mathbf{c}_2 \\ k\mathbf{c}_1 &= \mathbf{c}_2. \end{aligned}$$

that is, the second column of $\mathbf{C}_r - \mathbf{I}_2$ must be a multiple of the first column. What would our referral rate r need to be to make this be the case? Well, notice that the first and second components of the first column of $\mathbf{C}_r - \mathbf{I}_2$ are negatives of one another. For $\mathbf{c}_2$ to be a multiple of $\mathbf{c}_1$, it must be the case that the first and second entries of $\mathbf{c}_2$ are negatives of one another, too. In other words, it must be the case that $r = 0.05$.

We can use Matlab to verify our analysis.

```
EDU>> C = [0.4 0.05; 0.6 0.95];
EDU>> [V,D] = eig(C)

V =

   -0.7071   -0.0830
    0.7071   -0.9965


D =

    0.3500         0
         0    1.0000
```

An interesting note here is that with a referral rate of 5%, our customer base in the long term is composed almost entirely of repeat customers.

Reading question 23: Explain the preceding claim in terms of the eigenvalues and eigenvectors of $\mathbf{C}_{r=0.05}$.

3.2.2 Customer attrition rate

No matter how hard you try, some of your repeat customers are going to jump ship and go to a competitor. The probability of this happening over the a certain period of time is known as the *customer attrition rate*, or *churn rate*. In our basic example, 95% of our repeat customers remain repeat customers from month to month. So the attrition rate in the basic example is 5%. Let's investigate our business model when we represent the attrition rate as a parameter a. Our

system of equations now looks like

$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.1 \\ 0.6 & 1-a \end{bmatrix} \begin{bmatrix} o_t \\ r_t \end{bmatrix}$$
$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \mathbf{C}_a \begin{bmatrix} o_t \\ r_t \end{bmatrix}.$$

We can ask ourselves a similar set of questions as we did when investigated how changing the referral rate affected our long term customer distribution. Probably the most important problem is to determine the minimum value of the attrition rate a at which our customer base will not go to zero in the long term. In the previous section, we saw two different methods for finding this special parameter value. For this particular case, let's think about the nontrivial solutions of $\mathbf{C}_a - \mathbf{I}_2$.

$$\begin{aligned} \mathbf{C}_a - \mathbf{I}_2 &= \begin{bmatrix} 0.4-1 & 0.1 \\ 0.6 & (1-a)-1 \end{bmatrix} \\ &= \begin{bmatrix} -0.6 & 0.1 \\ 0.6 & -a \end{bmatrix}. \end{aligned}$$

For $\lambda = 1$ to be an eigenvector of $\mathbf{C}_a$, the systems of equations $(\mathbf{C}_a - \mathbf{I}_2)\mathbf{v} = \mathbf{0}$ must have a nontrivial solution, and this happens only if the first and second columns of $\mathbf{C}_a - \mathbf{I}_2$ are multiples of one another. Piecing all of this together, for $\lambda = 1$ it must be the case that $a = 0.1$. Stated differently, no less than 90% of our repeat customers must remain repeat customers month after month in order for our business to survive.

Again, a couple quick Matlab commands can confirm what we've found symbolically.

```
EDU>> C = [0.4 0.1; 0.6 1-.1];
EDU>> [V,D] = eig(C)

V =

   -0.7071   -0.1644
    0.7071   -0.9864


D =

    0.3000         0
         0    1.0000
```

Reading question 24: Discuss the distribution of our customer base in the long term using the eigenvalues and eigenvectors of $\mathbf{C}_{a=0.1}$

3.2.3 Customer conversion rate

I literally can't imagine a more reasonable goal than to convert one-time customers into repeat customers. The percentage of a one-time customers than become repeat customer over the course of a month is called the *conversion rate*. We'll keep assuming, as we did in the basic model, that a one-time customer either remains a one-time customer or becomes a repeat customer over the course of a month. Let's represent our conversion rate with a parameter c. Then our model becomes

$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \begin{bmatrix} 1-c & 0.1 \\ c & 0.95 \end{bmatrix} \begin{bmatrix} o_t \\ r_t \end{bmatrix}$$
$$\begin{bmatrix} o_{t+1} \\ r_{t+1} \end{bmatrix} = \mathbf{C}_c \begin{bmatrix} o_t \\ r_t \end{bmatrix}.$$

Again, let's consider whether there exist nontrivial solutions to $(\mathbf{C}_c - \mathbf{I}_2)\mathbf{v} = \mathbf{0}$. We know that the columns of

$$\begin{aligned} \mathbf{C}_c - \mathbf{I}_2 &= \begin{bmatrix} (1-c)-1 & 0.1 \\ c & 0.95-1 \end{bmatrix} \\ &= \begin{bmatrix} -c & 0.1 \\ c & -0.05 \end{bmatrix} \end{aligned}$$

What's happening here? There's no way to choose $c \neq 0$ such that the first column is a multiple of the second column! If we choose $c = 0$, then obviously the first column $\mathbf{c}_1$ and the second column $\mathbf{c}_2$ are multiples of each, namely through $\mathbf{c}_1 = 0\mathbf{c}_2$.