Applications of Linear Algebra
(Work In Progress April 27, 2018)

Justin Wyss-Gallifent

Contents

Chapter 1

A Review of the Basics

Contents

Here is a brief summary of the critical definitions and theorems necessary. No proofs are provided for theorems - check any introductory linear algebra text. We assume that all vectors are real, all scalars are real, etc. because that's all that is necessary for the remainder of the text.

1.1 Matrices and Vectors

Definition 1.1.0.1. A *matrix* is a rectangular array of numbers. We say it is $n \times m$ if it has n rows and m columns, and that its *dimensions* are $n \times m$.

Example 1.1. The following is a 3×5 matrix:

$$\begin{bmatrix} 1 & 0 & -1 & 2 & 3 \\ 0 & 1.2 & 8 & 0 & 1 \\ -10 & 0 & 1 & 1 & 4 \end{bmatrix}$$

Definition 1.1.0.2. A *vector* is an $n \times 1$ matrix.

Matrices are generally denoted by upper-case letters A, B, etc. while vectors are generally denoted by lower-case letters with a bar over them $\bar{b}$, $\bar{x}$, etc.

Example 1.2. We might write:

$$A = \begin{bmatrix} 1 & 0 & -1 & 2 & 3 \\ 0 & 1.2 & 8 & 0 & 1 \\ -10 & 0 & 1 & 1 & 4 \end{bmatrix} \text{ and } \bar{v} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 3 \end{bmatrix}$$

Definition 1.1.0.3. A matrix is *square* if $n = m$.

If a matrix is denoted by A then the entry in row i and column j will typically be denoted a_{ij} or $a_{(i,j)}$.

If a vector is denoted by $\bar{b}$ then the entry in row i will typically be denoted by b_i.

Definition 1.1.0.4. If A is an $n \times m$ matrix and $\bar{b}$ is an $m \times 1$ vector then we may define $A\bar{b}$ by

$$A\bar{b} = \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + ... + a_{1m}b_m \\ a_{21}b_1 + a_{22}b_2 + ... + a_{2m}b_m \\ ... \\ a_{n1}b_1 + a_{n2}b_2 + ... + a_{nm}b_m \end{bmatrix}$$

Example 1.3. We have:

$$\begin{bmatrix} 1 & 3 & -1 \\ 0 & 2 & -2 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \end{bmatrix} = \begin{bmatrix} (1)(5) + (3)(6) + (-1)(7) \\ (0)(5) + (2)(6) + (-2)(7) \end{bmatrix} = \begin{bmatrix} 16 \\ -2 \end{bmatrix}$$

Definition 1.1.0.5. If A is an $n \times m$ matrix and B is an $m \times p$ matrix then if the columns of B are denoted $\bar{b}_1, ..., \bar{b}_p$ then we may define AB by

$$AB = \begin{bmatrix} A\bar{b}_1 & A\bar{b}_2 & ... & A\bar{b}_p \end{bmatrix}$$

Example 1.4. We have:

$$\begin{aligned} \begin{bmatrix} 1 & 3 & -1 \\ 0 & 2 & -2 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 6 & -2 \\ 7 & 1 \end{bmatrix} &= \left[\begin{bmatrix} 1 & 3 & -1 \\ 0 & 2 & -2 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \end{bmatrix} \quad \begin{bmatrix} 1 & 3 & -1 \\ 0 & 2 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix} \right] \\ &= \begin{bmatrix} 16 & -7 \\ -2 & -6 \end{bmatrix} \end{aligned}$$

Notice that AB may be defined when BA is not (because of the dimensions). Even if both AB and BA are defined they may be different sizes. Even when they're the same size they may be have different entries.

Definition 1.1.0.6. The *main diagonal* of a matrix A is the entries a_{11}, a_{22}, ... , a_{nn}.

Definition 1.1.0.7. The *identity matrix* I_n is the $n \times n$ matrix with 1s on the main diagonal and 0s elsewhere. When the size is clear or implied we simply write I.

Definition 1.1.0.8. If A is a square matrix then the *transpose* of A, denoted A^T, is the matrix whose (i, j) entry equals a_{ji}. That is, it is obtained by reflecting A in the main diagonal.

Definition 1.1.0.9. A square matrix A is *symmetric* if $A^T = A$.

Definition 1.1.0.10. If A is an $n \times n$ square matrix then A is *invertible* if there is another $n \times n$ matrix, denoted A^{-1}, such that $AA^{-1} = I$ and $A^{-1}A = I$.

Example 1.5. Observe that

$$\begin{bmatrix} 5 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 7 \\ -2 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

so that

$$\begin{bmatrix} 5 & 7 \\ 2 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} 3 & 7 \\ -2 & 5 \end{bmatrix} \text{ and } \begin{bmatrix} 3 & 7 \\ -2 & 5 \end{bmatrix}^{-1} = \begin{bmatrix} 5 & 7 \\ 2 & 3 \end{bmatrix}$$

and both matrices are invertible.

Most matrix inverses are not nearly this pretty, nor are inverses easy to find.

The exception is the 2×2 case.

Theorem 1.1.0.1. If

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

then A is invertible iff $ad - bc \neq 0$ in which case

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Proof. Omitted. □

Not all matrices are invertible but "most" are, where "most" means something rigorous and meaningful.

Definition 1.1.0.11. If $\bar{v}, \bar{w} \in \mathbb{R}^n$ then we define the *dot product*

$$\bar{v} \cdot \bar{w} = \bar{v}^T \bar{w} = v_1 w_1 + ... + v_n w_n$$

Definition 1.1.0.12. If $\bar{v} \in \mathbb{R}^n$ then the *magnitude* or *norm* or length of $\bar{v}$ is defined by

$$||\bar{v}|| = \sqrt{v_1^2 + ... + v_n^2}$$

Definition 1.1.0.13. A *diagonal matrix* is a square matrix A such that $a_{ij} = 0$ for $i \neq j$.

1.2 Determinants

Definition 1.2.0.1. If A is an $n \times m$ matrix then the *matrix minor* denoted by A_{ij} is the $(n-1) \times (m-1)$ matrix obtained by removing row i and column j from A.

Definition 1.2.0.2. The *determinant* of an square matrix A denoted $\det(A)$ or just $\det A$ is defined recursively as follows:

- If A is 2×2 then
$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$
- If A is larger than 2×2 then
$$\det(A) = +a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + a_{13}\det(A_{13}) - ... \pm a_{1n}\det(A_{1n})$$

Example 1.6. For a 2×2

$$\det \begin{bmatrix} 5 & 3 \\ -2 & 6 \end{bmatrix} = (5)(6) - (3)(-2) = 36$$

Example 1.7. For a 3×3

$$\begin{aligned} \det \begin{bmatrix} 1 & 2 & -3 \\ 0 & 5 & 1 \\ -2 & 4 & 7 \end{bmatrix} &= +1 \det \begin{bmatrix} 5 & 1 \\ 4 & 7 \end{bmatrix} - 2 \det \begin{bmatrix} 0 & 1 \\ -2 & 7 \end{bmatrix} + (-3) \det \begin{bmatrix} 0 & 5 \\ -2 & 4 \end{bmatrix} \\ &= +1(31) - 2(2) + (-3)(10) \\ &= -3 \end{aligned}$$

Theorem 1.2.0.1. A square matrix A is invertible iff $\det(A) \neq 0$.

Proof. Omitted. □

Example 1.7 Revisited. The matrix:

$$\begin{bmatrix} 1 & 2 & -3 \\ 0 & 5 & 1 \\ -2 & 4 & 7 \end{bmatrix}$$

has determinant $-3 \neq 0$ and hence is invertible.

Mathematically speaking since the chances of having $\det(A) = 0$ are very small it is in this sense that we can say that "most" matrices are invertible since "most"matrices have nonzero determinant.

1.3 Systems of Equations

Definition 1.3.0.1. A linear system of m equations in the variables $x_1, ..., x_n$ given by

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_1 \\ ... &= ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n &= b_m \end{aligned}$$

may be represented by the matrix equation

$$A\bar{x} = \bar{b}$$

Example 1.8. The system of equations

$$\begin{aligned} 2x_1 + 3x_2 - 1x_3 &= 7 \\ -1x_1 + 7x_2 + 4x_3 &= -2 \end{aligned}$$

may be rewritten as

$$\begin{bmatrix} 2 & 3 & -1 \\ -1 & 7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -2 \end{bmatrix}$$

Theorem 1.3.0.1. The matrix equation $A\bar{x} = \bar{b}$ has either no solutions, one solution, or infinitely many solutions. There is one solution iff A is invertible and in that case the solution is given by $\bar{x} = A^{-1}\bar{b}$.

Proof. Omitted. □

Example 1.9. The matrix equation

$$\begin{bmatrix} 5 & 7 \\ 2 & 3 \end{bmatrix} \bar{x} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

has exactly one solution because the matrix is invertible as we saw earlier. The solution is given by

$$\bar{x} = \begin{bmatrix} 5 & 7 \\ 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ -2 & 5 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -9 \end{bmatrix}$$

Definition 1.3.0.2. If A is an $n \times n$ matrix then λ is an *eigenvalue* for A if there is a nonzero vector $\bar{v}$ such that $A\bar{v} = \lambda\bar{v}$. In this case $\bar{v}$ is the corresponding *eigenvector* . The set of all eigenvectors for a given eigenvalue is called the *eigenspace* of that eigenvector.

Notice that any multiple of an eigenvector is also an eigenvector.

Example 1.10. The matrix

$$A = \begin{bmatrix} 4 & 7 \\ 1 & -2 \end{bmatrix}$$

has two eigenvalues. One is $\lambda_1 = 5$ with eigenvector $\begin{bmatrix} 7 \\ 1 \end{bmatrix}$ because:

$$\begin{bmatrix} 4 & 7 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \end{bmatrix} \begin{bmatrix} 35 \\ 5 \end{bmatrix} = 5 \begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

The other eigenvalue is $\lambda_2 = -3$ with eigenvector $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. This can be easily checked.

Definition 1.3.0.3. If A is an $n \times n$ matrix then the *characteristic polynomial* of A denoted char(A), is defined by

$$\text{char}(A) = \det(\lambda I - A)$$

Theorem 1.3.0.2. The eigenvalues of a matrix A are the roots of the characteristic polynomial.

Proof. Omitted. □

Since the characteristic polynomial has degree n this tells us that an $n \times n$ matrix has n eigenvalues, counting multiplicity.

Example 1.11. If we have:

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 1 & 2 & 3 \\ 0 & 4 & 3 \end{bmatrix}$$

then

$$\begin{aligned} \operatorname{char}(A) &= \det(\lambda I - A) \\ &= \det \begin{bmatrix} \lambda - 1 & -2 & 1 \\ -1 & \lambda - 2 & -3 \\ 0 & -4 & \lambda - 3 \end{bmatrix} \\ &= \lambda^3 - 6\lambda^2 - 3\lambda + 16 \end{aligned}$$

The eigenvalues of the matrix are roots of this, $\lambda_1 \approx 6.0593$, $\lambda_2 \approx -1.6549$ and $\lambda_3 \approx 1.5956$.

1.4 Linear Independence

Definition 1.4.0.1. A set of vectors $\{\bar{v}_1, ..., \bar{v}_n\}$ is *linearly independent* if $a_1\bar{v}_1 + ... + a_n\bar{v}_n = \bar{0}$ implies $a_1 = ... = a_n = 0$.

As a consequence of this a set of just two vectors is linearly independent iff neither is a multiple of the other.

Example 1.12. The set of vectors

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \\ -1 \end{bmatrix} \right\}$$

is a linearly independent set.

A classic way of thinking of linear independence is that it is impossible to write any one of the vectors as a linear combination of the other vectors.

One of the consequences of having a linearly independent set is that if some vector $\bar{v}$ is a linear combination of that set then only that specific linear combination works.

Definition 1.4.0.2. If a set is not linearly independent then it is *linearly dependent.*

A classic way of thinking of linear dependence is that one vector may be written as a linear combination of the others.

Example 1.13. The set of vectors

$$\left\{\begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}\right\}$$

is linearly dependent. Observe that

$$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

1.5 Vector Spaces and Bases

Definition 1.5.0.1. A *vector space* is a nonempty set V of vectors such that the following properties hold:

1. $\bar{0} \in V$.
2. If $\bar{u}, \bar{v} \in V$ then $\bar{u} + \bar{v} \in V$.
3. $\bar{u} \in V$ then $-\bar{u} \in V$.
4. If $\bar{u} \in V$ and $c \in \mathbb{R}$ then $c\bar{u} \in V$.

Note that 1 actually follows from 2 and 3 together but it's worth listing on its own anyway.

Definition 1.5.0.2. Given a set of vectors $S = \{\bar{v}_1, ..., \bar{v}_n\}$ then the *span* of S denoted span(S) is the set of all linear combinations of vectors in S. More rigorously

$$\text{span}(S) = \left\{ a_1\bar{v}_1 + ... + a_n\bar{v}_n \,\middle|\, a_1, ..., a_n \in \mathbb{R} \right\}$$

Example 1.14. If

$$S = \left\{\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \\ -1 \end{bmatrix}\right\}$$

Then

$$\text{span}(S) = \left\{ a_1 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} + a_2 \begin{bmatrix} 0 \\ 5 \\ -1 \end{bmatrix} \;\middle|\; a_1, b_1 \in \mathbb{R} \right\}$$

Theorem 1.5.0.1. Given a set of vectors $S = \{\bar{v}_1, ..., \bar{v}_n\}$ the span of S is a vector space.

Proof. Omitted. □

Definition 1.5.0.3. If V is a vector space then a *basis* for V is a linearly independent set B of vectors such that $V = \text{span}(B)$.

Example 1.15. The set

$$B = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \right\}$$

is a basis for $\mathbb{R}^3$.

Essentially a basis for a vector space V is a set of building blocks B such that each vector in V can be written uniquely as a linear combination of vectors in B.

Definition 1.5.0.4. If A is an $m \times n$ matrix then the *column space of* A denoted $\text{col}(A)$ is the span of the columns of A.

Theorem 1.5.0.2. Every vector space has a basis and the number of vectors in a basis of a vector space is independent of the choice of basis. That is, every basis has exactly the same number of vectors as every other basis.

Proof. Omitted. □

Definition 1.5.0.5. For a vector space V the *dimension* of V denoted $\dim(V)$, is defined as the number of vectors in a basis of V.

Theorem 1.5.0.3. If A is an $n \times n$ matrix and λ is an eigenvalue with eigenspace V then the dimension of the eigenspace is less than or equal to the multiplicity of λ as a root of the characteristic polynomial.

Proof. Omitted. □

Example 1.16. The matrix

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 10 & 12 & -30 \\ 5 & 5 & -13 \end{bmatrix}$$

Has characteristic polynomial

$$\lambda^3 - \lambda^2 - 8\lambda + 12 = (\lambda - 2)^2(\lambda + 3)$$

Consequently the eigenspace for $\lambda_1 = 2$ has dimension either 1 or 2. In this case it's 2 but that's a bit more work.

1.6 Orthogonality and Orthonormality

Definition 1.6.0.1. Two vectors are *orthogonal* if their dot product equals zero. A set of vectors $\{\bar{v}_1, ..., \bar{v}_n\}$ is orthogonal if $\bar{v}_i \cdot \bar{v}_j = 0$ for all $i \neq j$.

Example 1.17. The set of vectors

$$\left\{ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} -5 \\ 2 \\ 1 \end{bmatrix} \right\}$$

is an orthogonal set of vectors.

Definition 1.6.0.2. A set of vectors $\{\bar{v}_1, ..., \bar{v}_n\}$ is *orthonormal* if $\bar{v}_i \cdot \bar{v}_j = 0$ for all $i \neq j$ and $||\bar{v}_i|| = 1$ for all i.

Example 1.18. The set of vectors in the previous example is orthonormal if each vector is divided by its magnitude:

$$\left\{ \begin{bmatrix} 1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}, \begin{bmatrix} 0 \\ 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix}, \begin{bmatrix} -5/\sqrt{30} \\ 2/\sqrt{30} \\ 1/\sqrt{30} \end{bmatrix} \right\}$$

is an orthogonal set of vectors.

Definition 1.6.0.3. A square matrix is *orthogonal* if the column vectors form an orthonormal set.

Example 1.19. The matrix

$$A = \begin{bmatrix} 1/\sqrt{6} & 0 & -5/\sqrt{30} \\ 2/\sqrt{6} & 1/\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & -2/\sqrt{5} & 1/\sqrt{30} \end{bmatrix}$$

is orthogonal.

Theorem 1.6.0.1. A square matrix A is orthogonal iff $A^TA = AA^T = I$. That is, if $A^T = A^{-1}$.

Proof. Omitted. □

Orthongonal matrices are great simply because $A^{-1} = A^T$ and so the inverse is really convenient.

1.7 Diagonalizable Matrices

Definition 1.7.0.1. An $n \times n$ matrix A is *diagonalizable* if there exists an $n \times n$ invertible matrix P and an $n \times n$ diagonal martix D such that

$$A = PDP^{-1}$$

.

Theorem 1.7.0.1. A matrix A is diagonalizable iff the dimension of each eigenspace equals the multiplicity of the corresponding eigenvalue in the characteristic polynomial.

Proof. Omitted. □

Theorem 1.7.0.2. If A is diagonalizable then the invertible matrix P is formed using the eigenvectors of A and the diagonal matrix D is formed using the eigenvalues of A. The eigenvector in column i corresponds to the eigenvalue in column i.

Proof. Omitted. □

Example 1.20. If

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 10 & 12 & -30 \\ 5 & 5 & -13 \end{bmatrix}$$

Then

$$A = PDP^{-1}$$

where

$$P = \begin{bmatrix} 0 & 0 & 0.4016 \\ -0.8944 & -0.9487 & -0.9006 \\ -0.4472 & -0.3162 & -0.1663 \end{bmatrix}$$

$$D = \begin{bmatrix} -3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

In this case the first column of P is an eigenvector corresponding to the eigenvalue $\lambda_1 = -3$ and the second and third columns of P are eigenvectors corresponding to the eigenvalue $\lambda_2 = 2$ which has multiplicity 2 and for which the dimension of the eigenspace is also 2.

Definition 1.7.0.2. An $n \times n$ matrix A is *orthogonally diagonalizable* if there exists an $n \times n$ orthogonal matrix Q and an $n \times n$ diagonal matrix D such that

$$A = QDQ^T$$

Theorem 1.7.0.3. A matrix A is orthogonally diagonalizable iff it is symmetric.

Proof. Omitted. □

Example 1.21. The matrix

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 3 \\ -1 & 3 & 4 \end{bmatrix}$$

is symmetric hence orthogonally diagonalizable.

Chapter 2

Leontief Input-Output Model

Contents

2.1 Introduction

2.1.1 Introduction

Wassily Leontief was an economist who was one of the first people to do computational analysis of economics. Moreover his work involved one of the first uses of a computer to produce this analysis, done in 1949 at Harvard. For his work he won a Nobel Prize in 1973.

Leontief based his approach on the idea that an economy is basically divided into sectors and each sector produces a product. In order to produce its product each sector requires input which must come from possibly all of the sectors, including itself.

Consequently, overall, the amount that the full economy must produce has to include any desired external demand as well as the internal demand which feeds back into the economy so that each sector can do its job.

As a pseudo-equation:

$$\text{Total Amount Produced} = \text{Internal Demand} + \text{External Demand}$$

To see this more clearly let's look at a basic example:

Example 2.1. Suppose there are three sectors each producing units of its product. In order for each sector to function it needs some of its own product as well as some of the other sectors' products. Suppose we have the following:

- To produce 1 unit of product 1 it takes 0.10 units of product 1, 0.20 units of product 2, and 0.25 units of product 3.
- To produce 1 unit of product 2 it takes 0.15 units of product 1, 0 units of product 2, and 0.40 units of product 3.
- To produce 1 unit of product 3 it takes 0.12 units of product 1, 0.30 units of product 2, and 0.20 units of product 3.

If the sectors are to produce p_1 units of product 1, p_2 units of product 2, and p_3 units of product 3, then what is the total internal requirement? This is the internal demand.

Consider for example how much product 1 is required in total:

- To produce p_1 units of product 1 requires $0.10p_1$ units of product 1.
- To produce p_2 units of product 2 requires $0.15p_2$ units of product 1.
- To produce p_3 units of product 3 requires $0.12p_3$ units of product 1.

Therefore in total we require $0.10p_1 + 0.15p_2 + 0.12p_3$ units of product 1.

Following this same approach we find that in total:

- We will need $0.10p_1 + 0.15p_2 + 0.12p_3$ units of product 1.
- We will need $0.20p_1 + 0p_2 + 0.30p_3$ units of product 2.
- We will need $0.25p_1 + 0.40p_2 + 0.20p_3$ units of product 3.

Since we must produce these quantities to satisfy internal demand we therefore have:

$$\begin{aligned}
p_1 &= 0.10p_1 + 0.15p_2 + 0.12p_3\\
p_2 &= 0.20p_1 + 0p_2 + 0.30p_3\\
p_3 &= 0.25p_1 + 0.40p_2 + 0.20p_3
\end{aligned}$$

This may also be written as:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = p_1 \begin{bmatrix} 0.10 \\ 0.20 \\ 0.25 \end{bmatrix} + p_2 \begin{bmatrix} 0.15 \\ 0 \\ 0.40 \end{bmatrix} + p_3 \begin{bmatrix} 0.12 \\ 0.30 \\ 0.20 \end{bmatrix}$$

which can be written nicely in matrix form as:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0.10 & 0.15 & 0.12 \\ 0.20 & 0 & 0.30 \\ 0.25 & 0.40 & 0.20 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

In addition suppose there is an external demand of d_1, d_2 and d_3 units for the three products respectively then the total that needs to be produced is:

$$\underbrace{\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}}_{\text{Total Produced}} = \underbrace{\begin{bmatrix} 0.10 & 0.15 & 0.12 \\ 0.20 & 0 & 0.30 \\ 0.25 & 0.40 & 0.20 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}}_{\text{Internal Demand}} + \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}}_{\text{External Demand}}$$

2.1.2 Generalization

Definition 2.1.2.1. The *Leontief Input-Output Model* is given by:

$$\bar{p} = M\bar{p} + \bar{d}$$

Definition 2.1.2.2. The matrix M is the *consumption matrix*.

Definition 2.1.2.3. The consumption matrix is made up of *consumption vectors*. The j^{th} column is the j^{th} consumption vector and contains the necessary input required from each of the sectors for Sector j to produce one unit of output.

Notice that in the consumption matrix the requirements for producing one unit of a given product becomes a column (rather than a row) of the matrix. This is often a source of confusion.

Definition 2.1.2.4. The vector $\bar{p}$ is the *production vector.*

Definition 2.1.2.5. The vector $\bar{d}$ is the *external demand vector.*

Definition 2.1.2.6. The vector $M\bar{p}$ is the *internal demand vector.*

Two associated definitions:

Definition 2.1.2.7. An economy is *open* if $\bar{d} \neq \bar{0}$ and *closed* if $\bar{d} = \bar{0}$.

In a closed economy all of the output that is produced by the various sectors is fed back in as input to those sectors - there is no external demand. If the economy is closed this has serious ramifications on M which will be discussed later. Closed economies are mathematically rare.

The terms *open* and *closed* are used in other ways in economics as well so be cautious.

2.1.3 Goal

The primary goal here is the following: We know the consumption matrix and the external demand and we wish to set the amounts that each sector must produce in order to satisfy both internal and external demand.

In other words we know M and $\bar{d}$ and we wish to know $\bar{p}$.

2.2 Solving Problems

If we blindly attempted to solve for $\bar{p}$ we might try:

$$\begin{aligned} M\bar{p} + \bar{d} &= \bar{p} \\ \bar{p} - M\bar{p} &= \bar{d} \\ (I - M)\bar{p} &= \bar{d} \end{aligned}$$

However at this point we make a few mathematical observations:

- If $I - M$ is invertible this has only one solution.
- If $I - M$ is invertible and $\bar{d} = \bar{0}$ then the only solution is $\bar{p} = \bar{0}$.

- If $I - M$ is not invertible then this may have none, one or infinitely many solutions.
- If $I - M$ is not invertible and $\bar{d} = \bar{0}$ then there are infinitely many solutions.

2.2.1 Method - Open Economy

Let's explore the case where $\bar{d} \neq \bar{0}$. There are two sub-cases, where $I - M$ is invertible and where it is not. Note that statistically speaking most matrices are invertible because most matrices have nonzero determinant.

Case: If $\bar{d} \neq \bar{0}$ and $I - M$ is invertible then we may simply solve:

$$\begin{aligned} M\bar{p} + \bar{d} &= \bar{p} \\ \bar{p} - M\bar{p} &= \bar{d} \\ (I - M)\bar{p} &= \bar{d} \\ \bar{p} &= (I - M)^{-1}\bar{d} \end{aligned}$$

It's worth noting that in most reasonable economies $(I - M)^{-1}\bar{d}$ is nonnegative (has nonnegative entries) for reasonable $\bar{d}$. Pathological examples are embedded in the exercises.

Example 2.1 Revisited.

Suppose we have our initial example - three sectors with consumption matrix:

$$M = \begin{bmatrix} 0.10 & 0.15 & 0.12 \\ 0.20 & 0 & 0.30 \\ 0.25 & 0.40 & 0.20 \end{bmatrix}$$

and suppose we have an external demand of:

$$\bar{d} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$$

Then the total amount that must be produced is given by:

$$\bar{p} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.10 & 0.15 & 0.12 \\ 0.20 & 0 & 0.30 \\ 0.25 & 0.40 & 0.20 \end{bmatrix} \right)^{-1} \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix} = \begin{bmatrix} 281.30 \\ 464.86 \\ 695.34 \end{bmatrix}$$

So the production of the three sectors should be set at these values.

Note: It's interesting to note that most of the production is for internal rather than external demand because external demand is only

$$\begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$$

so that

$$\begin{bmatrix} 281.30 \\ 464.86 \\ 695.34 \end{bmatrix} - \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix} = \begin{bmatrix} 181.30 \\ 264.86 \\ 395.34 \end{bmatrix}$$

is being used up internally.

This is because the internal demands are so high - this economy is not very efficient!

Here is an example with smaller internal demands; It's a much more efficient economy:

Example 2.2. Suppose we have three sectors with consumption matrix:

$$M = \begin{bmatrix} 0.01 & 0.002 & 0.04 \\ 0.02 & 0.004 & 0 \\ 0 & 0.01 & 0.02 \end{bmatrix}$$

and suppose we have an external demand of:

$$\bar{d} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$$

Then the total amount that must be produced is given by:

$$\bar{p} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.01 & 0.002 & 0.04 \\ 0.02 & 0.004 & 0 \\ 0 & 0.01 & 0.02 \end{bmatrix} \right)^{-1} \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix} = \begin{bmatrix} 113.873 \\ 203.09 \\ 308.195 \end{bmatrix}$$

So the production of the three sectors should be set at these values. Notice that most of this production goes directly to the external demand.

Case: If $\bar{d} \neq \bar{0}$ and $I - M$ is not invertible then there may be no solutions or infinitely many solutions.

An example can easily be constructed economically; If sector 1 required 1 unit of its own product to make 1 unit of its own product then no solution could be found if the external demand for product 1 were nonzero, since sector 1 could never product enough.

Likewise imagine two sectors who were fulfilling some external demand for two products with a single solution. If, suddenly a third sector appeared which had the same property as above, but if the external demand for that third product were 0, then that third sector could do whatever it liked, using up its own product, yielding infinitely many solutions.

We won't encounter this example much so we'll leave it there for now.

2.2.2 Method - Closed Economy

Let's explore the case where $\bar{d} = \bar{0}$. There are two sub-cases, where $I - M$ is invertible and where it is not.

Note that having $\bar{d} = \bar{0}$ means that there is no external demand and so the internal demand must balance exactly with production. This is economically rare and consequently mathematically unlikely to have solutions as we will see.

Case: If $\bar{d} = 0$ and $I - M$ is invertible then the only solution is $\bar{p} = \bar{0}$. This is fairly boring so we'll stop there.

Case: If $\bar{d} = 0$ and $I - M$ is not invertible observe that $I - M$ is not invertible iff there is a $\bar{p} \neq \bar{0}$ with $(I - M)\bar{p} = \bar{0}$ iff there is a $\bar{p} \neq \bar{0}$ with $M\bar{p} = 1\bar{p}$ iff M has 1 as an eigenvalue.

This means that in this case $\bar{p}$ may be any eigenvector corresponding to the eigenvalue 1.

Example 2.3. Consider the consumption matrix

$$M = \begin{bmatrix} 0.1 & 0.4 & 0 \\ 0.2 & 0.4 & 0.9 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}$$

This matrix has an eigenvalue of 1 with corresponding unit (length 1) eigenvector

$$\bar{p} = \begin{bmatrix} 0.3605 \\ 0.8111 \\ 0.4606 \end{bmatrix}$$

The fact that any multiple of this vector is an eigenvector indicates that the three sections can produce in combination any multiple of this.

For example they can produce any of the following:

$$\begin{bmatrix} 3.605 \\ 8.111 \\ 4.606 \end{bmatrix} \text{ or } \begin{bmatrix} 36.05 \\ 81.11 \\ 46.06 \end{bmatrix} \text{ or } \begin{bmatrix} 360.5 \\ 811.1 \\ 460.6 \end{bmatrix} \text{ or } \begin{bmatrix} 3605.0 \\ 8111.0 \\ 4606.0 \end{bmatrix} \text{ or } \begin{bmatrix} 7.21 \\ 16.222 \\ 9.212 \end{bmatrix}$$

This makes sense because the economy can simply scale up everything simultaneously.

2.3 Notes About $(I - M)^{-1}$

2.3.1 Interpretation of Entries

Observe that the equation

$$\bar{p} = (I - M)^{-1}\bar{d}$$

gives the relationship between the external demand and the amount that the sectors must produce.

However the entries in $(I - M)^{-1}$ itself have their own interpretation. To understand what they mean, suppose that we have some M and some $\bar{d}$. For the sake of simplicity assume M is 3×3.

We know that production must be set at

$$\bar{p} = (I - M)^{-1}\bar{d}$$

Let's investigate what happens if the external demand for Product 1 changes by $+1$.

The new external demand is

$$\bar{d} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

How does $\bar{p}$ change? Well

$$\begin{aligned}
\bar{p}_{new} &= (I-M)^{-1}\left(\bar{d} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) \\
&= (I-M)^{-1}\bar{d} + (I-M)^{-1}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \bar{p} + (I-M)^{-1}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}$$

Note that the vector

$$(I-M)^{-1}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

contains the first column of $(I-M)^{-1}$. Which shows us that the first colum in $(I-M)^{-1}$ indicates how the production must change in each of the three sectors if the external demand in the first sector changes by +1.

Consider this example:

Example 2.4. Suppose

$$M = \begin{bmatrix} 0.1 & 0.15 & 0.12 \\ 0.2 & 0 & 0.3 \\ 0.25 & 0.4 & 0.2 \end{bmatrix}$$

If we calculate we find:

$$(I-M)^{-1} = \begin{bmatrix} 1.2660 & 0.3128 & 0.3072 \\ 0.4375 & 1.2850 & 0.5473 \\ 0.6144 & 0.7400 & 1.6200 \end{bmatrix}$$

So now if the external demand for Product 1 changes by +1 the production in the three sectors must change by +1.266, +0.4375 and +0.6144 respectively.

This argument extends to the following:

Fact 2.3.1.1. If the external demand for sector j changes by +1 then column j of $(I-M)^{-1}$ indicates how production must change in all sectors in order to compensate.

Or entry-by-entry:

Fact 2.3.1.2. If the external demand for sector j changes by $+1$ then the i^{th} entry of column j of $(I-M)^{-1}$ indicates how the production in sector i must change in order to compensate.

In addition it's easy to see:

Fact 2.3.1.3. This change is linear in nature in that if the external demand for sector j changes by, for example, $+2$ then we can simply double column j.

Example 2.5. If an economy with five sectors has consumption matrix

$$M = \begin{bmatrix} 0.0200 & 0 & 0.1000 & 0.0200 & 0 \\ 0.0600 & 0 & 0.0400 & 0 & 0.0700 \\ 0 & 0.0150 & 0 & 0.0110 & 0 \\ 0 & 0.0220 & 0 & 0 & 0.0800 \\ 0.0300 & 0 & 0.0320 & 0.0100 & 0 \end{bmatrix}$$

then

$$(1-M)^{-1} = \begin{bmatrix} 1.0206 & 0.0020 & 0.1022 & 0.0216 & 0.0019 \\ 0.0634 & 1.0008 & 0.0486 & 0.0025 & 0.0703 \\ 0.0010 & 0.0153 & 1.0008 & 0.0110 & 0.0020 \\ 0.0039 & 0.0221 & 0.0039 & 1.0009 & 0.0816 \\ 0.0307 & 0.0008 & 0.0351 & 0.0110 & 1.0009 \end{bmatrix}$$

so for example if the external demand for the fourth sector changes by $+1$ then the production of the five sectors will need to change by

$$\text{Column } 4 = \begin{bmatrix} 0.0216 \\ 0.0025 \\ 0.0110 \\ 1.0009 \\ 0.0110 \end{bmatrix}$$

respectively. For example sector 1 must change production by $+0.02155$, sector 2 must change production by $+0.002506$, and so on.

Notice that the production of sector 4 must go up by more than 1. This makes sense since it must produce enough to cover the new external demand for $P4$ plus enough to cover the internal demands of all the sectors (including itself) as they work together in harmony to increase production.

And for example if the external demand for the sector 2 changes by -2 then the production of the five sectors will need to change by

$$-2(\text{Column } 2) = -2 \begin{bmatrix} 0.0216 \\ 0.0025 \\ 0.0110 \\ 1.0009 \\ 0.0110 \end{bmatrix} = \begin{bmatrix} 0.0431 \\ 0.0050 \\ 0.0221 \\ 2.0019 \\ 0.0220 \end{bmatrix}$$

respectively.

2.3.2 Calculation of Entries of an Inverse

Calculating a matrix inverse can be fairly intensive so it's useful to remember a sneaky formula for calculating entries one by one. This can be useful when coupled with the above meaning of the entries in $(I - M)^{-1}$.

Definition 2.3.2.1. If A is an $n \times n$ matrix then the (i, j)-*cofactor* of A is

$$C_{ij} = (-1)^{i+j} \det(A_{ij})$$

where A_{ij} is the matrix A with row i and column j removed.

Theorem 2.3.2.1. Let A be an invertible $n \times n$ matrix, then

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$$

where adj(A) is the *adjugate* of A and is defined as follows. Note the peculiar subscript order.

$$\text{adj}(A) = \begin{bmatrix} C_{11} & C_{21} & ... & C_{n1} \\ C_{12} & C_{22} & ... & C_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n} & C_{2n} & ... & C_{nn} \end{bmatrix}$$

Proof. Omitted. □

In a brief and compact form the adjugate method for the matrix inverse states that the (i, j) entry of A^{-1} may be found by:

$$\begin{aligned}\left(A^{-1}\right)_{ij} &= \frac{1}{\det(A)} C_{ji} \\ &= \frac{1}{\det(A)}(-1)^{i+j}\det(A_{ij}) \\ &= (-1)^{i+j}\frac{\det(A_{ji})}{\det(A)}\end{aligned}$$

where A_{ji} is the matrix A with row j and column i removed.

Or in a more expanded form:

$$A^{-1} = \frac{1}{\det(A)}\begin{bmatrix} +\det(A_{11}) & -\det(A_{21}) & +\det(A_{31}) & \dots \\ -\det(A_{12}) & +\det(A_{22}) & -\det(A_{32}) & \dots \\ +\det(A_{13}) & -\det(A_{23}) & +\det(A_{33}) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Example 2.6. Suppose the consumption matrix for three sectors is:

$$M = \begin{bmatrix} 0.10 & 0.15 & 0.12 \\ 0.20 & 0 & 0.30 \\ 0.25 & 0.40 & 0.20 \end{bmatrix}$$

Suppose the external demand for sector 3 change by +1. How must production in sector 2 change?

This value is stored in row 2 of column 3 of $(I-M)^{-1}$. That is, the $(2,3)$-entry of $(I-M)^{-1}$. To find this value first note that

$$I - M = \begin{bmatrix} 0.90 & -0.15 & -0.12 \\ -0.20 & 1.00 & -0.30 \\ -0.25 & -0.40 & 0.80 \end{bmatrix}$$

For this matrix (call it A) we need $(A^{-1})_{23}$ and we know

$$(A^{-1})_{23} = (-1)^{3+2}\frac{\det(A_{32})}{\det(A)}$$

Well

$$\det(A_{32}) = \det\begin{bmatrix} 0.90 & -0.12 \\ -0.20 & -0.30 \end{bmatrix} = -0.2940$$

and

$$\begin{aligned}\det(A) &= +\,(0.9)\,[(1)(0.8) - (-0.3)(-0.4)]\\ &\quad -(-0.15)\,[(-0.2)(0.8) - (-0.3)(-0.25)]\\ &\quad +(-0.12)\,[(-0.2)(-0.4) - (1)(-0.25)]\\ &= 0.5372\end{aligned}$$

So then our answer is

$$(A^{-1})_{23} = (-1)^{2+3}\frac{\det(A_{32})}{\det(A)} = -\frac{-0.2940}{0.5372} = 0.5473$$

meaning that if the external demand for sector 3 changes by +1 then the production in sector 2 must change by +0.5473. Other sectors must change production too, of course, but this lets us know about just this sector.

2.3.3 Expressing as an Infinite Sum

There is an interesting fact about certain matrix inverses which is worth mentioning here because it leads to an interesting observation.

We know from basic Taylor series that

$$\tfrac{1}{1-x} = 1 + x + x^2 + x^3 + ... \qquad \text{for } |x| < 1$$

which can also be written as

$$(1-x)^{-1} = 1 + x + x^2 + x^3 + ... \qquad \text{for } |x| < 1$$

What this is saying is that $1 + x + x^2 + x^3 + ...$ is the multiplicative inverse of $1 - x$ when $|x| < 1$.

There is a similar fact about matrices:

Fact 2.3.3.1. Under certain conditions for a square matrix M we have:

$$(1 - M)^{-1} = I + M + M^2 + M^3 + ...$$

Fact 2.3.3.2. The actual conditions on M are that the above is true iff

$$I = \left[\sum_{k=0}^{\infty} M^k\right](I - M)$$

which is true iff

$$I = \lim_{n\to\infty}\sum_{k=0}^{n} M^k - M^{k+1}$$

The right side of this equals

$$\lim_{n\to\infty} I - M^{n+1}$$

so that we get our desired result iff

$$\lim_{n\to\infty} M^{n+1} = 0$$

which occurs iff the maximum absolute value of all the eigenvalues of M is less than 1 (this is called the *operator norm* of the matrix M).

This will be true for all realistic consumption matrices.

If the values in M are small and especially if the matrix is very sparse, meaning if it has lots of zeros, then the sum converges very quickly and finding the sum up until a certain point can be an efficient way to approximate the inverse.

Example 2.7. For example if

$$\begin{bmatrix} 0.100 & 0 & 0.050 & 0 \\ 0.010 & 0 & 0.020 & 0.025 \\ 0 & 0.080 & 0 & 0 \\ 0.030 & 0 & 0 & 0.020 \end{bmatrix}$$

Then if we calculate $I + M + M^2 + ...$ we find that $I + M + M^2 + ... + M^4$ $I + M + M^2 + ... + M^5$ match to four decimal places:

$$I + M + M^2 + M^3 + M^4 + M^5 = \begin{bmatrix} 1.1112 & 0.0045 & 0.0556 & 0.0001 \\ 0.0120 & 1.0017 & 0.0206 & 0.0256 \\ 0.0010 & 0.0801 & 1.0017 & 0.0020 \\ 0.0340 & 0.0001 & 0.0017 & 1.0204 \end{bmatrix}$$

So we can reasonably assume that the sum has "settled" and that this would be a good approximation for $(I - M)^{-1}$, and in fact this matrix actually agrees with $(I - M)^{-1}$ to every digit shown.

So now if we have a very sparse matrix M as above then if we're given some $\bar{d}$ we can find the corresponding $\bar{p}$ approximately and quickly.

Example 2.8. If M is as above and

$$\bar{d} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Then the corresponding $\bar{p}$ can be approximated:

$$\begin{aligned}
\bar{p} &= (I - M)^{-1}\bar{d} \\
\bar{p} &\approx (I + M + M^2 + M^3 + M^4 + M^5)\bar{d} \\
\bar{p} &\approx \begin{bmatrix} 1.1112 & 0.0045 & 0.0556 & 0.0001 \\ 0.0120 & 1.0017 & 0.0206 & 0.0256 \\ 0.0010 & 0.0801 & 1.0017 & 0.0020 \\ 0.0340 & 0.0001 & 0.0017 & 1.0204 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix} \\
\bar{p} &\approx \begin{bmatrix} 12.8741 \\ 21.7938 \\ 31.7434 \\ 41.2103 \end{bmatrix}
\end{aligned}$$

2.3.4 Meaning of the Infinite Sum

What this infinite sum means is that in our original solution:

$$\bar{p} = (I - M)^{-1}\bar{d}$$

We could have rewritten this as:

$$\bar{p} = (I + M + M^2 + M^3 + ...)\bar{d}$$

Why is this interesting? Consider the original question from another perspective. We wish to produce external demand $\bar{d}$. To do so would suggest perhaps $\bar{p} = \bar{d}$. However this does not take into account the fact that we need to produce not just $\bar{d}$ but also enough to feed the internal demand, so perhaps $\bar{p} = \bar{d} + M\bar{d}$. But we also need to feed the internal demand to feed that internal demand, this is $M(M\bar{d}) = M^2\bar{d}$, and so on, so really:

$$\bar{p} = \bar{d} + M\bar{d} + M^2\bar{d} + M^3\bar{d} + ...$$

Which is exactly the same as the equation above.

2.4 Matlab

To enter a matrix in Matlab we can use semicolons to separate rows and either spaces or commas to separate columns. Alternately we can use newlines to separate rows.

The following all assign the same matrix:

```
>> A = [1 0 -2;-2 5 0];

>> A = [1,0,-2;-2,5,0];

>> A = [
1 0 -2
-2 5 0];
```

The inverse of a matrix can be found as follows:

```
>> A = [1 0 -2;-2 5 0;1 2 3];
>> inv(A)
ans =
    0.4545   -0.1212    0.3030
    0.1818    0.1515    0.1212
   -0.2727   -0.0606    0.1515
```

The identity matrix doesn't need to be typed in all the way:

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
```

We can then do something like this:

```
>> M = [1 0 -2;-2 5 0;1 2 3];
>> inv(eye(3)-M)
ans =
   -0.5000    0.2500   -0.5000
   -0.2500   -0.1250   -0.2500
    0.5000         0         0
```

We can then do a calculation such as this from an earlier example:

```
>> M = [
0.10 0.15 0.12
0.20 0 0.30
0.25 0.40 0.20];
>> d = [100;200;300];
>> inv(eye(3)-M)*d
ans =
  281.2995
  464.8608
  695.3365
```

We can test long sums of powers of matrices easily, like this earlier example:

```
>> M = [
0.1 0 0.05 0
0.01 0 0.02 0.025
0 0.08 0 0
0.03 0 0 0.02];
>> eye(4)+M+M^2+M^3+M^4
ans =
    1.1112    0.0044    0.0556    0.0001
    0.0120    1.0016    0.0206    0.0256
    0.0010    0.0801    1.0016    0.0020
    0.0340    0.0001    0.0017    1.0204
```

Eigenvalues and eigenvectors can be found together but we have to know how to interpret the result. In the following the matrix `d` contains one eigenvalue per entry on the diagonal and the matrix `p` contains the eigenvectors where the first column contains an eigenvector corresponding to the first entry in `d`, and so on:

```
>> M = [
0.1 0.4 0
0.2 0.4 0.9
0.7 0.2 0.1];
>> [p,d] = eig(M)
p =
  -0.3605 + 0.0000i  -0.2835 - 0.4119i  -0.2835 + 0.4119i
  -0.8111 + 0.0000i   0.6614 + 0.0000i   0.6614 + 0.0000i
  -0.4606 + 0.0000i  -0.3780 + 0.4119i  -0.3780 - 0.4119i
d =
   1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -0.2000 + 0.4359i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.2000 - 0.4359i
```

The determinant is easy:

```
>> A = [1 0 -2;-2 5 0;1 2 3];
>> det(A)
ans =
    33
```

2.5 Exercises

Exercise 2.1. Which of the following consumption matrices could yield a nonzero production vector for a closed economy? Justify.

$$M_1 = \begin{bmatrix} 0.2 & 0.1 & 0.3 \\ 0.5 & 0.7 & 0.3 \\ 0.3 & 0.2 & 0.4 \end{bmatrix} \quad M_2 = \begin{bmatrix} 0.04 & 0.02 & 0.06 \\ 0.10 & 0.14 & 0.06 \\ 0.06 & 0.04 & 0.08 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.1 \\ 0.1 & 0.1 & 0.0 \end{bmatrix}$$

Exercise 2.2. The following consumption matrix can yield a nonzero production vector for a closed economy. Find two different interesting corresponding nonzero production vectors.

$$M = \begin{bmatrix} 0.3157 & 0.947 \\ 0.6314 & 0.1263 \end{bmatrix}$$

Note: Due to approximations in technology you might not get an eigenvalue of exactly 1 but the intention is that 1 is an eigenvalue.

Exercise 2.3. The following consumption matrix can yield a nonzero production vector for a closed economy. Find two different interesting corresponding nonzero production vectors.

$$M = \begin{bmatrix} 0.5956 & 0.2978 & 0.1787 \\ 0.2978 & 0 & 0.4169 \\ 0.8934 & 0.1191 & 0.05956 \end{bmatrix}$$

Note: Due to approximations in technology you might not get an eigenvalue of exactly 1 but the intention is that 1 is an eigenvalue.

Exercise 2.4. Suppose an economy has two sectors producing products Product 1 and Product 2 respectively.

- It takes 0.10 units of Product 1 and 0.05 units of Product 2 to produce 1 unit of Product 1.
- It takes 0.06 units of Product 1 and 0.12 units of Product 2 to produce 1 units of Product 2.

(a) What should the total production be set at in order to satisfy an external demand of 20 units of Product 1 and 30 units of Product 2?

(b) Find $(I - M)^{-1}$ by hand and interpret each entry in the first column.

Exercise 2.5. Suppose an economy has two sectors producing products Product 1 and Product 2 respectively.

- It takes 0.22 units of Product 1 and 0.15 units of Product 2 to produce 1 unit of Product 1.

- It takes 0.16 units of Product 1 and 0.26 units of Product 2 to produce 1 units of Product 2.

(a) What should the total production be set at in order to satisfy an external demand of 120 units of Product 1 and 150 units of Product 2?

(b) Would you say this economy is efficient or not and why?

(c) Suppose the sectors scale down their input requirements to 10% of what they were. Repeat (a) and (b).

(d) Find $(I-M)^{-1}$ by hand and interpret each entry in the first row.

Exercise 2.6. Suppose the consumption matrix for a certain economy is

$$M = \begin{bmatrix} 0.02 & 0 & 0.05 \\ 0.1 & 0.08 & 0 \\ 0 & 0.12 & 0.04 \end{bmatrix}$$

Using the adjugate method - if the demand for sector 2 changes by -2 how must the production of sector 3 respond?

Exercise 2.7. Suppose an economy has three sectors producing products Product 1, Product 2 and Product 3 respectively.

- It takes 0.20 units of Product 1, 0.15 units of Product 2, and 0.10 units of Product 3 to produce 1 unit of Product 1.
- It takes 0.10 units of Product 1, 0.05 units of Product 2, and 0.12 units of Product 3 to produce 1 unit of Product 2.
- It takes 0.14 units of Product 1 and 0.08 units of Product 2 to produce 1 unit of Product 3.

(a) What should the total production be set at in order to satisfy an external demand of 100 units of Product 1, 120 units of Product 2 and 150 units of Product 3?

(b) Using the adjugate method calculate how production in sector 1 must respond if the demand for sector 3 changes by $+1$. How about by $+2$? How about by -1?

(c) Find $(I-M)^{-1}$ using technology.

Exercise 2.8. Suppose an economy has three sectors producing products Product 1, Product 2 and Product 3 respectively.

- It takes 0.02 units of Product 1, 0.06 units of Product 2, and 0.10 units of Product 3 to produce 1 unit of Product 1.
- It takes 0.40 units of Product 2 and 0.04 units of Product 3 to produce 1 unit of Product 2.

- It takes 0.18 units of Product 1, 0.01 units of Product 2, and 0.10 units of Product 3 to produce 1 unit of Product 3.

(a) What should the total production be set at in order to satisfy an external demand of 200 units of Product 1, 180 units of Product 2 and 175 units of Product 3?

(b) Using the adjugate method calculate how production in sector 3 must change if the demand for sector 2 changes by +1. How about by +2? How about by −1?

(c) Find $(I - M)^{-1}$ using technology.

Exercise 2.9. Suppose the consumption matrix for two sectors is given by:

$$M = \begin{bmatrix} 0.10 & 0.06 \\ 0.05 & 0.12 \end{bmatrix}$$

(a) Find the smallest i so that $I+M+...+M^i$ is the same as $I+M+...+M^{i-1}$ when rounded to the fourth decimal digit.

(b) Write down the matrix $I + M + ... + M^i$ using your i from (a).

(c) What can you say about $(I - M)^{-1}$?

Exercise 2.10. Suppose the consumption matrix for four sectors is given by:

$$M = \begin{bmatrix} 0.02 & 0 & 0.02 & 0.04 \\ 0 & 0.01 & 0.05 & 0.04 \\ 0.03 & 0 & 0.01 & 0.02 \\ 0.02 & 0.05 & 0.01 & 0 \end{bmatrix}$$

Using the infinite series method find a quick approximation for how production in sector 1 must change if the demand for sector 1 changes by +1.
Note: I'll let you decide how far to take your powers of M, just explain why you made whatever choice you did.

Exercise 2.11. In any sensible economy for each i the (i, i)-entry in $(I - M)^{-1}$ is always greater than 1. Explain why this is, economically speaking.

Exercise 2.12. Suppose for some consumption matrix M you find that $I + M + M^2 + ...$ does not converge. Intuitively speaking what does this say about the production issues associated to this economy? If it helps, give an example and use it to clarify.

Exercise 2.13. Suppose M is a consumption matrix with the property that all entries are between 0 and 1 inclusive except for some i for which $m_{ii} > 1$. Use a mathematical argument to explain why $\lim_{i\to\infty} M^i$ will not converge and hence $I + M + M^2 + ...$ will not converge.

Exercise 2.14. Suppose the consumption matrix for an economy with two sectors is given by

$$M = \begin{bmatrix} 1.02 & 0.06 \\ 0.05 & 0.01 \end{bmatrix}$$

(a) From an economic standpoint why is this M unrealistic?
Hint: Consider what the value of m_{11} means.

(b) Find $(I - M)^{-1}$.

(c) The previous question implies that $I + M + M^2 + ...$ does not converge, and yet $(I - M)^{-1}$ exists. Why does this not contradict the equality given in the chapter?

Exercise 2.15. Suppose the consumption matrix for an economy with two sectors is given by

$$M = \begin{bmatrix} 0.06 & 1.02 \\ 0.05 & 0.10 \end{bmatrix}$$

(a) This matrix is very similar to the matrix in the previous question yet this one gives reasonable results whereas the previous question does not. Explain this difference in economic terms. In other words why is it economically reasonable that $m_{12} > 1$ but not that $m_{11} > 1$?

(b) Find $(I - M)^{-1}$.

(c) Find $\bar{p}$ which corresponds to $\bar{d} = \begin{bmatrix} 100 \\ 200 \end{bmatrix}$.

(d) Show with some calculation that it seems that $I + M + M^2 + ...$ does converge to $(I - M)^{-1}$ in this case.

Exercise 2.16. Suppose the consumption matrix for an economy with two sectors is given by the following where all of a, b, c, d are between 0 and 1 inclusive

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

(a) Find a formula for $(I - M)^{-1}$.

(b) Explain why this result is reasonable if and only if

$$\left(\frac{1-a}{b}\right)\left(\frac{1-d}{c}\right) > 1$$

That is, if the inequality is not satisfied what can you say about the values in the inverse and why are those values economically nonsensical?

(c) Give some examples of how this makes sense in terms of the economy and the sectors. This is a tricky question with an interesting and commonsense answer. Think about how a and b are related to one another, what this means in economic terms, and how this feeds into the inequality you find. Similarly for c and d.

Exercise 2.17. Suppose instead of being given the consumption matrix M for an economy we are given the matrix $(I - M)^{-1}$. Call this the *marginal production response matrix*. (I have no idea if it has another name!)

(a) How can you find M?

(b) Apply this method to find M if you're given the marginal production response matrix

$$\begin{bmatrix} 1.06 & 0.02 & 0.04 \\ 0.02 & 1.06 & 0.09 \\ 0.11 & 0.02 & 1.03 \end{bmatrix}$$

Exercise 2.18. Suppose an economy has two sectors producing products Product 1 and Product 2 respectively.

- It takes 0.10 units of Product 1 and 0.05 units of Product 2 to produce 1 unit of Product 1.
- It takes 0.06 units of Product 1 and 0.12 units of Product 2 to produce 1 units of Product 2.

Suppose the total production is fixed at 1000 units of Product 1 and 1500 units of Product 2 of which some (as much as is needed) is used internally and the rest (normally called the external demand) is stored as surplus. How much of each is there as surplus?

Exercise 2.19. Suppose an economy has two sectors producing products Product 1 and Product 2 respectively.

- It takes 0.10 units of Product 1 and 0.20 units of Product 2 to produce 1 unit of Product 1.
- It takes x units of Product 1 and 0.05 units of Product 2 to produce 1 units of Product 2.

(a) How large can x be for this to be economically sensible, meaning it can find a solution for any external demand? Justify.

(b) If the 0.20 is replaced by y, what relationship between x and y would be economically sensible?

Chapter 3

Computer Graphics

Contents

3.1 Introduction

3.1.1 Chapter Goal

In computer graphics the natural representation of a point is as a vector, meaning the natural way store the point (x, y) is as $[x; y]$ (in two dimensions) or (x, y, z) is as $[x; y; z]$.

The goal of this section is to try to figure out how work with points in 2D and 3D in such a way that standard movements (translations and rotations) as well as projections (used to turn 3D pictures into 2D representations) can all be managed using matrix multiplication.

More specifically if $\bar{p}$ is a point in two dimensions then we would like to represent a rotation (for example) as a matrix R such that the vector $R\bar{p}$ is the new point after it has been rotated.

The reason for this is twofold:

- Matrix multiplication is easy to calculate.
- If we have multiple points $\bar{p}_1, ..., \bar{p}_k$ then we can apply M to all of them simultaneously by simply putting $\bar{p}_1, ..., \bar{p}_k$ in a matrix because

$$M\left[\bar{p}_1 \ldots \bar{p}_k\right] = \left[M\bar{p}_1 \ldots M\bar{p}_k\right]$$

- If we have two operations and they are represented by the matrices A and B then the operation "A then B" can be represented easily by the matrix BA in the sense that the product $BA\bar{p} = B(A\bar{p})$ is the point which results from doing A then B to $\bar{p}$.

3.1.2 Brief Review on Linearity

Recall that a function f from $\mathbb{R}^n$ to $\mathbb{R}^m$ is linear if and only if it can be represented by matrix multiplication where the matrix equals $[f(\bar{e}_1)\ldots f(\bar{e}_n)]$.

Some consequences of this are:

- If we have a function and know it's linear then we can construct the matrix as instructed.
- If we have a function and claim it's linear one approach is to find $f(\bar{e}_i)$, construct the matrix and show that the matrix multiplication actually does what f does. The argument here is that f is linear if and only if the matrix would represents, and since it does, f is linear.
- If we have a function and claim it's not linear one approach is to find $f(\bar{e}_i)$, construct the matrix and show that the matrix multiplication does not do what f does. The argument here is that f is linear if and only if the matrix would represents, and since it doesn't, f isn't linear.
- If we have a function and don't know if it's linear we can't simply construct the matrix and say it is. It's necessary that the matrix multiplication does what f does to finish the job.

3.2 Translations in 2D and Lower

3.2.1 Translation Problem

The first thing we'd like to do is translate a point, meaning shift it horizontally and/or vertically. This means finding a matrix T such that $T\bar{p}$ translates the point.

However this will never work, the reason being that if $\bar{p} = \bar{0}$ then no matter what we choose for T we will get $T\bar{p} = T\bar{0} = \bar{0}$ meaning the origin is never going to be translated no matter what choice we have for T.

The problem is deeper than this, the problem is not just with the origin, the problem is that translations are not linear and matrix multiplications are, so it seems we're out of luck.

So what can we do?

3.2.2 Stepping back to 1D

To see how we can fix this it's helpful to step back to one dimension where the analogy would be that we want to somehow represent the shift $x \to x + 1$ by a matrix multiplication.

Instead of representing x simply as a single variable (a 1×1 vector $[x]$) if we represent it by $\begin{bmatrix} x \\ 1 \end{bmatrix}$ then observe that for any a we have:

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} x + a \\ 1 \end{bmatrix}$$

What is going on here?

The matrix

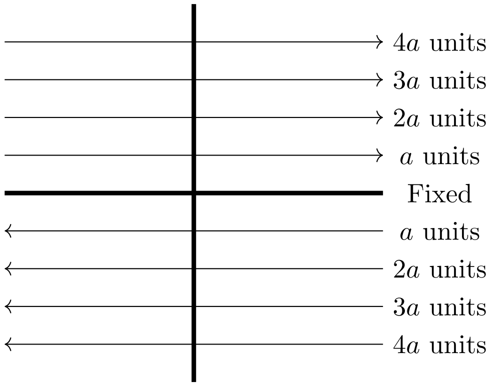
$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

is a *shear transformation.* If we look at what it does to any point:

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ay \\ y \end{bmatrix}$$

we see that each horizontal line is preserved as whole but moved to the left or right proportional to the particular y value. So for example the line $y = 1$ is moved to the right by a units, the line $y = 2$ is moved to the right by $2a$ units, the line $y = -3$ is moved to the right by $-3a$ units, and so on.

Visually:



What this means then is that a point x in one dimension can be represented by a vector $\begin{bmatrix} x \\ 1 \end{bmatrix}$ and translation by a units can be represented by the vector

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

Example 3.1. Translation by 3 units is represented by the matrix

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$$

and then to shift the point $x = 7$ we do:

$$\begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$$

and see that it's shifted to $x = 10$.

3.2.3 Back to 2D and Building a Matrix

So then for our 2D point what we'll do is represent our point by the vector:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

For any a, b we claim that there is a linear transformation f with:

$$f\left(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\right) = \begin{bmatrix} x+a \\ y+b \\ 1 \end{bmatrix}$$

If such an linear transformation exists then it can be represented by a matrix M and that matrix is dictated by what it does to the standard basis. More specifically if such an M exists then it would equal:

$$\left[f\left(\begin{bmatrix}1\\0\\0\end{bmatrix}\right) \quad f\left(\begin{bmatrix}0\\1\\0\end{bmatrix}\right) \quad f\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\right) \right]$$

In order to find these, note that the translation would shift $(1, 0)$ to $(1 + a, b)$, $(0, 1)$ to $(a, 1 + b)$ and $(0, 0)$ to (a, b). In other words:

$$\begin{aligned} f\left(\begin{bmatrix}1\\0\\1\end{bmatrix}\right) &= \begin{bmatrix}1+a\\b\\1\end{bmatrix} \\ f\left(\begin{bmatrix}0\\1\\1\end{bmatrix}\right) &= \begin{bmatrix}a\\1+b\\1\end{bmatrix} \\ f\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\right) &= \begin{bmatrix}a\\b\\1\end{bmatrix} \end{aligned}$$

It follows then that by linearity we must have:

$$\begin{aligned} f\left(\begin{bmatrix}1\\0\\0\end{bmatrix}\right) &= f\left(\begin{bmatrix}1\\0\\1\end{bmatrix} - \begin{bmatrix}0\\0\\1\end{bmatrix}\right) \\ &= f\left(\begin{bmatrix}1\\0\\1\end{bmatrix}\right) - f\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\right) \\ &= \begin{bmatrix}1+a\\b\\1\end{bmatrix} - \begin{bmatrix}a\\b\\1\end{bmatrix} \\ &= \begin{bmatrix}1\\0\\0\end{bmatrix} \end{aligned}$$

and

$$\begin{aligned}
f\left(\begin{bmatrix}0\\1\\0\end{bmatrix}\right) &= f\left(\begin{bmatrix}0\\1\\1\end{bmatrix} - f\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\\
&= f\left(\begin{bmatrix}0\\1\\1\end{bmatrix}\right) - f\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\\
&= \begin{bmatrix}a\\1+b\\1\end{bmatrix} - \begin{bmatrix}a\\b\\1\end{bmatrix}\\
&= \begin{bmatrix}0\\1\\0\end{bmatrix}
\end{aligned}$$

Therefore if such a linear transformation f exists it must be represented by the matrix

$$M = \left[f\left(\begin{bmatrix}1\\0\\0\end{bmatrix}\right) \quad f\left(\begin{bmatrix}0\\1\\0\end{bmatrix}\right) \quad f\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\right] = \begin{bmatrix}1&0&a\\0&1&b\\0&0&1\end{bmatrix}$$

So we check that this M does in fact work:

$$M\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}1&0&a\\0&1&b\\0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix} = \begin{bmatrix}x+a\\y+b\\1\end{bmatrix}$$

Therefore we have a matrix that does the desired job and so in general for any a, b we write:

$$T(a,b) = \left[M\begin{bmatrix}1\\0\\0\end{bmatrix} \quad M\begin{bmatrix}0\\1\\0\end{bmatrix} \quad M\begin{bmatrix}0\\0\\1\end{bmatrix}\right] = \begin{bmatrix}1&0&a\\0&1&b\\0&0&1\end{bmatrix}$$

Example 3.2. Translation by +2 units in the x direction and -5 units in the y direction is represented by the matrix

$$T(2,-5) = \begin{bmatrix}1&0&2\\0&1&-5\\0&0&1\end{bmatrix}$$

So then if we translate the points $(7,1)$ and $(-3,0)$ we would represent the points by vectors $[7;1;1]$ and $[-3;0,1]$ and then:

$$T(2,-5)\begin{bmatrix} 7 & -3 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 7 & -3 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 9 & -1 \\ -4 & -5 \\ 1 & 1 \end{bmatrix}$$

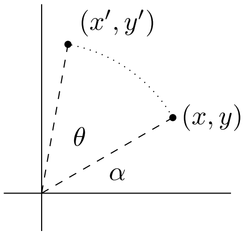
resulting in the points $(9,-4)$ and $(-1,-5)$.

3.3 Rotations in 2D

First we'll deal with rotations around the origin (which is easier) and then we'll see how we can rotate around any point.

Unless otherwise specified all rotations in 2D are counterclockwise.

To rotate around the origin by θ radians consider that we want to take the point (x,y) to (x',y') as shown:



First note that:

$$\begin{aligned} x &= \sqrt{x^2+y^2}\cos\alpha \\ y &= \sqrt{x^2+y^2}\sin\alpha \end{aligned}$$

It then follows that:

$$\begin{aligned} x' &= \sqrt{x^2+y^2}\cos(\alpha+\theta) \\ &= \sqrt{x^2+y^2}\,[\cos\alpha\cos\theta - \sin\alpha\sin\theta] \\ &= x\cos\theta - y\sin\theta \end{aligned}$$

And that:

$$
\begin{aligned}
y' &= \sqrt{x^2+y^2}\sin(\alpha+\theta) \\
&= \sqrt{x^2+y^2}\,[\sin\alpha\cos\theta+\sin\theta\cos\alpha] \\
&= y\cos\theta + x\sin\theta \\
&= x\sin\theta + y\cos\theta
\end{aligned}
$$

With our extra 1, we claim that there is a linear tranformation f with:

$$
f\left(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\right) = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{bmatrix}
$$

If such a linear transformation exists then it can be represented by a matrix R and that matrix is dictated by what it does to the standard basis. More specifically if such an R exists then it would equal:

$$
\left[f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) \quad f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right) \quad f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \right]
$$

From above, our rotation would rotate $(1,0)$ to $(\cos\theta,\sin\theta)$, $(0,1)$ to $(-\sin\theta,\cos\theta)$, and $(0,0)$ to $(0,0)$. In other words:

$$
\begin{aligned}
f\left(\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\right) &= \begin{bmatrix} \cos\theta \\ \sin\theta \\ 1 \end{bmatrix} \\
f\left(\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}\right) &= \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 1 \end{bmatrix} \\
f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

It follows then by linearity we must have:

$$
\begin{aligned}
f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) &= f\left(\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \\
&= f\left(\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}\right) - f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \\
&= \begin{bmatrix} \cos\theta \\ \sin\theta \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}
\end{aligned}
$$

and

$$
\begin{aligned}
f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right) &= f\left(\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - f\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \\
&= f\left(\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}\right) - f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \\
&= \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}
\end{aligned}
$$

Therefore if such a linear transformation f exists it must be represented by the matrix:

$$
R = \begin{bmatrix} f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right) & f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right) & f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

So we check that this R does in fact work:

$$
R\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{bmatrix}
$$

Therefore we have a matrix that does the desired job and so in general for any θ we write:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example 3.3. Rotation around the origin by $\pi/6$ radians is given by the matrix

$$R(\pi/6) = \begin{bmatrix} \cos(\pi/6) & -\sin(\pi/6) & 0 \\ \sin(\pi/6) & \cos(\pi/6) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To rotate the point $(5, 3)$ we do:

$$\begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} (5\sqrt{3} - 3)/2 \\ (5 + 3\sqrt{3})/2 \\ 1 \end{bmatrix}$$

to get the point $\big((5\sqrt{3} - 3)/2, (5 + 3\sqrt{3})/2\big)$.

3.4 Combining Translations and Rotations

Now that we have matrices for translations and matrices for rotations we can combine these to get matrices for other transformations.

Example 3.4. Suppose we wish to first translate by -2 units in the x-direction and by 7 units in the y-direction and then rotate by $\pi/2$ radians.

The translation is:

$$T(-2, 7) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix}$$

while the rotation is:

$$R(\pi/2) = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) & 0 \\ \sin(\pi/2) & \cos(\pi/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore the matrix which does the translation and then the rotation will be the following, note that the translation, which happens first, is on the right:

$$R(\pi/2)T(-2,7) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & -7 \\ 1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

Notice that the translation, which happens first, goes on the right, because when we apply the combination to a point such as $(5,3)$ we would do:

$$\begin{aligned} R(\pi/2)T(-2,7)\begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & -7 \\ 1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -10 \\ 3 \\ 1 \end{bmatrix} \end{aligned}$$

so that the matrix multiplication for the translation happens before the matrix multiplication for the rotation.

Example 3.5. The classic example is rotation around a point other than the origin. Suppose we wish to rotate around the point $(4,7)$ by $\pi/6$ radians What we'll do is first shift the plane 4 units left and 7 units down, thereby placing our desired rotation point at the origin, rotate by $\pi/6$, and then shift back.

So we wish to do $T(-4,-7)$ then $R(\pi/6)$ then $T(4,7)$. In other words, the following; note the order because of which one we wish to do first:

$$T(4,7)R(\pi/6)T(-4,-7)$$

which is

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi/6) & -\sin(\pi/6) & 0 \\ \sin(\pi/6) & \cos(\pi/6) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -7 \\ 0 & 0 & 1 \end{bmatrix}$$

or

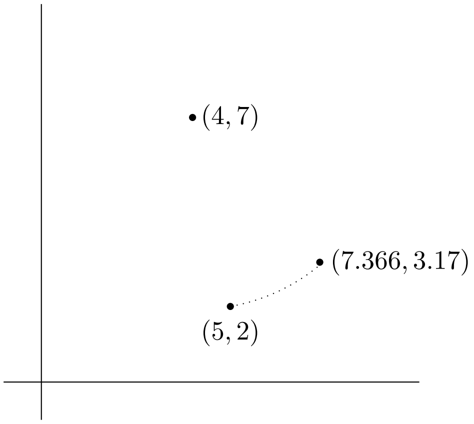
$$\begin{bmatrix} \sqrt{3}/2 & -1/2 & 15/2 - 2\sqrt{3} \\ 1/2 & \sqrt{3}/2 & 5 - 7\sqrt{3}/2 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 0.866 & -0.500 & 4.036 \\ 0.500 & 0.866 & -1.062 \\ 0 & 0 & 1.000 \end{bmatrix}$$

To summarize this new matrix rotates the plane around the point $(4, 7)$ by $\pi/6$ radians.

Then for example if we rotate the point $(5, 2)$ we get:

$$\begin{bmatrix} \sqrt{3}/2 & -1/2 & 15/2 - 2\sqrt{3} \\ 1/2 & \sqrt{3}/2 & 5 - 7\sqrt{3}/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix} \approx \begin{bmatrix} 0.866 & -0.500 & 4.036 \\ 0.500 & 0.866 & -1.062 \\ 0 & 0 & 1.000 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7.366 \\ 3.17 \\ 1 \end{bmatrix}$$

We can see that this works:



3.5 Moving to 3D

We'll avoid giving too much detail here since it should be fairly clear at this point where all of this comes from.

In analogy to two dimensions a point in 3D will be represented by a vector with an additional entry:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

A translation by a, b, c units in the x, y, z directions respectively will then be given by the matrix

$$T(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation is a tricky business because we now have to rotate around axes and an axis could be any line. However it's fairly easy to write down rotations around the x, y and z axes in the direction dictated by the right-hand rule in the sense that following the rotation with the fingers of the right hand points the thumb in the positive axis direction.

In general when we talk about rotation around an arbitrary axis we'll make sure that the axis has direction and that the rotation obeys the right-hand rule with regards to this direction.

Rotation around the z-axis is easiest because it comes from the two-dimensional case where we're moving x and y but not z. The following matrix leaves the z value alone and rotates the positive x-axis toward the positive y-axis thereby obeying our right-hand-rule wishes:

$$RZ(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If we simply swap some positions we get rotation around the x-axis. The following matrix leaves the x value alone and rotates the positive y-axis toward the positive z-axis thereby obeying our right-hand-rule wishes:

$$RX(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

When we rotate around the y-axis we need to be careful. Obeying the right-hand-rule wishes for the y-axis insists that the positive z-axis should rotate toward the positive x-axis.

This means when we adapt $RZ(\theta)$ we need to interchange the x and z positions, yielding:

$$RY(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 3.6. To rotate around the y-axis by $7\pi/6$ radians we use the matrix:

$$RY(7\pi/6) = \begin{bmatrix} \cos(7\pi/6) & 0 & \sin(7\pi/6) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(7\pi/6) & 0 & \cos(7\pi/6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\sqrt{3}/2 & 0 & -1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 0 & -\sqrt{3}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

At this point just like in 2D we can combine translations and rotations by simply multiplying matrices.

To rotate about an axis which is parallel to either the x, y or z axis we simply translate, rotate, then translate back, as in 2D.

Example 3.7. To rotate around the axis given by the line $x = 1$, $y = 4$ with upwards direction by $\pi/4$ radians we first shift $T(-1,-4,0)$, then do $RZ(\pi/4)$, then do $T(1,4,0)$:

$$\begin{aligned} &T(1,4,0)RZ(\pi/4)T(-1,-4,0) \\ &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 & 3\sqrt{2}/2+1 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 & 4-5\sqrt{2}/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Example 3.8. To rotate around the axis given by the line $y = 2$, $z = -1$ with orientation opposite to the x-axis by $3\pi/4$ radians we first shift $T(0,-2,1)$, then do $RX(-3\pi/4)$, then do $T(0,2,-1)$:

$$T(0,2,-1)RX(-3\pi/4)T(0,-2,1)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & -\sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sqrt{2}/2 & \sqrt{2}/2 & 3\sqrt{2}/2+2 \\ 0 & -\sqrt{2}/2 & -\sqrt{2}/2 & \sqrt{2}/2-1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To rotate about axes which are not parallel to one of the three main axes is trickier and is explored in the exercises.

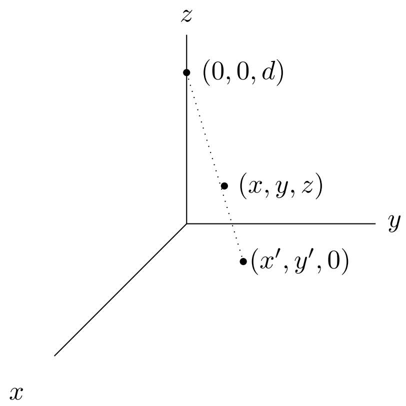
3.6 Perspective Projection in 3D

The last issue we'd like to address is the fact that computer graphics, while composed of 3D points, are rendered in 2D on your computer screen. This is done with a sense of perspective, meaning that things that are farther away look smaller than things that are closer.

Can we accomplish this with a matrix multiplication? The short answer is no, however there is an easy fix, and we'll see why and how as we build a simple example.

3.6.1 Perspective Projection from $z = d > 0$

For our example we'll assume that the *center of perspective* (in simple terms, the viewpoint) is positioned at a position $z = d > 0$ along the positive z-axis, that the object lies between the center of perspective and that we want to project our points to the xy-plane as illustrated by the following picture:

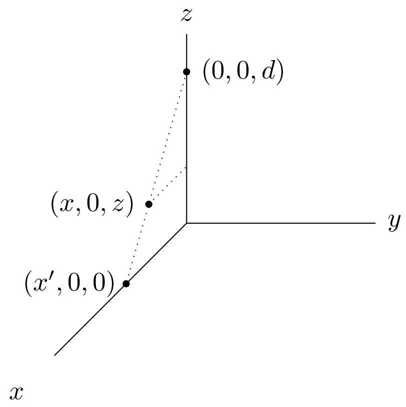


The goal is to find $(x', y', 0)$ in terms of (x, y, z).

Note: The idea here is that the object is between the center of perspective and the xy-plane but in reality it's only necessary that the object and the xy-plane be in a negative direction from the center of perspective. The final mathematics works out even if the object is on the other side of the xy-plane from the center of perspective, in effect the object just "reverse projects" back to the xy-plane.

In any case, back to our job of finding $(x', y', 0)$ in terms of (x, y, z).

To do so first consider the projection of the point $(x, 0, z)$:



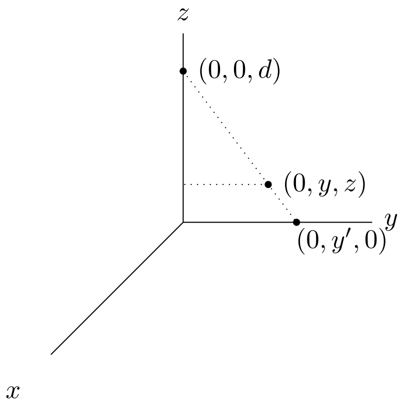
Similar triangles tells us that

$$\begin{aligned} \frac{x}{d-z} &= \frac{x'}{d} \\ x' &= \frac{dx}{d-z} \\ x' &= \frac{x}{1-z/d} \end{aligned}$$

Therefore we'd need

$$\begin{bmatrix} x \\ 0 \\ z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} \frac{x}{1-z/d} \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Next consider the projection of the point $(0, y, z)$:



Similar triangles tells us that

$$\begin{aligned} \frac{y}{d-z} &= \frac{y'}{d} \\ y' &= \frac{dy}{d-z} \\ y' &= \frac{y}{1-z/d} \end{aligned}$$

Therefore we'd need

$$\begin{bmatrix} x \\ 0 \\ z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ \frac{y}{1-z/d} \\ 0 \\ 1 \end{bmatrix}$$

In summary since x and y project independently of one another then we hope that the mapping f is linear, where:

$$f\left(\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\right) = \begin{bmatrix} \frac{x}{1-z/d} \\ \frac{y}{1-z/d} \\ 0 \\ 1 \end{bmatrix}$$

If such a linear transformation exists then it can be represented by a matrix M and that matrix is dictated by what it does to the standard basis. More specifically if such an M exists then:

$$M = \left[f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right) \quad f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right) \quad f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}\right) \quad f\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) \right]$$

In order to find these, note that f being linear gives us the following:

$$\begin{aligned}
f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right) &= f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) - f\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right) &= f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}\right) - f\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\
f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}\right) &= f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}\right) - f\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}$$

So that if such an M exists then:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

However this matrix does not do what f does, and so f is not linear.

So what are we to do? The answer is related to our fourth coordinate which has, up until this point, always been a 1.

Instead of trying to send:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} \frac{x}{1-z/d} \\ \frac{y}{1-z/d} \\ 0 \\ 1 \end{bmatrix}$$

Consider this matrix product:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ 1 - z/d \end{bmatrix}$$

The result of this product is a multiple of the desired result. If we divide the result by $1 - z/d$ we get:

$$\frac{1}{1 - z/d} \begin{bmatrix} x \\ y \\ 0 \\ 1 - z/d \end{bmatrix} = \begin{bmatrix} \frac{x}{1-z/d} \\ \frac{y}{1-z/d} \\ 0 \\ 1 \end{bmatrix}$$

as desired.

The solution therefore is to treat all points of the form:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

with $W \neq 0$ as equivalent to the point

$$\begin{bmatrix} X/W \\ Y/W \\ Z/W \\ 1 \end{bmatrix}$$

When we do this we say we are using *homogeneous coordinates*. These are heavily used in computer graphics applications as we are discovering.

In other words we do translations and rotations as discussed and when we need to project we use the projection matrix:

$$P(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix}$$

and then post-process the data by dividing by the final coordinate for each point.

Example 3.9. Consider the cube with vertices:

$$(-3,-3,-3),(3,-3,-3),(3,3,-3),(-3,3,-3),(-3,-3,3),(3,-3,3),(3,3,3),(-3,3,3)$$

If we treat these as vectors we can put them all together in a matrix:

$$A = \begin{bmatrix} -3 & 3 & 3 & -3 & -3 & 3 & 3 & -3 \\ -3 & -3 & 3 & 3 & -3 & -3 & 3 & 3 \\ -3 & -3 & -3 & -3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

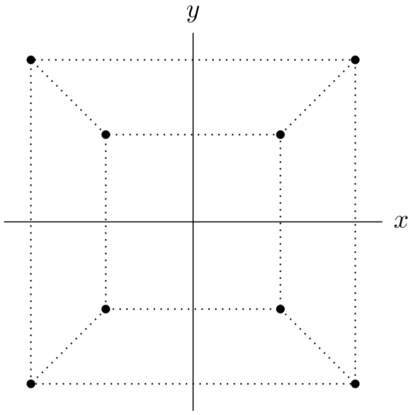
we can project them all at once. If $d = 10$ then we get:

$$\begin{aligned} P(10)A &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/10 & 1 \end{bmatrix} \begin{bmatrix} -3 & 3 & 3 & -3 & -3 & 3 & 3 & -3 \\ -3 & -3 & 3 & 3 & -3 & -3 & 3 & 3 \\ -3 & -3 & -3 & -3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -3.0 & 3.0 & 3.0 & -3.0 & -3.0 & 3.0 & 3.0 & -3.0 \\ -3.0 & -3.0 & 3.0 & 3.0 & -3.0 & -3.0 & 3.0 & 3.0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.3 & 1.3 & 1.3 & 1.3 & 0.7 & 0.7 & 0.7 & 0.7 \end{bmatrix} \end{aligned}$$

Each of these columns is then individually scaled so that the bottom entry is 1. Here is the result approximated:

$$\begin{bmatrix} -2.308 & 2.308 & 2.308 & -2.308 & -4.286 & 4.286 & 4.286 & -4.286 \\ -2.308 & -2.308 & 2.308 & 2.308 & -4.286 & -4.286 & 4.286 & 4.286 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \end{bmatrix}$$

We then extract the top three vectors as our new point. If we plot these in the xy plane we see the result. Here I've connected the corners by lines for added effect:



At this point we can do really fancy stuff.

Example 3.10. Suppose we wanted to take our cube from above, rotate it around the x-axis by $\pi/6$, the y-axis by $\pi/3$ and the z-axis by $\pi/12$, then shift it in the z-direction by -1, then project it with $d = 10$.

The five necessary matrices are:

$$RX(\pi/6) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{3}/2 & -1/2 & 0 \\ 0 & 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RY(\pi/3) = \begin{bmatrix} 1/2 & 0 & -\sqrt{3}/2 & 0 \\ 0 & 1 & 0 & 0 \\ \sqrt{3}/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RZ(\pi/12) = \begin{bmatrix} (\sqrt{2}+\sqrt{6})/4 & (\sqrt{2}-\sqrt{6})/4 & 0 & 0 \\ (\sqrt{6}-\sqrt{2})/4 & (\sqrt{2}+\sqrt{6})/4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T(0,0,-1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P(10) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/10 & 1 \end{bmatrix}$$

The result of doing each of these in the correct order is:

$$P(10)T(0,0,-1)RZ(\pi/12)RY(\pi/3)RX(\pi/6) \approx \begin{bmatrix} 0.4830 & 0.1941 & 0.8539 & 0 \\ 0.1294 & 0.9486 & -0.2888 & 0 \\ 0 & 0 & 0 & 0 \\ 0.0866 & -0.0250 & -0.0433 & 1.1000 \end{bmatrix}$$

And applied to the corners of the cube:

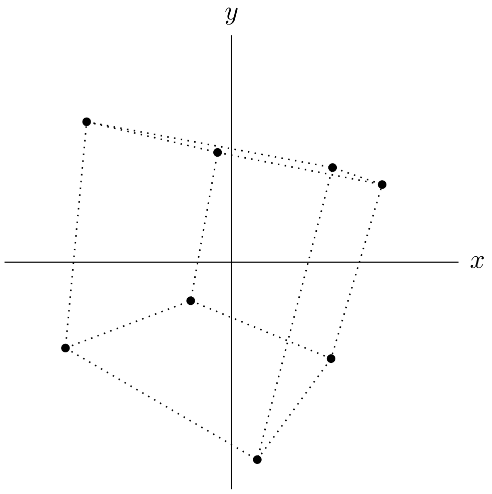
$$P(10)T(0,0,-1)RZ(\pi/12)RY(\pi/3)RX(\pi/6) \begin{bmatrix} -3 & 3 & 3 & -3 & -3 & 3 & 3 & -3 \\ -3 & -3 & 3 & 3 & -3 & -3 & 3 & 3 \\ -3 & -3 & -3 & -3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\approx \begin{bmatrix} -4.5928 & -1.6950 & -0.5303 & -3.4281 & 0.5303 & 3.4281 & 4.5928 & 1.6950 \\ -2.3674 & -1.5910 & 4.1005 & 3.3241 & -4.1005 & -3.3241 & 2.3674 & 1.5910 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0451 & 1.5647 & 1.4147 & 0.8951 & 0.7853 & 1.3049 & 1.1549 & 0.6353 \end{bmatrix}$$

and then scaled:

$$\begin{bmatrix} -4.3946 & -1.0833 & -0.3749 & -3.8299 & 0.6753 & 2.6271 & 3.9768 & 2.6681 \\ -2.2653 & -1.0168 & 2.8985 & 3.7137 & -5.2217 & -2.5474 & 2.0499 & 2.5044 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$$

Here's the picture:



3.6.2 Perspective Projection from Other Places

In order to get a perspective from another location we need to specify a few things, namely the plane on which we're projecting, a perpendicular axis on which the center of perspective lies, and how far away that center of perspective is. We also need to know which way is "up" for the view. That "up" direction will be given by a vector in the plane.

We then find the transformation which moves that plane to the xy-plane, puts the axis along the positive z-axis, and rotates around the positive z-axis so that the "up" direction points in the positive y-direction. We then find out how far the center of perspective is from the plane and use that for d.

Then use the resulting transformation to move the objects and lastly we do as in the previous section.

Notce that the result lies in the xy-plane and we don't need to move it all back, because getting it into the xy-plane is the end result for viewing.

This is calculation-heavy and sensitive and is explored in the exercises.

3.7 Matlab

Here are all the matrices in Matlab, along with necessary `syms`.

```
% 2D Stuff
syms TT(a,b) R(t)
TT(a,b)=[1 0 a;0 1 b;0 0 1];
R(t)=[cos(t) -sin(t) 0;sin(t) cos(t) 0;0 0 1];
% 3D Stuff
syms T(a,b,c) RX(t) RY(t) RZ(t) P(d)
T(a,b,c)=[1 0 0 a;0 1 0 b;0 0 1 c;0 0 0 1];
RX(t)=[1 0 0 0 ; 0 cos(t) -sin(t) 0;0 sin(t) cos(t) 0 ; 0 0 0 1];
RY(t)=[cos(t) 0 sin(t) 0;0 1 0 0;-sin(t) 0 cos(t) 0 ;0 0 0 1];
RZ(t)=[cos(t) -sin(t) 0 0;sin(t) cos(t) 0 0 ; 0 0 1 0; 0 0 0 1];
P(d)=[1 0 0 0;0 1 0 0;0 0 0 0;0 0 -1/d 1];
```

We enter a set of points as the transpose of a vector only because it makes it easier to enter the points one by one:

```
>> P=transpose([1,2,1;-3,0,1])
P =
     1    -3
     2     0
     1     1
```

It's then easy to apply transformations to points:

```
>> P=transpose([1,2,1;-3,0,1]);
>> NEWP=TT(4,5)*R(pi/6)*P
NEWP =
[  3^(1/2)/2 + 3, 4 - (3*3^(1/2))/2]
[ 3^(1/2) + 11/2,               7/2]
[              1,                 1]
```

If you would prefer approximations because they're easier to read, simply wrap the result in `vpa` which stands for *variable precision arithmetic*, and give it the number of digits as a second argument:

```
>> P=transpose([1,2,1;-3,0,1]);
>> NEWP=vpa(TT(4,5)*R(pi/6)*P,4)
NEWP =
[ 3.866, 1.402]
[ 7.232,   3.5]
[   1.0,   1.0]
```

We can plot these in 2D (plot not shown here) using the following, which takes out the x and y-coordinates and plots them against one another:

```
>> scatter(NEWP(1,:),NEWP(2,:))
```

In 3D, when we need to post-process the final matrix so that the fourth row is all 1s, we can do this easily. Here is the full process where we define two points, apply some rotations and a projection, and then post-process and plot. In addition the final two command set the x and y-axes to be proportionally sized and set the ranges of those axes.

```
>> PTS=transpose([1 2 3 0;0 -1 3 2]);
>> NEWPTS=P(10)*RY(pi/4)*RZ(pi/4)*PTS;
>> for i=1:size(NEWPTS,2);
NEWPTS(:,i)=NEWPTS(:,i)/NEWPTS(4,i);
end;
>> scatter(NEWPTS(1,:),NEWPTS(2,:),'filled')
>> axis square
>> axis([-5 5 -5 5])
```

3.8 Exercises

Exercise 3.1. In two dimensions write down the translation matrix which shifts $+8$ in the x-direction and -3 in the y-direction. Apply this matrix to the points $(1, 2)$ and $(0, -10)$.

Exercise 3.2. In three dimensions write down the translation matrix which shifts $+1$ in the x-direction, 2 in the y-direction and -7 in the y-direction. Apply this matrix to the points $(1, 2, 0)$ and $(1, 4, -3)$.

Exercise 3.3. In two dimensions write down the rotation matrix which rotates around the origin by $7\pi/6$ radians. Apply this matrix to the points $(1, 2)$ and $(0, -10)$.

Exercise 3.4. In two dimensions write down the rotation matrix which rotates around the origin by $2\pi/3$ radians clockwise. Apply this matrix to the points $(0, 3)$ and $(1, -1)$.

Exercise 3.5. In two dimensions write down the matrix which rotates around the origin by $\pi/6$ radians and then translates by -3 in the x-direction and 5 in the y-direction. Apply this matrix to the points $(0, 3)$ and $(1, -1)$.

Exercise 3.6. In two dimensions write down the matrix which translates by -3 in the x-direction and 5 in the y-direction and then rotates around the origin by $\pi/6$ radians. Apply this matrix to the points $(0, 3)$ and $(1, -1)$.

Exercise 3.7. In two dimensions find the rotation matrix which will rotate around the point $(-3, 2)$ by an angle of $\pi/4$ radians. Apply this matrix to the points $(4, 5)$ and $(0, 0)$.

Exercise 3.8. In two dimensions find the rotation matrix which will rotate around the point $(6, -3)$ by an angle of $7\pi/6$ radians. Apply this matrix to the points $(-2, 1)$ and $(0, 0)$.

Exercise 3.9. In two dimensions find the image of the three points

$$(6, 3), (4, 1), (7, 1)$$

under rotation around the point $(8, 2)$ by $\pi/3$ radians. Sketch the original points and the images.

Exercise 3.10. In two dimensions find the image of the three points

$$(6, 3), (4, 1), (7, 1)$$

under rotation around the point $(0, 7)$ by $\pi/4$ radians. Sketch the original points and the images.

Exercise 3.11. In two dimensions write down the matrix (simplified) for rotation around the point (a, b) by θ radians.

Exercise 3.12. In two dimensions find the matrix transformation composed of one translation followed by one rotation which moves the line segment joining $(4, 3)$ to $(7, 2)$ so that the point $(4, 3)$ moves to the origin and the segment lies along the positive x-axis. Then find the image of the origin under this transformation.

Exercise 3.13. In two dimensions find the matrix transformation composed of one translation followed by one rotation which moves the line segment joining $(4, 3)$ to $(7, 2)$ so that the point $(7, 2)$ moves to the origin and the segment lies along the positive y-axis. Then find the image of the point $(-2, 1)$ under this transformation.

Exercise 3.14. Reflection through a line through the origin followed by reflection through another line through the origin results in a rotation around the origin. Show this as follows:

(a) Write down the matrix which reflects in the x-axis.

(b) Use this to find the matrix which reflects in the line which makes an angle of θ degrees with the positive x-axis.

(c) Compose two of these, simplify, and explain why the result is a rotation around the origin.

Exercise 3.15. In three dimensions find the image of the three points

$$(1, 2, 3), (-2, 3, 1), (3, 2, 2)$$

under rotation around the y-axis by $\pi/4$.

Exercise 3.16. In three dimensions find the image of the three points

$$(0, 3, 1), (-6, 3, 1), (3, 1, 10)$$

under shifting by $(+2, -3, -6)$ followed by rotation around the z-axis by $7\pi/6$.

Exercise 3.17. In three dimensions find the image of the three points

$$(1, 2, 3), (-2, 3, 1), (3, 2, 2)$$

under the perspective projection with center of perspective at $z = 10$.

Exercise 3.18. In three dimensions find the image of the three points

$$(4,3,1),(-3,4,2),(5,-1,6)$$

under the perspective projection with center of perspective at $z=20$.

Exercise 3.19. In three dimensions we can rotate around an axis parallel to the x-axis by translating the desired axis so that it's on top of the x-axis, rotating, and then translating back. Using this method find the rotation matrix which will rotate around the line $y=2$, $z=5$ with direction identical to the x-axis by $3\pi/4$ radians.

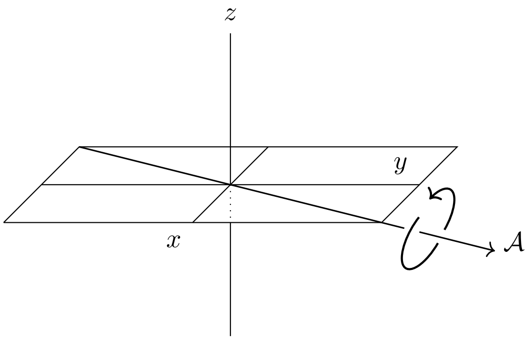
Exercise 3.20. In three dimensions we can rotate around an axis parallel to the y-axis by translating the desired axis so that it's on top of the y-axis, rotating, and then translating back. Using this method find the rotation matrix which will rotate around the line $x=-2$, $z=4$ with direction opposite to the y-axis by $2\pi/3$ radians.

Exercise 3.21. In three dimensions we can rotate around an axis parallel to the z-axis by translating the desired axis so that it's on top of the z-axis, rotating, and then translating back. Using this method find the rotation matrix which will rotate around the line $x=1$, $y=-5$ with direction identical to the z-axis by $4\pi/3$ radians.

Exercise 3.22. Consider the axis $\mathcal{A}$ lying along the vector

$$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

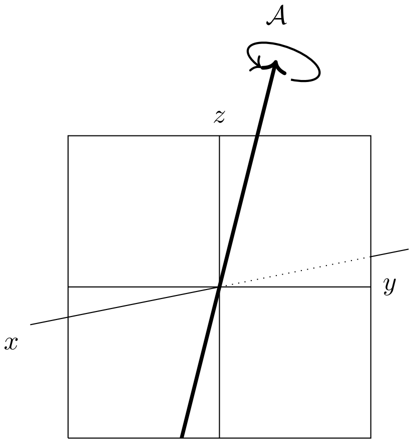
By rotating around the z-axis by an appropriate amount this axis can be placed on top of the x-axis. Use this fact to find the rotation matrix which will rotate by an angle of $\pi/6$ around $\mathcal{A}$.



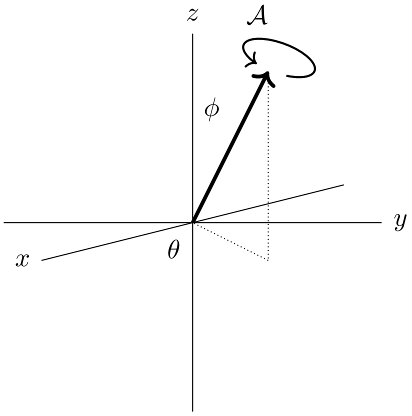
Exercise 3.23. Consider the axis $\mathcal{A}$ lying along the vector

$$\begin{bmatrix} 0 \\ 1 \\ \sqrt{3} \end{bmatrix}$$

By rotating around the x-axis by an appropriate amount this axis can be placed on top of the z-axis. Use this fact to find the rotation matrix which will rotate by an angle of $\pi/3$ around $\mathcal{A}$.



Exercise 3.24. If a desired axis of rotation $\mathcal{A}$ is given in terms of the spherical coordinate angles ϕ and θ:



Then rotation around $\mathcal{A}$ by α radians can be easily obtained by first rotating around the z-axis by $-\theta$ radians (at this point the desired axis is in the xz-plane), then rotating around the y-axis by $-\phi$ radians (at this point the desired axis is on top of the positive z-axis). We then rotate around the z-axis by α radians. Finally we undo the first two rotations.

Calculate (and simplify) the matrix for the rotation if $\phi = \pi/6$, $\theta = \pi/3$ and $\alpha = 3\pi/4$.

Exercise 3.25. Projection onto one of the other two coordinate planes (the xz-plane and the yz-plane) can easily be accomplished by rotation. For example if we wish to project onto the xz-plane what we do is first rotate the y-axis to the z-axis (which is a rotation around the x-axis), then project, then rotate back.

(a) Write down the projection matrix which does this.

(b) Use this to project the three points

$$(1, 2, 3), (4, -1, 0), (5, 2, 3)$$

with center of perspective at $y = 10$.

Exercise 3.26. Projection onto one of the other two coordinate planes (the xz-plane and the yz-plane) can easily be accomplished by rotation. For example if we wish to project onto the yz-plane what we do is first rotate the x-axis to the z-axis (which is a rotation around the y-axis), then project, then rotate back.

(a) Write down the projection matrix which does this.

(b) Use this to project the three points

$$(1, 2, 3), (4, -1, 0), (5, 2, 3)$$

with center of perspective at $x = 10$.

Exercise 3.27. Design/construct a three-dimensional object using points. Store those points in a $4 \times k$ matrix and then apply a series of interesting transformations. Find the resulting points and plot. In your submission plot the points before the transformations, give the series of transformations both by description and by matrix and give the overall resulting matrix. Finally plot the points after the transformations.

Exercise 3.28. As the center of perspective moves along the z-axis to infinity the projection matrix becomes $P_\infty = \lim_{d\to\infty} \bar{P}_d$.

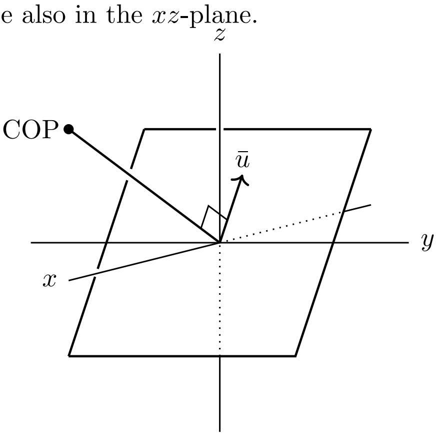
(a) Calculate this matrix.

(b) Find

$$P_\infty \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

and explain why it makes sense from a geometric viewpoint. Pictures will probably help.

Exercise 3.29. Suppose we wish our view to work as in the following picture. In this picture assume the COP is at $(5, 0, 3)$, the plane which is visible (and which is perpendicular to the line joining the COP to the origin) is the plane we wish to act as the xy-plane, the vector $\bar{u}$ is the vector which indicates which way is up, meaning which needs to act as the y-axis. This vector lies in the plane and in this case also in the xz-plane.



Our goal is to construct the projection matrix which projects onto this plane as if it were the xy-plane. Do this using the following steps:

(a) Write down the matrix which rotates around the y-axis an appropriate amount so that the COP lies on the positive z-axis and the plane lies on the xy-plane.

(b) Assuming this has been done, write down the matrix which rotates around the z-axis an appropriate amount so that $\bar{u}$ lies on the positive y-axis.

(c) Multiply these two in the correct order followed by the appropriate $P(d)$ to finish the job.

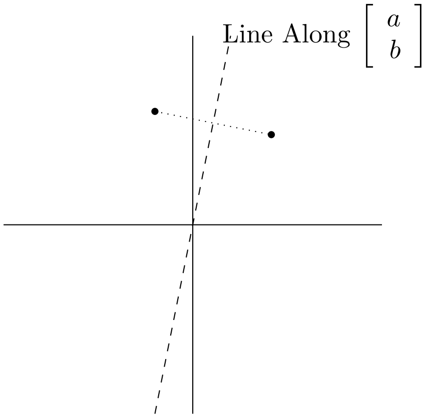
(d) Apply the matrix to the Matlab box, post-process and plot. Print and attach.

Exercise 3.30. Use the approach of the proceeding problem to find the projection matrix corresponding to the viewpoint where the COP is at $(10, 10, 10)$ looking directly toward the origin. The projection plane contains the origin and the "up" direction is given by the vector $[2, -1, -1]$.

Exercise 3.31. In two dimensions find the generic rotation matrix which will rotate by an angle of θ around the point (x_0, y_0).

Exercise 3.32. In three dimensions suppose we projected to the plane $z = z_0$ instead of the plane $z = 0$. To which (x, y) would the point (x_0, y_0, z_0) be sent? Justify.

Exercise 3.33. In two dimensions find the matrix which would reflect the xy-plane in the line through the origin with direction vector $\begin{bmatrix} a \\ b \end{bmatrix}$.



Exercise 3.34. Prove that if points in the xy-plane are treated simply as vectors

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

that the following are not linear:

(a) Rotation around a point which is not the origin.

(b) Translation.

Exercise 3.35. Prove that the matrices representing the following pairs of transformations are inverses of each other:

(a) $T(a, b)$ and $T(-a, -b)$.

(b) $R(\theta)$ and $R(-\theta)$.

Exercise 3.36. Prove that $RX(\pi)RY(\pi)RZ(\pi) = I$. What interesting inverse facts can you extract from this?

Exercise 3.37. Show (in 2D and in 3D) using the corresponding matrices that a translation followed by a translation equals a translation.

Exercise 3.38. Is it true or false that for all a, b, θ we have the following. Provide evidence.

$$R(\theta)T(a, b) = T(a, b)R(\theta)$$

Exercise 3.39. Write down the 3×3 matrix which reflects the plane through the origin, sending each point to its opposite. Then use this to find the matrix which reflects the plane through the point (x_0, y_0).

Exercise 3.40. It's possible to do a 2D version of projection, where projection is done with the COP at $y = d$ and projection is onto the x-axis. Develop this. Specifically, what would the projection matrix look like, how would it work, would post-processing be necessary and so on?

Chapter 4

Least Squares

Contents

4.1 Introduction

Let's go back to the matrix equation

$$A\bar{x} = \bar{b}$$

We know that a unique solution exists if A is invertible and if A is not invertible then there are either no solutions or infinitely many solutions.

Specifically the question that we'd like to address here is what can we do if there are no solutions at all? On answer might be to just stop, however maybe we could ask the question - what's the nearest solution we could find?

In other words if we can't find $\bar{x}$ so that $A\bar{x} = \bar{b}$, can we find some $\bar{x}$ so that $A\bar{x}$ is as close as possible to $\bar{b}$?

More rigorously can we find some specific $\hat{x}$ such that:

$$\text{For all } \bar{x} \text{ we have } ||A\hat{x} - \bar{b}|| \leq ||A\bar{x} - \bar{b}||$$

4.2 Reminder - Solutions and Column Space

First let's recall:

Definition 4.2.0.1. Given an $n \times m$ matrix A the *column space* of A is the subspace of $\mathbb{R}^n$ given by:

$$\mathrm{Col}(A) = \mathrm{span}\,\{\text{Columns of } A\}$$

Fact 4.2.0.1. The column space of A is exactly the vectors $\bar{b}$ such that $A\bar{x} = \bar{b}$ has at least one solution.

To reinforce this, a simple example will do:

Example 4.1. The equation

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \\ -1 & 0 \end{bmatrix} \bar{x} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

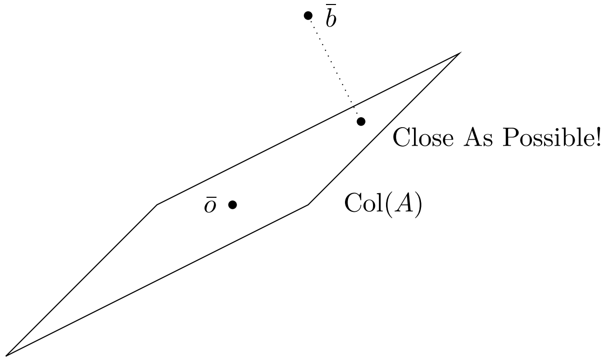
has a solution if and only if there are x_1, x_2 with

$$\begin{aligned}
\begin{bmatrix} 1 & 2 \\ 0 & 3 \\ -1 & 0 \end{bmatrix} \bar{x} &= \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\
\begin{bmatrix} 1 & 2 \\ 0 & 3 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\
\begin{bmatrix} 1x_1 + 2x_2 \\ 0x_1 + 3x_2 \\ -1x_1 + 0x_2 \end{bmatrix} &= \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\
x_1 \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} &= \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\
\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} &\in \mathrm{col} \left\{ \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} \right\}
\end{aligned}$$

4.3 The Intuition and Theory

So the situation we're in is that $\bar{b}$ is not in $\mathrm{Col}(A)$ and we wish to find $\hat{x}$ so that $A\hat{x}$, which is in $\mathrm{Col}(A)$, is as close as possible to $\bar{b}$.

Here's a picture of the situation:



This picture suggests that we can obtain a solution by projecting $\bar{b}$ onto Col(A) to get $\text{Pr}_{\text{Col}(A)}\bar{b}$ and then finding $\hat{x}$ to solve the equation:

$$A\hat{x} = \text{Pr}_{\text{Col}(A)}\bar{b}$$

Assuming this is correct, the problem with this approach, practically, is that calculating $\text{Pr}_{\text{Col}(A)}\bar{b}$ requires having an orthogonal basis for Col(A) and this is procedurally intense especially when A is large so what we'll do is find a sneaky way to find $\hat{x}$ a different way. Just to be clear, we will solve this equation, but we won't solve it by finding $\text{Pr}_{\text{Col}(A)}\bar{b}$.

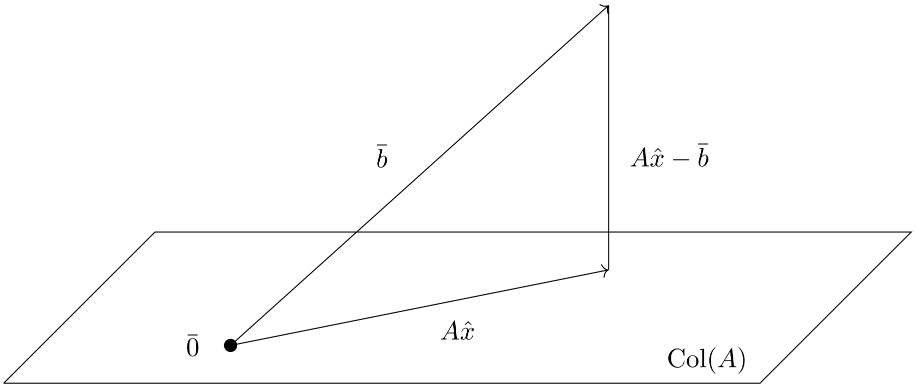
4.4 Theory: Least Squares Solution

The approach is based on the following two things:

(a) There is a sneaky way of finding $\hat{x}$ such that $A\hat{x} = \text{Pr}_{\text{Col}(A)}\bar{b}$.

(b) The $\hat{x}$ we find actually does satisfy $||A\hat{x} - \bar{b}|| \leq ||A\bar{x} - \bar{b}||$ for all $\bar{x}$.

First, let's see how we can do part (a). Finding $\hat{x}$ such that $A\hat{x} = \text{Pr}_{\text{Col}(A)}\bar{b}$ means the vector $A\hat{x} - \bar{b}$ is perpendicular to Col(A) as illustrated by this picture:



From here note that $A\hat{x}-\bar{b}$ being perpendicular to Col(A) is equivalent to

$$\begin{aligned} A\bar{x}\cdot(A\hat{x}-\bar{b}) &= 0 && \text{for all } \bar{x} \\ (A\bar{x})^T(A\hat{x}-\bar{b}) &= 0 && \text{for all } \bar{x} \\ \bar{x}^T A^T(A\hat{x}-\bar{b}) &= 0 && \text{for all } \bar{x} \\ A^T(A\hat{x}-\bar{b}) &= \bar{0} \\ A^T A\hat{x} &= A^T\bar{b} \end{aligned}$$

(Note that $\bar{x}^T\bar{y}=0$ for all $\bar{x}$ means $\bar{y}=\bar{0}$ because no nonzero vector can be perpendicular to every vector.)

So we simply solve this final equation instead. Notice that this final equation must have a solution since we're effectively solving $A\hat{x}=\text{Pr}_{\text{Col}(A)}\bar{b}$. It may have more than one, though, and it will have more than one precisely when the columns of A are linearly independent (because this always the case for systems of equations that have solutions) which is precisely when A^TA is invertible (because we're solving $A^TA\hat{x}=A^T\bar{b}$ to get the job done.)

Second, how about part (b). We want to make sure that finding $\hat{x}$ such that $A\hat{x}=\text{Pr}_{\text{Col}(A)}\bar{b}$ actually satisfies $||A\hat{x}-\bar{b}||\le||A\bar{x}-\bar{b}||$ for all $\bar{x}$.

Well as we've seen this $\hat{x}$ is such that $A\hat{x}-\bar{b}$ is perpendicular to Col(A).

For any $\bar{x}$, since $A\bar{x}$ and $A\hat{x}$ are both in Col(A), so is $A\bar{x}-A\hat{x}$, so then $A\bar{x}-A\hat{x}$ is perpendicular to $A\hat{x}-\bar{b}$.

From here observe that:

$$(A\hat{x}-\bar{b})+(A\bar{x}-A\hat{x})=A\bar{x}-\bar{b}$$

and since the two on the left are perpendicular by the Pythagorean Theorem we have:

$$||A\hat{x} - \bar{b}||^2 + ||A\bar{x} - A\hat{x}||^2 = ||A\bar{x} - \bar{b}||^2$$

and therefore

$$||A\hat{x} - \bar{b}|| \leq ||A\bar{x} - \bar{b}||$$

4.5 Practical: Least Squares Solution

Definition 4.5.0.1. If the columns of A are linearly independent then the *least squares solution* to

$$A\bar{x} = \bar{b}$$

is given by the solution $\hat{x}$ to

$$A^T A\hat{x} = A^T\bar{b}$$

This solution is exactly

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

If the columns of A are not linearly independent then there is no least squares solution.

Definition 4.5.0.2. The *least squares error* is the difference

$$||A\hat{x} - \bar{b}||$$

which measures how far our $A\hat{x}$ is from the desired $\bar{b}$.

Example 4.2. Consider the system of equations

$$\begin{aligned} x + 2y &= 6 \\ x + y &= 4 \\ x - y &= 1 \end{aligned}$$

I've chosen this so it's clear that there is no solution. The first two equations have solution $x = 2$, $y = 2$ but this fails in the third, so there is no solution to all three.

Rephrased as a matrix equation:

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \bar{x} = \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix}$$

We instead solve the least-squares equation:

$$
\begin{aligned}
\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}^T \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \hat{x} &= \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}^T \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \\
\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \hat{x} &= \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \\
\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \hat{x} &= \begin{bmatrix} 11 \\ 15 \end{bmatrix} \\
\hat{x} &= \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 11 \\ 15 \end{bmatrix} \\
\hat{x} &= \begin{bmatrix} 18/7 \\ 23/14 \end{bmatrix}
\end{aligned}
$$

The least-squares error is given by:

$$
\begin{aligned}
||A\hat{x} - \bar{b}|| &= \left|\left| \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 18/7 \\ 23/14 \end{bmatrix} - \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \right|\right| \\
&= \left|\left| \begin{bmatrix} -1/7 \\ 3/14 \\ -1/14 \end{bmatrix} \right|\right| \\
&= \frac{929}{3476} \\
&\approx 0.2673
\end{aligned}
$$

This is as small as $||A\bar{x} - \bar{b}||$ can be for any $\bar{x}$. You can test this to convince yourself by plugging in some other $\bar{x}$, maybe some close to $\hat{x}$ and some not. Note that $\hat{x} \approx \begin{bmatrix} 2.57143 \\ 1.64286 \end{bmatrix}$.

$$
\begin{aligned}
\left|\left| A \begin{bmatrix} 2.6 \\ 1.6 \end{bmatrix} - \bar{b} \right|\right| &\approx 0.28284 > 0.26726 \\
\left|\left| A \begin{bmatrix} 2.55 \\ 1.62 \end{bmatrix} - \bar{b} \right|\right| &\approx 0.27911 > 0.26726 \\
\left|\left| A \begin{bmatrix} 3 \\ 2 \end{bmatrix} - \bar{b} \right|\right| &\approx 1.41421 > 0.26726
\end{aligned}
$$

Example 4.3. Consider the system

$$\begin{aligned} x + 2y &= 3 \\ x + 2y &= 5 \end{aligned}$$

Clearly this has no solutions. As a matrix equation we have

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

Notice that the columns of A are not linearly independent. If we attempt the least-squares approach:

$$\begin{aligned} \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}^T \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \widehat{\begin{bmatrix} x \\ y \end{bmatrix}} &= \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}^T \begin{bmatrix} 3 \\ 5 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \widehat{\begin{bmatrix} x \\ y \end{bmatrix}} &= \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} \\ \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix} \widehat{\begin{bmatrix} x \\ y \end{bmatrix}} &= \begin{bmatrix} 8 \\ 16 \end{bmatrix} \end{aligned}$$

We see that there are infinitely many solutions.

4.6 Picture of a Simple Case

In closing, a really simple example can help nail down what we've done.

Consider the matrix equation

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} [x_1] = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Obviously there is no solution. Graphically Col(A) is the set of multiples of $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and there is no solution since $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$ is not a multiple of $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$.

When we solve the least-squares problem as follows:

$$\begin{aligned} \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} 2 \\ 1 \end{bmatrix} \widehat{[x_1]} &= \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ [5]\,\widehat{[x_1]} &= [6] \\ \widehat{[x_1]} &= [6/5] \end{aligned}$$

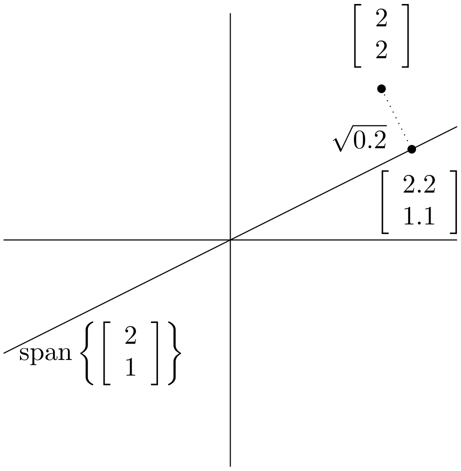
So that

$$A\widehat{[x_1]} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} [6/5] = \begin{bmatrix} 2.4 \\ 1.2 \end{bmatrix}$$

which is in $\text{Col}(A)$ and is as close as possible to $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$, with that distance being the least squares error:

$$\left\| \begin{bmatrix} 2.4 \\ 1.2 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} 0.4 \\ 0.2 \end{bmatrix} \right\| = \sqrt{0.4^2 + 0.2^2} = \sqrt{0.2}$$

as shown here:


$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$
$\sqrt{0.2}$
$\begin{bmatrix} 2.2 \\ 1.1 \end{bmatrix}$
$\text{span}\left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\}$

4.7 Matlab

The transpose of a matrix can be done either with the `transpose` command or an apostrophe:

```
>> A = [1 2;1 1;1 -1];
>> transpose(A)
ans =
     1     1     1
     2     1    -1
>> A’
ans =
     1     1     1
     2     1    -1
```

Practically speaking a least-squares problem can be solved easily in Matlab:

```
>> A = [1 2;1 1;1 -1];
>> b = [6;4;1];
>> inv(A’*A)*A’*b
ans =
    2.5714
    1.6429
```

The `norm` command is useful for the least-squares error:

```
>> A = [1 2;1 1;1 -1];
>> b = [6;4;1];
>> x=inv(A’*A)*A’*b;
>> norm(A*x-b)
ans =
    0.2673
```

4.8 Exercises

Exercise 4.1. Find the least-squares solution and least-squares error for the matrix equation

$$\begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 2 \\ 5 & 1 & 1 \\ 2 & 2 & 0 \end{bmatrix} \bar{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Exercise 4.2. Find the vector in col$A$ closest to $\bar{b}$ where:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & -3 \\ 2 & 6 \end{bmatrix} \text{ and } \bar{b} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Exercise 4.3. Using least squares, find the vector in

$$\text{span} \left\{ \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 3 \\ 0 \\ 3 \end{bmatrix} \right\} \text{ closest to } \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Exercise 4.4. Assuming the dimensions all work out, is it possible for the matrix equation $A^T A\bar{x} = A^T\bar{b}$ to have no solutions? Explain.

Exercise 4.5. Could a system of equations with more variables than equations have a unique least squares solution? Explain.

Exercise 4.6. Using least squares, find the vector in

$$\text{span} \left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\} \text{ closest to } \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

Exercise 4.7. Use the method of least-squares to find the point on the line $y = 3x$ closest to $(2, 3)$.

Exercise 4.8. Suppose A is invertible so that $A\bar{x} = \bar{b}$ actually has a single solution but you use the method of least-squares anyway. Show that the solution you get via least-squares is the actual solution. Hint: Manipulate the least-squares formula.

Exercise 4.9. Show that the following does not have a unique least-squares solution by attempting to find such a solution and explaining where the process fails:

$$\begin{aligned} x + 2y &= 4 \\ x + 2y &= 3 \end{aligned}$$

Exercise 4.10. Consider the following matrix equation $A\bar{x} = \bar{b}$ with:

$$\begin{bmatrix} 1 & 2 \\ -1 & 1 \\ 0 & 3 \end{bmatrix} \bar{x} = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}$$

(a) Find the least-squares solution the long way:

(i) Find an orthogonal basis for $\text{Col}(A)$. You can do this by calling the columns $\bar{c}_1$ and $\bar{c}_2$ and then using $\{\bar{c}_1, \bar{c}_2 - \text{Pr}_{\bar{c}_1}\bar{c}_2\}$ as a basis.

(ii) Find $\text{Pr}_{\text{Col}(A)}\bar{b}$. This equals the sum of the projections of $\bar{b}$ onto each of the basis vectors. Call this $\hat{b}$.

(iii) Solve $A\bar{x} = \hat{b}$.

(b) Find the least-squares solution using the easy method and verify that they're the same.

Exercise 4.11. Repeat the previous question with:

$$\begin{bmatrix} 2 & -2 \\ 1 & 4 \\ 1 & 2 \end{bmatrix} \bar{x} = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

Exercise 4.12. Explain why a least-squares problem always has a solution. Your answer should touch on the issue of the column space and what is really going on under the hood.

Exercise 4.13. Assuming it exists we know that the least-squares solution is given by:

$$\hat{x} = \left(A^T A\right)^{-1} A^T \bar{b}$$

What is mathematically wrong with the following attempt to simplify this? Specifically, which equals signs are not valid and why?

$$\hat{x} = \left(A^T A\right)^{-1} A^T \bar{b} = A^{-1} \left(A^T\right)^{-1} A^T \bar{b} = A^{-1}\bar{b}$$

Exercise 4.14. Suppose we're solving for the least squares solution to $A\bar{x} = \bar{b}$. Why will switching the order of the columns in A have no effect on the solution?

Chapter 5

Curve Fitting

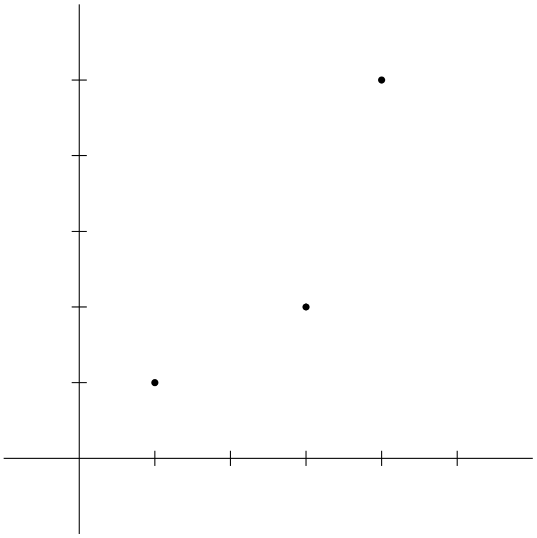
Contents

5.1 Straight Line Fitting

5.1.1 Introductory Example

A classic application of the method of least squares is illustrated by the following example:

Example 5.1. Consider the three points $(1, 1)$, $(3, 2)$ and $(4, 5)$. As we can see these do not lie on a straight line:



But suppose we want want to find a line that's really close to the points, whatever that might mean. How can we apply the above method to do this?

Let's look at the problem. We're trying (and failing) to find a line $y = mx + b$ such that all three points line on it. This means that we want the following to be true:

$$\begin{aligned} 1 &= m(1) + b \\ 2 &= m(3) + b \\ 5 &= m(4) + b \end{aligned}$$

Or, as a matrix equation:

$$\begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

Since we can't solve this (the points don't lie on a line) let's see what the least-squares solution is:

$$\widehat{\begin{bmatrix} m \\ b \end{bmatrix}} = \left(A^T A\right)^{-1} A^T \bar{b} = \begin{bmatrix} 17/14 \\ -4/7 \end{bmatrix}$$

This means $y = \frac{17}{14}x - \frac{4}{7}$ is somehow the best line. What does this mean?

Looking back at our matrix equation the vector

$$\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

contained the y-values that we wanted to get but could not. Instead the y-values that we did get, those contained in the vector

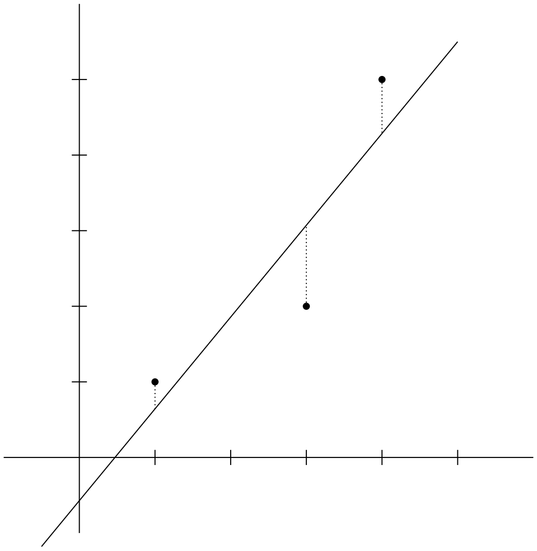
$$\begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \widehat{\begin{bmatrix} m \\ b \end{bmatrix}}$$

are those that minimize

$$\left\| \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \widehat{\begin{bmatrix} m \\ b \end{bmatrix}} - \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \right\|$$

meaning we're minimizing the sum of the squares of the differences between the y-values we wanted and the y-values we obtained.

This can be nicely illustrated by the following picture where we've minimized the sum of the squares of the dotted distances shown:



An interesting note about the previous example is that there are two things going on at once. First, we're finding a best-fit line where "best-fit" means that the sum of the squares of the vertical distances from the points to the line is minimum. Second, we're attempting a matrix equation which is really a three-dimensional problem with no actual solution but with a least-squares solution.

5.1.2 Least Squares Line

We can summarize this as a definition and theorem:

Theorem 5.1.2.1. Given a set of points (x_1, y_1), ..., (x_n, y_n) with not all of the x_i equal the *least squares line* is the line obtained by finding the least squares solution to

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

This line minimizes the sum of the squares of the distances between the y-values of the points and the y-values on the line.

5.2 More General Curve Fitting

Least squares doesn't only work for finding a straight line but it can work for finding any function in which the function is linear in the unknown variables.

What this means is as long as the function you're trying to fit has the form:

$$f(x) = a_1 f_1(x) + a_2 f_2(x) + ...a_n f_n(x)$$

Where the $f_i(x)$ are known, then least squares may be used to find the a_i.

Example 5.2. Consider the points $(-1, 2)$, $(0, 0)$, $(1, 2)$ and $(2, 3)$. These almost follow a parabola. Suppose we want to find a function $f(x) = ax^2 + bx + c$ (a parabola) which does a good job of fitting these four points.

Ideally we'd like the function to actually pass through these points, meaning:

$$\begin{aligned} a(-1)^2 + b(-1) + c &= 2 \\ a(0)^2 + b(0) + c &= 0 \\ a(1)^2 + b(1) + c &= 2 \\ a(2)^2 + b(2) + c &= 3 \end{aligned}$$

As a matrix equation:

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

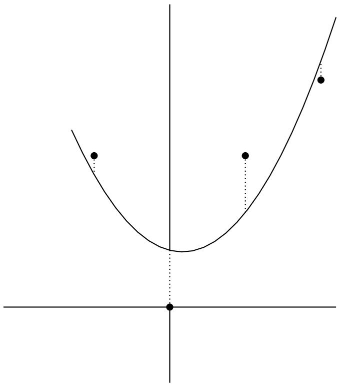
This has no solution but we can find a least-squares solution:

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{bmatrix} \widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{bmatrix}^T \begin{bmatrix} 2 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 18 & 8 & 6 \\ 8 & 6 & 2 \\ 6 & 2 & 4 \end{bmatrix} \widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} = \begin{bmatrix} 16 \\ 6 \\ 7 \end{bmatrix}$$

$$\widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} = \begin{bmatrix} 0.75 \\ -0.25 \\ 0.75 \end{bmatrix}$$

So that the parabola $f(x) = 0.75x^2 - 0.25x + 0.75$ provides the best fit, where best means minimizes the sum of the squares of the vertical distaince.



Just to really understand what we cannot do, and why:

Example 5.3. Consider the points $(-1, 1)$, $(1, 3)$ and $(2, 10)$. Suppose we believe these points follow the function $f(x) = ax + \sin(bx)$. This is all well and good except the corresponding system of equations is:

$$\begin{aligned} a(-1) + \sin(-b) &= 1 \\ a(1) + \sin(b) &= 3 \\ a(2) + \sin(2b) &= 10 \end{aligned}$$

Unfortunately this cannot be written as a matrix equation and so the method of least squares cannot be applied.

5.3 More General Surface Fitting

Least squares doesn't just work when the function is of one variable. The only requirement is that the fuction be linear in the variables we wish to find, so again if the function has the form:

$$f(x_1,...,x_k) = a_1f_1(x_1,...,x_k) + a_2f_2(x_1,...,x_k) + ...a_nf_n(x_1,...,x_k)$$

Where the $f_i(x_1,...,x_n)$ are known.

Example 5.4. Consider the points $(1,1,6)$, $(3,1,22)$, $(5,4,95)$, $(-2,0,10)$. Suppose we believe these points follow an elliptical paraboloid of the form $f(x,y) = ax^2 + by^2 + c$. This would mean that we have:

$$\begin{aligned} a(1)^2 + b(1)^2 + c &= 6 \\ a(3)^2 + b(1)^2 + c &= 22 \\ a(5)^2 + b(4)^2 + c &= 95 \\ a(-2)^2 + b(0)^2 + c &= 10 \end{aligned}$$

As a matrix equation:

$$\begin{bmatrix} 1 & 1 & 1 \\ 9 & 1 & 1 \\ 25 & 16 & 1 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 6 \\ 22 \\ 95 \\ 10 \end{bmatrix}$$

This has no solution but we can find a least-squares solution:

$$\begin{aligned} \begin{bmatrix} 1 & 1 & 1 \\ 9 & 1 & 1 \\ 25 & 16 & 1 \\ 4 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 1 & 1 \\ 9 & 1 & 1 \\ 25 & 16 & 1 \\ 4 & 0 & 1 \end{bmatrix} \widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} &= \begin{bmatrix} 1 & 1 & 1 \\ 9 & 1 & 1 \\ 25 & 16 & 1 \\ 4 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 6 \\ 22 \\ 95 \\ 10 \end{bmatrix} \\ \begin{bmatrix} 723 & 410 & 39 \\ 410 & 258 & 18 \\ 39 & 18 & 4 \end{bmatrix} \widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} &= \begin{bmatrix} 2619 \\ 1548 \\ 133 \end{bmatrix} \\ \widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} &\approx \begin{bmatrix} 2.0048 \\ 2.7083 \\ 1.5155 \end{bmatrix} \end{aligned}$$

So that the elliptical paraboloid $f(x, y) = 2.0048x^2 + 2.7083y^2 + 1.5155$ provides the best fit, where best means minimizes the sum of the squares of the vertical distances.

5.4 Real World Modeling and Predictions

5.4.1 Choosing a Function

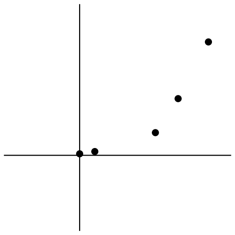
In real-world data modeling if you're using least-squares, especially with two variables, the first step would be to come up with a best-guess as to what the function might be. One thing to note is that the creation of a function (a model for the data) does not mean finding out what function the data follows. The data is just data, it doesn't necessarily obey any model at all. Whether a model is good or not is simply based upon whether it delivers on whatever we need it to do. Generally you would get data, build a model using some method for some reason, test it (on more data, or in the field) and then adjust accordingly.

Here's an example illustrating how you might start:

Example 5.5. Suppose you are analyzing average health insurance costs as a function of time. You collect the following sparse data in the form (t, d) where t is years after 2000 and d is average yearly cost taken over all individuals. In reality you'd have a LOT more data than this:

$$(0, 2), (2, 5), (10, 30), (13, 75), (15, 150)$$

The first thing you'd do is plot this, and you get the following where the axes ratio is not to scale:



There are any number of functions that might fit this data. The two that might leap out at you are exponential functions and quadratic functions. Might as well try both.

For an exponential function you might suggest $f(t) = a + be^t$ but then you note that e^{15} is really big, much bigger than 150, so perhaps a different base is better. For a guess you might want a base b so that $b^{15} \approx 150$ so $b \approx \sqrt[15]{150} \approx 1.3966$ so you figure that you'll give 1.4 a try and so you go with $f(t) = a + b(1.4)^t$.

If this were the case then you'd have:

$$\begin{aligned} f(0) &= a + b(1.4)^0 = 2 \\ f(2) &= a + b(1.4)^2 = 5 \\ f(10) &= a + b(1.4)^{10} = 30 \\ f(13) &= a + b(1.4)^{13} = 75 \\ f(15) &= a + b(1.4)^{15} = 150 \end{aligned}$$

And the corresponding matrix equation would be:

$$\begin{bmatrix} 1 & (1.4)^0 \\ 1 & (1.4)^2 \\ 1 & (1.4)^{10} \\ 1 & (1.4)^{13} \\ 1 & (1.4)^{15} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 30 \\ 75 \\ 150 \end{bmatrix}$$

The least-squares solution to this is

$$\widehat{\begin{bmatrix} a \\ b \end{bmatrix}} = \begin{bmatrix} 1.8982 \\ 0.9463 \end{bmatrix}$$

which gives you the function:

$$f(t) = 1.8982 + 0.9463(1.4)^t$$

which has the least-squares error:

$$\left\| \begin{bmatrix} 1 & (1.4)^0 \\ 1 & (1.4)^2 \\ 1 & (1.4)^{10} \\ 1 & (1.4)^{13} \\ 1 & (1.4)^{15} \end{bmatrix} \begin{bmatrix} 1.8982 \\ 0.9463 \end{bmatrix} - \begin{bmatrix} 2 \\ 5 \\ 30 \\ 75 \\ 150 \end{bmatrix} \right\| \approx 2.7601$$

For a quadratic function you might suggest $f(t) = at^2 + bt + c$. If this were the case then you'd have:

$$\begin{aligned}
f(0) &= a(0)^2 + b(0) + c = 2 \\
f(2) &= a(2)^2 + b(2) + c = 5 \\
f(10) &= a(10)^2 + b(10) + c = 30 \\
f(13) &= a(13)^2 + b(13) + c = 75 \\
f(15) &= a(15)^2 + b(15) + c = 150
\end{aligned}$$

And the corresponding matrix equation would be:

$$\begin{bmatrix} 0 & 0 & 1 \\ 4 & 2 & 1 \\ 100 & 10 & 1 \\ 169 & 13 & 1 \\ 225 & 15 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 30 \\ 75 \\ 150 \end{bmatrix}$$

The least-squares solution to this is

$$\widehat{\begin{bmatrix} a \\ b \\ c \end{bmatrix}} = \begin{bmatrix} 1.3577 \\ -11.7172 \\ 10.9079 \end{bmatrix}$$

which gives you the function:

$$f(t) = 1.3577t^2 - 11.7172t + 10.9079$$

which has the least-squares error:

$$\left\| \begin{bmatrix} 0 & 0 & 1 \\ 4 & 2 & 1 \\ 100 & 10 & 1 \\ 169 & 13 & 1 \\ 225 & 15 & 1 \end{bmatrix} \begin{bmatrix} 1.3577 \\ -11.7172 \\ 10.9079 \end{bmatrix} - \begin{bmatrix} 2 \\ 5 \\ 30 \\ 75 \\ 150 \end{bmatrix} \right\| \approx 21.9904$$

You see that the least-squares error for the exponential is smaller and so your exponential function does a better job of modeling the data than your quadratic function does.

Note that this doesn't mean it's correct (that is, the data is not necessarily exponential), it just means out of two models, one fits the data better than the other.

5.4.2 Predicting

Once we have obtained our least squares function we can then use it to make predictions.

Example 5.6. In the previous health care example if you wanted to predict what the health care costs might be in 2020 you could use your quadratic model:

$$f(20) = 1.3577(20)^2 - 11.7172(20) + 10.9079 \approx 319.6439$$

If you're interested in when the costs might hit 400, you can solve $f(t) = 400$. Doing so yields a positive and negative solutions. We ignore the negative because we're focusing on the future with cost increasing as t increases. The positive solution is approximately 21.7851.

5.5 Matlab

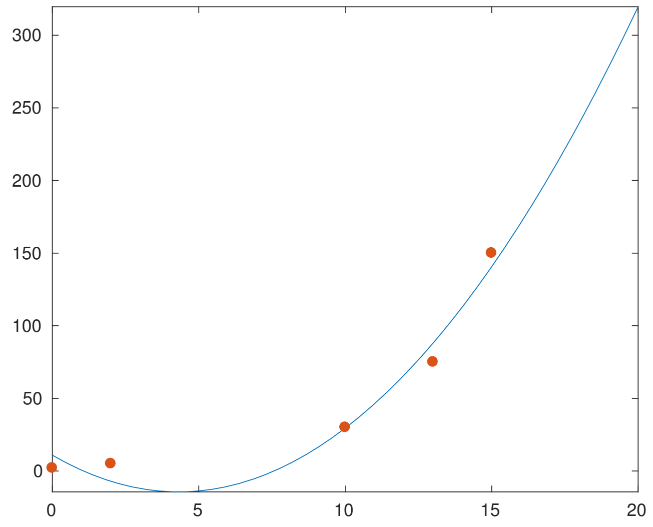
When you've found a function if you want to play with it in Matlatb you can do so easily:

```
>> f(t) = 1.3577*t^2-11.7172*t+10.9079;
>> vpa(f(27),4)
ans =
684.3
>> vpa(solve(f(t)==500),4)
ans =
 -15.15
  23.78
```

It can be useful to plot a resulting function along with a line. The easiest way to do this is with the `hold on` command, which holds on to the current figure when you draw the next figure. The following code:

```
>> syms f(t)
>> f(t) = 1.3577*t^2-11.7172*t+10.9079;
>> pts = transpose([0,2;2,5;10,30;13,75;15,150]);
>> fplot(f,[0,20])
>> hold on
>> scatter(pts(1,:),pts(2,:),'filled')
```

Shows the following:



A few things to note: The `fplot` command does a function plot and we've given it `[0,20]` for the domain. The `scatter` command does a scatter plot and requires all the x-coordinates followed by the y-coordinates. The matrix `pts` contains all the points with each point being a column, so then `pts(1,:)` pulls out the first row (the `:` gets the entire column) which consists of the x-coordinates and `pts(2,:)` pulls out the second row which consists of the y-coordinates.

5.6 Exercises

Exercise 5.1. Given the points $(-1,0)$, $(1,2)$, $(2,2)$

(a) Find the least-squares line $f(x) = mx + b$ for the points.

(b) Plot the points and the line and explain using your picture what exactly has been minimized.

Exercise 5.2. Given the points:

$$(1,1),\ (5,2),\ (6,2),\ (8,3)$$

(a) Find the least-squares line $f(x) = mx + b$ for the points.

(b) Plot $f(x)$ along with the points.

(c) Use $f(x)$ to estimate y when $x = 20$.

Exercise 5.3. Given the points:

$$(-1,3),\ (0,1),\ (1,2),\ (3,9)$$

(a) Find the least-squares parabola $f(x) = ax^2 + bx + c$ for the points.

(b) Plot $f(x)$ along with the points.

(c) Use $f(x)$ to estimate all x so that $f(x) = 10$.

Exercise 5.4. Given the points:

$$(-2,0),\ (0,3),\ (4,4)$$

(a) Find the least-squares exponential $f(x) = ae^x + b$ for the points.

(b) Plot $f(x)$ along with the points.

(c) Use $f(x)$ to estimate $f'(1)$.

Exercise 5.5. Given the points:

$$(-2,6.3),\ (3,1.2),\ (5,7.1),\ (8,-2.8),\ (9,-0.05)$$

(a) Find the least-squares $f(x) = a + b\sin x$ for the points.

(b) Plot $f(x)$ along with the points.

(c) Use $f(x)$ to estimate $f(\pi/2)$.

Exercise 5.6. Given the points:

$$(-3,-2,45)\ (2,-2,30),\ (0,1,6),\ (-2,3,55),\ (6,5,230)$$

(a) Find the least-squares paraboloid $f(x,y) = ax^2 + by^2$ for the points.

(b) Use $f(x, y)$ to estimate $f(3, 5)$.

Exercise 5.7. Here is an interesting question - given the points $(0, 0)$, $(0, 1)$, $(1, 1)$ if we're looking for a best-fit line it's possible to look both for $y = mx + b$ and for $x = ny + c$. Neither has an exact solution but both have least-squares solution. Find each of these. Show that these don't yield the same line. Plot the points and both lines. From a geometric perspective of minimizing distance from the line, what is going on here?

Exercise 5.8. Suppose you would like to estimate the orbit of a certain object around the origin. Observations are made of both an angle and a distance. You collect five observations as follows where the first value is degrees and the second is in millions of miles:

$$(23^\circ, 152),\ (50^\circ, 135),\ (100^\circ, 102),\ (110^\circ, 110),\ (152^\circ, 137)$$

The equation of an ellipse in polar coordinates can be given by the following for some A and B:

$$Ar^2 \cos^2 \theta + Br^2 \sin^2 \theta = 1$$

(a) Find the least-squares best-fit ellipse.

(b) Use this to predict the distance of the object when $\theta = 225^\circ$.

(c) What is the furthest that the object ever gets from the origin?

Exercise 5.9. Repeating data points has an impact on the method of least squares. To visualize this, imagine we're trying to best-fit a straight line to a set of points. If a point appears more than once then the square of the distance to the line is being counted more than once and hence carries more weight in the method. To test this out find the least-squares line which best fits each of the following sets of points. Which line is closer to the point $(3, 2)$?

(a) The points $(1, 1)$, $(2, 1)$, $(3, 2)$

(b) The points $(1, 1)$, $(2, 1)$, $(3, 2)$, $(3, 2)$

Exercise 5.10. Consider the set of $n + 2$ points:

$$(1, 1), (2, 1), \underbrace{(3, 2), (3, 2), ..., (3, 2)}_{n \text{ times}}$$

Suppose you wish to best-fit these to a line $y = mx + b$ using least-squares.

(a) Write down the corresponding matrix equation.

(b) Solve for $\hat{x}_n$ using the method of least squares. Make sure you simplify; the answer should not be complicated.

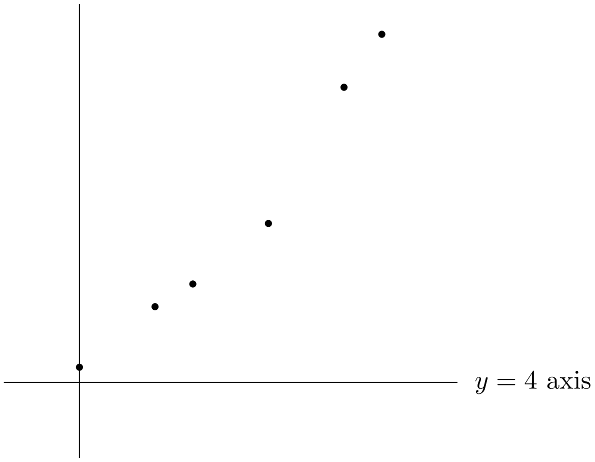
(c) Find $\lim_{n\to\infty} \hat{x}_n$.

(d) The line corresponding to your answer in (c) passes through $(3, 2)$. Why does this make sense?

Exercise 5.11. This problem loosely follows the data modeling example from class. Suppose you collect the following data points:

$$(0, 4.2), (2, 5), (3, 5.3), (5, 6.1), (7, 7.9), (8, 8.6)$$

When you plot these you see:



(a) Use least-squares to fit the function $f(x) = mx + b$.

(b) Use least-squares to fit the function $f(x) = ax^2 + bx + c$.

(c) If the data were to fit the function $f(x) = a + bc^x$, make an educated guess for c and then use least-squares to fit the function. Hint: One idea might be to ignore a and b and suggest that $f(x) \approx c^x$ especially for bigger x, but there are other options for guessing c.

(d) Calculating the least-squares error for each, which seems to provide the best fit?

(e) Use that function to predict $f(10)$.

(f) Use that function to predict which x would yield $f(x) = 50$.

Exercise 5.12. The following sets of points each approximately follows a familiar function which is linear in some unknowns. First plot the points. Then make an educated sensible guess as to the form of the function. Finally use the method of least squares to find a best-fit function and estimate the y-value corresponding to the given x value. The problems work from easier to more difficult.

(a) Two unknowns, $x = 10$.

$$(-1, 8.5), (1, 2.5), (2, 0.53), (3, -1.5), (6, -7.4), (7, -9.5)$$

(b) Two unknowns, $x = 20$.

$$(-3, 20), (-1, 3.5), (1, 3.6), (2, 9.5), (5, 52), (7, 100)$$

(c) Three unknowns, $x = 2$.

$$(-1, -5.4), (0, -1.9), (1, 0.51), (3, 2.5), (4, 2.0), (6, -1.9)$$

(d) Two unknowns, $x = -10$.

$$(-4, 6.1), (-1, 2.9), (0, 4.5), (2, 6.3), (6, 4.0), (7, 5.8), (8, 6.6), (10, 3.5), (11, 2.6)$$

(e) Three unknowns, $x = 20$.

$$(-2, -0.69), (-1, 1.5), (0, 2.5), (1, 1.8), (2, 0.08), (4, 0), (5, 2.2), (6, 3.7), (8, 1.9), (9, 0.49)$$

Exercise 5.13. For which of the following function templates will the method of least squares work and for which will it not. Explain. For one of the ones for which it will not work cite an example and show in detail what goes wrong. Your answer to this second part should touch on the issue of linear vs nonlinear systems.

(a) $f(x) = ax^2 + bx^c$

(b) $f(x) = ae^x + bx$

(c) $f(x) = e^{ax} + bx$

(d) $f(x) = a\sin(x) + b\cos(x) + c$

(e) $f(x) = a\sin(bx) + c$

Chapter 6

Team Ranking

Contents

6.1 Introduction

Suppose there are three sports teams called T1, T2 and T3 and they play a number of games against one another with point outcomes for each game. We wish to assign some sort of numerical rank to each team so that we can decide who is best, second best, and third best, and in some sense by how much.

Just to see how this might be complicated, consider: If T1 beats T2, T2 beats T3 and T1 beats T3 then it's fairly clear how to rank the teams in terms of who is best. But what if T1 beats T2, T2 beats T3 and T3 beats T1, then who is best? Suppose T1 beats T2 by 5, T2 beats T3 by 3 and T3 beats T1 by just 1, now what? As we can tell, this isn't entirely clear at all.

The method we present is Massey's Method, developed by Kenneth Massey while he was an undergraduate.

6.2 Method

6.2.1 Building a System of Equations

What could we mean to assign each team a ranking?

One idea is that for two teams i and j with numerical rankings r_i and r_j it seems reasonable that if they play a game in which team i beats team j by p points then we could have $r_i - r_j = p$. For example if team i beats team j by 3 points then it seems reasonable that $r_i - r_j = 3$ because team i is 3 points better than team j.

For losses and ties this works just fine. For example if team i and team j tie then we can write either $r_i = r_j$ or $r_i - r_j = 0$ and if team j beats team i by 5 then we can write either $r_j - r_i = 5$ or $r_i - r_j = -5$.

By way of a juicy example let's suppose that there are four teams T1, T2, T3, T4. During a particular season we have the following where a win by a negative number really indicates a loss:

- T1 plays T2 and wins by 2.
- T1 plays T2 again and wins by 1.
- T1 plays T4 and wins by -2 (loses by 2).
- T2 plays T3 and wins by 1.
- T2 plays T4 and wins by 3.
- T3 plays T4 and wins by -3 (loses by 3).

If the rankings are r_1, r_2, r_3, r_4 then we ask that:

$$\begin{aligned}
r_1 - r_2 &= 2 \\
r_1 - r_2 &= 1 \\
r_1 - r_4 &= -2 \\
r_2 - r_3 &= 1 \\
r_2 - r_4 &= 3 \\
r_3 - r_4 &= -3
\end{aligned}$$

This is akin to solving the matrix equation

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -2 \\ 1 \\ 3 \\ -3 \end{bmatrix}$$

This system has no solution.

6.2.2 Trying to Apply Least Squares

So what can we do? We can instead find the least-squares solution. What would this mean exactly? It would mean finding the team rank values such that if the teams had these rankings and played the games they did, that those game outcomes, as a vector, would be as close as possible to the actual game outcomes.

Onwards! If the above is $A\bar{r} = \bar{p}$ then we solve $A^T A\hat{r} = A^T\bar{p}$ instead:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \hat{r}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -2 \\ 1 \\ 3 \\ -3 \end{bmatrix}$$

which gives us:

$$\begin{bmatrix} 3 & -2 & 0 & -1 \\ -2 & 4 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \hat{r} = \begin{bmatrix} 1 \\ 1 \\ -4 \\ 2 \end{bmatrix}$$

6.2.3 Encountering the Problem

Before we proceed further let's call this new matrix equation $M\hat{r} = \bar{q}$ and make some notes about M. For simplicity suppose there are T teams and G games were played.

- The matrix M is $T \times T$.
- The entry m_{tt} equals the number of games that team t played. For example and $m_{33} = 2$ because T3 played a total of 2 games.

  This is because

  $$\begin{aligned} m_{tt} &= \left(A^T\right)_{t1} a_{1t} + \left(A^T\right)_{t2} a_{2t} + \left(A^T\right)_{t3} a_{3t} + ... + \left(A^T\right)_{tG} a_{Gt} \\ &= a_{1t}^2 + a_{2t}^2 + a_{3t}^2 + ... + a_{Gt}^2 \end{aligned}$$

  and for any $1 \leq g \leq G$ we have $a_{gt} = \pm 1$ if and only if, in game g, team t is involved, meaning that this sum equals the total number of games team t played.
- For $s \neq t$ the entry m_{st} equals negative of the number of games s and t played. For example $m_{21} = -2$ because T2 played T1 twice.

  This is because

  $$\begin{aligned} m_{st} &= \left(A^T\right)_{s1} a_{1t} + \left(A^T\right)_{s2} a_{2t} + \left(A^T\right)_{s3} a_{3t} + ... + \left(A^T\right)_{sG} a_{Gt} \\ &= a_{1s}a_{1t} + a_{2s}a_{2t} + a_{3s}a_{3t} + ... + a_{Gs}a_{Gt} \end{aligned}$$

  and $a_{gs}a_{gt} = -1$ if and only if, in game g, both teams s and t are involved (one has a 1 in the row entry and the other has a -1), meaning that this sum equals the negative of the total number of games team s played against team t.
- The matrix M is symmetric.

- Any single row of M can be calculated from the other rows because any team's game history can be found from knowing every other team's game histories. More explicitly each row is the negation of the sum of the other rows.
- Any two (or more) rows of M cannot be calculated from the remaining rows because it is impossible to know how those two teams performed without knowing something about at least one of them.

The last two points are very important, they say that the rows are linearly dependent and so the matrix is singular and therefore has infinitely many solutions (it definitely has solutions because it's a least-squares problem). Moreover they say that if we were to remove a single row that the remaining rows would be linearly independent.

Notice also that in the vector $\bar{q}$ any entry is the negative of the sum of the remaining entries since the total points won and lost among all teams is zero.

So in conclusion this new least-squares matrix equation has infinitely many solutions, and this is something we need to fix.

We could also have seen this back in the original matrix A in that each row sums to zero because each row has a 1 and a -1 in it. Consequently any column is simply the negation of the sum of the remaining columns. Therefore the columns are not linearly independent and so the least-squares method will yield multiple solutions.

6.2.4 Fixing the Problem

The way we'll solve this is to modify M and modify $\bar{q}$. We will replace the final row r_T (in this case r_4) with all 1s and the bottom entry of $\bar{q}$ with 0. This effectively adds the requirement that $r_1+r_2+r_3+...+r_T=0$, which seems like a perfectly reasonable requirement in the sense that it only adds a requirement which doesn't get in the way of a reasonable team ranking. For the sake of simplicity we'll also call this new matrix M and new vector $\bar{q}$ and never look back.

Notice with this new row we see that for any of rows 1,2,3,...,$T-1$ that $r_i \cdot r_T = 0$ so that row T is perpendicular to each of the other rows and is therefore not a linear combination of them. Therefore the new matrix is invertible since the remaining rows we already knew to be linearly independent as noted in the bullet points above.

Thus we solve

$$\begin{bmatrix} 3 & -2 & 0 & -1 \\ -2 & 4 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \hat{r} = \begin{bmatrix} 1 \\ 1 \\ -4 \\ 0 \end{bmatrix}$$

and we find

$$\hat{r} = \begin{bmatrix} 19/26 \\ 9/26 \\ -41/26 \\ 1/2 \end{bmatrix} \approx \begin{bmatrix} 0.730769 \\ 0.346154 \\ -1.57692 \\ 0.5 \end{bmatrix}$$

So now we can assign numerical rankings to the teams as follows, in decreasing order:

$$\begin{aligned} &\text{Team 1 has ranking } r_1 = 0.730769 \\ &\text{Team 4 has ranking } r_4 = 0.5 \\ &\text{Team 2 has ranking } r_2 = 0.346154 \\ &\text{Team 3 has ranking } r_3 = -1.57692 \end{aligned}$$

Note that, for example, T4 is significantly worse than the other three, all of which are rather close together.

Does this seem reasonable given their records?

6.2.5 Massey Method Summary

In summary the method is very direct:

I. Write down the system of equations corresponding to the games played and the points won or lost where the rankings are the unknowns.

II. Convert to a system of equations $A\bar{r} = \bar{p}$.

III. Write down the least-squares system $A^T A\hat{r} = A^T\bar{p}$.

IV. Change the lower row of the new matrix on the left to all 1s and the bottom entry of the new vector on the right to 0.

V. Solve.

6.2.6 Shortcut

In reality our observations earlier suggest that we don't need the intermediate step of filling in A at all.

Instead we can go straight to $M\hat{r} = \bar{q}$.

For M the final row is all 1s. Other than that row, m_{ii} equals the total number of games team i played and for $i \neq j$, m_{ij} equals the negative of the number of games i and j played.

For $\bar{q}$ the final entry is 0. Other than that entry, q_i equals the total number of points that team i earned.

Example 6.1. Suppose we have five teams who play a series of twelve games with the following outcomes:

- T1 plays T2 and wins by 3.
- T1 plays T3 and ties.
- T1 plays T4 and wins by -2.
- T2 plays T3 and wins by 1.
- T2 plays T3 and wins by 2.
- T2 plays T4 and wins by -3.
- T2 plays T5 and wins by -5.
- T3 plays T4 and ties.
- T3 plays T4 and wins by -1.
- T3 plays T5 and ties.
- T4 plays T5 and wins by 4.
- T4 plays T5 and wins by -3.

We can immediately fill in the following:

$$\begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 5 & -2 & -1 & -1 \\ -1 & -2 & 6 & -2 & -1 \\ -1 & -1 & -2 & 6 & -2 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \hat{r} = \begin{bmatrix} 1 \\ -8 \\ -4 \\ 7 \\ 0 \end{bmatrix}$$

which we solve to get:

$$\hat{r} = \begin{bmatrix} -0.002899 \\ -1.438 \\ -0.6261 \\ 1.055 \\ 1.012 \end{bmatrix}$$

In decreasing order we have rankings:

$$\begin{aligned} &\text{Team 4 has ranking } r_4 = 1.055 \\ &\text{Team 5 has ranking } r_5 = 1.012 \\ &\text{Team 1 has ranking } r_1 = -0.002899 \\ &\text{Team 3 has ranking } r_3 = -0.6261 \\ &\text{Team 2 has ranking } r_2 = -1.438 \end{aligned}$$

6.3 Commentaries

6.3.1 Ranking is Relative But...

It's important to understand that the rankings are relative to one another. For example a ranking of 4/3 has no absolute numerical meaning until it is compared to another ranking at which point we can say which is better and we can give some sense of how much better.

However the working premise was that the difference between two teams' ranking $r_i - r_j$ ought to equal the number of points that Team i wins by in a game against Team j and so it's possible to use this difference to guess at how two teams might perform against one another.

Example 6.2. In the previous example T1 never played T5. However their relative rankings are -0.002899 and 1.1012 respectively. It follows that we can guess that if they did play that the score difference would be $-0.002899 - 1.1012 = -1.104099$ meaning T5 would win by 1.104099 points.

6.3.2 Ties

Massey's Method already takes into account ties since a tie between teams i and j can be entered as $r_i - r_j = 0$ or $r_j - r_i = 0$.

6.3.3 Multiple Games

Since least-squares is sensitive to repeated equations in curve fitting if two teams play more than once then Massey's Method will account for this simply by making sure both equations have been entered into the mix.

6.3.4 Weighting Games

It's possible to give a game more significance in two straightfoward ways. First, the point difference can be increased, so for example a win by 5 points could be recorded as a win by more than 5 points if we want to give that game more significance. Alternately the game could be recorded two or more times. The difference between these is explored in the exercises.

6.3.5 Disconnected Sets of Games

The Massey Method fails to return a result when the teams may be divided into two (or more) subsets none of whom play one another.

This is because the matrix $A^T A$ (and in fact even just the matrix A) is divided into two (or more) blocks. When we create M with the lower row all being 1 the lower row of any upper block is still a linear combination of rows from the same block.

We can't replace two (or more) entire rows by 1s because then they are linearly dependent. The solution would be to replace the appropriate parts of M corresponding to the lower row of each block which has the practical result of just applying the Massey Method to each subset of games.

This is explored further in the exercises.

6.4 Matlab

Nothing new here.

6.5 Exercises

Exercise 6.1. Suppose four teams play a number of games with the following results:

- Team 1 beats Team 2 by 7 points.
- Team 1 beats Team 3 by -8 points.
- Team 2 beats Team 3 by 0 points.
- Team 2 beats Team 3 by 15 points.
- Team 2 beats Team 4 by 5 points.
- Team 3 beats Team 4 by -1 points.

(a) Find the team rankings.

(b) Even though Team 1 did not play Team 4 if they were to play what result might you expect? Hint: The working assumption is that we wished $r_i - r_j$ equals the amount that team i wins by.

Exercise 6.2. Suppose four teams play a number of games with the following results:

- Team 1 beats Team 2 by 17 points.
- Team 1 beats Team 3 by 21 points.
- Team 2 beats Team 3 by 10 points.
- Team 2 beats Team 3 by 0 points.
- Team 2 beats Team 4 by -14 points.
- Team 3 beats Team 4 by 8 points.

(a) Find the team rankings.

(b) Even though Team 2 did not play Team 4 if they were to play what result might you expect?

Exercise 6.3. If the Massey matrix equation for a set of teams is:

$$\begin{bmatrix} 3 & -1 & -2 & 0 \\ -1 & 4 & -2 & -1 \\ -2 & -2 & 4 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \hat{r} = \begin{bmatrix} 6 \\ 2 \\ 10 \\ 0 \end{bmatrix}$$

(a) How many teams are there?

(b) How many games did Team 4 play?

(c) How many games did Team 1 play and against whom?

(d) How many total games were played?

Exercise 6.4. Before replacing the final row of the Massey matrix by all 1s, the sum of the diagonal entries be even. Why?

Exercise 6.5. Suppose four teams play a number of games with the following results:

- Team 1 beats Team 2 by 5 points. Team 1 at home.
- Team 1 beats Team 3 by 10 points. Team 3 at home.
- Team 2 beats Team 3 by -10 points. Team 2 at home.
- Team 2 beats Team 3 by 7 points. Team 3 at home.
- Team 2 beats Team 4 by -14 points. Team 2 at home.
- Team 3 beats Team 4 by 21 points. Team 3 at home.

(a) Find the team rankings, ignore the "at home" aspect.

(b) Suppose when a team plays at home it has a point advantage. Suppose history suggests that to factor this into the scores if a team wins at home the score should be lowered by 25% before calculating the ranking. Find the new team rankings; do they change?

Exercise 6.6. It seems reasonable that if T1 beats T2 by k points, T2 beats T3 by k points and T3 beats T1 by k points that all three teams should have the same ranking. Show that this is the case.

Exercise 6.7. Suppose two teams play two games. In the first game Team 1 beats Team 2 by α points. In the second game Team 2 beats Team 1 by β points. Both α and β could be zero or negative. Find the rankings of the two teams.

Exercise 6.8. Suppose three teams play three games with the following results:

- Team 1 beats Team 2 by 17 points.
- Team 2 beats Team 3 by 22 points.
- Team 3 beats Team 1 by α points.

(a) Assuming α is unknown find the team rankings.

(b) What would α need to be so that Team 3 ranks higher than Team 1?

Exercise 6.9. Is it better for a team to beat another team twice by one point each time or once by two points? Provide evidence to support your assertion. Note: This question is given in an easier version in the next question; this question is more exploratory.

Exercise 6.10. Is it better for a team to beat another team twice by one point each time or once by two points?

(a) Find the rankings in the following scenario:

- T1 beats T2 by 3 point,
- T2 beats T3 by 2 points,
- T1 beats T3 by 2 points.

(b) Find the rankings in the following scenario:

- T1 beats T2 by 3 point,
- T2 beats T3 by 2 points,
- T1 beats T3 by 1 point.
- T1 beats T3 by 1 point.

(c) What is your conclusion?

Exercise 6.11. This problem explores the same idea as the previous problem.. It looks at the difference between winning 1 game by n points versus winning n games by 1 point.

(a) Suppose T1 beats T2 by n points, T2 beats T3 by 1 point and T3 beats T1 by 1 point. Find the team rankings $\bar{r}_n$ and find $\lim\limits_{n\to\infty} \bar{r}_n$.

(b) Suppose T2 beats T2 by 1 point but does this n times, T2 beats T3 by 1 point and T3 beats T1 by 1 point. Find the team rankings $\bar{r}_n$ and find $\lim\limits_{n\to\infty} \bar{r}_n$. This should simplify a lot!

(c) What do you notice about your two limits? Which seems better for T1?

Exercise 6.12. Suppose three teams T1,T2,T3 play games as follows:

- T1 beats T2 by α.
- T2 beats T3 by β.
- T3 beats T1 by γ.

Note that α, β, γ may be zero or negative.

(a) Show that if the three are to be ranked equally that it must be true that $\alpha = \beta = \gamma$.

(b) Is it possible for exactly two of the teams to be ranked the same if all of α, β, γ are different from one another? If not, explain why not. If so, give an example.

(c) Is it possible to choose α, β, γ so that the spread of rankings is arbitrarily large? Explain.

Exercise 6.13. Suppose six teams play games as follows:

- T1 beats T2 by 2.
- T2 beats T3 by 3.
- T4 beats T5 by 2.
- T5 beats T6 by 3.

(a) Show that the Massey method as given will fail to rank the teams.

(b) Rank the teams by applying the Massey method to each subset T1, T2, T3 and T4, T5, T6.

(c) Suppose later on T3 plays a game against T4 and wins by 1. Apply the Massey Method to rank all six teams together.

(d) Does the way the rankings change after the T3-T4 game make sense intutively? Explain.

Exercise 6.14. The Massey method modifies $A^T A\bar{r} = A^T\bar{p}$ by replacing the lowest row of $A^T A\bar{r}$ by all 1s and by replacing the lowest entry of $A^T\bar{p}$ by 0. This is the same as insisting that $r_1 + ... + r_n = 0$. What would happen if we had set this equal to something else? Modify the first question and instead use $r_1 + ... + r_n = 1$ and then try it with $r_1 + ... + r_n = 4$. What do you find in each case? What do you think the general pattern is?

Exercise 6.15. Suppose a collection of more than two teams play a series of games. If the games and/or scores between exactly two of the teams changes can this affect the ranking value of teams other than those two? Provide evidence to support your assertion.

Exercise 6.16. Give some nontrivial examples of situations where the team rankings produced by the Massey method do not require least-squares to solve; that is, there is an exact solution the initial desired system of equations. Is it still necessary to insist that the sum of all of the rankings is zero?

Chapter 7

Markov Chains

Contents

7.1 Introduction

Markov chains have many applications but we'll start with one which is easy to understand.

7.1.1 The Problem

Suppose there are two states (think countries, or US states, or cities, or whatever) 1 and 2 with a total population of 1 distributed as 0.7 in State 1 and 0.3 in State 2.

Suppose that at the end of the year 10% of the people in State 1 move out of State 1 and into State 2 (the rest remain) and 5% of the people in State 2 move out of State 2 and into State 1 (the rest remain).

What will the new population distribution be? We can find this out easily:

State 1 now has $0.7 - 0.10(0.7) + 0.05(0.3) = 0.6450$

State 2 now has $0.3 - 0.05(0.3) + 0.10(0.7) = 0.3550$

There's another way to write this, however. If we rephrase this as 90% of the people in State 1 stay in State 1 and 5% of the people in State 2 move to State1, and similarly for State 2 then we get:

State 1 now has $0.90(0.7) + 0.05(0.3) = 0.6450$

State 2 now has $0.10(0.7) + 0.93(0.3) = 0.3550$

Now then, suppose this happened again the next year. Again, this is easy to find out:

State 1 now has $0.90(0.6450) + 0.05(0.3550) = 0.5983$

State 2 now has $0.10(0.6450) + 0.95(0.3550) = 0.4017$

Suppose this keeps happening year-by-year for years. This calculation would not only be annoying to repeat but perhaps it's possible to gain some insight without doing it over and over.

7.1.2 The Problem Rephrased with Matrices and Vectors

For starters let's notice that we can convert the calculations very easily to linear algebra. If the populations start with the vector:

$$\begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$$

Then after one year the new populations are:

$$\begin{bmatrix} 0.90 & 0.05 \\ 0.10 & 0.95 \end{bmatrix} \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.90(0.7) + 0.05(0.3) \\ 0.10(0.7) + 0.95(0.3) \end{bmatrix} = \begin{bmatrix} 0.6450 \\ 0.3550 \end{bmatrix}$$

In other words if the population is $\bar{x}_0$ then after one year the population is:

$$\bar{x}_1 = T\bar{x}_0$$

Where

$$T = \begin{bmatrix} 0.90 & 0.05 \\ 0.10 & 0.95 \end{bmatrix}$$

Furthermore after another year the population is:

$$\bar{x}_2 = T\bar{x}_1 = T(T\bar{x}_0) = T^2\bar{x}_0$$

and in general after n years:

$$\bar{x}_n = T^n\bar{x}_0$$

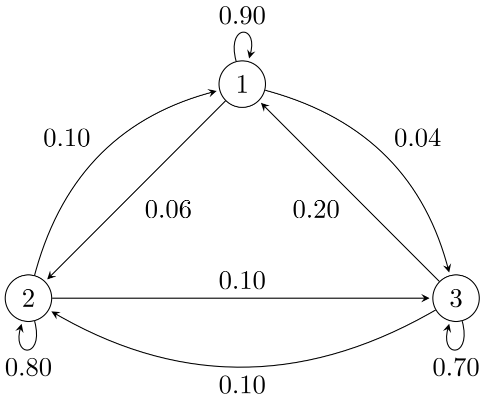
7.1.3 Higher Dimensions

The first nice thing to notice about moving to matrices and vectors is that the problem generalizes easily to higher dimensions.

First let's make one thing clear that it's easy to mess up. In the matrix T (called the transition matrix) column j contains the information about where the population in state j goes to. You could think of as an "exit vector" from state j. More specifically the (i, j) row (i^{th} entry in the j^{th} column) contains the proportion of people who moved from state j to state i in exactly one iteration.

So then when we move to more states we simply add rows and columns as needed for the number of states we have.

Example 7.1. Consider the following diagram which shows the populations shifting between three states:



Let's determine the steady state of this situation in terms of how the populations will distribute in the long term.

The transition matrix is:

$$T = \begin{bmatrix} 0.90 & 0.10 & 0.20 \\ 0.06 & 0.80 & 0.10 \\ 0.04 & 0.10 & 0.70 \end{bmatrix}$$

So now for a given initial population distribution we can find out the distribution after one, two and five iterations easily: If

$$\bar{x}_0 = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}$$

then

$$\begin{aligned} T\bar{x}_0 &\approx \begin{bmatrix} 0.2500 \\ 0.2360 \\ 0.5140 \end{bmatrix} \\ T^2\bar{x}_0 &\approx \begin{bmatrix} 0.3514 \\ 0.2552 \\ 0.3934 \end{bmatrix} \\ T^5\bar{x}_0 &\approx \begin{bmatrix} 0.5007 \\ 0.2692 \\ 0.2302 \end{bmatrix} \end{aligned}$$

7.1.4 Long Term Behavior Experiment

So what happens in the long term? Let's take a look as n gets large for our original distribution:

$$\begin{aligned}
\bar{x}_0 &= \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \\
T\bar{x}_0 &= \begin{bmatrix} 0.6450 \\ 0.3550 \end{bmatrix} \\
T^2\bar{x}_0 &= \begin{bmatrix} 0.5983 \\ 0.4017 \end{bmatrix} \\
T^{10}\bar{x}_0 &\approx \begin{bmatrix} 0.4055 \\ 0.5945 \end{bmatrix} \\
T^{100}\bar{x}_0 &\approx \begin{bmatrix} 0.3333 \\ 0.6667 \end{bmatrix} \\
T^{1000}\bar{x}_0 &\approx \begin{bmatrix} 0.3333 \\ 0.6667 \end{bmatrix}
\end{aligned}$$

Oh, that's interesting. It looks like the population settles whereby 1/3 is in State 1 and 2/3 is in State 2. Moreover it really suggests two things:

- $\lim\limits_{n\to\infty} T^n\bar{x}_0 = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix}$ for this particular $\bar{x}_0$.
- $T\begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix}$

The second of these is true, and it's easy to check.

As for the first, we might ask a preliminary question: What if we'd started with a very different distribution of people? Let's try with 0.05 and 0.95 in State 1 and State 2 respectively!

$$\begin{aligned}
\bar{x}_0 &= \begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix} \\
T\bar{x}_0 &= \begin{bmatrix} 0.0925 \\ 0.9075 \end{bmatrix} \\
T^{10}\bar{x}_0 &= \begin{bmatrix} 0.2776 \\ 0.7224 \end{bmatrix} \\
T^{100}\bar{x}_0 &\approx \begin{bmatrix} 0.3333 \\ 0.6667 \end{bmatrix} \\
T^{1000}\bar{x}_0 &\approx \begin{bmatrix} 0.3333 \\ 0.6667 \end{bmatrix}
\end{aligned}$$

Oh wow! Could it be that the population always approaches this same distribution?

7.2 Steady States and Limits

7.2.1 Formal Stuff

It appears (no proof yet!) in our simple example that there is a vector which is fixed by T and to which perhaps all other states approach as $n \to \infty$. However is this always the case?

Let's lay everything out formally:

Definition 7.2.1.1. A *probability vector* is a vector whose entries lie between 0 and 1 (inclusive) and add to 1. An example would be a distribution of the population.

Definition 7.2.1.2. A *transition matrix* (also known as a *stochastic matrix*) is a matrix in which each column is a probability vector. An example would be the matrix representing how the populations shift year-to-year where the (i, j) entry contains the fraction of people who move from state j to state i in one iteration.

Definition 7.2.1.3. A probability vector $\bar{x}$ is a *steady-state vector* for a transition matrix T if $T\bar{x} = \bar{x}$. Notice that a steady-state vector is an eigenvector corresponding to the eigenvalue $\lambda = 1$.

Definition 7.2.1.4. A *regular transition matrix* is a transition matrix T such that there is some integer $k \geq 1$ such that all entries of T^k are nonzero. For the simplest case if all the entries of T itself are nonzero then T is a regular transition matrix.

Theorem 7.2.1.1. If T is a regular transition matrix then it has $\lambda = 1$ as an eigenvalue and there is a unique steady-state eigenvector $\bar{x}_*$. Moreover for any probability vector $\bar{x}_0$, $\lim\limits_{k\to\infty} T^k \bar{x}_0 = \bar{x}_*$.

Proof. This proof is hard and is omitted. Later in the chapter there is a proof for the 2 × 2 case. □

Corollary 7.2.1.1. If T is a regular transition matrix then $\lim\limits_{k\to\infty} T^k$, where the columns are all identitical and all equal $\bar{x}_*$.

Proof. Suppose T is $n \times n$, then we know from the theorem that for each $1 \leq i \leq n$ we have

$$\lim_{k\to\infty} T^k \bar{e}_i = \bar{x}_*$$

so that

$$\lim_{k\to\infty} T^k \left[\bar{e}_1 \ \ \dots \ \ \bar{e}_n\right] = \left[\bar{x}_* \ \ \dots \ \ \bar{x}_*\right]$$
$$\lim_{k\to\infty} T^k I = \left[\bar{x}_* \ \ \dots \ \ \bar{x}_*\right]$$

which gives the result. □

It follows from the corrollary that computationally speaking if we want to approximate the steady state vector for a regular transition matrix T that all we need to do is look at one column from T^k for some very large k.

Fact 7.2.1.1. If T is a transition matrix but is not regular then there is no guarantee that the results of the Theorem will hold! They might, but no guarantee.

Fact 7.2.1.2. A transition matrix which is not regular may have more than one steady state vector and there are no guarantees about limiting behavior.

Fact 7.2.1.3. If you use Matlab or Wolfram Alpha to find this eigenvector be aware that it will almost certainly not give you a probability vector. For example Matlab typically gives a unit vector. However since any multiple of an eigenvector is an eigenvector you can simply divide by the sum of the values to get an eigenvector which is a probability vector.

7.2.2 Full Theory Example

Example 7.2. Consider the example from earlier with transition matrix

$$T = \begin{bmatrix} 0.90 & 0.10 & 0.20 \\ 0.06 & 0.80 & 0.10 \\ 0.04 & 0.10 & 0.70 \end{bmatrix}$$

Notice that T is regular because all entries of T^1 are nonzero.

Now then, the eigenvalues of this matrix are $\lambda_1 = 1$, $\lambda_2 \approx 0.763246$, $\lambda_3 \approx 0.636754$.

Note that in reality we don't need these - we know (from the theorem) that $\lambda = 1$ is an eigenvalue and so all we really need is the associated steady-state vector, meaning an eigenvector which is a probability vector, meaning the entries add to 1.

Matlab tells us an eigenvector for $\lambda_1 = 1$ is approximately:

$$\begin{bmatrix} 0.886659 \\ 0.39013 \\ 0.248264 \end{bmatrix}$$

This is not a probability vector, in fact this is a vector whose magnitude is 1 but the sum of the entries is not. So to make it a probability vector we divide through by the sum (because any nonzero multiple of an eigenvector is also an eigenvector) to get:

$$\begin{bmatrix} 0.581397 \\ 0.255815 \\ 0.162791 \end{bmatrix}$$

Which means that in the long term:

- 58.1397% of the population will end up in 1.
- 25.5815% of the population will end up in 2.
- 16.2791% of the population will end up in 3.

As per the corollary we could also have found these approximately by taking a sufficiently large value of T and looking at a column:

$$T^{1000} \approx \begin{bmatrix} 0.581395 & 0.581395 & 0.581395 \\ 0.255814 & 0.255814 & 0.255814 \\ 0.162791 & 0.162791 & 0.162791 \end{bmatrix}$$

7.3 Transition Matrices and Regularity

The definition of T being regular is that there is some power T^k for which none of the entries are zero. What does this actually mean, though, and what does it mean for T to be non-regular?

We can answer this by investigating the meaning of the entries in T^2, T^3, etc.

Let's look at T^2. If we write:

$$T = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{bmatrix}$$

First we know that the (i, j) entry in T represents the proportion of people who move from state j to state i in one interation.

The (i, j)-entry in T^2 equals:

$$t_{i1}t_{1j} + t_{i2}t_{2j} + ... + t_{in}t_{nj}$$

But what does this value represent?

Well notice that:

- $t_{i1}t_{1j}$ represents the proportion of people who move from state j to state 1 and then from state 1 to state i.
- $t_{i2}t_{2i}$ represents the proportion of people who move from state j to state 2 and then from state 2 to state i.
- ...until...
- $t_{in}t_{nj}$ represents the proportion of people who move from state j to state n and then from state n to state i.

It follows that the (i, j)-entry in T^2 represents the proportion of people who move from state j to state i in exactly two iterations.

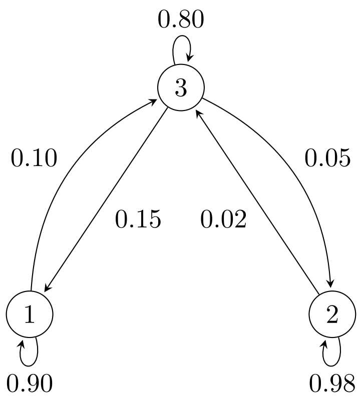
Similarly the (i, j)-entry in T^3 represents the proportion of people who move from state j to state i in exactly three iterations and in general the (i, j)-entry in T^k represents the proportion of people who move from state j to state i in exactly k iterations.

What this means is that for a transition matrix to be regular that there is some iterative step (the k value) for which it can be said that in exactly k iterations some people move from every state to every other state. Note that this is not the same as saying that the population reaches every state from each state eventually. For example if $T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ then everyone alternates states but the matrix is not regular.

However we can certainly say that if some states are not reachable from others (ever) then the transition matrix will definitely not be regular since if state i is not reachable from j (ever) then the (i, j) entry of T^k will be zero for all k.

Here some examples to flesh out these ideas:

Example 7.3. Consider:



The transition matrix for this is:

$$T = \begin{bmatrix} 0.9 & 0 & 0.15 \\ 0 & 0.98 & 0.05 \\ 0.1 & 0.02 & 0.8 \end{bmatrix}$$

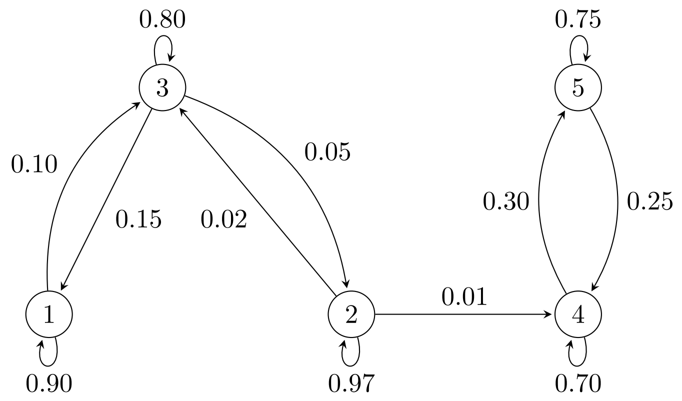
Observe the following two:

$$T^2 = \begin{bmatrix} 0.825 & 0.003 & 0.255 \\ 0.005 & 0.9614 & 0.089 \\ 0.17 & 0.0356 & 0.656 \end{bmatrix} \text{ and } T^5 = \begin{bmatrix} 0.6894 & 0.02171 & 0.4117 \\ 0.03618 & 0.9118 & 0.1662 \\ 0.2744 & 0.06646 & 0.4222 \end{bmatrix}$$

Some things we can observe:

- The matrix is regular because T^2 has no zeros.
- The $(3, 2)$ entry in T^2 is 0.0356 and indicates that in exactly two iterations 0.0356 of the population of state 2 will have moved to state 3.
- The $(3, 1)$ entry in T^5 is 0.2744 and indicates that in exactly five iterations 0.2744 of the population of state 1 will have moved to state 3.
- The $(2, 1)$ entry in T is 0 since we can't get from state 1 to state 2 in exactly one iteration but the $(2, 1)$ entry in T^2 is 0.005 since we can do it in exactly two iterations.

Example 7.4. Consider:



Some things we can observe:

- This matrix is not regular because it is impossible to ever get from states 4,5 to states 1,2,3. Consequently the (i, j) entries of T^k will be zero for $i = 1, 2, 3$ and $j = 4, 5$ for all k.
- It's certainly not possible to get from state 1 to state 5 in one iteration but it is possible in four. Consequently the $(5, 1)$ entry in T would be zero but the $(5, 1)$ entry in T^4 would not be. It would in fact be $(0.10)(0.05)(0.01)(0.30) = 0.000015$.
- To directly calculate the $(3, 1)$ entry in T^3 we look at the paths from state 1 to state 3 which are three iterations long. There are five:

  $$\begin{gathered} 1 \to 3 \to 1 \to 3 \\ 1 \to 3 \to 2 \to 3 \\ 1 \to 3 \to 3 \to 3 \\ 1 \to 1 \to 1 \to 3 \\ 1 \to 1 \to 3 \to 3 \end{gathered}$$

  Consequently the $(3, 1)$ entry in T^3 equals:

  $$\begin{aligned} &(0.10)(0.15)(0.10) \\ +&(0.10)(0.05)(0.02) \\ +&(0.10)(0.80)(0.80) \\ +&(0.90)(0.90)(0.10) \\ +&(0.90)(0.10)(0.80) \\ &= 0.2186 \end{aligned}$$

Interestingly if we write down the transition matrix for this:

$$T = \begin{bmatrix} 0.9 & 0 & 0.15 & 0 & 0 \\ 0 & 0.97 & 0.05 & 0 & 0 \\ 0.1 & 0.02 & 0.8 & 0 & 0 \\ 0 & 0.01 & 0 & 0.7 & 0.25 \\ 0 & 0 & 0 & 0.3 & 0.75 \end{bmatrix}$$

and if we find $T^k\bar{x}_0$ for any $\bar{x}_0$ and for some very large k (emulating a limit):

$$T^k\bar{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.455 \\ 0.545 \end{bmatrix}$$

So this shows (experimentally) that there is a steady-state vector to which all other states converge.

What is happening here is that in the long term everything moves out of states 1,2,3 and into 4,5 which act like their own little Markov chain and $\begin{bmatrix} 0.455 \\ 0.545 \end{bmatrix}$ is the steady state for this chain.

Example 7.5. Consider the simple example with 1 and 2 where, each year, 100% of the population moves to the other state.

The transition matrix here is:

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

which has $T^2 = I$, $T^3 = T$, etc. and is hence not regular.

Moreover any initial population will flip year by year.

If the initial population distribution is $\begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$ then the following year it will be $\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$ and then $\begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$ and so on, and this will happen (this flip) for any distribution it starts with. Basically this distribution never settles unless it begins (and stays with) $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

So there is a single steady-state vector, specifically $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$, but no other state will converge to it.

7.4 Steady State Proof for The Two-Dimensional Case

Theorem 7.4.0.1. If T is a 2×2 regular transition matrix then T has a steady-state vector $\bar{x}_*$ and moreover for any vector $\bar{x}_0$ we have $\lim\limits_{n \to \infty} T^n \bar{x}_0 = \bar{x}_*$.

Proof. A 2×2 matrix whose columns add to 1 has the form

$$T = \begin{bmatrix} a & 1-b \\ 1-a & b \end{bmatrix}$$

where $0 \leq a \leq 1$ and $0 \leq b \leq 1$.

Notice first that if $a = b = 1$ then

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and if $a = b = 0$ then

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and neither of these are regular so we can safely ignore these cases.

Calculation shows that the eigenvalue-eigenvector pairs of T are:

$$\left\{1, \begin{bmatrix} b-1 \\ a-1 \end{bmatrix}\right\} \text{ and } \left\{a+b-1, \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right\}$$

Notice that $\lambda = 1$ is an eigenvalue with eigenvector

$$\begin{bmatrix} b-1 \\ a-1 \end{bmatrix}$$

If we divide through by the sum $(a-1)+(b-1)$ (which is not zero because we've excluded $a = b = 1$) we get the probability eigenvector:

$$\begin{bmatrix} \frac{b-1}{(a-1)+(b-1)} \\ \frac{a-1}{(a-1)+(b-1)} \end{bmatrix}$$

We can see here that the sum is now 1 and each value is between 0 and 1, so we have a steady-state vector which we'll denote $\bar{x}_*$.

Next we need to show that for any probability vector $\bar{x}_0$ we have the long-term behavior $\lim_{n\to\infty} T^n\bar{x}_0 = \bar{x}_*$.

First, it follows from our eigenpairs that the diagonalization of T is:

$$T = \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & a+b-1 \end{bmatrix} \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix}^{-1}$$

and therefore:

$$\begin{aligned} \lim_{n\to\infty} T^n &= \lim_{n\to\infty} \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & a+b-1 \end{bmatrix}^n \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix} \lim_{n\to\infty} \begin{bmatrix} 1 & 0 \\ 0 & (a+b-1)^n \end{bmatrix} \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b-1 & -1 \\ a-1 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{b-1}{a+b-2} & \frac{b-1}{a+b-2} \\ \frac{a-1}{a+b-2} & \frac{a-1}{a+b-2} \end{bmatrix} \end{aligned}$$

Notice that

$$\lim_{n\to\infty} (a+b-1)^n = 0$$

because $-1 < a+b-1 < 1$ (because we've excluded $a = b = 0$ and $a = b = 1$).

Now then if $\bar{x}_0$ is any probability vector then if we write:

$$\bar{x}_0 = \begin{bmatrix} c \\ 1-c \end{bmatrix}$$

with $0 \le c \le 1$ then calculation shows that:

$$\lim_{n\to\infty} T^n\bar{x}_0 = \begin{bmatrix} \frac{b-1}{a+b-2} & \frac{b-1}{a+b-2} \\ \frac{a-1}{a+b-2} & \frac{a-1}{a+b-2} \end{bmatrix} \begin{bmatrix} c \\ 1-c \end{bmatrix} = \begin{bmatrix} \frac{b-1}{a+b-2} \\ \frac{a-1}{a+b-2} \end{bmatrix} = \bar{x}_*$$

as desired.

□

7.5 Matlab

We addressed earlier how we can find the eigenvalues and eigenvectors of a matrix but it's worth revisiting, especially since we need to get a vector which sums to 1.

Here's an example from earlier.

```
>> T = [
0.90 0.10 0.20
0.06 0.80 0.10
0.04 0.10 0.70];
```

We can find all the eigenpairs as before:

```
>> [p,d] = eig(T)
p =
    0.8867    0.8071   -0.4960
    0.3901   -0.5105   -0.3137
    0.2483   -0.2966    0.8097
d =
    1.0000         0         0
         0    0.7632         0
         0         0    0.6368
```

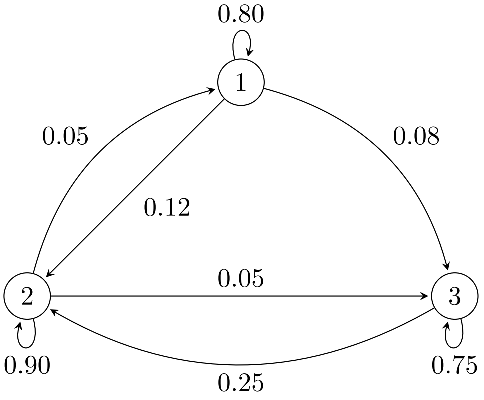
What we'd like to do is take the first column of `p`, which is the eigenvector corresponding to $\lambda = 1$, the first entry in the diagonal matrix `d`, and divide by its magnitude. Here is how we can do it in one go:

```
>> v = p(:,1)/sum(p(:,1))
v =
    0.5814
    0.2558
    0.1628
```

This notation might be familiar. Writing `p(:,1)` takes every row in the first column of `p`. So what we're doing is taking that vector and dividing it by its own sum.

7.6 Exercises

Exercise 7.1. Consider the following population movement diagram.



(a) Write down the corresponding transition matrix T.

(b) What does the fact that T is not symmetric say about the population movements?

(c) If the population distribution starts at $\bar{x}_0 = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.2 \end{bmatrix}$ what will it be after one iteration? How about after two iterations?

(d) Calculate the steady state distribution $\bar{x}^*$.

(e) Using software (your choice) find the smallest value of k such that $T^k\bar{x}_0$ agrees with $\bar{x}^*$ to four decimal places.

Exercise 7.2. Given the transition matrix:

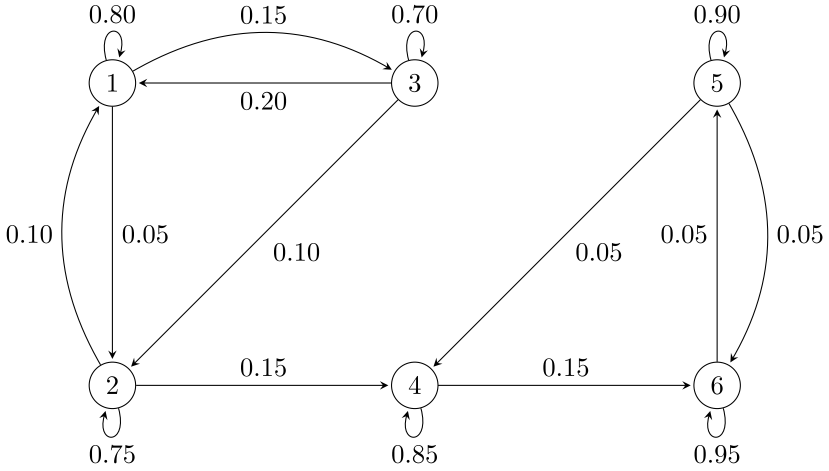
$$T = \begin{bmatrix} 0.9 & 0 & 0.05 & 0 & 0.15 & 0.05 \\ 0 & 0.8 & 0.05 & 0.2 & 0.05 & 0.05 \\ 0.01 & 0.1 & 0.5 & 0 & 0 & 0.05 \\ 0.02 & 0.1 & 0.1 & 0.75 & 0 & 0.05 \\ 0.07 & 0 & 0.1 & 0 & 0.7 & 0.05 \\ 0 & 0 & 0.2 & 0.05 & 0.1 & 0.75 \end{bmatrix}$$

(a) Draw the corresponding population movement diagram.

(b) Is this matrix regular? Justify.

Exercise 7.3. Draw a population movement diagram whose transition matrix is 3 × 3, regular and symmetric. Also give the transition matrix.

Exercise 7.4. Given the following population movement diagram:



(a) Write down the transition matrix T for this.

(b) Find each of the following without actually taking powers of T:

  (a) The $(3, 2)$ entry of T.

  (b) The $(3, 2)$ entry of T^2.

  (c) The $(1, 3)$ entry of T^2. Be careful!

  (d) The smallest k such that the $(5, 3)$ entry of T^k is nonzeo and what that value is.

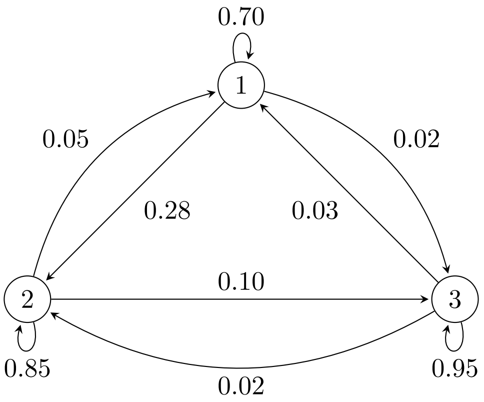
  (e) All (i, j) such that the (i, j) entry of T^k is zero for all k.

  (f) Explain intuitively what will happen in the long term to any initial population distribution. Justify intuitively. This question can be answered to various degrees of detail, the most basic being - in which area(s) do the populations tend to eventually move and why?

Exercise 7.5. Find a population movement diagram and the corresponding transition matrix T such that $T^7 = I$ but $T^k \neq I$ for $k < 7$.

Exercise 7.6. We know that a regular transition matrix has a steady state to which all states converge and that a nonregular transition matrix does not necessarily have to. Give an example of a population movement diagram and the corresponding transition matrix T which is 5×5 and nonregular but which does have a steady state to which all states converge and find this steady state.

Exercise 7.7. Consider the following population movement diagram:



(a) Write down the corresponding transition matrix T.

(b) If the population distribution starts at $\bar{x}_0 = \begin{bmatrix} 0.1 \\ 0 \\ 0.9 \end{bmatrix}$ what will it be after one iteration? How about after two iterations? How about after five iterations?

(c) Find the steady state distribution $\bar{x}^*$.

(d) Find the smallest value of k such that $T^k \bar{x}_0$ agrees with $\bar{x}^*$ to four decimal places.

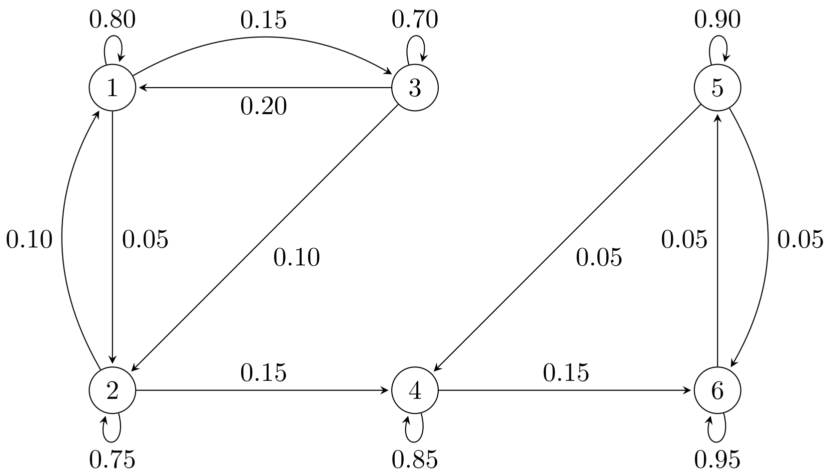
Exercise 7.8. Given the transition matrix:

$$T = \begin{bmatrix} 0.9 & 0 & 0.05 & 0 & 0.15 & 0.05 \\ 0 & 0.8 & 0.05 & 0.2 & 0.05 & 0.05 \\ 0.01 & 0.1 & 0.5 & 0 & 0 & 0.05 \\ 0.02 & 0.1 & 0.1 & 0.75 & 0 & 0.05 \\ 0.07 & 0 & 0.1 & 0 & 0.7 & 0.05 \\ 0 & 0 & 0.2 & 0.05 & 0.1 & 0.75 \end{bmatrix}$$

(a) Draw the corresponding population movement diagram.

(b) Find T^2. Is T regular?

Exercise 7.9. Consider the following population movement diagram:



(a) Without doing any calculation what you think will happen to the population in the long term? Justify informally.

(b) Write down the corresponding transition matrix T.

(c) Even though T is not regular (don't prove this) it still has an eigenvalue of $\lambda = 1$ with a corresponding steady state. Find the corresponding probability eigenvector.

(d) How does your answer to (c) fit with your intuition in (a)?

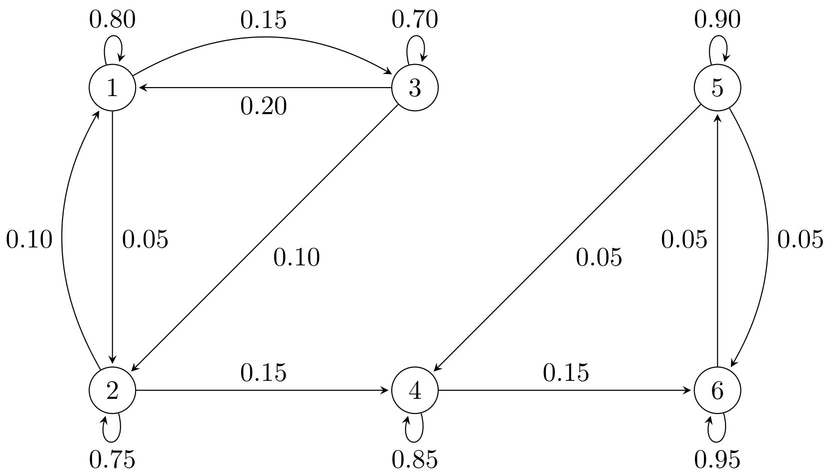
Exercise 7.10. Find all eigenvalues of the following transition matrix:

$$T = \begin{bmatrix} a & 0 & 1-c \\ 1-a & b & 0 \\ 0 & 1-b & c \end{bmatrix}$$

Exercise 7.11. It seems like $T^k\bar{x}_0$ gets very close to $\bar{x}^*$ very quickly but this doesn't have to be the case. Find an example of a 2×2 regular transition matrix and an initial state $\bar{x}_0$ such that all of the entries in $\bar{x}_{1000} = T^{1000}\bar{x}_0$ still differ from those in $\bar{x}^*$ at the first decimal place.

Exercise 7.12. Write down an example of a 3×3 symmetric and regular transition matrix and draw the corresponding population movement diagram.

Exercise 7.13. Given the following population movement diagram:



(a) Write down the transition matrix T for this.

(b) Find each of the following without actually taking powers of T:

(i) The $(3,2)$ entry of T^2.

(ii) The $(1,3)$ entry of T^2. Be careful!

(iii) The smallest k such that the $(5,3)$ entry of T^k is nonzeo and what that value is.

(iv) All (i,j) such that the (i,j) entry of T^k is zero for all k.

(c) Is T regular? Explain.

Exercise 7.14. Find a population movement diagram and the corresponding transition matrix T such that $T^7 = I$ but $T^k \neq I$ for $k < 7$.

Exercise 7.15. Give an example of a non-regular transition matrix with no steady state vector and an example of a non-regular transition matrix with a steady state.

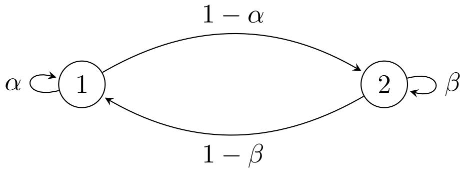
Exercise 7.16. Give an example of a transition matrix whose corresponding population movement diagram is separated into two non-connected components.

Exercise 7.17. Give an example of a transition matrix whose corresponding population movement diagram is separated into two non-connected components, one of which has a steady-state to which all states converge and one of which does not.

Exercise 7.18. Define the following terms: Probability vector, transition matrix, regular transition matrix.

Exercise 7.19. For any given n is it possible to construct a transition matrix T such that some entry in T^k is 0 for $k = 1, ..., n-1$ but that entry is nonzero in T^n? Explain why or why not.

Exercise 7.20. Consider:



(a) Write down the corresponding transition matrix T.

(b) Assuming that $0 < \alpha < 1$ and $0 < \beta < 1$, find the limiting steady state vector for T.

(c) If remove the above restrictions on α and β does T have to have a limiting steady state vector? If it does not have to, can it? Justify.

Exercise 7.21. Suppose that T is regular and so T^k has all nonzero entries for some k. Explain why T^j has all nonzero entries for $j > k$.

Chapter 8

Google Pagerank

Contents

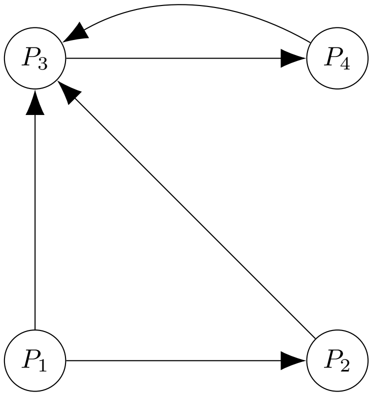
8.1 Introduction

One of the principal requirements that Google has to deal with is ranking web pages. A web page should be ranked higher by some sort of criteria. So how can we go about doing this? Given a web page, the basic idea might be to look at how many pages are linking *to* this page; The more the better. However then we have to appreciate how important *those* pages are, since being linked to by a useless page is not as important as being linked to by an important one, and so the problem goes back and back.

8.2 Relationship to Markov Chains

Let's examine a very basic internet, see how it connects to Markov Chains, and how the Google Pagerank method works.

Example 8.1. Suppose the internet consists of only four pages, P_1, P_2, P_3, P_4, linked as follows, where, for example, an arrow from P_1 to P_3 indicates a link from P_1 to P_3.



Notice that this looks very much like the diagram of a Markov Chain. If that's the case why don't we just assign probability values to the directions like we did with population diagrams? For example if a web page has two outbound links we could assign each a probability value of 0.5 and so on? If we did this then the steady-state vector would correspond to where a web surfer would end up in the long term, and this seems like a reasonable way to assign value to web pages.

One obvious problem is that a web page may have no outbound links. If that's the case we wouldn't know what to assign for the probabilities.

Another obvious problem is that this method doesn't really act like a web surfer. Web surfers don't just follow links, they also jump to other pages independently of whether they were.

The Google Pagerank (GP) algorithm takes the following approach:

(1) We assume that a Random Websurfer (RW) starts at some page.

(2) If the page has outbound links then there is an 85% chance that WR chooses one of those links and those links are equally likely. There is a 15% chance that RW chooses a page at random from all possible pages.

(3) If the page has no outbound links then there is a 100% chance that RW chooses a page at random from all possible pages.

(4) RW will continue to do this forever.

After reading this it becomes fairly clear that this is exactly a Markov Chain. The picture above is not an exact representation of the movement of RW because we need to take into account the 15% chance that RW randomly chooses a page.

We could connect every page to every other page in the diagram but that would be a bit silly so instead we just recognize that the connections are there.

Example 8.1 Revisited.

- There is a weight of 0.15/4 connecting P_i to P_j for all i, j.
- There is an additional weight of $0.85(1/n)$ connecting P_i to P_j provided that P_i links to P_j and that P_i links to n pages total.

This is particularly easy to see in terms of two separate matrices:

$$T = \frac{0.15}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + 0.85\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.0375 & 0.0375 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.0375 & 0.0375 \\ 0.4625 & 0.8875 & 0.0375 & 0.8875 \\ 0.0375 & 0.0375 & 0.8875 & 0.0375 \end{bmatrix}$$

Check your sanity - this should be a transition matrix. Not only that but it's a regular transition matrix because the first part of the sum forces all entries of T^1 to be nonzero. Consequently it obeys our theorem, having an eigenvalue of $\lambda = 1$ and a corresponding probability eigenvector.

The corresponding probability eigenvector is the *ranking vector* :

$$\begin{bmatrix} 0.0375 \\ 0.0534 \\ 0.4711 \\ 0.4379 \end{bmatrix}$$

We therefore rank the pages according to the probability that RW will end up there in the long run:

- P_1 has pagerank 0.0375
- P_2 has pagerank 0.0534
- P_3 has pagerank 0.4711
- P_4 has pagerank 0.4379

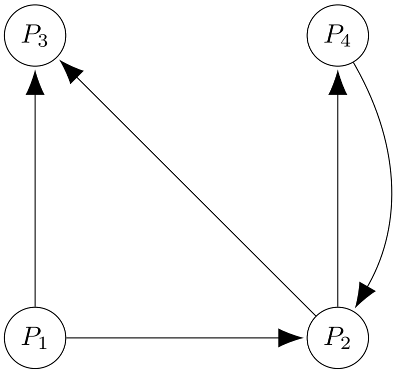
Think about why this makes sense in the context of the picture.

- The page P_3 is important because lots of pages link to it.
- The page P_4 only has P_3 linking to it but P_3 itself is important, so P_4 is too. Not quite as important though.

- Page P_1 seems least important since no other pages link to it.
- Page P_2 is only slightly more important than P_1 because it does have one page linking to it, that being P_1, but P_1 is not that important.

Here is a second example in which a page has no outbound link.

Example 8.2. Consider the internet:



Here we have:

$$T = \frac{0.15}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + 0.85\begin{bmatrix} 0 & 0 & 1/4 & 0 \\ 1/2 & 0 & 1/4 & 0 \\ 1/2 & 1/2 & 1/4 & 1 \\ 0 & 1/2 & 1/4 & 0 \end{bmatrix}$$

Notice the column of 1/4. Since page 3 has no outbound links there is a 100% chance that RW will choose a page at random. Since 15% of that is accounted for in the first matrix we simply account for the other 85% in the second one.

We find the corresponding probability eigenvector to be:

$$\begin{bmatrix} 0.0999 \\ 0.3557 \\ 0.2435 \\ 0.2510 \end{bmatrix}$$

- P_1 has pagerank 0.0999
- P_2 has pagerank 0.3557
- P_3 has pagerank 0.2435
- P_4 has pagerank 0.2510

8.3 General Pagerank Matrix

In general then for an internet with n pages we have:

$$T = \frac{0.15}{n}\,[n \times n \text{ matrix of 1s}] + 0.85\,[\bar{v}_1\ \ \bar{v}_n\ \ \ldots\ \ \bar{v}_n]$$

Where $\bar{v}_i$ is given by:

- If page i has k outbound links then the j^{th} entry of $\bar{v}_i$ equals $1/k$ if page i has an outbound link to page j.
- If page i has no outbound links then every entry of $\bar{v}_i$ equals $1/n$.

8.4 Scalability

It's important that we understand that we never need to find the eigenvalues since we know that $\lambda = 1$ is there. This is good because finding the eigenvalues of an $n \times n$ matrix requires finding the roots of a polynomial of degree n and there is no closed formula for the roots of a polynomial of degree 5 or more.

Knowing that $\lambda = 1$ is an eigenvalue then requires us "only" to solve a system of n equations where n is the number of web pages on the internet.

8.5 Matlab

There's nothing particularly new related to Matlab in this chapter but it's worth noting that we can write a function m-file which creates the matrix for us. This is a slightly more sophisticated use of Matlab. The idea is that we'll first create a vector which indicates the links. In the following each row is a link from the first page to the second:

```
>> links = [1,2;1,3;2,3;3,4;4,3]
links =
     1     2
     1     3
     2     3
     3     4
     4     3
```

We also create a scalar containing the total number of pages:

```
>> pagecount = 4;
```

And then the following Matlab function m-file does the job:

```
function m = creategpmatrix(links,pagecount)
  % Usage:
  % links = [1,2;1,3;2,3;3,4;4,3];
  % pagecount = 4;
  % creategpmatrix(links,pagecount)
  part1 = ones(pagecount,pagecount);
  part2 = zeros(pagecount,pagecount);
  linksize = size(links);
  numlinks = linksize(1);
  for i = [1:numlinks]
    part2(links(i,2),links(i,1)) = 1;
  end
  for i = [1:pagecount]
    if (sum(part2(:,i)) > 0)
      part2(:,i) = part2(:,i)/sum(part2(:,i));
    else
      part2(:,i) = ones(pagecount,1)/pagecount;
    end
  end
  m = 0.15/pagecount*part1 + 0.85*part2;
end
```

As follows:

```
>> creategpmatrix(links,pagecount)
ans =
    0.0375    0.0375    0.0375    0.0375
    0.4625    0.0375    0.0375    0.0375
    0.4625    0.8875    0.0375    0.8875
    0.0375    0.0375    0.8875    0.0375
```

If you're curious about what's going on in the function m-file, let's walk through it.

First the command `ones(pagecount,pagecount)` creates a square matrix filled with 1s and the command `zeros(pagecount,pagecount)` creates a square matrix filled with 0s, both of the appropriate size. Next the number of links is calculated.

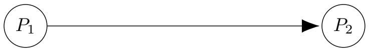
The first `for` loop goes through the links, here `length(links)` is the number of rows in the `links` matrix, hence the number of outbound links (since each link is a row). For each link from P_i to P_j (which are found in `link(i,1)` and `link(i,2)`) we place a 1 in the (j, i) entry of `part2`.

The second `for` loop goes through each column of `part2`. If the sum is nonzero, meaning there are outbound links, then it divides each column by its sum. If the sum is zero, meaning there are no outbound links, then it assigns each entry in the column to be the same and add to 1, which pretends that the page is linked to every other page equally.

Finally we assign the matrix `0.15/pagecount*part1 + 0.85*part2` to return.

8.6 Exercises

Exercise 8.1. Consider this mini-internet:

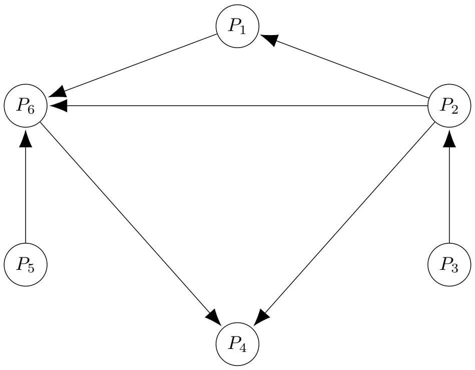


(a) Try to rank the pages in order of importance without doing any calculation.

(b) Find the pagerank of each of the pages.

(c) If there are any disparities between your answer to (a) and (b) explain (if you can) what the cause of this disparity might be.

Exercise 8.2. Consider this mini-internet:



(a) Try to rank the pages in order of importance without doing any calculation.

(b) Find the pagerank of each of the pages.

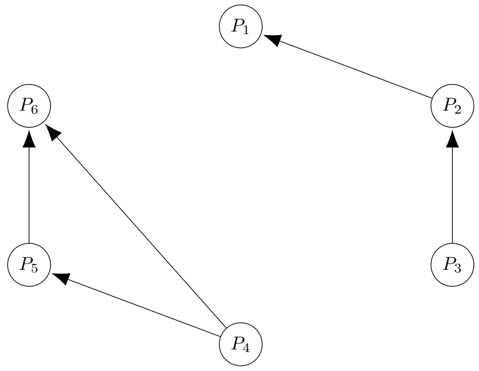
(c) If there are any disparities between your answer to (a) and (b) explain (if you can) what the cause of this disparity might be.

Exercise 8.3. Suppose that the rule that there is a 15% probability that RW jumps to a random page were removed, and instead the full 100% (instead of 85%) from each node were distributed across all outbound links. If there are no outbound links then there is still a 100% probability that RW jumps to a random page. This could lead to the possiblility that T is not regular.

(a) Give an example of an internet with a non-regular T such that there is no $\bar{x}^*$ such that $T^k\bar{x}_0$ converges to $\bar{x}^*$ for all $\bar{x}_0$.

(b) Give an example of an internet with a non-regular T such that there is some $\bar{x}^*$ such that $T^k\bar{x}_0$ converges to $\bar{x}^*$ for all $\bar{x}_0$ but that this causes serious problems with the ranking of the pages. Hint: Can you design an internet where some of the pages will end up with a pagerank of zero?

Exercise 8.4. The Google Pagerank algorithm works even if the internet is disconnected. Consider this example:



Find the pagerank of each of the pages.

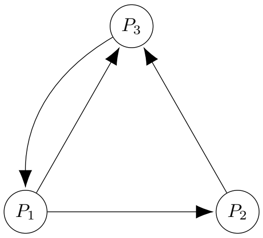
Exercise 8.5. Given the transition matrix T we generally find the pagerank vector by solving the eigenvector equation $(A - I)\bar{x} = \bar{0}$, meaning we find the eigenvector corresponding to the eigenvalue $\lambda = 1$. However it's also possible simply to pick an arbitrary $\bar{x}_0$ and then find $T^k\bar{x}_0$ for successive values of k until successive entries of $T^k\bar{x}_0$ all differ by a predetermined number. Starting with $\bar{x}_0 = [1, 0, 0, 0, 0, 0]^T$ and using the T from the previous problem, find the smallest k so that all entries in $T^k\bar{x}_0$ and $T^{k-1}\bar{x}_0$ are equal up to and including the fourth decimal place.

Exercise 8.6. In an internet with n pages all of which link to one another it makes sense that all of the pages have the same pagerank. Show that this is the case - find the matrix T and show that the vector with all entries equal (and adding to 1) is an eigenvector corresponding to eigenvalue $\lambda = 1$.

Exercise 8.7. A valid question is whether the 85/15 split has an impact not just on the pagerank but on the order of the pages in terms of ranking. For example if we used 60/40 instead would a higher ranked page using 85/15 still be higher ranked using 60/40.

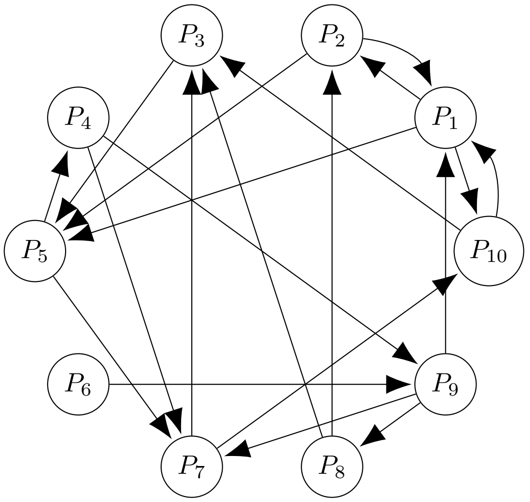
(a) Justify informally why it seems reasonable that the order of the pages in terms of ranking would not be affected.

(b) Test this assumption on the following internet by finding the pageranks using both 85/15 and 60/40 splits.



(c) The above internet can also be analyzed with a 1/0 split. Find the pageranks using this split.

Exercise 8.8. Find the pagerank of the pages in the following internet. You will definitely want to use technology for this!



Exercise 8.9. Explain why having outbound links on your webpage will not affect your Google Pagerank.

Exercise 8.10. Write down the transition matrix for an internet with n pages for which the only links are from page 1 to page 2, page 2 to page 3, ... page $n-1$ to page n.

Exercise 8.11. Why does the Google Pagerank produce more reasonable results than simply assigning a page a ranking in accordance with the number of pages that link to it?

Chapter 9

Singular Value Decomposition

Contents

9.1 Introduction

Factoring a matrix means writing the matrix as a product of other matrices. For example if we have a matrix A and we manage to write it as $A = BC$ for some B and C then we've factored it into a product of two other matrices.

There are many ways to factor a matrix and many of them are extremely useful. For example if a matrix A is diagonalizable then we can write the matrix as $A = PDP^{-1}$. This is useful because the entries in D are the eigenvalues and the columns of P are the eigenvectors.

Another really useful factorization of a matrix is the *singular value decomposition* which is a way of factoring a matrix which is used in areas like data compression, matrix approximation, pseudoinverses, signal analysis, handwriting and facial recognition, the list goes on.

In this chapter we define the *singular value decomposition* and see what mathematical properties it has.

9.2 Definitions

Definition 9.2.0.1. For any $m \times n$ real matrix A the *singular value decomposition* (SVD) of A is a factorization of A as:

$$A = U\Sigma V^T$$

where:

- U is an $m \times m$ orthogonal matrix.
- Σ is an $m \times n$ rectangular diagonal matrix.
- V is an $n \times n$ orthogonal matrix.

Just to be sure everything's clear, here are some auxilliary definitions:

Definition 9.2.0.2. An $n \times n$ matrix is *orthogonal* if its columns are unit vectors all perpendicular to one another. This is equivalent to saying that the columns form an orthonormal basis of $\mathbb{R}^n$.

Definition 9.2.0.3. An $m \times n$ *rectangular diagonal matrix* (which may or may not be square) is a matrix in which the only nonzero entries can be in the $(1, 1)$, $(2, 2)$, ... positions. If there happen to be more rows than columns then the additional rows are all zeros and likewise if there are more columns than rows.

9.3 Constructing the SVD

9.3.1 Preliminaries

Theorem 9.3.1.1. The following hold:

(a) An $n \times n$ symmetrix matrix has n unit eigenvectors all perpendicular to one another, hence those unit eigenvectors, when place in a matrix, produce an orthogonal matrix.

(b) For any matrix A the matrices A^TA and AA^T are symmetric.

(c) For any matrix A the matrices AA^T and A^TA share the same nonzero eigenvalues.

(d) For any matrix A the eigenvalues for AA^T and A^TA are all nonnegative.

Proof. We have:

(a) This follows from the Spectral Theorem, proof omitted.

(b) Easy, since $(AA^T)^T = (A^T)^T A^T = AA^T$ and $(A^T A)^T = A^T(A^T)^T = A^T A$.

(c) To see this suppose $\lambda \neq 0$ is an eigenvalue of AA^T with eigenvector $\bar{v} \neq 0$. Then $AA^T\bar{v} = \lambda\bar{v}$ and so $A^T AA^T\bar{v} = A^T\lambda\bar{v}$ and so $A^T A(A^T\bar{v}) = \lambda(A^T\bar{v})$. Thus $\lambda \neq 0$ is also an eigenvalue of $A^T A$ with eigenvector $A^T\bar{v}$. Notice that $A^T\bar{v} \neq 0$ since otherwise $AA^T\bar{v} = \lambda\bar{v}$ reduces to $\bar{0} = \lambda\bar{v}$ which is impossible because $\lambda \neq 0$ and $\bar{v} \neq \bar{0}$.

A similar argument shows that every nonzero eigenvalue of $A^T A$ is an eigenvalue of AA^T. The value 0 may be an eigenvalue of neither, both or just one of $A^T A$ and AA^T.

(d) To see this for $A^T A$ (and hence for AA^T) note if $\{\bar{v}_1, ..., \bar{v}_n\}$ is an orthonormal basis for $\mathbb{R}^n$ consisting of the eigenvectors of $A^T A$ and if $\lambda_1, ..., \lambda_n$ are the associated eigenvalues. Then for each i:

$$||A\bar{v}_i||^2 = (A\bar{v}_i)^T A\bar{v}_i = \bar{v}_i^T A^T A\bar{v}_i = \bar{v}_i^T \lambda_i \bar{v}_i = \lambda_i \bar{v}_i^T \bar{v}_i = \lambda_i$$

□

9.3.2 Obtaining the Factors

Given an $m \times n$ matrix A, a singular value decomposition $A = U\Sigma V^T$ may be obtained by assigning the following:

- The matrix U is $m \times m$ and the columns of U are an orthonormal basis of $\mathbb{R}^m$ consisting of unit eigenvectors of AA^T. These are the *left singular vectors* of A. This follows from Theorem 9.3.1.1 (a) and (b)
- The matrix Σ is $m \times n$ and the diagonal entries of Σ are the *singular values* of A; these are the square roots of the shared eigenvalues of $A^T A$ and AA^T. This follows from Theorem 9.3.1.1 (c) and (d)
- The matrix V is $n \times n$ and the columns of V are an orthonormal basis of $\mathbb{R}^n$ consisting of unit eigenvectors of $A^T A$. These are the *right singular vectors* of A. This follows from Theorem 9.3.1.1 (a) and (b)
- Typically the singular values in Σ are arranged in nonincreasing order. The columns/eigenvectors in each of U and V must be ordered to match the corresponding singular values in Σ.

Example 9.1. For example if

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

(a) We calculate:

$$AA^T = \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix}$$

which has eigenvalue and unit eigenvector pairs (these are the left singular vectors of A):

$$\left(6, \begin{bmatrix} -0.89 \\ -0.45 \end{bmatrix}\right), \left(1, \begin{bmatrix} 0.45 \\ -0.89 \end{bmatrix}\right)$$

(b) We calculate:

$$A^T A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 5 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

which has eigenvalue and unit eigenvector pairs (these are the right singular vectors of A):

$$\left(6, \begin{bmatrix} -0.37 \\ -0.91 \\ 0.18 \end{bmatrix}\right), \left(1, \begin{bmatrix} -0.45 \\ 0 \\ -0.89 \end{bmatrix}\right), \left(0, \begin{bmatrix} -0.82 \\ 0.41 \\ 0.41 \end{bmatrix}\right)$$

(c) Notice that these two have the same nonzero eigenvalues, all of which are positive. The singular values are the square roots of these:

$$\sqrt{\{6, 1\}} = \{2.45, 1\}$$

(d) Consequently $A = U\Sigma V^T$ where U is 2×2 and Σ is 2×3 and V is 3×3 with:

$$\begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \end{bmatrix} = \underbrace{\begin{bmatrix} -0.89 & -0.45 \\ -0.45 & 0.89 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 2.45 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} -0.37 & -0.45 & -0.82 \\ -0.91 & 0 & 0.41 \\ 0.18 & -0.89 & 0.41 \end{bmatrix}^T}_{V^T}$$

Example 9.2. For example if

$$A = \begin{bmatrix} 1.0 & 2.0 & 0 \\ 3.0 & 1.0 & 4.0 \\ 1.0 & 1.0 & -1.0 \\ 0 & 1.0 & 2.0 \end{bmatrix}$$

(a) We calculate:

$$AA^T = \begin{bmatrix} 5.0 & 5.0 & 3.0 & 2.0 \\ 5.0 & 26.0 & 0 & 9.0 \\ 3.0 & 0 & 3.0 & -1.0 \\ 2.0 & 9.0 & -1.0 & 5.0 \end{bmatrix}$$

which has eigenvalue and unit eigenvector pairs (these are the left singular vectors of A):

$$\left(30.4702, \begin{bmatrix} -0.2079 \\ -0.9171 \\ -0.0103 \\ -0.3400 \end{bmatrix}\right), \left(6.5221, \begin{bmatrix} 0.7246 \\ -0.1139 \\ 0.6615 \\ -0.1560 \end{bmatrix}\right),$$

$$\left(2.0078, \begin{bmatrix} -0.3584 \\ 0.3788 \\ 0.2666 \\ -0.8106 \end{bmatrix}\right), \left(0, \begin{bmatrix} -0.5507 \\ -0.501 \\ 0.7009 \\ 0.4506 \end{bmatrix}\right)$$

(b) We calculate

$$A^TA = \begin{bmatrix} 11.0 & 6.0 & 11.0 \\ 6.0 & 7.0 & 5.0 \\ 11.0 & 5.0 & 21.0 \end{bmatrix}$$

which has eigenvalue and unit eigenvector pairs (these are the right singular vectors of A):

$$\left(30.4702, \begin{bmatrix} -0.5380 \\ -0.3049 \\ -0.7859 \end{bmatrix}\right), \left(6.5221, \begin{bmatrix} 0.4090 \\ 0.7208 \\ -0.5596 \end{bmatrix}\right), \left(2.0078, \begin{bmatrix} 0.7371 \\ -0.6225 \\ -0.2631 \end{bmatrix}\right)$$

(c) Notice that these two have the same nonzero eigenvalues, all of which are positive. The singular values are the square roots of these:

$$\sqrt{\{30.4702, 6.5221, 2.0078\}} = \{5.52, 2.554, 1.417\}$$

(d) Consequently

$$\begin{bmatrix} 1.0 & 2.0 & 0 \\ 3.0 & 1.0 & 4.0 \\ 1.0 & 1.0 & -1.0 \\ 0 & 1.0 & 2.0 \end{bmatrix} = U\Sigma V^T$$

where U is 4×4 and Σ is 4×3 and V is 3×3 with:

$$U = \begin{bmatrix} -0.2079 & 0.7246 & -0.3584 & -0.5507 \\ -0.9171 & -0.1139 & 0.3788 & -0.0501 \\ -0.0103 & 0.6615 & 0.2666 & 0.7009 \\ -0.3400 & -0.1560 & -0.8106 & 0.4506 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 5.52 & 0 & 0 \\ 0 & 2.554 & 0 \\ 0 & 0 & 1.417 \\ 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.5380 & 0.4090 & 0.7371 \\ -0.3049 & 0.7208 & -0.6225 \\ -0.7859 & -0.5596 & -0.2631 \end{bmatrix}$$

Side note: It may seem strange that we've kept the fourth column of U which corresponds to the eigenvalue 0, which we sort of aren't caring about. It also might seem more sensible to want Σ to be square. In fact if we were to simply throw out the fourth column of U and the fourth row of Σ we still have $A = U\Sigma V^T$. Some sources do this. In this book we don't, if only because leaving it in is more standard and is what software like Matlab returns.

9.4 Matlab

Calculations of the Singular Value Decomposition can be easily done in Matlab:

```
>> A=[
1 2 0
3 1 4
1 1 -1
0 1 2
]
A =
     1     2     0
     3     1     4
     1     1    -1
     0     1     2
>> [U,S,V] = svd(A)
U =
   -0.2079    0.7246   -0.3584   -0.5507
   -0.9171   -0.1139    0.3788   -0.0501
   -0.0103    0.6615    0.2666    0.7009
   -0.3400   -0.1560   -0.8106    0.4506
S =
    5.5200         0         0
         0    2.5538         0
         0         0    1.4170
         0         0         0
V =
   -0.5380    0.4090    0.7371
   -0.3049    0.7208   -0.6225
   -0.7859   -0.5596   -0.2631
```

9.5 Exercises

Exercise 9.1. Find the singular value decomposition of each of the following matrices. First do this by computing both AA^T and A^TA, finding the eigenvalue/eigenvector pairs of each, finding the corresponding singular values, and putting the results together. Then check your answer via technology.

(a) $A = \begin{bmatrix} 1.0 & 2.0 & -3.0 \\ 0 & 1.0 & 1.0 \\ 1.0 & 2.0 & 5.0 \\ -1.0 & 0 & 2.0 \end{bmatrix}$

(b) $A = \begin{bmatrix} -1.0 & 0 & 2.0 & 2.0 & 2.0 \\ 0 & 2.0 & 3.0 & 0 & 1.0 \\ 1.0 & 2.0 & -2.0 & 1.0 & 2.0 \end{bmatrix}$

(c) $A = \begin{bmatrix} 1.0 & 2.0 & -3.0 \\ 0 & 1.0 & 1.0 \\ 1.0 & 2.0 & 5.0 \\ -1.0 & 0 & 2.0 \end{bmatrix}$

(d) $A = \begin{bmatrix} 0.1 & 0.2 & 0.9 & 0.3 \\ 0.9 & 0.2 & 0 & 0.2 \\ 0.2 & 0.2 & 0.3 & 0.1 \\ 0 & 0.3 & 0.7 & 0.6 \end{bmatrix}$

Chapter 10

Matrix Approximation

Contents

10.1 Introduction

For almost all of our uses of the singular value decomposition we'll be thinking of an $m \times n$ matrix as made up of columns where each column represents something useful. For now just think about the matrix as containing n points in $\mathbb{R}^m$.

Thinking this way, it turns out that the SVD contains a lot of useful information about the points.

Before we proceed we should note that we will only need the matrices U and Σ but not V so typically we'll just write V^T in our SVD instead of actually writing V explicitly.

The general approach will be as follows:

(I) Get some geometric inspiration for what the left-singular vectors (the columns of U) represent.

(II) Get some algebraic evidence and figure out what the singular values (the entries in Σ) represent.

(III) Put together a conclusion and solidify this with examples.

Before proceeding, two definitions:

Definition 10.1.0.1. The *variance* of a set of vectors equals the sum of the squares of the norms of the vectors.

Definition 10.1.0.2. The *norm* of a matrix, denoted $||A||$, equals the square root of the sum of the squares of the entries, meaning it equals the square root of the variance of the columns of the matrix.

It follows that that the norm of a matrix equals the square root of the variance of the columns.

It also follows that two matrices are close to one another (in every entry) if the norm of their difference is small.

10.2 Geometric Inspiration for U

Let's start of with an example:

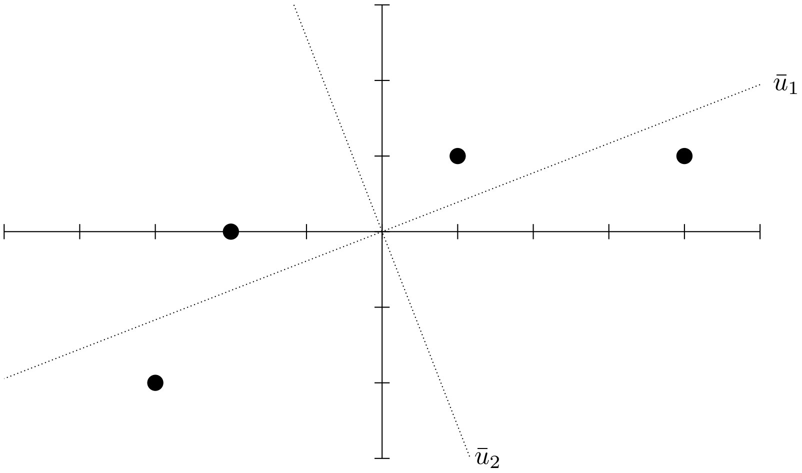
Example 10.1. Consider the matrix:

$$A = \begin{bmatrix} 4 & 1 & -2 & -3 \\ 1 & 1 & 0 & -2 \end{bmatrix}$$

This matrix has the following SVD:

$$A = \begin{bmatrix} -0.9320 & -0.3625 \\ -0.3625 & 0.9320 \end{bmatrix} \begin{bmatrix} 5.855 & 0 & 0 & 0 \\ 0 & 1.312 & 0 & 0 \end{bmatrix} V^T$$

If we plot the columns of A as points along with the lines generated by each of the the columns of U (call these $\bar{u}_1$ and $\bar{u}_2$) we see:



What do we see? It looks like the data is really spread out in the $\bar{u}_1$ direction and not so spread out in the $\bar{u}_2$ direction.

What is this example suggesting? Well, it looks like:

(a) Column $\bar{u}_1$ (associated to s_1, the larger singular value) is indicating the direction in which the columns of A are most spread out.

(b) Column $\bar{u}_2$ (associated to s_2, the smaller singular value) the one associated to the second largest singular value, is indicating the direction perpendicular to $\bar{u}_1$ in which the columns of A are second most spread out.

It may seem reasonable to suggest that if there are more than two columns in U that they tell us how the columns of A are most spread out in perpendicular directions of decreasing importance.

10.3 Algebraic Evidence

If we look at the SVD of an $m \times n$ matrix A:

$$A = U\Sigma V^T$$

where the dimensions are $m \times m$, $m \times n$ (with k singular values $s_1, ..., s_k$) and $n \times n$ respectively then a relatively straightforward calculation shows that the the columns of A are calculated by:

$$\begin{aligned}
\bar{a}_1 &= s_1 v_{11}\bar{u}_1 + s_2 v_{12}\bar{u}_2 + ... + s_k v_{1k}\bar{u}_k \\
\bar{a}_2 &= s_1 v_{21}\bar{u}_1 + s_2 v_{22}\bar{u}_2 + ... + s_k v_{2k}\bar{u}_k \\
\vdots \quad & \quad \vdots \\
\bar{a}_n &= s_1 v_{n1}\bar{u}_1 + s_2 v_{n2}\bar{u}_2 + ... + s_k v_{nk}\bar{u}_k
\end{aligned}$$

First, this tell us that each column of A is built out of the columns in U, which makes sense since the columns of U form a basis for $\mathbb{R}^n$.

However since $s_1 \geq s_2 \geq ... \geq s_k \geq 0$ this is suggesting exactly what we've seen geometrically, that $\bar{u}_1$ is most important, $\bar{u}_2$ next most and so on.

In addition this tells us that if we create Σ' by taking Σ and replacing some of the s_i by 0, and if we then recalculate $A' = U\Sigma' V^T$, the result will be the orthogonal projection of the columns of A onto the columns of U corresponding to the s_i which remain because essentially we are ignoring the directions associated to the s_i we eliminated.

To help clarify let's revisit our initial example:

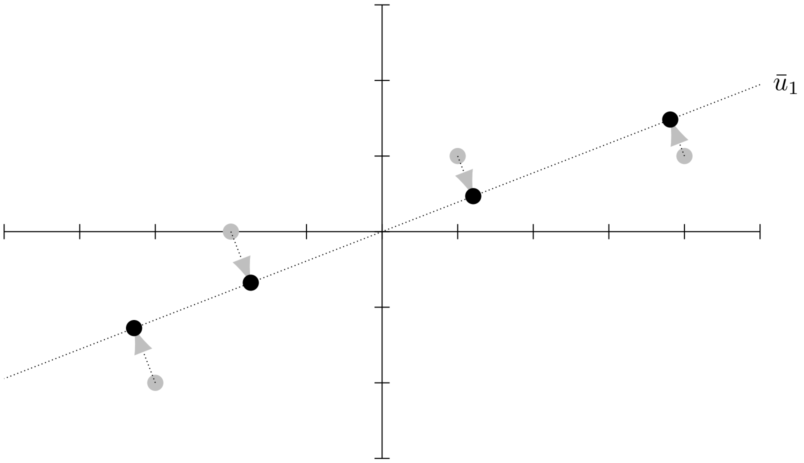
Example 10.2.

$$A = \begin{bmatrix} 4 & 1 & -2 & -3 \\ 1 & 1 & 0 & -2 \end{bmatrix} = \begin{bmatrix} -0.9320 & -0.3625 \\ -0.3625 & 0.9320 \end{bmatrix} \begin{bmatrix} 5.855 & 0 & 0 & 0 \\ 0 & 1.312 & 0 & 0 \end{bmatrix} V^T$$

If we set $s_2 = 0$ and recalculate the matrix product:

$$\begin{bmatrix} -0.9320 & -0.3625 \\ -0.3625 & 0.9320 \end{bmatrix} \begin{bmatrix} 5.855 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} V^T = \begin{bmatrix} 3.812 & 1.2060 & -1.7370 & -3.281 \\ 1.483 & 0.4693 & -0.6757 & -1.276 \end{bmatrix}$$

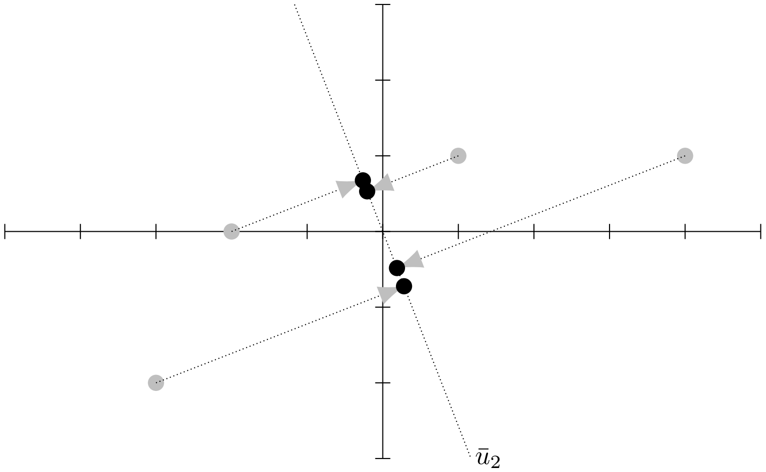
The result is the projection of the points onto $\bar{u}_1$:



If we set $s_1 = 0$ and recalculate the matrix product:

$$\begin{bmatrix} -0.932 & -0.3625 \\ -0.3625 & 0.932 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1.312 & 0 & 0 \end{bmatrix} V^T = \begin{bmatrix} 0.1878 & -0.2064 & -0.2628 & 0.2815 \\ -0.4829 & 0.5307 & 0.6757 & -0.7236 \end{bmatrix}$$

The result is the projection of the points onto $\bar{u}_2$:



Moreover the calculation from earlier also tells us the meaning of the s_i. If we take the square of the norm of each side, keeping in mind that the vectors in U and in V are orthonormal, we see:

$$\begin{aligned}
||\bar{a}_1||^2 &= s_1^2 v_{11}^2 + s_2^2 v_{12}^2 + ... + s_k^2 v_{1k}^2 \\
||\bar{a}_2||^2 &= s_1^2 v_{21}^2 + s_2^2 v_{22}^2 + ... + s_k^2 v_{2k}^2 \\
\vdots \quad & \quad \vdots \\
||\bar{a}_n||^2 &= s_1^2 v_{n1}^2 + s_2^2 v_{n2}^2 + ... + s_k^2 v_{nk}^2
\end{aligned}$$

And then if we add these equations:

$$||A||^2 = ||\bar{a}_1||^2 + ||\bar{a}_2||^2 + ... + ||\bar{a}_n||^2 = s_1^2 + s_2^2 + ... + s_k^2$$

This tells us that the total variation in all of the columns of A equals the sum of the squares of the singular values.

Moreover since each s_i corresponds to $\bar{u}_i$, this tells us that the total variation in all of the columns of A in the direction of $\bar{u}_i$ equals s_i^2 and that the proportion of the total variation in all of the columns of A in the direction of $\bar{u}_i$ equals

$$\frac{s_i^2}{s_1^2 + s_2^2 + ... + s_k^2}$$

10.4 Summary and Matrix Comment

So where does this leave us? In summary so far, for an $m \times n$ matrix A when we find the singular value decomposition $A = U\Sigma V^T$ then:

(a) The $\bar{u}_i$ break down the directions in which the data is spread out.

(b) The order of the directions $\bar{u}_1, ..., \bar{u}_n$ gives us the order in which the variance of the data goes from largest to smallest

(c) In the direction $\bar{u}_i$ the variance of the data equals s_i^2 and the proportion of the total variance of the data equals $s_i^2/(s_1^2 + s_2^2 + ... + s_k^2)$.

(d) If we create Σ' by taking Σ and replacing some of the s_i by 0, and if we then recalculate $A' = U\Sigma' V^T$, the result will be the orthogonal projection of the columns of A onto the columns of U corresponding to the s_i which remain.

As a consequence of the above:

(a) We can simplify the data (meaning keeping only the large-scale trends with the most variance) by retaining only the large singular values, thereby keeping only the primary directions of variance in which the columns of A.

(b) If we retain only $s_1, ..., s_j$ (with $j < k$) then the total variance in the data that is preserved can be calculated as:

$$\frac{s_1^2 + ... + s_j^2}{s_1^2 + ... + s_k^2}$$

(c) If we wish to simplify the data while retaining a certain amount of variance we can choose how many singular values to preserve accordingly.

(d) The columns in the matrix A' will generally be close to the columns in A, which translates to the matrix A' being close to A because $||A - A'||$ is small.

Let's try to summarize with some detailed examples.

Example 10.3. Consider the matrix with SVD:

$$A = \begin{bmatrix} -1 & 1 & 2 \\ -1 & 2 & 2 \end{bmatrix} = \begin{bmatrix} -0.6287 & -0.7777 \\ -0.7777 & 0.6287 \end{bmatrix} \begin{bmatrix} 3.8287 & 0 & 0 \\ 0 & 0.5840 & 0 \end{bmatrix} V^T$$

The total variance in the data is:

$$3.8287^2 + 0.5840^2 = 15$$

The left-singular vector of A corresponding to $s_1 = 3.8287$ captures

$$\frac{3.8287^2}{3.8287^2 + 0.5840^2} = 0.9773 = 97.73\%$$

of the total variance in the data, meaning that the data is extremely spread out in that direction.

The left-singular vector of A corresponding to $s_2 = 0.5840$ captures

$$\frac{0.5840^2}{3.8287^2 + 0.5840^2} = 0.0227 = 2.27\%$$

of the total variance in the data, meaning that the data is not very spread out in that direction.

We can see that if we let Σ' be Σ but with the smaller singular value set to zero and recalculate:

$$\begin{aligned} A' = U\Sigma' V^T &= \begin{bmatrix} -0.6287 & -0.7777 \\ -0.7777 & 0.6287 \end{bmatrix} \begin{bmatrix} 3.8287 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \\ &= \begin{bmatrix} -0.8841 & 1.3730 & 1.7683 \\ -1.0937 & 1.6984 & 2.1873 \end{bmatrix} \end{aligned}$$

The result is almost A in the sense that the columns of A have been rebuilt using only the first building-block vector. Take a second to look at the values in A and A'. They're close to one another:

$$A = \begin{bmatrix} -1 & 1 & 2 \\ -1 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} -0.8841 & 1.3730 & 1.7683 \\ -1.0937 & 1.6984 & 2.1873 \end{bmatrix}$$

Example 10.4. Consider the matrix:

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

The singular value decomposition $A = U\Sigma V^T$ has:

$$\begin{aligned} A &= U\Sigma V^T \\ &= \begin{bmatrix} -0.5098 & 0.6299 & -0.586 \\ -0.4949 & -0.7719 & -0.3991 \\ -0.7037 & 0.08654 & 0.7052 \end{bmatrix} \begin{bmatrix} 10.1205 & 0 & 0 & 0 \\ 0 & 2.8502 & 0 & 0 \\ 0 & 0 & 0.6722 & 0 \end{bmatrix} V^T \end{aligned}$$

The total variance in the data is:

$$10.1205^2 + 2.8502^2 + 0.6722^2 = 111$$

The variance captured by each of the three directions is:

- The left-singular vector of A corresponding to $s_1 = 10.12$ captures

$$\frac{10.1205^2}{10.1205^2 + 2.8502^2 + 0.6722^2} = 0.9227 = 92.27\%$$

of the total variance.

- The left-singular vector of A corresponding to $s_2 = 2.85$ captures

  $$\frac{2.8502^2}{10.1205^2 + 2.8502^2 + 0.6722^2} = 0.0732 = 7.32\%$$

  of the total variance.

- The left-singular vector of A corresponding to $s_3 = 0.6722$ captures

  $$\frac{0.6722^2}{10.1205^2 + 2.8502^2 + 0.6722^2} = 0.0041 = 0.41\%$$

  of the total variance.

Moreover if we wanted to ignore the last direction and project all the points onto the first two directions we simply take Σ, change the 0.6722 to 0 to get Σ', and recalculate.

$$A' = U\Sigma' V^T = \begin{bmatrix} 0.8905 & 1.7265 & 4.2251 & 2.8669 \\ -1.0746 & -0.1863 & 2.1533 & 4.9094 \\ 0.1318 & 1.3291 & 4.7290 & 5.1602 \end{bmatrix}$$

The new matrix is said to preserve

$$\frac{10.1205^2 + 2.8502^2}{10.1205^2 + 2.8502^2 + 0.6722^2} = 0.9959 = 99.59\%$$

of the variance of the original matrix. Really compare the values in the A and A'. They are quite close!

$$A = \begin{bmatrix} 1 & 2 & 4 & 3 \\ -1 & 0 & 2 & 5 \\ 0 & 1 & 5 & 5 \end{bmatrix} \approx \begin{bmatrix} 0.8905 & 1.7265 & 4.2251 & 2.8669 \\ -1.0746 & -0.1863 & 2.1533 & 4.9094 \\ 0.1318 & 1.3291 & 4.7290 & 5.1602 \end{bmatrix}$$

Example 10.5. Consider the matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 & 0 & 5 & 2 \\ 3 & 4 & -1 & 2 & 3 & 4 \\ 5 & 4 & 1 & 2 & 0 & 0 \\ 6 & 4 & 1 & 2 & 3 & -1 \\ 6 & 3 & 1 & 2 & 3 & 4 \end{bmatrix}$$

Suppose we wanted to simplify the columns of this matrix as much as possible while retaining 95% of the variance.

We calculate the singular value decomposition $A = U\Sigma V^T$ and find that the singular values are $\{15.0759, 5.8017, 4.2418, 2.1720, 1.5318\}$

The total variance in the data is:

$$15.0759^2 + 5.8017^2 + 4.2418^2 + 2.1720^2 + 1.5318^2 = 286$$

Noting that:

$$\frac{15.0759^2 + 5.8017^2}{15.0759^2 + 5.8017^2 + 4.2418^2 + 2.1720^2 + 1.5318^2} = 0.9124$$

and

$$\frac{15.0759^2 + 5.8017^2 + 4.2418^2}{15.0759^2 + 5.8017^2 + 4.2418^2 + 2.1720^2 + 1.5318^2} = 0.9753$$

We see that we will need to preserve the largest three singular values and we will retain 97.53% of the variance.

If Σ' is Σ but with the smallest two replaced by zero then our simpler matrix is:

$$A' = U\Sigma' V^T = \begin{bmatrix} 1.163 & 1.728 & 2.755 & -0.006 & 5.214 & 1.916 \\ 3.618 & 3.171 & -0.361 & 1.829 & 2.643 & 4.425 \\ 5.335 & 3.419 & 0.310 & 2.007 & 0.579 & -0.262 \\ 6.012 & 4.081 & 1.771 & 1.924 & 2.428 & -0.632 \\ 5.172 & 4.147 & 0.425 & 2.202 & 3.269 & 3.564 \end{bmatrix}$$

Notice that this matrix is fairly close to the original A.

10.5 Centering Comment

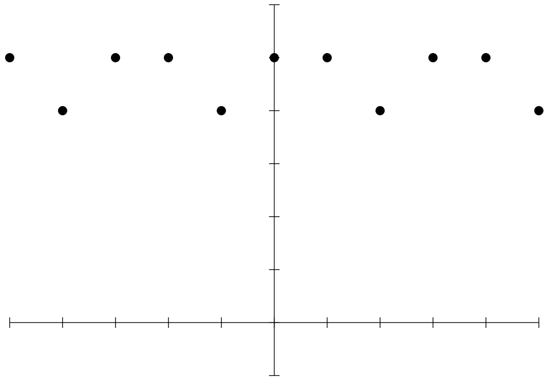
In the real world if data is analyzed this way usually the first step is to center it around the origin by subtracting the average of all columns of A from each column of A. This is because, if the data is not centered around the origin in a particularly nasty way, the vectors in U don't really make sense for the data.

We are completely ignoring that in order to avoid an extra step and to keep the values in A nice, as mostly the data we are using is already essentially around the origin in all the examples. There is one homework question which steps through the analysis with the centering process.

Example 10.6. The points in this matrix:

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

When plotted look like this:



The singular value decomposition has:

$$A = \begin{bmatrix} -0.0155 & 1.0 \\ 1.0 & 0.0155 \end{bmatrix} \begin{bmatrix} 15.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} V^T$$

These vectors don't make intuitive sense at all for the data.

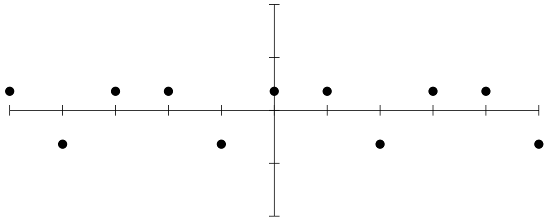
If we center the data, subtracting the average of all points

$$\begin{bmatrix} 0 \\ 4.6364 \end{bmatrix}$$

from each point, we get the centered data:

$$\begin{bmatrix} -5.0 & -4.0 & -3.0 & -2.0 & -1.0 & 0 & 1.0 & 2.0 & 3.0 & 4.0 & 5.0 \\ 0.364 & -0.636 & 0.364 & 0.364 & -0.636 & 0.364 & 0.364 & -0.636 & 0.364 & 0.364 & -0.636 \end{bmatrix}$$

Which, when plotted, looks like:



The singular value decomposition now has:

$$A = \begin{bmatrix} -1.0 & 0.0186 \\ 0.0186 & 1.0 \end{bmatrix} \begin{bmatrix} 10.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.58 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} V^T$$

which makes much more sense if we think of these vectors centered at the center of the data.

10.6 Formal Theorem and Proof

We really need to verify that it works, meaning that $A = U\Sigma V^T$ where U, Σ and V are defined as earlier.

Here is the case where $m = n$. The case where $m \neq n$ requires a bit more fiddling with extra columns and rows.

Theorem 10.6.0.1. Suppose A is an $n \times n$ matrix. Then A may be written as $A = U\Sigma V^T$ where U is $n \times n$ orthogonal, Σ is $n \times n$ diagonal and V is $n \times n$ orthogonal.

Proof. Since V is to be orthogonal $V^T = V^{-1}$ so the goal can be rewritten as

$$AV = U\Sigma$$

Moreover this is equivalent to having

$$A\bar{v}_i = \sigma_i \bar{u}_i$$

for each i where $\bar{v}_i$ and $\bar{u}_i$ are the columns of V and U respectively.

Let $\{\bar{v}_1, ..., \bar{v}_n\}$ be an orthonormal basis of eigenvectors for $A^T A$ with nonnegative eigenvalues $\lambda_1, ..., \lambda_n$ so that $A^T A\bar{v}_i = \lambda_i \bar{v}_i$ for each i and let $\sigma_i = \sqrt{\lambda_i}$ for each i.

Let $\{\bar{u}_1, ..., \bar{u}_n\}$ be defined by

$$\bar{u}_i = \frac{A\bar{v}_i}{\sqrt{\lambda_i}}$$

and observe that for every i, j we have:

$$\begin{aligned}
\bar{u}_i \cdot \bar{u}_j &= \left(\frac{A\bar{v}_i}{\sqrt{\lambda_i}}\right) \cdot \left(\frac{A\bar{v}_j}{\sqrt{\lambda_j}}\right) \\
&= \frac{1}{\lambda_i}\left(A\bar{v}_i \cdot A\bar{v}_j\right) \\
&= \frac{1}{\lambda_i}\left(\bar{v}_i^T A^T A\bar{v}_j\right) \\
&= \frac{1}{\lambda_i}\left(\bar{v}_i^T \lambda_i \bar{v}_j\right) \\
&= \bar{v}_i \cdot \bar{v}_j
\end{aligned}$$

so that $\{\bar{u}_1, ..., \bar{u}_n\}$ are orthonormal too. More over we have:

$$A\bar{v}_i = \sqrt{\lambda}\bar{u}_i$$

and the proof is complete.

Note that by construction the columns of U form an orthonormal eigenvector basis for A^TA and also since

$$AA^T\bar{v}_i = AA^T\left(\frac{A\bar{u}_i}{\sqrt{\lambda_i}}\right) = A\left(\frac{\lambda_i\bar{u}_i}{\sqrt{\lambda_i}}\right) = A\sqrt{\lambda_i}\bar{u}_i = \lambda_i\bar{v}_i$$

we see that the columns of V form an orthonormal eigenvector basis for AA^T.

□

10.7 Matlab

Throwing out a singular value and recalculating is easy too:

```
>> A=[
1 2 0
3 1 4
1 1 -1
0 1 2
]
A =
     1     2     0
     3     1     4
     1     1    -1
     0     1     2
>> [U,S,V] = svd(A)
U =
   -0.2079    0.7246   -0.3584   -0.5507
   -0.9171   -0.1139    0.3788   -0.0501
   -0.0103    0.6615    0.2666    0.7009
   -0.3400   -0.1560   -0.8106    0.4506
S =
    5.5200         0         0
         0    2.5538         0
         0         0    1.4170
         0         0         0
V =
   -0.5380    0.4090    0.7371
   -0.3049    0.7208   -0.6225
   -0.7859   -0.5596   -0.2631
>> SP=S;SP(3,3)=0;
>> U*SP*transpose(V)
ans =
    1.3743    1.6839   -0.1336
    2.6044    1.3341    4.1412
    0.7216    1.2351   -0.9006
    0.8466    0.2851    1.6979
```

10.8 Exercises

Exercise 10.1. Find the (vector) direction in which the following set of points have the most variance, then plot the points and the corresponding line.

$$(1, 2), (4, 3), (-1, -3), (-5, -8)$$

Exercise 10.2. Find the (vector) direction in which the following set of points have the most variance, then plot the points and the corresponding line.

$$(1, -1), (3, -2), (6, -4), (10, -4)$$

Exercise 10.3. For each of the following matrices find the SVD and identify the direction in which the column data is most spread out, as well as the variance and proportion of total variance in that direction.

(a) $A = \begin{bmatrix} 5.0 & 6.0 & 5.0 & 5.0 & 6.0 \\ 6.0 & 5.0 & 5.0 & 6.0 & 5.0 \\ 5.0 & 6.0 & 5.0 & 6.0 & 5.0 \end{bmatrix}$

(b) $A = \begin{bmatrix} 0 & 1.0 & 1.0 & 2.0 & 2.0 \\ 1.0 & 2.0 & 2.0 & 0 & 1.0 \\ 7.0 & 8.0 & 8.0 & 8.0 & 9.0 \\ 8.0 & 9.0 & 9.0 & 8.0 & 7.0 \\ 7.0 & 7.0 & 7.0 & 9.0 & 7.0 \end{bmatrix}$

(c) $A = \begin{bmatrix} 4.0 & 5.0 & 0 & 0 & 0 \\ 5.0 & 5.0 & 0 & 0 & 0 \\ 0 & 1.0 & 7.0 & 8.0 & 8.0 \\ 0 & 0 & 8.0 & 7.0 & 7.0 \end{bmatrix}$

(d) $A = \begin{bmatrix} 2.0 & 3.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 3.0 & 3.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 & 5.0 & 5.0 & 5.0 \end{bmatrix}$

Exercise 10.4. Given the matrix

$$A = \begin{bmatrix} 2.0 & 3.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 3.0 & 3.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 & 5.0 & 5.0 & 5.0 \end{bmatrix}$$

(a) Find the singular value decomposition of A.

(b) Find an approximation A' to A preserving the largest two singular values only. What proportion of the total variance is preserved?

Exercise 10.5. Given the matrix

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 & 5 \\ 0 & 1 & 1 & -1 & 6 \\ -1 & 0 & -2 & 1 & 4 \\ 2 & 2 & 1 & 2 & -5 \\ 2 & 1 & 0 & 0 & -5 \end{bmatrix}$$

(a) Find the singular value decomposition of A.

(b) Find an approximation A' to A preserving the largest three singular values only. What proportion of the total variance is preserved?

Exercise 10.6. Find an approximation to the following matrix which preserves at least 95% of the total variance using as few singular values as possible:

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Exercise 10.7. Consider the points stored in the columns of this matrix:

$$\begin{bmatrix} 2.0 & -0.1 & 5.9 & 2.3 & 2.9 & -5.1 & 1.9 & 3.6 \\ 3.2 & 4.8 & -2.2 & 5.5 & 0.9 & -1.7 & 3.0 & 3.6 \\ 17.0 & 13.0 & 1.5 & 21.0 & 3.7 & -18.0 & 15.0 & 15.0 \end{bmatrix}$$

These points mostly lie close to a plane through the origin.

(a) Find the equation of this plane by finding the two directions with the most variance, finding the normal vector for the plane, then finding the equation.

(b) Using this equation predict which z-value would correspond to $x = 10$ and $y = -3$.

Exercise 10.8. Consider the points stored in the columns of this matrix:

$$\begin{bmatrix} 1.9 & 0.9 & 2.6 & 6.1 & 3.4 & -0.5 & 4.5 & 6.0 \\ 5.3 & 11.0 & 3.9 & 7.0 & 8.2 & 9.2 & 12.0 & 2.6 \\ -0.5 & 1.6 & -1.6 & -13.0 & -10.0 & 3.3 & -18.0 & -8.8 \end{bmatrix}$$

These points mostly lie close to a plane but not through the origin.

(a) Find the mean of all the points. This is basically where all the points are centered.

(b) Center the points around the origin by subtracting this mean from each point.

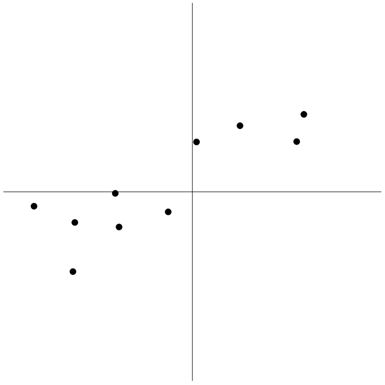
(c) Find the two directions with the most variance and use these to find the normal vector for the plane.

(d) Find the equation of the plane using the normal vector and also the mean.

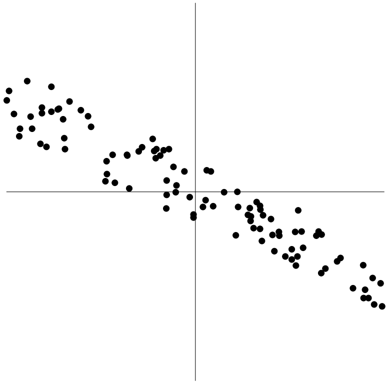
(e) Using this equation predict x so that $(x, 1, -2)$ is on the plane, predict y so that $(0, y, 10)$ is on the plane, and predict z so that $(3, 4, z)$ is on the plane.

Exercise 10.9. Suppose n points are placed as columns in a $2 \times n$ matrix. If $A = U\Sigma V^T$ is the SVD and if the points are plotted below, say as much as you can about the matrices U and Σ

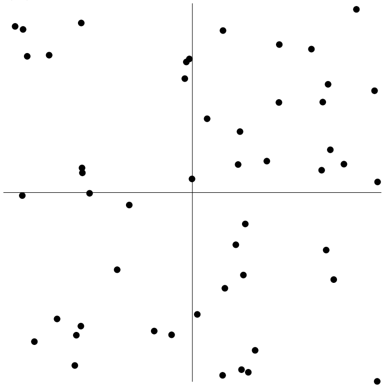
(a) $n = 10$ points:



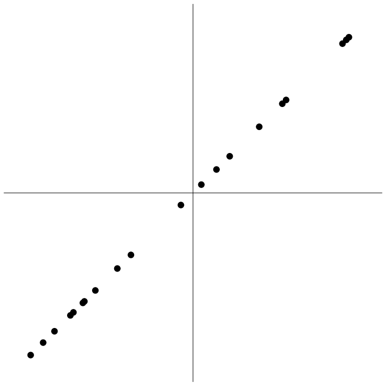
(b) $n = 100$ points:



(c) $n = 50$ points:



(d) $n = 20$ points:



Exercise 10.10. Suppose a 100×100 matrix has 100 singular values whose squares add to 9512 and in decreasing order are $\{82.2, 27.6, 23.3, 19.1, 16.3, 8.5, 5.3, ...\}$.

How many singular values must be preserved in order to keep 90% of the data variance? How about 80%?

Exercise 10.11. Explain the mathematical process by which you would approximate a 200×200 matrix while keeping 99% of the variance.

Chapter 11

Image Compression

Contents

11.1 Image Representation

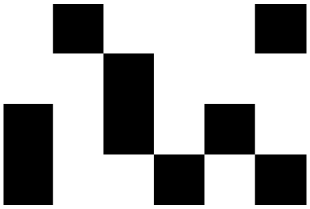
There are a variety of ways to store image data but when graphics are displayed pixel-by-pixel then we need to store data for each pixel.

Typical approaches:

- For a color image combine Red, Green and Blue, each with an integer between 0 and 255 inclusive. This gives a total of $256^3 = 16777216$ colors available.
- For a grayscale image assign a single integer between 0 and 255 inclusive, where 0 indicates Black and 255 indicates White.
- For a grayscale image assign a single real number between 0 and 1 inclusive, where 0 indicates Black and 1 indicates White.

In order to facilitate easy mathematical calculations (no integer truncation, etc.) we're going to stick with the third option.

Example 11.1. For example the following image consists of no greyscale, only black and white:



This image is be represented by the matrix:

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

11.2 Image Compression

We saw in the previous chapter that we can use the singular value decomposition of a matrix to find an approximation to that matrix which preserves a certain amount of the original matrix's variance.

It follows that if the matrix represents an image then an approximation to that matrix can be thought of as an image which preserves a certain amount of the original image's variance.

Example 11.2. If we find the SVD of this matrix we have:

$$A = \begin{bmatrix} -0.51 & 0.57 & -0.50 & 0.40 \\ -0.66 & -0.29 & -0.19 & -0.66 \\ -0.40 & -0.64 & 0.18 & 0.63 \\ -0.38 & 0.42 & 0.82 & -0.05 \end{bmatrix} \begin{bmatrix} 3.198 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.678 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.235 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6576 & 0 & 0 \end{bmatrix} V^T$$

If we set the smallest singular value to zero to create Σ' and compute:

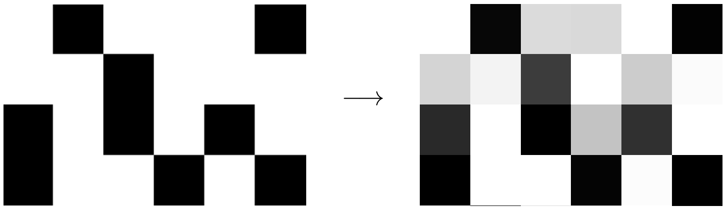
$$A' = U\Sigma' V^T = \begin{bmatrix} 1.103 & 0.030 & 0.856 & 0.848 & 1.122 & 0.011 \\ 0.831 & 0.951 & 0.236 & 1.248 & 0.800 & 0.982 \\ 0.162 & 1.047 & -0.226 & 0.762 & 0.191 & 1.018 \\ -0.012 & 0.997 & 1.017 & 0.018 & 0.986 & -0.001 \end{bmatrix}$$

Before we charge ahead and look at the resulting image we notice an issue. The values in this new matrix are not necessarily between 0 and 1.

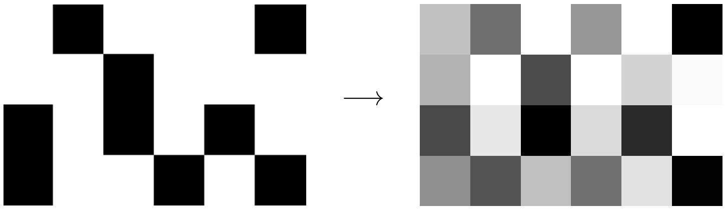
There are two possible approaches to this. The first would be to renormalize the values between 0 and 1. However there's a good argument against this, that being that (for example) a value of 1 is supposed to represent completely White so if we had a value such as 1.1 then rescaling would interpret the first as not completely White but rather as slightly more grey, which is not accurate at all.

Instead we'll take the second approach and simply leave the values alone. Values above 1 will be treated as White (like 1) and values below 0 will be treated as black (like 0).

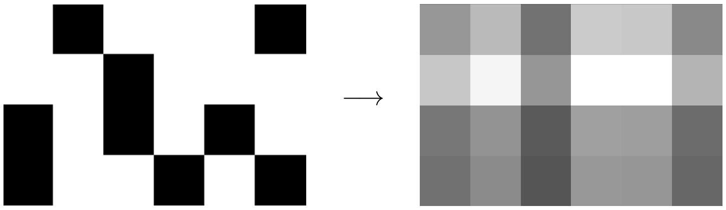
Example 11.3. With this in mind our previous example, preserving three singular values (and 97.12% of the matrix data variance), would be, along with the original:



If we do it again preserving two singular values (and 86.95% of the matrix data variance), along with the original:



and then one singular value (and 68.18% of the matrix data variance), along with the original:

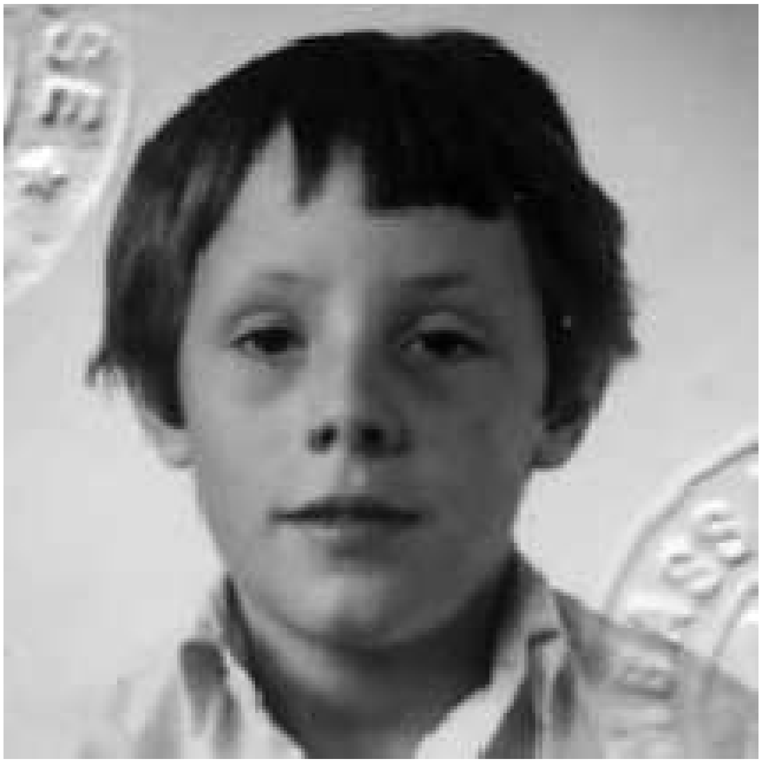


As the number of singular values decreases the image loses variance and becomes more uniform while still trying to retain as much variance as possible.

Here is an even better example:

Example 11.4. The following image is represented by a 200×200 matrix A:

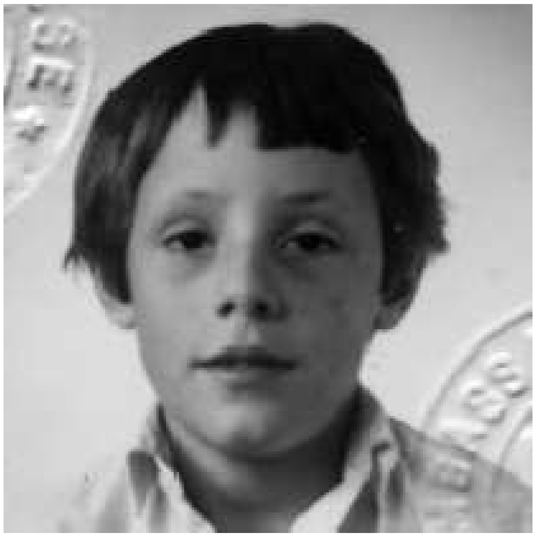
This is Justin, age 8.



If we do a SVD for the matrix A we see that there are 200 singular values. We won't list them all but here are the ones that are greater than 0.5:

```
114.6751, 23.1226, 17.6162, 10.0497,
8.8713, 7.4690, 7.0125, 5.5884, 4.7874, 4.4234, 4.1422, 3.7599, 3.1233,
2.9382, 2.8150, 2.6498, 2.3993, 2.1972, 2.0984, 1.9532, 1.8977, 1.7923,
1.7188, 1.5803, 1.4746, 1.3644, 1.3258, 1.3239, 1.1909, 1.1655, 1.0972,
1.0582, 1.0382, 1.0117, 0.9170, 0.9015, 0.8496, 0.7791, 0.7556, 0.7100,
0.6853, 0.6507, 0.6407, 0.6081, 0.5906, 0.5732, 0.5585, 0.5356, 0.5013
```

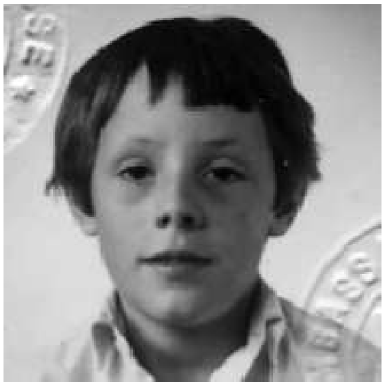
If we zero out all but those highest 100, recreate the matrix and view we get:



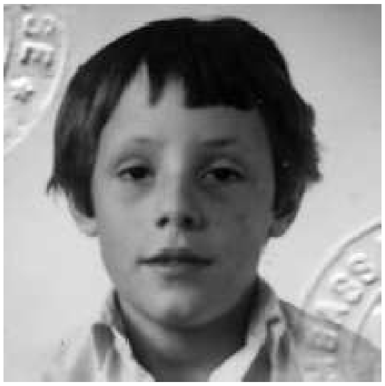
It's really hard to see any difference here.

We can do this preserving any number of values.

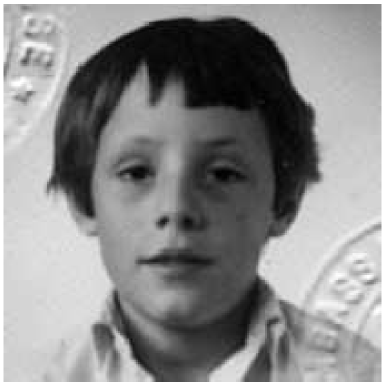
Here are the images resulting from preserving 200 (all), 100, 50, 20, 10, 5 and 1 singular values:



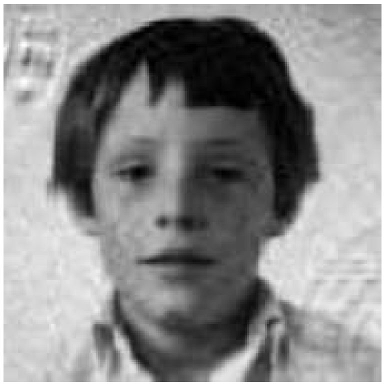
200 values



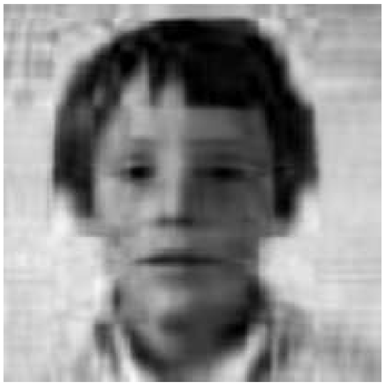
100 values



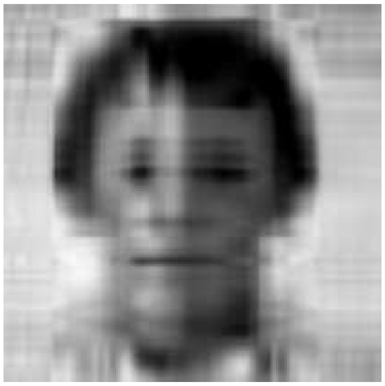
50 values



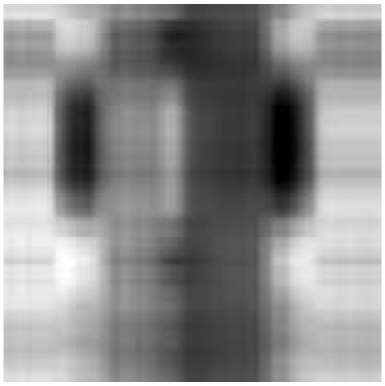
20 values



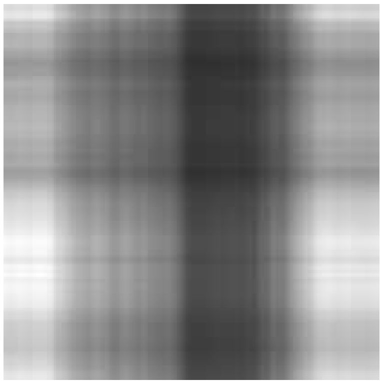
10 values



5 values



2 values



1 value

We can see that the rank 50 version is still quite good, it's only when we get to rank 20 that noticeable deterioration of image quality starts to take place.

The beautiful thing about the final rank 1 version is that the entire image is constructed out of a single vertical vector whose values correspond to shades of grey. Each column is a multiple of that value and you can see that easily!

11.3 Image Quality

The image quality can be defined as the proportion of variance preserved in the new image. Recall that this the sum of the squares of the retained singular values divided by the sum of the squares of all of the singular values.

In the Justin case if we look at the version preserving 50 singular values (the last pretty good one) then we find that the image quality can be defined as:

$$\frac{s_1^2 + ... + s_{20}^2}{s_1^2 + ... + s_{200}^2} = 0.999769$$

meaning that we've preserved 99.9769% of the original image variance. Loosely speaking this image is 99.9769% as good as the original.

Here are the values for all of the versions:

# Singular Values	Variance Preserved	Quality Percentage
200	1.00	100
100	0.999985	99.9985
50	0.999769	99.9769
20	0.997387	99.7387
10	0.991623	99.1623
5	0.979279	97.9279
2	0.945426	94.5426
1	0.908489	90.8489

This is interesting because it gives us a sense of how much variance we would need to preserve to keep a reasonable-looking picture, albeit for only one example. While 90% seems like a lot, in this case the resulting picture (the final one) is not good at all, and we'd probably aim for perhaps 99.5% or more.

If we set a threshold at 99.5% then a quick calculation shows that we need at least 14 singular values to do the job.

11.4 Data Savings

Importantly we need to address why we would do this at all.

Here we'll look at $n \times n$ (square) images. Images which are not square take a little tweaking and are addressed in the exercises.

If we wish to save a grayscale $n \times n$ image in a matrix A using one value per pixel then we need to save n^2 values.

Suppose instead we use the SVD and preserve $k < n$ singular values. In the previous chapter we saw that this means we're essentially recalculating the matrix A' as follows:

$$\begin{aligned}
\bar{a}'_1 &= s_1 v_{11}\bar{u}_1 + s_2 v_{12}\bar{u}_2 + ... + s_k v_{1k}\bar{u}_k \\
\bar{a}'_2 &= s_1 v_{21}\bar{u}_1 + s_2 v_{22}\bar{u}_2 + ... + s_k v_{2k}\bar{u}_k \\
\vdots & \qquad\quad \vdots \\
\bar{a}'_n &= s_1 v_{n1}\bar{u}_1 + s_2 v_{n2}\bar{u}_2 + ... + s_k v_{nk}\bar{u}_k
\end{aligned}$$

If we think of the right hand side as simply linear combinations of $\bar{u}_1$, ..., $\bar{u}_k$ then all we really need to save are the vectors $\bar{u}_1, ..., \bar{u}_k$, each in $\mathbb{R}^n$ and therefore consisting of n values each for a total of kn values, and the coefficients to find each $\bar{a}'_i$, consisting of k values each for a total of kn values again. Thus in sum total we need to save $2kn$ values.

The value $\frac{2kn}{n^2} = \frac{2k}{n}$ can be throught of as the compression ratio and provided $2nk < n^2$, or $k < \frac{n}{2}$, we've saved space.

Example 11.5. If we again focus on the Justin example, preserving 20 singular values, we can store the image using $2kn = 2(20)(200) = 8000$ values instead of $200^2 = 40000$ values for a compression ratio of $\frac{8000}{40000} = 0.20 = 20\%$.

Preserving 50 singular values, quite a decent picture, we can store the image using $2kn = 2(50)(200) = 20000$ values instead of $200^2 = 40000$ values for a compression ratio of $\frac{20000}{40000} = 0.50 = 50\%$.

11.5 Matlab

A matrix with values between 0 and 1 can be displayed in Matlab as follows, where 0 is black and 1 is white:

```
>> A = [1 0.5 1 1 0.9 0;1 1 0 1 1 1;0 0.1 0 1 0 1;0 1 1 0 1 0.3];
>> imshow(A,’border’,’tight’,’InitialMagnification’,1000)
```

Note that the `’InitialMagnification’,1000` setting makes the picture 1000% of the original size, otherwise our image (which is only a few pixels wide and high) would be too small to see.

The `’border’,’tight’` just gets rid of the border around the image. This isn’t really necessary unless you’re going to save the image and don’t want to save the border.

We can then do a singular value decomposition on it and manipulate it as before. For example the matrix above has four singular values so let’s zero out the smallest and redisplay:

```
>> [U,S,V] = svd(A);
>> SP = S;SP(4,4)=0;
>> AP = A*SP*transpose(V);
>> imshow(AP,’border’,’tight’,’InitialMagnification’,1000)
```

Now then, if we’d like to read in a graphics file we can do so quickly. We use `imread` to read the image into a matrix. Next we use `rgb2gray` to ensure the image is grayscale (otherwise the matrix might have an extra dimension containing red, blue and green values). After that we use `double` to convert the values from `uint8` (the default unsigned integer values, which we can’t directly use svd on) to double-precision real numbers (more than sufficient for our purposes but Matlab default) Lastly we scale them between 0 and 1 by subtracting the minimum of all of them (making the minimum 0) and then dividing by the maximum of all of them (making the maximum 1).

All together:

```
>> A = imread(’justin.jpg’);
>> A = rgb2gray(A);A = double(A);A = A-min(A(:));A = A/max(A(:));
```

If we’d like to view it:

```
>> imshow(A,’border’,’tight’);
```

Now then, if our file has lots of singular values and we want to zero out a bunch of them, a for loop can help. For example the above `justin.jpg` file is the image from this chapter and has 200 singular values. If we only want to keep the first 100 and plot:

```
>> [U,S,V] = svd(A);
>> SP = S;for i=[101:200];SP(i,i)=0;end;
>> AP = U*SP*transpose(V);
>> imshow(AP,'border','tight');
```

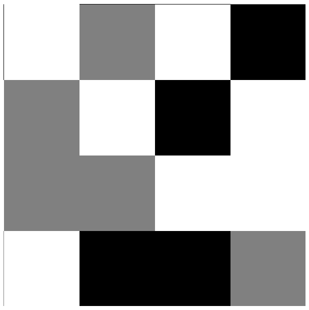
Okay, so how about calculating the proportion of variance? Well, we can use the `diag` command to extract the values out of our Σ and play with the resulting vector of values. For example here's the proportion of variance in the first 100 singular values of the original Justin picture from this chapter, loaded earlier:

```
>> svals = diag(S);
>> vpa(sum(svals(1:100).^2)/sum(svals.^2),6)
ans =
0.999985
```

Notation Note: The reason we use `.^` instead of `^` is that `diag(S)` and hence `svals` and `svals(1:100)` are all vectors which are treated as matrices in Matlab. Simply doing, for example, `svals^2` would attempt in this case to multiply a 200×1 matrix by itself, which doesn't work. What we really want to do is take each entry in that matrix and square it, and this is what `.^2` does. Basically it applies the `^2` operation element-by-element.

11.6 Exercises

Exercise 11.1. Consider the following image:

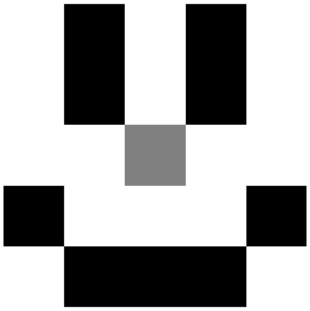


(a) Enter this image into a 4×4 matrix A. Assume the shades you see are only 0, 0.5 and 1.

(b) Find the SVD for A.

(c) Simplify the matrix preserving 3, 2 and then 1 singular value. Draw the best image representation you can of each matrix.

Exercise 11.2. Consider the following image:



(a) Enter this image into a 5×5 matrix A. Assume the shades you see are only 0, 0.5 and 1.

(b) Find the SVD for A.

(c) Simplify the matrix preserving 4, 3, 2 and then 1 singular value. Draw the best image representation you can of each matrix.

Exercise 11.3. Find a square image on the internet of reasonable size; 200×200 or thereabouts is good.

(a) Load this into the Matlab matrix A, convert to grayscale if necessary and scale the values between 0 and 1.

(b) Find the SVD for A.

(c) How many singular values are there?

(d) Simplify the image by preserving only 75% of the singular values. What is the percentage of data quality preserved? Print the result.

(e) Simplify the image by preserving only 50% of the singular values. What is the percentage of data quality preserved? Print the result.

(f) Simplify the image by preserving only 25% of the singular values. What is the percentage of data quality preserved? Print the result.

(g) What is the minimum number of singular values that must be preserved in order to preserve 99.9% of the data quality? What level of data compression would this achieve? Simplify the image accordingly and print the result.

Exercise 11.4. Try to find two images, both 200×200, both photographs, one requiring as few singular values as possible and one requiring as many singular values as possiboe, both to achieve 99.9% image variance. Identify, if you can, what about the images leads to this disparity. Lots of darks and lights? Regular patterns? Anything you can find!

Exercise 11.5. Assume for each of the following that the singular values have been given for a 10×10 image. Determine the minimum number of singular values in decreasing order that would need to be preserved to keep 99.9% of the image variance and what the resulting data compression ratio would be.

(a) $\{80.2608, 63.3520, 20.5871, 8.4696, 2.8841, 1.6763, 0.7962, 0.6926, 0.6553, 0.6520\}$

(b) $\{138.5481, 49.5181, 16.5869, 4.9396, 3.2379, 1.5248, 1.1992, 1.0277, 1.0208, 0.9786\}$

(c) $\{446.1649, 163.4218, 43.8892, 13.7979, 4.7417, 1.3437, 1.0500, 0.6422, 0.4532, 0.4484\}$

(d) $\{209.4851, 28.1916, 22.8720, 11.8304, 6.3254, 2.7847, 1.2043, 0.9255, 0.8765, 0.8722\}$

Exercise 11.6. Suppose we simplify an $m \times n$ (not necessarily square) image using k singular values. What is the resulting data compression ratio? What criteria on k would make this worth doing?

Chapter 12

Character Recognition

Contents

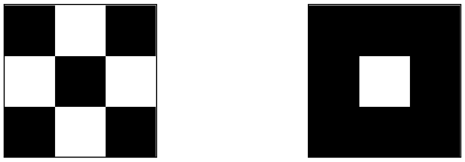
12.1 Introduction

The goal of this chapter is to introduce a very simple but reasonably effective method of recognizing characters. This can (and will) be implemented in a practical way (Matlab) but the real key is to understand what is going on mathematically.

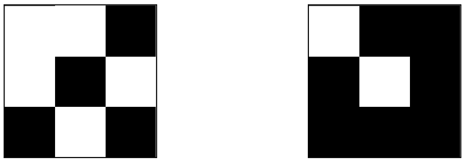
12.2 Simple Distance Checking

12.2.1 An Example

Before diving into complicated characters let's just look at the two characters X and O. To keep things really simple let's write each of them in a 3×3 grid:



Now consider these two, which we would intuitively accept as also X and O:



How can we say mathematically what we see intutively?

The obvious way might be to say that if we treat them as matrices where 0 indicates black and 1 indicates white then we have four matrices:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

What we'll actually do is unroll the matrices into vectors by simply putting each column underneath the previous one:

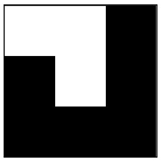
$$\bar{x} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \bar{x}_? = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \bar{o} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \bar{o}_? = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Then notice the following distances:

$$\begin{aligned}
||\bar{x} - \bar{x}_?|| &= 1 \\
||\bar{x} - \bar{o}_?|| &= 2.4495 \\
||\bar{o} - \bar{o}_?|| &= 1 \\
||\bar{o} - \bar{x}_?|| &= 2.4495
\end{aligned}$$

Clearly and for obvious reasons the Xs are closer to each other than they are to the Os and the Os are closer to each other than they are to the Xs.

So what we could do would be really simplistic - given an unknown character we could simply ask how far it is from our X and from our O and decide appropriately. For example consider this:



If we convert this to a matrix and then unroll it to a vector:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \bar{u} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Then we find:

$$\begin{aligned}
||\bar{x} - \bar{u}|| &= 2.2361 \\
||\bar{o} - \bar{u}|| &= 1.4142
\end{aligned}$$

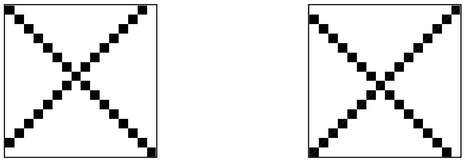
Since our unknown character is closer to an O, we call it an O. Voila!

Note: This could all have been done without unrolling the matrices but the unrolling will be necessary for what comes next.

12.2.2 There are Problems

Problems with this abound:

- Shifting a character slighly in a larger grid creates a huge distance. For example the following two Xs stored in 16 × 16 grids are distance 7.7460 apart when they are converted into vectors because one is shifted from the other in a way that misaligns the values.



- Rotating a character slightly does the same.
- There are many different versions of any character.

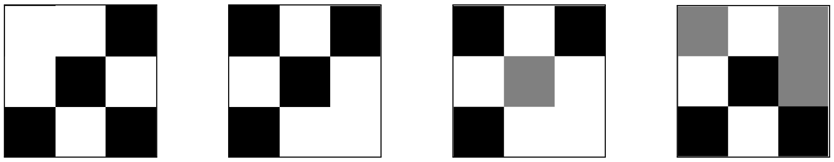
12.3 Developing a Robust SVD Method

12.3.1 Introduction

What we would really like to ask is more generally what a particular character looks like. In other words what are the mean features, what are more minor features, and so on.

12.3.2 The Essentials of a Character

Consider these four sort-of Xs. None is exactly an X but all of them would be accepted as such:



The vectors (unrolled matrices) corresponding to these are:

$$\bar{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \bar{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \bar{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0.5 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \bar{x}_4 = \begin{bmatrix} 0.5 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

If we put these together in a matrix and find the SVD:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0.5 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0.5 \\ 1 & 1 & 1 & 0.5 \\ 0 & 1 & 1 & 0 \end{bmatrix} = U\Sigma V^T$$

where

$$U = \begin{bmatrix} -0.18 & -0.62 & 0.42 & -0.10 & 0 & -0.11 & 0.24 & -0.35 & 0.44 \\ -0.49 & -0.08 & -0.19 & 0.01 & 0 & -0.36 & 0.09 & -0.45 & -0.62 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -0.49 & -0.08 & -0.19 & 0.01 & 0 & -0.35 & -0.63 & 0.28 & 0.36 \\ -0.07 & 0.19 & 0.03 & -0.98 & 0 & 0 & 0 & 0 & 0 \\ -0.49 & -0.08 & -0.19 & 0.01 & 0 & 0.84 & -0.05 & -0.10 & 0.03 \\ -0.05 & -0.17 & -0.68 & -0.05 & 0 & -0.13 & 0.59 & 0.27 & 0.23 \\ -0.43 & 0.09 & 0.49 & 0.06 & 0 & -0.03 & 0.35 & 0.62 & -0.21 \\ -0.26 & 0.72 & 0.07 & 0.16 & 0 & -0.11 & 0.24 & -0.35 & 0.44 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 4.1 & 0 & 0 & 0 \\ 0 & 1.3 & 0 & 0 \\ 0 & 0 & 0.57 & 0 \\ 0 & 0 & 0 & 0.35 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

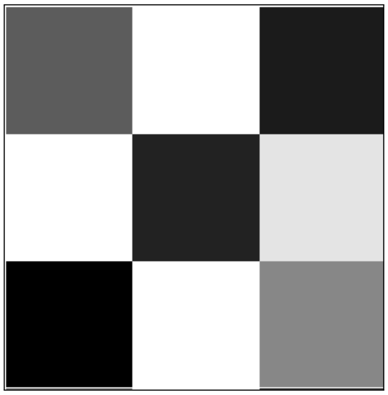
$V =$ Omitted - Not Relevant

Since each column of M represents some variation on the character X, the vector $\bar{u}_1$ becomes the most important building block for all of those columns, meaning for all of those Xs. In other words that single column accounts for most of the variation in all the variations on the character X.

Subsequent columns $\bar{u}_2$, ... are less relevant, as indicated by the singular values.

In fact if we take the appropriate multiple of $\bar{u}_1$ such that the values are scaled between 0 and 1, we get:

$$\begin{bmatrix} 0.36 \\ 1 \\ 0 \\ 1 \\ 0.13 \\ 1 \\ 0.11 \\ 0.89 \\ 0.53 \end{bmatrix} \rightarrow \begin{bmatrix} 0.36 & 1 & 0.11 \\ 1 & 0.13 & 0.89 \\ 0 & 1 & 0.53 \end{bmatrix}$$



This is very clearly what we visually accept as an X.

Of course we may not want to only consider $\bar{u}_1$. If we were to take $\bar{u}_2$ into account we might suggest that:

$$\text{An X is mostly} \begin{bmatrix} -0.18 \\ -0.49 \\ 0 \\ -0.49 \\ -0.07 \\ -0.49 \\ -0.05 \\ -0.43 \\ -0.26 \end{bmatrix} \text{with a bit of} \begin{bmatrix} -0.62 \\ -0.08 \\ 0 \\ -0.08 \\ 0.19 \\ -0.08 \\ -0.17 \\ 0.09 \\ 0.72 \end{bmatrix} \text{thrown in.}$$

So perhaps when we're thinking about the building blocks for an X we might use both $\bar{u}_1$ and $\bar{u}_2$. We'll discuss how many of the $\bar{u}_i$ we might use a bit later.

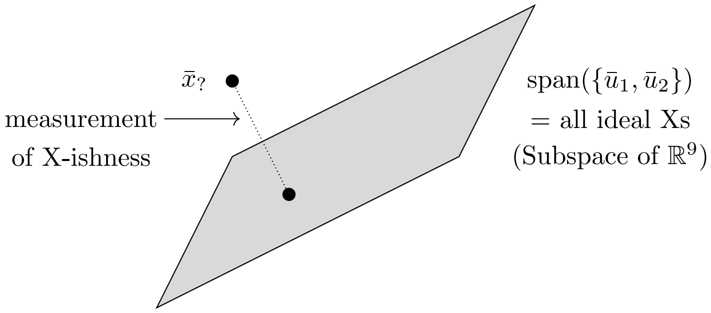
12.3.3 Comparing Another Character

Continuing the above example, suppose now we had another character and we wished to ask if we think it's an X.

We could suggest that if a (nonzero) character can be built out of $\bar{u}_1$ and $\bar{u}_2$ then it's definitely an X.

For a character we're testing it's almost certainly the case that we cannot build it exactly out of $\bar{u}_1$ and $\bar{u}_2$. So what we could ask instead if we can "almost" build this character out of $\bar{u}_1$ and $\bar{u}_2$.

A nice way to ask this in linear algebra would be to put this new character in a vector $\bar{x}_?$ and ask how far it is from the subspace spanned by $\bar{u}_1$ and $\bar{u}_2$. If it's not far, then it's almost an X. If it is far, then it's probably not an X.


$\bar{x}_?$
measurement of X-ishness
span$(\{\bar{u}_1, \bar{u}_2\})$
= all ideal Xs
(Subspace of $\mathbb{R}^9$)

(Note: All ideal Xs are not exactly the subspace spanned by $\bar{u}_1$ and $\bar{u}_2$ but rather the subset of that subspace in which all entries in the vectors are in the interval $[0, 1]$. However this alters nothing in the process so we generally gloss over it.)

In other words the distance from $\bar{x}_?$ to this subspace can be thought of as a measurement of the "X-ishness" of $\bar{x}_?$:

Since the vectors $\bar{u}_1$ and $\bar{u}_2$ form an orthonormal basis for the subspace they span, calculating this distance is easy.

For example if we take our actual, proper, original X (stored in the vector $\bar{x}$) from the beginning of the chapter we find:

$$\begin{aligned}\text{dist}(\bar{x}_?, \text{span}(\{\bar{u}_1, \bar{u}_2\})) &= \left|\left|\bar{x}_? - \text{Proj}_{\text{span}(\{\bar{u}_1, \bar{u}_2\})}\bar{x}_?\right|\right| \\ &= ||\bar{x}_? - ((\bar{x}_? \cdot \bar{u}_1)\bar{u}_1 + (\bar{x}_? \cdot \bar{u}_2)\bar{u}_2)|| \\ &= 0.6340\end{aligned}$$

Of course distance is relative but we might conclude that this is close enough.

On the other hand if we take our actual, proper, original O (stored in the vector $\bar{o}$) from the beginning of the chapter we find:

$$\begin{aligned}\text{dist}(\bar{o}, \text{span}(\{\bar{u}_1, \bar{u}_2\})) &= \left|\left|\bar{o} - \text{Proj}_{\text{span}(\{\bar{u}_1,\bar{u}_2\})}\bar{o}\right|\right| \\ &= ||\bar{o} - ((\bar{o} \cdot \bar{u}_1)\bar{u}_1 + (\bar{o} \cdot \bar{u}_2)\bar{u}_2)|| \\ &= 0.9790\end{aligned}$$

So certainly our original X is more of an X than our original O is.

12.3.4 Comprehensive SVD Summary

The general approach will be to construct a basis (called a *character basis*) for every character in our alphabet and then for any unknown character examine how close it is to the subspace spanned by each character basis and pick the one it's closest to.

More specifically for any given character α we take some number n of representative versions of α and construct the matrix for each. Each of these matrices gets unrolled into a vector in $\mathbb{R}^m$ where m is the number of pixels in each version. Call these vectors

$$\bar{\alpha}_1,\ \bar{\alpha}_2,\ ...,\ \bar{\alpha}_n$$

Place these vectors into a matrix and find the singular value decomposition:

$$[\bar{\alpha}_1 \quad \bar{\alpha}_2 \quad ... \quad \bar{\alpha}_n] = U\Sigma V^T$$

From the resulting U we pick a reasonable collection of vectors as determined by the singular values. Let's say we pick k of them, and we form a basis which encapsulates what it means to be that character. These vectors form the *character basis* for α. For convenience we put them in a matris called the *character basis matrix*:

$$B(\alpha) = [\bar{u}_1 \quad \bar{u}_2 \quad ... \quad \bar{u}_k]$$

This basis then spans the subspace $\text{col}(B(\alpha))$ called the *character subspace*.

Now then, for an unknown other character unrolled into a vector $\bar{x}$ we can measure its distance to α by measuring how close the vector is to $\text{col}(B(\alpha))$. This is essentially asking how easy it is to construct $\bar{x}$ (the character) out of the essential building blocks that make up α.

This distance is calculated by:

$$\text{dist}_\alpha\beta = \left|\left|\bar{x} - \text{Pr}_{\text{col}(B(\alpha))}\bar{x}\right|\right|$$

Luckily there's a shortcut for this when the matrix $B(\alpha)$ has orthonormal columns which it does in this case because our matrix $B(\alpha)$ was taken as columns of U:

$$\begin{aligned}
\text{Pr}_{\text{col}(B(\alpha))}\bar{x} &= (\bar{u}_1 \cdot \bar{x})\bar{u}_1 + ... + (\bar{u}_k \cdot \bar{x})\bar{u}_k \\
&= \bar{u}_1^T\bar{x}\bar{u}_1 + ... + \bar{u}_k^T\bar{x}\bar{u}_k \\
&= [\bar{u}_1 \ \ ... \ \ \bar{u}_k] \begin{bmatrix} \bar{u}_1^T\bar{x} \\ \vdots \\ \bar{u}_k^T\bar{x} \end{bmatrix} \\
&= [\bar{u}_1 \ \ ... \ \ \bar{u}_k] \begin{bmatrix} \bar{u}_1^T \\ \vdots \\ \bar{u}_k^T \end{bmatrix} \bar{x} \\
&= B(\alpha)B(\alpha)^T\bar{x}
\end{aligned}$$

(Note: Line 3 follows line 2 directly from the definition of $A\bar{x}$ as the linear combination of the columns of A using the weights in $\bar{x}$.)

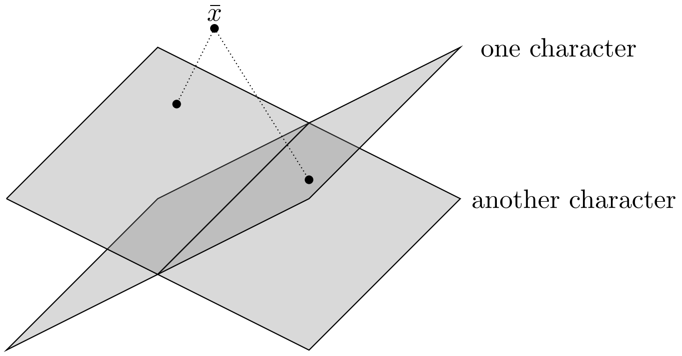
Consequently the distance between the unknown character vector and the known character basis is simply:

$$\left|\left|\bar{x} - B(\alpha)B(\alpha)^T\bar{x}\right|\right|$$

If we start the process with an entire collection of characters then an unknown character can be compared against each of them in turn and we can evaluate which one it's closest to.

More accurately we're taking each character basis for our known characters and seeing how close we can get to the unknown character using each character basis. The character basis that does the best job is the one we choose.

A crude visualization would be the following.



In reality there are many more subspaces, the dimension of those subspaces are higher and the dimension of the space they are in is also higher.

For example if we had 26 characters each using 16 × 16 resolution and if we used three columns of each U then we would have twenty-six (one per character) three-dimensional (basis for each) subspaces of $\mathbb{R}^{256}$ (256 pixels).

12.3.5 Choices

In the previous subsection we had to make two choices, the number of sample digits to use and the number of singular values to preserve. It's worth looking at this a bit more.

First, how many representative versions of a character should we take? It may be tempting to take as many as possible and as long as we don't take too many outliers (weird and esoteric version) then that's usually fine. If we use too many varied versions then the SVD gets confused about what's important in terms of variance.

Second, how many singular values should we preserve when we construct the character basis and character basis matrix? Again it may be temping to take more singular values but in reality this starts to cause problems. Why is this?

If our collection of characters is $m \times n$ (meaning we have n versions with m pixels each) then our U will be $m \times m$.

As we take more and more nonzero singular values, the character basis starts to span more and more of $\mathbb{R}^m$. As we do so, when we test a new character $\bar{d}$ the distance between $\bar{d}$ and $\text{span}(B(\alpha))$ starts to decrease. Consequently it becomes harder to compare a new character to a set of established characters.

Intuitively what's happening is that if we account for enough variation to draw every α then we have so many character basis vectors in A that every character (not just α) can potentially be built out of those character basis vectors. Consequently every character (not just α) begins to look like an α.

So figuring out how many singular values to preserve really comes down to testing and seeing what actually works best when tested in the real world and double-checked by another approach, such as a human.

12.3.6 Barebones Summary and Partial Example

The barebones summary goes like this:

(a) For each character α, β, ... ,ω that we'd like to have in our database, take a bunch of sample characters, unroll them into vectors, put those vectors in a matrix, find the SVD and take the most significant columns of U. This gives us our character basis matrices $B(\alpha)$, $B(\beta)$, ..., $B(\omega)$.

(b) Take a character we'd like to identify, unroll it into a vector $\bar{x}$, then check the values:

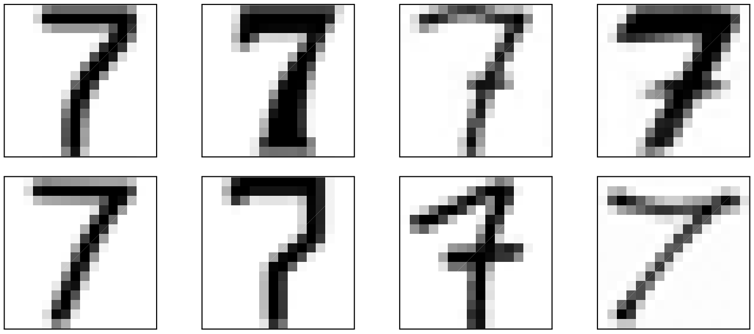
$$\begin{aligned}\text{Distance from } \bar{x} \text{ to } \alpha &= \left|\left|\bar{x} - B(\alpha)B(\alpha)^T\bar{x}\right|\right|\\ \text{Distance from } \bar{x} \text{ to } \beta &= \left|\left|\bar{x} - B(\beta)B(\beta)^T\bar{x}\right|\right|\\ \vdots &= \vdots\\ \text{Distance from } \bar{x} \text{ to } \omega &= \left|\left|\bar{x} - B(\omega)B(\omega)^T\bar{x}\right|\right|\end{aligned}$$

and choose the letter (α or β or ... or ω) corresponding to the smallest value.

Example 12.1. Suppose we wanted to do simple character recognition on the digits 0 through 9. We decide to use 16×16 resolution and we decide to preserve $k = 3$ columns of U.

First we would need to get some samples of each digit.

Here are eight different versions of the number 7 at a resolution of 16×16:



We convert each of these into a matrix, then unroll each to a vector (256 entries!), then put them all in a matrix (256×8) and find the SVD.

From the corresponding U we take the first $k = 3$ vectors and construct the character basis matrix:

$$B(7) = [\bar{u}_1 \;\; \bar{u}_2 \;\; \bar{u}_3]$$

This is a 256×3 matrix so we haven't explicitly written it out.

We would then repeat this for the digits 0,1,2,3,4,5,6,8,9 to get basis matrices $B(0)$, ..., $B(9)$ all together.

Now then, given a digit we'd like to identify we would unroll it into a vector $\bar{x}$ and check:

$$\begin{aligned}
\text{Distance from } \bar{x} \text{ to } 0 &= \left|\left|\bar{x} - B(0)B(0)^T\bar{x}\right|\right| \\
\text{Distance from } \bar{x} \text{ to } 1 &= \left|\left|\bar{x} - B(1)B(1)^T\bar{x}\right|\right| \\
\vdots &= \vdots \\
\text{Distance from } \bar{x} \text{ to } 9 &= \left|\left|\bar{x} - B(9)B(9)^T\bar{x}\right|\right|
\end{aligned}$$

and we would choose the digit (0 or 1 or ... or 9) with the smallest value.

As a side note for our 7 example above the singular values are

$$\{37.6566, 6.3112, 4.6349, 3.9108, 3.5626, 3.1247, 2.0267, 1.8722\}$$

So the first singular value captures most of the variance. If we take the corresponding $\bar{u}_1$ and multiply it by the appropriate scalar to get all values between 0 and 1 we get:

12.4 Comments

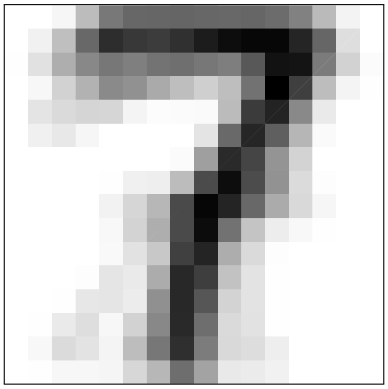
12.4.1 Visualizing the Basis

As we've commented earlier, the first column from U forms the fundamental building block for the character.

With the previous example here is $\bar{u}_1$ rolled into a matrix and scaled so all the values are between 0 and 1:

$$\begin{bmatrix}
1 & 1 & 0.96 & 0.74 & 0.47 & 0.41 & 0.4 & 0.39 & 0.4 & 0.41 & 0.4 & 0.42 & 0.51 & 0.73 & 0.95 & 1 \\
1 & 0.94 & 0.75 & 0.41 & 0.18 & 0.22 & 0.24 & 0.19 & 0.11 & 0.052 & 0.03 & 0.03 & 0.15 & 0.4 & 0.87 & 1 \\
0.99 & 0.88 & 0.68 & 0.53 & 0.47 & 0.5 & 0.45 & 0.44 & 0.47 & 0.5 & 0.34 & 0.07 & 0.079 & 0.48 & 0.82 & 0.98 \\
1 & 0.97 & 0.82 & 0.7 & 0.54 & 0.57 & 0.67 & 0.74 & 0.81 & 0.74 & 0.45 & 0 & 0.25 & 0.74 & 0.96 & 0.99 \\
1 & 0.89 & 0.85 & 0.83 & 0.84 & 0.95 & 0.99 & 0.98 & 0.98 & 0.73 & 0.23 & 0.14 & 0.54 & 0.92 & 1 & 1 \\
1 & 0.94 & 0.91 & 0.96 & 1 & 1 & 1 & 1 & 0.89 & 0.41 & 0.15 & 0.37 & 0.71 & 0.98 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.98 & 0.62 & 0.17 & 0.27 & 0.58 & 0.83 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0.99 & 0.94 & 0.93 & 0.74 & 0.3 & 0.053 & 0.3 & 0.57 & 0.86 & 0.99 & 1 & 1 \\
1 & 1 & 1 & 1 & 0.95 & 0.84 & 0.72 & 0.29 & 0.024 & 0.14 & 0.41 & 0.67 & 0.85 & 0.96 & 1 & 1 \\
1 & 1 & 1 & 1 & 0.97 & 0.83 & 0.68 & 0.29 & 0.05 & 0.45 & 0.77 & 0.93 & 0.97 & 0.99 & 1 & 1 \\
1 & 1 & 1 & 1 & 0.98 & 0.89 & 0.78 & 0.25 & 0.14 & 0.68 & 0.86 & 0.99 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0.99 & 0.9 & 0.91 & 0.67 & 0.16 & 0.25 & 0.76 & 0.89 & 0.99 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0.9 & 0.9 & 0.92 & 0.56 & 0.16 & 0.34 & 0.83 & 0.89 & 0.99 & 1 & 1 & 1 & 1 \\
1 & 1 & 0.92 & 0.87 & 0.97 & 0.82 & 0.53 & 0.16 & 0.43 & 0.85 & 0.88 & 0.99 & 1 & 1 & 1 & 1 \\
1 & 0.97 & 0.86 & 0.89 & 0.95 & 0.74 & 0.47 & 0.19 & 0.49 & 0.85 & 0.86 & 0.93 & 1 & 1 & 1 & 1 \\
1 & 0.99 & 0.97 & 0.97 & 0.96 & 0.84 & 0.72 & 0.43 & 0.64 & 0.91 & 0.92 & 0.94 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Here is is an image, we immediately see what we've got!



We might ask what the second and third columns of U look like but this question is much tricker. Because the first column of U is the primary building block it essentially adds the critical structure to the image, which is adding white where it needs to be (remember 0 is black and 1 is white, so we think of white as being added) and consequently we can scale the first column of U so that all values are nonnegative.

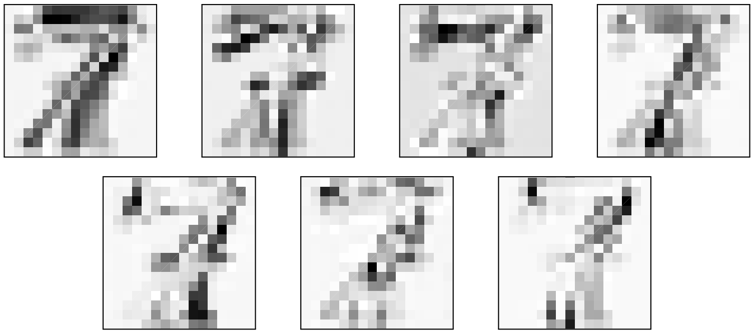
Subsequent columns add variation, however, and the values in those columns are usually a combination of positive and negative values.

For example here is $\bar{u}_2$ rolled into a matrix:

$$\begin{bmatrix}
0 & 0 & 0 & 0.05 & 0.12 & 0.13 & 0.13 & 0.13 & 0.13 & 0.12 & 0.11 & 0.11 & 0.12 & 0.07 & 0.01 & 0 \\
0 & -0.04 & -0.03 & 0.08 & 0.16 & 0.16 & 0.16 & 0.14 & 0.12 & 0.11 & 0.11 & 0.12 & 0.15 & 0.09 & 0.02 & 0 \\
-0.01 & -0.09 & -0.08 & -0.01 & 0.05 & 0.05 & 0.04 & 0.04 & 0.05 & 0.05 & 0.02 & 0.07 & 0.07 & -0.03 & -0.08 & -0.02 \\
0 & -0.01 & -0.02 & -0.03 & -0.02 & -0.06 & -0.1 & -0.08 & -0.06 & -0.09 & -0.1 & 0.03 & 0.07 & 0 & -0.03 & 0 \\
0 & -0.04 & -0.05 & -0.04 & -0.01 & -0.01 & -0.02 & -0.02 & -0.02 & -0.08 & -0.06 & 0.09 & 0.12 & 0.02 & 0 & 0 \\
0 & -0.03 & -0.04 & -0.02 & 0 & 0 & 0 & 0 & -0.04 & -0.11 & 0.03 & 0.16 & 0.09 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.02 & -0.11 & -0.03 & 0.15 & 0.14 & 0.05 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.03 & -0.03 & -0.1 & -0.06 & 0.11 & 0.12 & 0.05 & -0.03 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.02 & -0.05 & -0.09 & -0.06 & 0.1 & 0.12 & 0.1 & 0.01 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.05 & -0.1 & 0.06 & 0.12 & 0.12 & 0.09 & 0.02 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.02 & -0.1 & -0.03 & 0.12 & 0.11 & 0.08 & 0.06 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.01 & -0.09 & -0.07 & 0.04 & 0.13 & 0.1 & 0.07 & 0.04 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.08 & -0.09 & 0 & 0.07 & 0.13 & 0.09 & 0.06 & 0.04 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.08 & -0.11 & -0.03 & 0.01 & 0.08 & 0.13 & 0.08 & 0.05 & 0.05 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.03 & -0.12 & -0.1 & 0 & 0.03 & 0.11 & 0.12 & 0.06 & 0.05 & 0.06 & 0.02 & 0 & 0 & 0 & 0 \\
0 & -0.01 & -0.03 & -0.03 & 0 & 0.02 & 0.06 & 0.06 & 0.01 & 0.02 & 0.03 & 0.02 & 0 & 0 & 0 & 0
\end{bmatrix}$$

In terms of building the 7s, the negative values here are were white is subtracted (or black is added) and the positive values here are where white is added (or black is subtracted) in order to get additional variation.

However it can be interesting to take the absolute values of these entries which gives a sense of where the most and least variation occurs after $\bar{u}_1$ has been taken into account. If we do this for $\bar{u}_2$ through $\bar{u}_8$ and then scale the values between 0 and 1 we get the following images:



12.4.2 Additonal Miscellaneous

In real world applications this produces reasonable but not acceptable results. There are several additional things that would be done if this were being actually implemented, including but not limited to:

- Cleaning images.
- Scaling images.
- Emphasizing black/white distinction.

- Centering characters.
- Rotating characters during testing.

12.5 Matlab

There are not a lot of new Matlab requirements for this section other than a few basic notes.

If we have an image stored in a matrix we can unroll it to a vector with the `reshape` command:

```
>> A = [1 4 7;2 5 8;3 6 9]
A =
     1     4     7
     2     5     8
     3     6     9
>> A = reshape(A,[9 1])
A =
     1
     2
     3
     4
     5
     6
     7
     8
     9
```

and then we can roll it back up again:

```
>> A = reshape(A,[3 3])
A =
     1     4     7
     2     5     8
     3     6     9
```

If we have a bunch of image matrices (already read and converted) we can put them together in a matrix easily. For example let's assume that $D1$ through $D9$ contain unrolled matrices for nine versions of the digit 7. To put these as columns into a matrix:

```
>> M = [D1 D2 D3 D4 D5 D6 D7 D8 D9];
```

Alternately you can add the columns to the matrix as you go:

```
>> M = [];
>> M = [M D1];
>> M = [M D2];
>> M = [M D3];
```

and so on...

To find the SVD and extract the first three columns:

```
>> [U,S,V] = svd(M);
>> B = U(:,1:3);
```

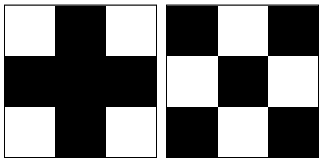
The notation here is that `:` takes all rows and `1:3` takes columns 1 through 3.

So now if we had another unrolled matrix `x` for a digit we could see how far it is from the above:

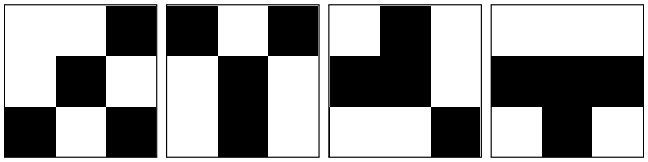
```
>> norm(x - B*transpose(B)*x)
```

12.6 Exercises

Exercise 12.1. Suppose your alphabet consists of the two 3×3 characters:

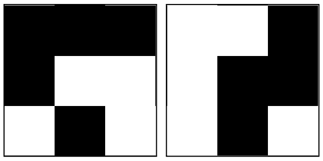


and you wish to identify each of the following characters:

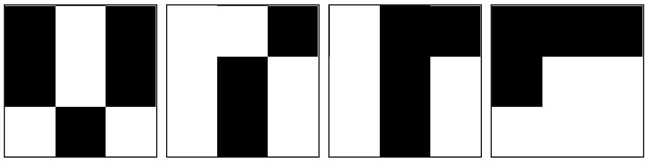


Convert these into matrices (white = 1 and black = 0) and unroll into vectors. Then identify each of the four characters using simple distance checking.

Exercise 12.2. Suppose your alphabet consists of the two 3×3 characters:

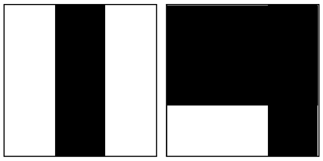


and you wish to identify each of the following characters:



Convert these into matrices (white = 1 and black = 0) and unroll into vectors. Then identify each of the four characters by using simple distance checking.

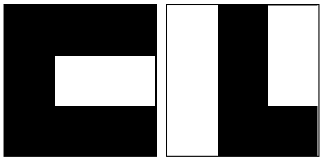
Exercise 12.3. Suppose your alphabet consists of the two 3×3 characters that look like a 1 and a 9 respectively:



(a) Design a character that looks like a 1 that will be recognized as a 9 using simple distance comparison. Show the calculations.

(b) Design a character that looks like a 9 that will be recognized as a 1 using simple distance comparison. Show the calculations.

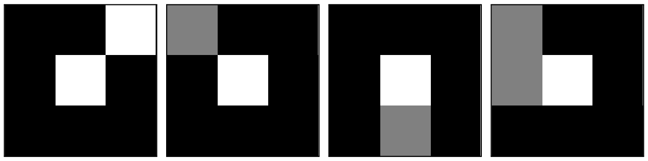
Exercise 12.4. Suppose your alphabet consists of the two 3×3 characters that look like a C and an L respectively:



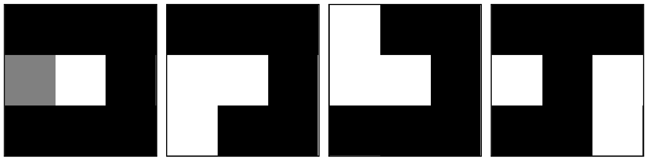
(a) Design a character that looks like a C that will be recognized as an L using simple distance comparison. Show the calculations.

(b) Design a character that looks like an L that will be recognized as a C using simple distance comparison. Show the calculations.

Exercise 12.5. Suppose you have the following four versions of the letter O:

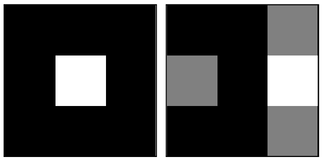


And you have the following four versions of the letter J:



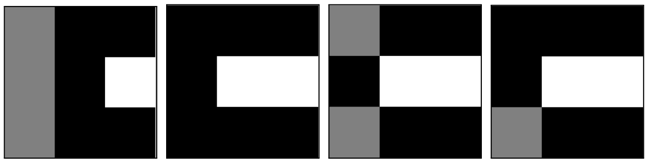
(a) Assuming values of 1.0 for white, 0.5 for gray and 0.0 for black, create letter basis matrices for O and for J using the two most important vectors from each associated U.

(b) Categorize the following two "unknown" characters by finding the distance between each character and the letter subspaces you constructed in (a).

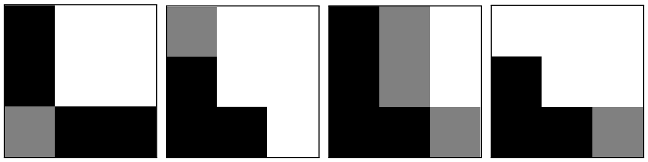


(c) For each letter basis matrix take the most important vector, multiply it by an appropriate scalar so all values lie between 0 and 1 with the largest value being 1, roll it back up to a matrix and plot. You are welcome to plot it by hand (reasonable shading) if you're not familiar enough with the Matlab commands. Do these look like an O and a J respectively?

Exercise 12.6. Suppose you have the following four versions of the letter C:

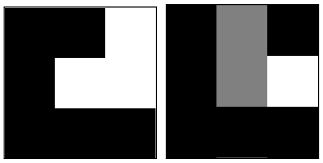


And you have the following four versions of the letter L:



(a) Assuming values of 1.0 for white, 0.5 for gray and 0.0 for black, create letter basis matrices for C and for L using the two most important vectors from each associated U.

(b) Categorize the following two "unknown" characters by finding the distance between each character and the letter subspaces you constructed in (a).



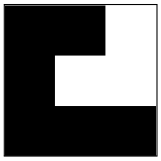
(c) For each letter basis matrix take the most important vector, multiply it by an appropriate scalar so all values lie between 0 and 1 with the largest value being 1, roll it back up to a matrix and plot. You are welcome to plot it by hand (reasonable shading) if you're not familiar enough with the Matlab commands. Do these look like an C and a L respectively?

Exercise 12.7. Suppose that in generating the basis matrix for one letter of your alphabet you took 256 versions of that letter with resolution 16×16 and just to be precise you took the entire of U as your letter basis matrix. Why would this be a bad idea?
Hint: Probably all the singular values will be distinct, why would this be a problem? Even if only most of them were, why would this be a problem?

Exercise 12.8. Suppose instead of using lots of different versions of the same letter to build a letter basis matrix you accidentally used the same version over and over.

(a) What would the letter basis matrix look like and why?

(b) If the letter looked like this 3×3 letter what would the leftmost vector in the letter basis matrix be?



Exercise 12.9. Suppose that your alphabet has two letters of size 2×2. When you build the letter basis matrices for them you find the following:

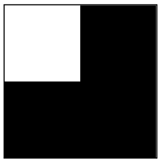
For the first letter when you do the SVD you find:

$$U = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & ... \\ \sqrt{2}/2 & \sqrt{2}/2 & ... \\ 0 & 0 & ... \\ 0 & 0 & ... \end{bmatrix}$$

For the second letter when you do the SVD you find:

$$U = \begin{bmatrix} \sqrt{3}/3 & \sqrt{2}/2 & ... \\ 0 & 0 & ... \\ \sqrt{3}/3 & -\sqrt{2}/2 & ... \\ \sqrt{3}/3 & 0 & ... \end{bmatrix}$$

Identify this letter as either the first letter or second letter:



Chapter 13

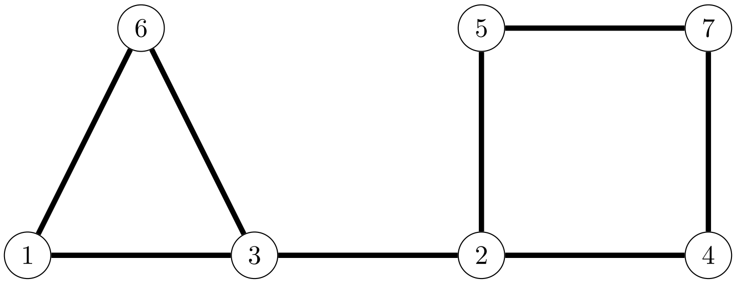
Graph Theory

Contents

13.1 Introduction

Let's get started with a simple example.

Example 13.1. Consider this picture which represents seven objects connected to one another:



This picture could represent a computer network, a network of friends, or the lines could represent roads between locations or borders between countries.

The study of structures like these is the heart of *graph theory* and in order to manage large graphs we need linear algebra.

13.2 Basic Definitions

Definition 13.2.0.1. A *graph* is a collection of *vertices* (nodes or points) connected by *edges* (line segments).

Definition 13.2.0.2. A graph is *simple* if has no multiple edges, (meaning two vertices can only be connected by one edge) and no loops (a vertex cannot have an edge connecting it to itself).

Definition 13.2.0.3. A graph is *connected* if it is in one single connected piece.

All the graphs we will look at will be simple connected graphs.

The example in the introduction is then a simple connected graph with seven vertices connected by eight edges.

Definition 13.2.0.4. The *degree* of a vertex is the number of edges connected to the vertex.

Definition 13.2.0.5. For a simple graph G with n vertices the *degree matrix* for G is the $n \times n$ diagonal matrix D such that d_{ii} equals the degree of the i^{th} vertex.

Definition 13.2.0.6. For a simple graph G the *adjacency matrix* is the symmetric matrix A such that a_{ij} equals 1 if vertices i and j are connected by an edge and 0 otherwise.

Definition 13.2.0.7. For a simple graph G the *Laplacian matrix* L is defined by $L = D - A$.

The term *Laplacian matrix* for a graph is actually very general. There are lots of different Laplacian matrices, this one is by far the most common and is technically the *unnormalized graph Laplacian matrix* but since it's the only one we will look at we will simply called it the *Laplacian matrix*.

Example 15.9 Revisited. For the graph given in the introduction we have:

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$L = D - A = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 2 \end{bmatrix}$$

Both the adjacency matrix and the Laplacian matrix contain all information about the graph and both can be used to analyze the graph.

13.3 Basic Graph Analysis

The adjacency matrix of a graph can give us some interesting facts about that graph.

Definition 13.3.0.1. A *walk* from vertex i to vertex j is an alternating series of connected vertices and edges that starts with vertex i and ends with vertex

j. There are no restrictions on repeating edges or vertices.

Theorem 13.3.0.1. If A is the $n \times n$ adjacency matrix of a graph with n vertices then for every integer $k \geq 1$, the ij-entry of A^k equals the number of walks of length k from vertex i to vertex j.

Proof. The proof proceeds by induction.

The $n = 1$ case is clear by definition of A.

Assume that the statement is true for A^k and look at the ij-entry of A^{k+1}. By the definition of matrix multiplication

$$(A^{k+1})_{ij} = (A^k)_{i1}a_{1j} + (A^k)_{i2}a_{2j} + ... + (A^k)_{in}a_{nj}$$

Since $(A^k)_{ik}$ equals the number of walks of length k from vertex i to vertex k and $a_{kj} = 1$ iff there is an edge from vertex k to vertex j (and 0 otherwise) it follows that the right side above equals the total number of walks of length $k + 1$ from vertex i to vertex j as desired. □

This Theorem gives us an interesting use of A^3. First, a definition:

Definition 13.3.0.2. The *trace* of a square matrix M, denoted $\text{tr}(M)$, equals the sum of the entries along the main diagonal.

Then we have the following:

Theorem 13.3.0.2. Thus the number of triangles in a graph equals $\frac{1}{6}\text{tr}(A^3)$.

Proof. A walk of length 3 from a vertex to itself is a triangle, and that triangle actually yields two walks, one in each direction. It follows that if a vertex i is contained in a triangle then $(A^3)_{ii} = 2$. From there we see that $\text{tr}(A^3)$ equals twice the number of vertices contained in triangles. However since each triangle contains three vertices it follows that $\text{tr}(A^3)$ equals six times the number of triangles. □

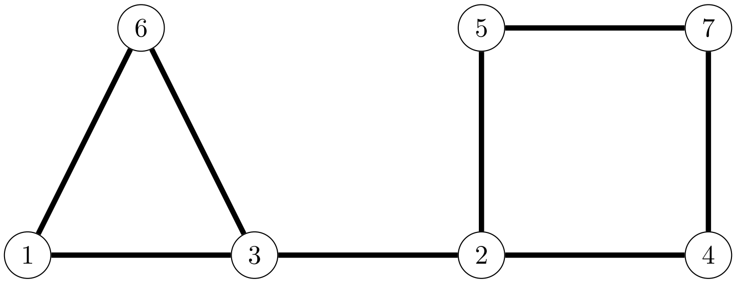
This same approach doesn't work for squares, pentagons, etc. Why not?

13.4 Graph Partitioning

13.4.1 Introduction to Partitioning

Consider the graph from the chapter opening:

Example 15.9 Revisited.



One way we might immediately describe this graph is that it is a square connected to a triangle. What we are doing when we see this is we are breaking the graph into those two subgraphs.

This process, of breaking a graph into two or more subgraphs, has generic uses when analyzing networks.

Consequently what we'd like to know is if there is a way of doing this easily.

In order to investigate we first need some more definitions.

Definition 13.4.1.1. Given a graph G with n vertices $V = \{1, 2, ..., n\}$ For an integer $k \geq 2$ a *k-partition* of G is an partition of the vertices into into k subsets V_1, ..., V_k such that the subsets do not overlap and their union is all of V. We will write $P = (V_1, V_2, ..., V_k)$. A 2-partition is often just called a *partition*.

Example 15.9 Revisited. For example the partition we intuitively saw with our starting graph could be denoted $P = (\{1, 3, 6\}, \{2, 4, 5, 7\})$.

Sometimes we describe a partition by describing which edge(s) would need to be removed in order to disconnect the graph into the resulting pieces. However we're not actually removing the edges, just indicating that they would do the job.

Example 15.9 Revisited. For example we might say that our starting graph's partition could be partitioned by removing the $(2, 3)$ edge.

Definition 13.4.1.2. For a partition $P = (V_1, V_2)$ of a graph G we define the *cut* of P, denoted $\text{cut}(P)$, as the number of edges joining a vertex in V_1 with a vertex in V_2.

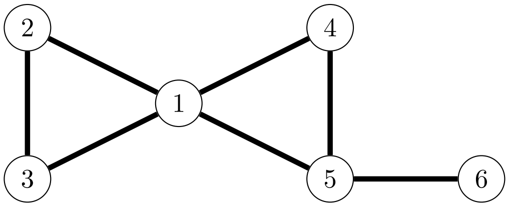
Example 15.9 Revisited. In our opening example we would have $\text{cut(P)} = 1$ because there is only one edge to count, the $(2, 3)$ edge.

So how might we want to partition a graph? One obvious way is:

Definition 13.4.1.3. A *minimum cut* is a partition P of a graph G in a manner that minimizes $\text{cut}(P)$. In other words it's the minimum number of edges we need to remove to partition the graph.

One problem with a minimum cut is that if there is a stray vertex connected to the rest of the graph by one edge then this would be a minimum cut. This tends to leave the subgraphs unbalanced which is somewhat unsatifactory.

Example 13.2. Consider this example:



A minimum cut can be achieved by removing the $(5, 6)$ edge. However the result (the bow-tie on the left and the single vertex on the right) isn't very satisfactory in a balanced sense.

The usual solution to this is to minimize $\text{cut}(P)$ with the added condition that we try to keep the number of vertices in each of the two remaining subgraphs as equal as possible.

In the above example we might remove the $(1, 4)$ and $(1, 5)$ edges which is more edges than just the $(5, 6)$ edge (two instead of one) but gains the advantage that the partition subsets have equal size.

It is this attempt at a balanced approach we will take, attempting to find a partition $P = (V_1, V_2)$ which minimizes $\text{cut}(P)$ while keeping $|V_1| \approx |V_2|$.

13.4.2 Introduction to the Fiedler Method

The Fiedler Method is an easy way to partition a graph. First we will state the method in its most fundamental form and give some simple examples. Lots of questions will remain unanswered.

Next we will will look at what the Fiedler method is actually doing. After that we can look at some more complicated examples.

Lastly we will go through a rigorous proof.

The Fiedler Method is named after Miroslav Fiedler, a Czech mathematician, who worked in graph theory and linear algebra. This method was presented by him in 1973.

13.4.3 Basic Fiedler Method

First, a few definitions and facts about the Laplacian Matrix $L = D - A$. The following are addressed in more detail later but let's just get them out right now.

Fact 13.4.3.1.

If G is a simple connected graph with n vertices and if L is the Laplacian matrix for G then L has n real eigenvalues satisfying

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$$

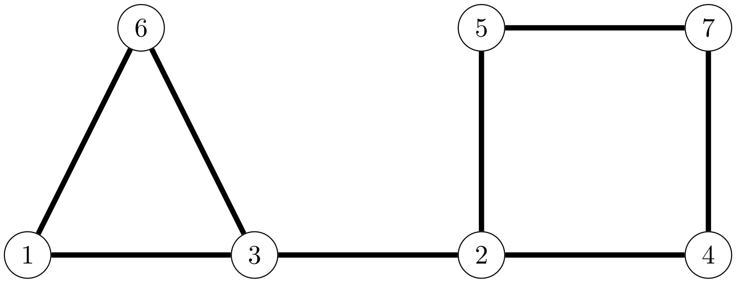
Definition 13.4.3.1. The *Fiedler Value* or the *algebraic connectivity* of a graph is the second smallest eigenvalue of its Laplacian matrix L.

The Fiedler Value gives a measurement as to how well connected the graph is. This value only has meaning when compared to something called the *vertex connectivity* which we won't go into.

Definition 13.4.3.2. A *Fiedler Vector* of a graph is an eigenvector corresponding to the Fiedler Value.

Notice that the eigenspace corresponding to the Fiedler Value may be multidimensional.

Example 15.9 Revisited. In our example:



we saw that:

$$L = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 2 \end{bmatrix}$$

the eigenvalues in order are:

$$0,\ 0.3588,\ 2.0000,\ 2.2763,\ 3.0000,\ 3.5892,\ 4.7757$$

Note that the Fiedler Value is 0.3588. A Fiedler Vector is an eigenvector corresponding to this. Any nonzero multiple of the following unit vector will suffice:

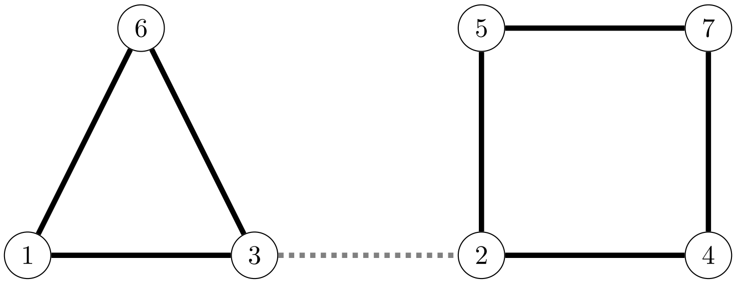
$$\bar{v} = \begin{bmatrix} 0.48 \\ -0.15 \\ 0.31 \\ -0.35 \\ -0.35 \\ 0.48 \\ -0.42 \end{bmatrix}$$

At its most basic, the Fiedler Method basically states that we can achieve a "reasonable" partition into two subgraphs by separating the vertices according to the sign of the values in a Fiedler Vector $\bar{v}$ where each entry corresponds to a vertex. This means we group together the vertices i with $v_i = +$ and we group together the vertices i with $v_i = -$. In the case that $v_i = 0$ we simply have to make a choice.

By "reasonable" we mean that an attempt is made to remove as few edges as possible while keeping the resulting subgraphs of approximately equal size.

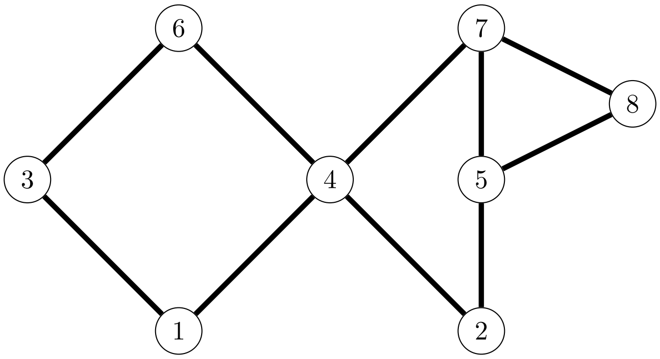
It's worth noting that the Fiedler Method is not perfect, as we'll see, but often the problems that arise can be easily accounted for.

Example 15.9 Revisited. In our example above $v_i = +$ for $i = 1, 3, 6$ and $v_i = -$ for $i = 2, 4, 5, 7$, so that $P = (\{1, 3, 6\}, \{2, 4, 5, 7\})$. This means we separate the vertices accordingly:



This is just what we predicted!

Example 13.3. Consider this graph:



We have

$$L = \begin{bmatrix} 2 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & 0 & 4 & 0 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 3 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

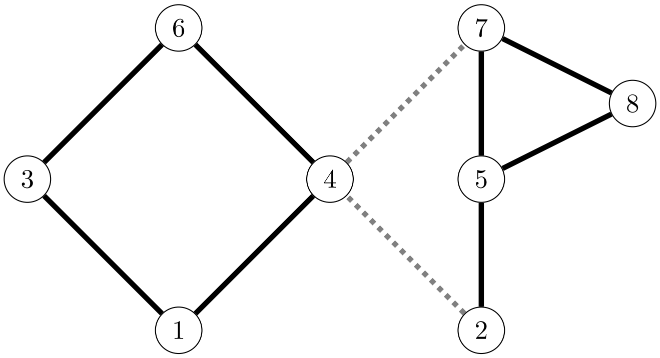
The eigenvalues in order are:

$$0,\ 0.4869,\ 1.6769,\ 2.0000,\ 2.7647,\ 3.4963,\ 4.0000,\ 5.5753$$

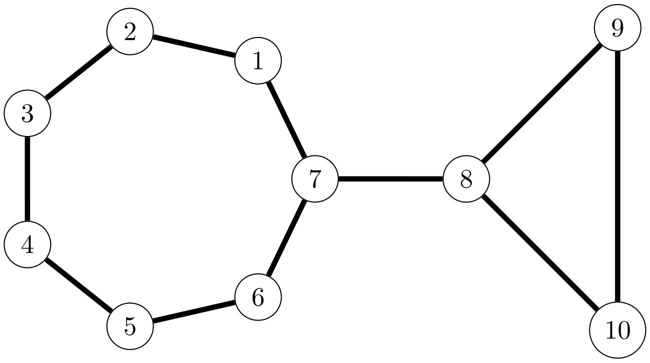
The Fiedler Value is therefore 0.4869. Since this is positive the graph is connected. A Fiedler Vector is an eigenvector corresponding to this.

$$\begin{bmatrix} 0.38 \\ -0.20 \\ 0.50 \\ 0.07 \\ -0.38 \\ 0.38 \\ -0.30 \\ -0.44 \end{bmatrix}$$

So a reasonable partition is achieved via $P = (\{1, 3, 4, 6\}, \{2, 5, 7, 8\})$. This requires removing the $(2, 4)$ and $(4, 7)$ edges:



Example 13.4. Consider this graph:



We have

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

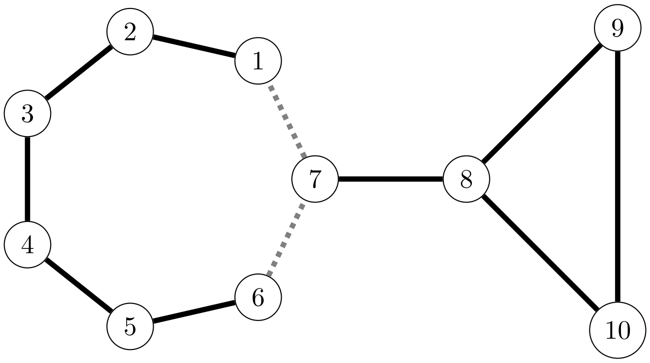
The eigenvalues in order are:

0, 0.2375, 0.7530, 1.0000, 2.4450, 2.5634, 3.0000, 3.4832, 3.8019, 4.7159

Thus the Fiedler Value is 0.2375. Since this is positive the graph is connected. A Fiedler Vector is an eigenvector corresponding to this. From Matlab:

$$\begin{bmatrix} 0.11 \\ 0.25 \\ 0.33 \\ 0.33 \\ 0.25 \\ 0.11 \\ -0.05 \\ -0.37 \\ -0.49 \\ -0.49 \end{bmatrix}$$

So a reasonable partition is achieved via $P = (\{1,2,3,4,5,6\},\{7,8,9,10\})$. This requires removing the $(1,7)$ and $(6,7)$ edges so $\text{cut}(P) = 2$.



Notice that obtaining a cut of 1 is possible but would leave a much more unbalanced graph. Notice that another partition actually does better than the

Fiedler Method, with $P = (\{1,2,3,4,5\},\{6,7,8,9,9\})$ having $\text{cut}(P) = 2$ and also $|V_1| = |V_2|$. We'll look at how this relates to the Fiedler Method later.

13.4.4 What are We Wishing For?

Earlier we commented that ideally for a partition $P = (V_1, V_2)$ of a graph G we would like to minimize $\text{cut}(P)$ while keeping $|V_1| \approx |V_2|$.

To formalize this first observe that a partition of a graph G with n vertices can be defined by choosing a vector $\bar{x} \in \mathbb{R}^n$ with each entry $x_i = \pm 1$. Having such a vector we can then create a partition by taking the vertices i with $x_i = +1$ as one subset and the vertices i with $x_i = -1$ as the other subset.

More formally

$$P = (\{i \,|\, x_i = +1\}, \{i \,|\, x_i = -1\})$$

Keeping the sizes of the subsets equal amounts to having $\sum\limits_{i=1}^{n} x_i = 0$ and keeping them close amounts to having $\sum\limits_{i=1}^{n} x_i \approx 0$

In what follows, the edge set of a graph only includes each edge once so for example if $(1,2) \in E$ then we don't count $(2,1)$ as different.

Lemma 13.4.4.1. For any partition $P = (V_1, V_2)$ of a graph G with edge set E we have

$$\text{cut}(P) = \frac{1}{4} \sum_{(i,j)\in E} (x_i - x_j)^2$$

Proof. Consider that

$$\begin{aligned}
\sum_{(i,j)\in E} (x_i - x_j)^2 &= \sum_{\substack{(i,j)\in E \\ x_i = -x_j}} (x_i - x_j)^2 + \sum_{\substack{(i,j)\in E \\ x_i = x_j}} (x_i - x_j)^2 \\
&= \sum_{\substack{(i,j)\in E \\ x_i = -x_j}} (\pm 2)^2 + \sum_{\substack{(i,j)\in E \\ x_i = x_j}} (0)^2 \\
&= 4\,\text{cut}(P)
\end{aligned}$$

□

The $\frac{1}{4}$ doesn't matter for minimizing so the goal can be rephrased as trying to minimize $\sum\limits_{(i,j)\in E} (x_i - x_j)^2$ with the conditions that $\sum\limits_{i=1}^{n} x_i \approx 0.$ and $x_i = \pm 1$.

Notice that this is computationally intensive and involves checking all possible combinations of the x_i.

For example if the graph has 10 vertices then there are $2^{10} = 1024$ possible $\bar{x}$.
and if the graph has 100 vertices then there are $2^{100} = 1267650600228229401496703205376$ possible $\bar{x}$.

In addition we need to decide how close we want $|V_1| \approx |V_2|$ when looking for a trade-off in minimizing the cut value.

What we do instead is relax the requirement somewhat.

13.4.5 What are We Getting?

Old Goal: Choose $\bar{x}$ to minimize $\sum\limits_{(i,j)\in E} (x_i - x_j)^2$ with the conditions that $\sum\limits_{i=1}^{n} x_i \approx 0$ and $x_i = \pm 1$.

New Goal: Choose $\bar{x}$ to minimize $\sum\limits_{(i,j)\in E} (x_i - x_j)^2$ with the conditions that $\sum\limits_{i=1}^{n} x_i = 0$ and $\sum\limits_{i=1}^{n} x_i^2 = n$.

Notice that the New Goal is a slightly weaker version of the Old Goal. The Old Goal would satisfy the New Goal but not necessarily the reverse.

What will the x_i values in this $\bar{x}$ mean? Consider the following observations:

- The first condition makes sure that all the x_i average to 0, meaning that they should be spread out around 0.
- The second condition prevents all the x_i from being too close to 0 and prevents any one x_i from being larger than $\sqrt{n}$.
- Minimizing the objective prevents any large gaps between clumps of x_i values. The reason for this is that the graph being connected guarantees that some vertex in the first clump must be connected to some vertex in the second clump and so a large gap between clumps would contribute a large value to the objective.
- From the previous bullets we can see that it's not reasonable to have a few values less than 0 and many values more than 0 (or the reverse) because these would not average out to 0 unless there were a large gap, which can't exist.
- If two vertices i and j are connected by an edge then minimizing the objective means keeping the corresponding x_i and x_j close so that they only contribute a small value to the objective.

- If two vertices i and j are not connected by an edge then the corresponding x_i and x_j can be further apart.

The practical upshot of all of this is that the x_i are spread over the interval $[-\sqrt{n}, \sqrt{n}]$ with no large gaps and with disconnected vertices having x_i values which tend further and with connected vertices having x_i values which tend closer.

What this means is that we can choose a place in $[-\sqrt{n}, \sqrt{n}]$ to split the vertices into two groups and that vertices in each group will tend to be clustered together.

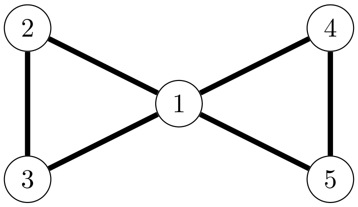
An obvious way to split is to take the vertices corresponding to positive values and those correspoding to negative values and to put 0 in either one group or the other but there are other choices, including splitting at the median.

13.4.6 More and Trickier Examples

We've seen what happens in clear-cut examples but the nature of the Fiedler Vector, whose values indicate a sort of connectedness, lends itself to more than just simple partitions. Now we will look at some examples in which:

- There is a 0 in the Fiedler Vector.
- Repeated values in the Fiedler Vector might yield choices.
- We might choose a k-partition with $k > 2$.
- The eigenspace corresponding to the Fiedler Value has dimension greater than 1.
- The Fiedler Vector can give insight into the graph's structure.
- The Fiedler Vector fails to be helpful at all!

Example 13.5. Consider this innocuous looking example:



The Laplacian matrix is

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

The eigenvalues in order are

$$0,\ 1,\ 3,\ 3,\ 5$$

so the Fiedler Value is 1. A Fiedler Vector is:

$$\begin{bmatrix} 0 \\ -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

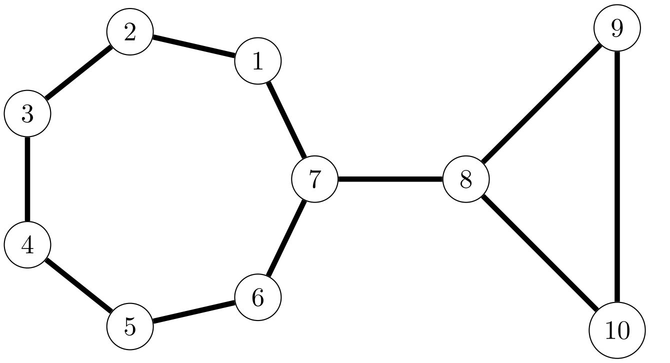
It's clear both from the graph and from the vector that the 1 vertex is difficult to categorize.

Even though the Fiedler Method doesn't explicitly tell us what to do with that vertex the way that the values are spread out makes our options fairly clear. We can either partition as $(\{2, 3, 1\}, \{4, 5\})$ or as $(\{2, 3\}, \{1, 4, 5\})$.

We can even see this sort of behavior (options!) arising when the Fiedler method does work.

Example 13.4 Revisited.

Consider the earlier example:

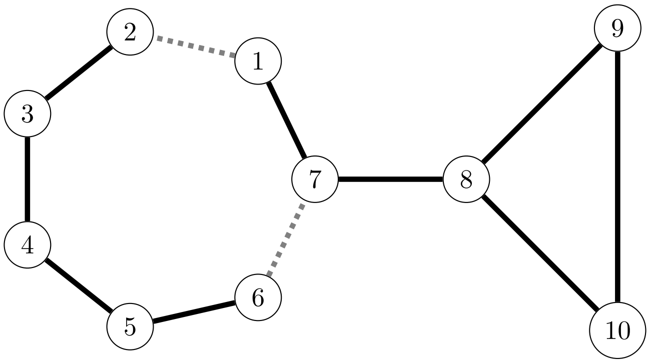


Here is the same Fiedler Vector we saw before except with the vector entries placed in increasing order (equal values chosen arbitrarily) with the vertex number (that is, the vector index) labeling each.

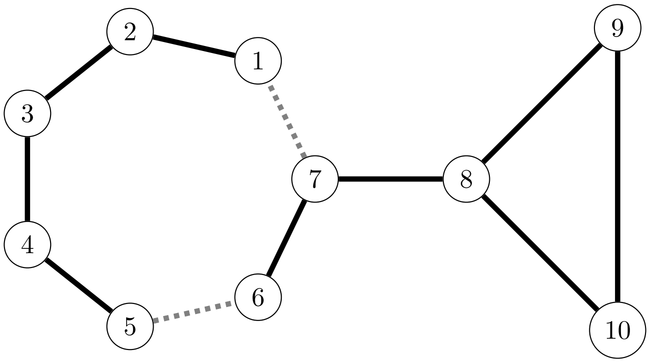
Vertex	Entry
9	-0.49
10	-0.49
8	-0.37
7	-0.05
1	0.11
6	0.11
2	0.25
5	0.25
3	0.33
4	0.33

Our choice to separate by the negative and positive values is a classic approach.

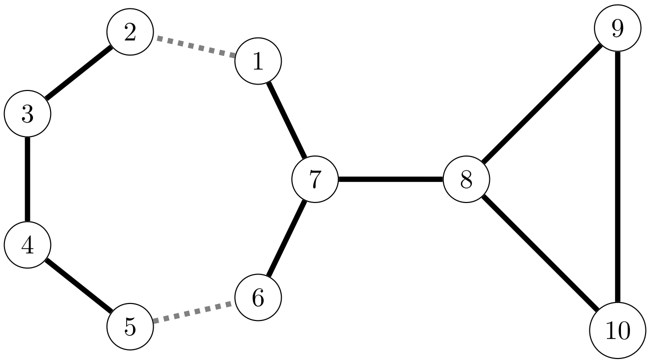
Another alternatives would be to take the smallest half of the entries. Since 0.11 appears twice we could split those two up, meaning we could take vertex 1 with the first half and overtex 6 with the second half, or the reverse, giving either: $(\{9, 10, 8, 7, 1\}, \{6, 2, 5, 3, 4\})$



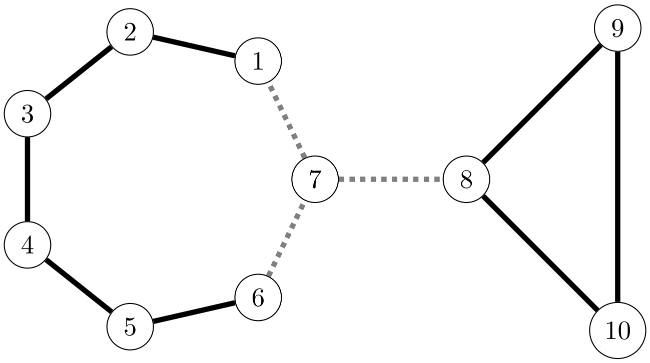
or $(\{9, 10, 8, 7, 6\}, \{1, 2, 5, 3, 4\})$



Or we could take both 1 and 6 with the first half, giving $(\{9, 10, 8, 7, 1, 6\}, \{2, 5, 3, 4\})$

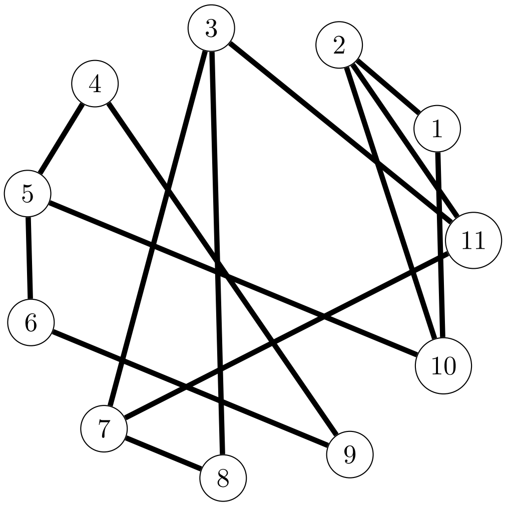


Or we could argue that since the 7 vertex has value very close to 0 perhaps it should just be left alone. Then we would partition into more than two subgraphs, giving $(\{9, 10, 8\}, \{7\}, \{1, 6, 2, 5, 3, 4\})$.



The Fiedler Vector can also help us figure out the structure of a graph which is not given in an obvious way.

Example 13.6. Consider this example:



The Laplacian matrix is not shown but the eigenvalues are

$$0.0000, 0.1483, 0.5858, 2.0000, 2.2170, 2.3820, 3.4142, 3.6913, 4.0000, 4.6180, 4.9434$$

and so the Fiedler Value is 0.1483.

The Fiedler Vector is:

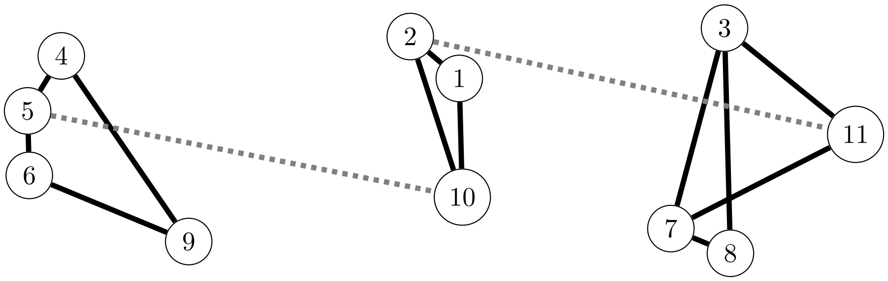
$$\begin{bmatrix} 0.0000 \\ 0.0726 \\ 0.3626 \\ -0.3626 \\ -0.2798 \\ -0.3626 \\ 0.3626 \\ 0.3917 \\ -0.3917 \\ -0.0726 \\ 0.2798 \end{bmatrix}$$

Sorted with the corresponding vertex numbers:

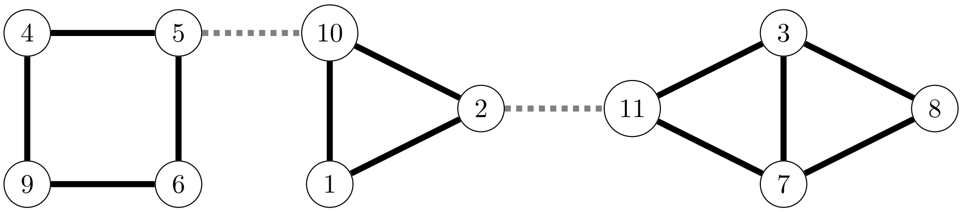
Vertex	Entry
9	-0.3917
6	-0.3626
4	-0.3626
5	-0.2798
10	-0.0726
1	0
2	0.0726
11	0.2798
7	0.3626
3	0.3626
8	0.3917

The values here are basically divided into three groupings according to the vertices $(\{9,4,6,5\},\{10,1,2\},\{11,3,7,8\})$ so it might make sense to partition the graph into three subgraphs.

Here is the graph redrawn with those groupings separated and with the cut edges as dotted lines. Basically I dragged the first group left and the third group right from the original graph. The underlying structure becomes much more clear now!

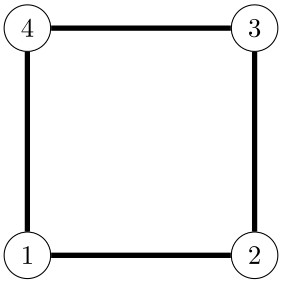


If we clean it up a bit:



Here's a particularly messy example which looks so nice at the start.

Example 13.7. Consider the simple square:



The Laplacian matrix for this graph is:

$$L = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

The eigenvalues are $\{0, 2, 2, 4\}$ so the Fiedler Value has multiplicity 2 and hence has a two dimensional subspace. This subspace is spanned by the two correponding vectors:

$$\begin{bmatrix} 0.7071 \\ 0 \\ -0.7071 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ 0.7071 \\ 0 \\ -0.7071 \end{bmatrix}$$

However this gives lots of confusing options:

If we use the first vector then vertices 1 and 3 are separate but depending on what we do with vertices 2 and 4 we could get either $(\{1\}, \{2, 3, 4\})$, $(\{1, 2\}, \{3, 4\})$, $(\{1, 4\}, \{2, 3\})$ or $(\{1, 2, 4\}, \{3\})$.

If we use the second vector then vertices 2 and 4 are separate but depending on what we do with vertices 1 and 3 we could get either $(\{2\}, \{1, 3, 4\})$, $(\{1, 2\}, \{3, 4\})$, $(\{2, 3\}, \{1, 4\})$ or $(\{1, 2, 3\}, \{4\})$.

Any linear combination using nonzero multiples of both vectors will lead to a Fiedler vector of the form:

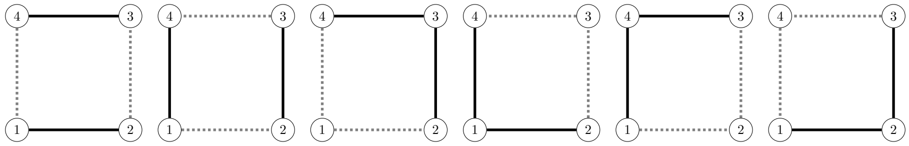
$$\begin{bmatrix} + \\ + \\ - \\ - \end{bmatrix} \text{ or } \begin{bmatrix} + \\ - \\ - \\ + \end{bmatrix} \text{ or } \begin{bmatrix} - \\ + \\ + \\ - \end{bmatrix} \text{ or } \begin{bmatrix} - \\ - \\ + \\ + \end{bmatrix}$$

These yield only the two partitions $(\{1, 2\}, \{3, 4\})$ and $(\{1, 4\}, \{2, 3\})$.

Thus in total there are six possibilities:

$$(\{1, 2\}, \{3, 4\}),\ (\{1, 4\}, \{2, 3\}),\ (\{1\}, \{2, 3, 4\}),\ (\{1, 2, 4\}, \{3\}),\ (\{2\}, \{1, 3, 4\}), \\ (\{1, 2, 3\}, \{4\})$$

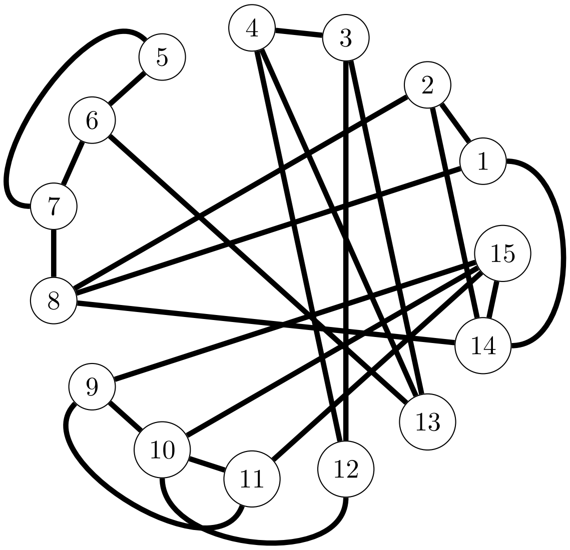
with corresponding pictures:



In this example the Fiedler method can't decide other than ensuring that either 1 and 3 are separate or 2 and 4 are separate, which actually seems reasonable, but beyond that options abound.

Here's an example where the Fiedler vector doesn't do the best job of partitioning the graph.

Example 13.8. Consider the graph:



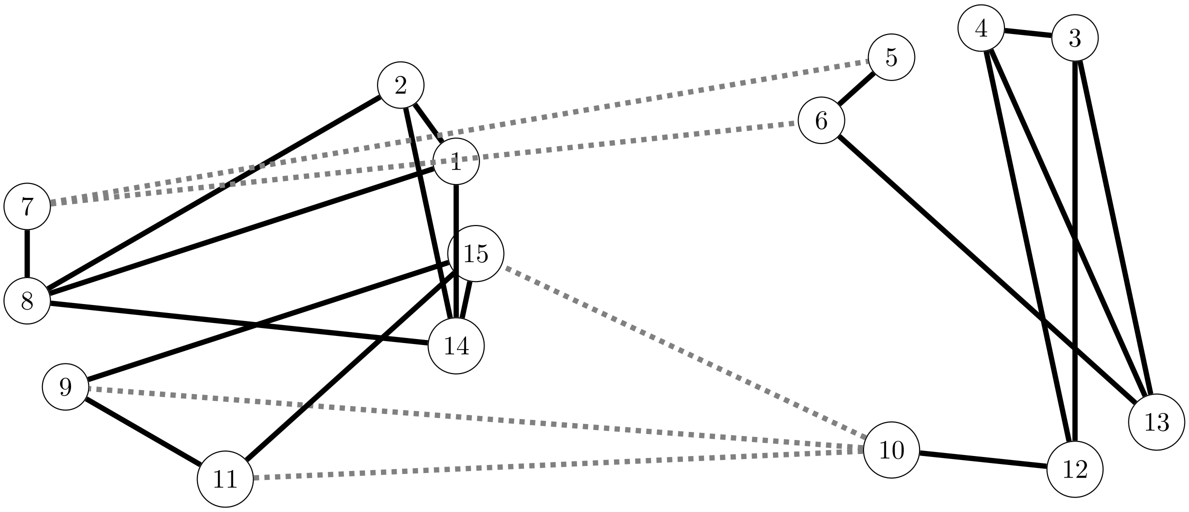
The Fiedler Value is 0.3424 and the Fiedler Vector is:

$$\begin{bmatrix} -0.3778 \\ -0.3778 \\ 0.3819 \\ 0.3819 \\ 0.0187 \\ 0.1027 \\ -0.0718 \\ -0.3121 \\ -0.0177 \\ 0.0520 \\ -0.0177 \\ 0.3070 \\ 0.3261 \\ -0.3142 \\ -0.0813 \end{bmatrix}$$

Sorted with the corresponding vertex numbers:

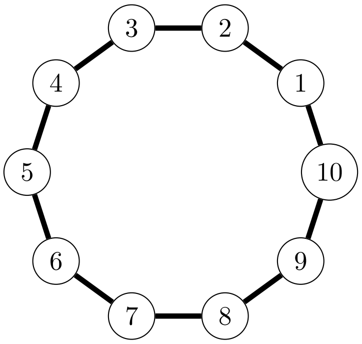
Vertex	Entry
1	-0.3778
2	-0.3778
14	-0.3142
8	-0.3121
15	-0.0813
7	-0.0718
9	-0.0177
11	-0.0177
5	0.0187
10	0.0520
6	0.1027
12	0.3070
13	0.3261
3	0.3819
4	0.3819

The Fiedler Method does a pretty mediocre job of dividing the graph into two subgraphs using $(\{1, 2, 14, 8, 15, 7, 9, 11\}, \{5, 10, 6, 12, 13, 3, 4\})$ as shown here:



13.4.7 Why Might the Fiedler Method Have Issues

The Fiedler vector tends to have issues with graphs in which it's difficult to measure distance between vertices. For example in the cycle:



It's clear that vertices 1 and 6 are far apart but are 1 and 2 close or not? Intuitively they are but by some measurement (around the wrong way) they're not. The mathematics in the Fiedler Method tends to stumble on things like this.

13.4.8 Why Does the Fiedler Vector Do This?

The final thing we need to address is why the Fiedler Method accomplishes our relaxed goal from a mathematical standpoint.

Lemma 13.4.8.1. Let G be a graph with n vertices and let L be its Laplacian matrix. Then L is orthogonally diagonalizable and the eigenvalues are all nonnegative and hence there exists an orthonormal basis of eigenvectors $\bar{v}_1$, $\bar{v}_2$, ..., $\bar{v}_n$ corresponding to eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

Proof. Since L is symmetric most of this follows from the Spectral Theorem. Proving that the eigenvalues are all nonnegative takes a bit more work but is omitted. □

From here on whenever we discuss the eigenvalues and eigenvectors of a Laplacian matrix for a graph we'll assume that it is an orthonormal basis from above.

Lemma 13.4.8.2. Let G be a connected graph. Then $\lambda_2 > 0$.

Proof. Omitted. While this is not difficult it takes a bit of time to write down and the proof is largely unrelated to and doesn't provide any insight into how we use it. □

Lemma 13.4.8.3. Let G be a graph with n vertices and let L be its Laplacian matrix. Then we have $\lambda_1 = 0$ and $\bar{v}_1 = \frac{1}{\sqrt{n}}\bar{1}$.

Proof. Since each row of L adds to 0 we have $L\bar{1} = \bar{0}$ and so $L\bar{1} = 0\bar{1}$ and so 0 is an eigenvalue with eigenvector $\bar{1}$ and hence with unit eigenvector $\frac{1}{\sqrt{n}}\bar{1}$. □

Lemma 13.4.8.4. Let G be a graph with n vertices and let L be its Laplacian matrix. For any eigenvalue $\lambda > 0$ of L the entries in any corresponding eigenvector $\bar{v}$ add to 0.

Proof. Let the entries of L be a_{ij}. If $A\bar{v} = \lambda\bar{v}$ then we have

$$\begin{aligned} a_{11}v_1 + a_{12}v_2 + ... + a_{1n}v_n &= \lambda v_1 \\ a_{21}v_1 + a_{22}v_2 + ... + a_{2n}v_n &= \lambda v_2 \\ ... &= ... \\ a_{n1}v_1 + a_{n2}v_2 + ... + a_{nn}v_n &= \lambda v_n \end{aligned}$$

The sum of this system on the left and right yields:

$$(a_{11} + a_{21} + ... + a_{n1})v_1 + (...)v_2 + ... + (...)v_n = \lambda(v_1 + ... + v_n)$$

Since the columns of L sum to zero the left size is zero and hence $v_1 + ... + v_n = 0$. □

Lemma 13.4.8.5. Let G be a graph with n vertices and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum\limits_{i=1}^{n} x_i = 0$ then $\bar{x} = \sum\limits_{i=2}^{n} w_i\bar{v}_i$ for appropriate w_i.

Proof. We know that since $\bar{v}_1, \bar{v}_2, ..., \bar{v}_n$ forms a basis for $\mathbb{R}^n$ that for appropriate w_i we may write:

$$\begin{aligned}
\bar{x} &= \sum_{i=1}^{n} w_i \bar{v}_i \\
&= w_1 \bar{v}_1 + \sum_{i=2}^{n} w_i \bar{v}_i \\
&= w_1 \frac{1}{\sqrt{n}} \bar{1} + \sum_{i=2}^{n} w_i \bar{v}_i
\end{aligned}$$

Now then we know that $\bar{x}$ and $\bar{v}_i$ (and hence $w_i \bar{v}_i$) are all in the subspace of $\mathbb{R}^n$ consisting of vectors whose entries add to 0. Consequently the entries of $w_1 \frac{1}{\sqrt{n}} \bar{1}$ must all add to 0 because subspaces are closed under linear combinations. But this implies $w_1 = 0$ as desired. □

Lemma 13.4.8.6. Let G be a graph with n vertices and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum_{i=1}^{n} x_i = 0$ then we have

$$\sum_{i=1}^{n} x_i^2 = \sum_{i=2}^{n} w_i^2$$

Proof. Observe that:

$$\begin{aligned}
\sum_{i=1}^{n} x_i^2 &= \bar{x}^T \bar{x} \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i \right]^T \left[\sum_{i=2}^{n} w_i \bar{v}_i \right] \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i^T \right] \left[\sum_{i=2}^{n} w_i \bar{v}_i \right] \\
&= \sum_{i=2}^{n} \sum_{j=2}^{n} w_i \bar{v}_i^T w_j \bar{v}_j \\
&= \sum_{i=2}^{n} \sum_{j=2}^{n} w_i w_j \bar{v}_i^T \bar{v}_j \\
&= \sum_{i=2}^{n} w_i^2
\end{aligned}$$

□

Lemma 13.4.8.7. Let G be a graph with n vertices and let E be the set of all edges of G. For any vector $\bar{x}$ we have

$$\bar{x}^T A\bar{x} = \sum_{(i,j)\in E} 2x_i x_j$$

Proof. We know that for any $\bar{x}$ by calculation that

$$\bar{x}^T A\bar{x} = \sum_{1\le i\le n, 1\le j\le n} a_{ij} x_i x_j$$

Since $a_{ij} = 1$ iff there is an edge between vertex i and vertex j and 0 otherwise that:

$$\bar{x}^T A\bar{x} = \sum_{(i,j)\in E} 2x_i x_j$$

Here the 2 appears because each pair i, j appears twice in the original sum but we're only counting it once in the set of all edges. □

Lemma 13.4.8.8. Let G be a graph with n vertices, let D be its degree matrix, and let E be the set of all edges of G. For any vector $\bar{x}$ we have

$$\bar{x}^T D\bar{x} = \sum_{(i,j)\in E} (x_i + x_j)^2$$

Proof. Let V be the set of all vertices of G. We know by straightforward calculation that

$$\bar{x}^T D\bar{x} = \sum_{i\in V} d_i x_i^2$$

An alternate way to calculate the degree of any vertex would be to look over the set of all edges and for each edge contribute +1 to the degree of each of the two vertices it connects. In order to have the total coefficient of each x_i^2 be the degree of vertex i this means that when we sum over all edges each edge between vertices i and j must contribute $+x_i^2 + x_j^2$ to the total sum.

Thus as desired

$$\bar{x}^T D\bar{x} = \sum_{i\in V} d_i x_i^2 = \sum_{(i,j)\in E} (x_i^2 + x_j^2)$$

□

Lemma 13.4.8.9. Let G be a graph with n vertices, let D be its degree matrix, let E be the set of all edges of G, and let L be its Laplacian matrix. Then for any vector $\bar{x}$ we have

$$\bar{x}^T L\bar{x} = \sum_{(i,j)\in E} (x_i - x_j)^2$$

Proof. We have:

$$\begin{aligned}
\bar{x}^T L\bar{x} &= \bar{x}^T(D-A)\bar{x} \\
&= \bar{x}^T D\bar{x} - \bar{x}^T A\bar{x} \\
&= \sum_{(i,j)\in E} (x_i^2 + x_j^2) - \sum_{(i,j)\in E} 2x_i x_j \\
&= \sum_{(i,j)\in E} (x_i - x_j)^2
\end{aligned}$$

□

Lemma 13.4.8.10. Let G be a graph with n vertices, let E be the set of all edges of G, and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum x_i = 0$ then we have:

$$\sum_{(i,j)\in E} (x_i - x_j)^2 = \sum_{i=2}^{n} w_i^2 \lambda_i$$

Proof. Observe that:

$$\begin{aligned}\sum_{(i,j)\in E}(x_i - x_j)^2 &= \bar{x}^T L \bar{x}\\ &= \left[\sum_{i=2}^{n} w_i \bar{v}_i\right]^T L \left[\sum_{i=2}^{n} w_i \bar{v}_i\right]\\ &= \left[\sum_{i=2}^{n} w_i \bar{v}_i^T\right] L \left[\sum_{i=2}^{n} w_i \bar{v}_i\right]\\ &= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i \bar{v}_i^T L w_j \bar{v}_j\\ &= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \bar{v}_i^T L \bar{v}_j\\ &= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \bar{v}_i^T \lambda_j \bar{v}_j\\ &= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \lambda_j \bar{v}_i^T \bar{v}_j\\ &= \sum_{i=2}^{n} w_i^2 \lambda_i\end{aligned}$$

□

Theorem 13.4.8.1. The entries in a Fiedler Vector obtain the desired goal.

Proof. The goal is to select $\bar{x}$ which minimizes $\sum\limits_{(i,j)\in E}(x_i - x_j)^2$ with the conditions that $\sum\limits_{i=1}^{n} x_i^2 = 1.$ and $\sum\limits_{i=1}^{n} x_i = 0$. Accordingly this means we wish to mimize $\sum\limits_{i=2}^{n} w_i^2 \lambda_i$ with the conditions that $\sum\limits_{i=2}^{n} w_i^2 = n$ and $\sum\limits_{i=1}^{n} x_i = 0$.

Given that $\lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$, this will be accomplished by setting $w_2 = \sqrt{n}$ and $w_3 = ... = w_n = 0$.

From here we get $\bar{x} = \sqrt{n}\bar{v}_2$. Since this $\bar{x}$ is an eigenvector of $\bar{L}$ corresponding to λ_2 the entries add to 0.

This vector is a Fiedler vector. Of course since $\bar{v}_2$ is simply a multiple of this which scales the values, we can use $\bar{v}_2$ itself instead.

□

13.5 Matlab

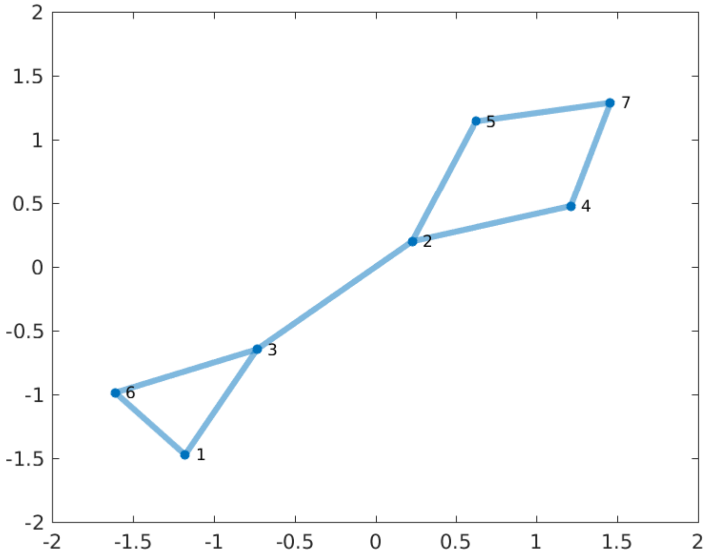
Matlab can plot a graph from the adjacency matrix. It does a pretty reasonable job of arranging the vertices so the graph is comprehensible. First, the following function m-file will create the adjacency matrix for a graph given a matrix of edges and a total number of vertices:

```
function M = createadjacency(v,n)
  % Create the Adjacency Matrix for a graph.
  % Usage:
  % createadjacency([1,2;2,3;1;4],5)
  % Will create a graph with 5 vertices
  % and edges joining 1-2, 2-3 and 1-4.
  M = zeros(n,n);
  for i = 1:length(v)
    M(v(i,1),v(i,2)) = 1;
    M(v(i,2),v(i,1)) = 1;
  end
end
```

In order to plot this graph in Matlab we first create the graph object and then we plot it. Here's the example which started the chapter:

```
>> A = createadjacency([1,3;1,6;3,6;3,2;2,5;5,7;7,4;2,4],7);
>> G = graph(A);
>> plot(G,'LineWidth',3)
```

This produces the image where I've thickened the lines a bit. There's currently no easy way to change the label size in Matlab.



The following function m-file will create the Laplacian matrix for a graph. It's just a slight modification on the one above:

```
function M = createlaplacian(v,n)
  % Create the Laplacian Matrix for a graph.
  % Usage:
  % createlaplacian([1,2;2,3;1;4],5)
  % Will create a graph with 5 vertices
  % and edges joining 1-2, 2-3 and 1-4.
  M = zeros(n,n);
  for i = 1:length(v)
    M(v(i,1),v(i,2)) = -1;
    M(v(i,2),v(i,1)) = -1;
  end
  for i = 1:n
    M(i,i) = -1*sum(M(:,i));
  end
end
```

Then we can find a Fiedler Vector easily. Here we examine the eigenvalues first, notice that the Fieder Value has multiplicity one so we can take any multiple of the corresponding eigenvector, we just look at the eigenvector Matlab gives:

```
>> L = createlaplacian([1,3;1,6;3,6;3,2;2,5;5,7;7,4;2,4],7);
>> [p,d] = eig(L);
>> diag(d)
ans =
   -0.0000
    0.3588
    2.0000
    2.2763
    3.0000
    3.5892
    4.7757
>> p(:,2)
ans =
    0.4801
   -0.1471
    0.3078
   -0.3482
   -0.3482
    0.4801
   -0.4244
```

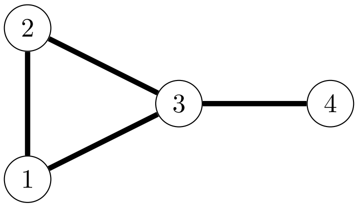
To order this vector and attach the index numbers is easy too:

```
>> L = createlaplacian([1,2;1,10;2,10;2,11;3,7;3,8;3,11;
4,5;4,9;5,6;5,10;6,9;7,8;7,11],11);
>> [p,d] = eig(L);
>> v = p(:,2);
>> sortrows(horzcat(v,[1:size(v)]’))
ans =
   -0.3917    9.0000
   -0.3626    6.0000
   -0.3626    4.0000
   -0.2798    5.0000
   -0.0726   10.0000
    0.0000    1.0000
    0.0726    2.0000
    0.2798   11.0000
    0.3626    7.0000
    0.3626    3.0000
    0.3917    8.0000
```

The `[1:size(v)]` command creates a horizontal vector with entries `1` up to the length of `v`. The `’` does the transpose so it’s vertical just like `v`. The `horzcat` command concatenates them horizontally, putting them together. The `sortrows` command sorts each row by the first column.

13.6 Exercises

Exercise 13.1. Consider the following graph:

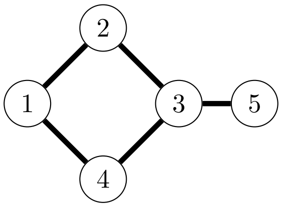


(a) Find the number of walks of length 3 from vertex 1 to vertex 3.

(b) Find the number of walks of length 20 from vertex 2 to vertex 4.

(c) There are no walks of length 3 from vertex 4 to itself. Rather than using A^3, explain intuitively why this is.

Exercise 13.2. Consider the following graph:



(a) Find the number of walks of length 3 from vertex 1 to vertex 2.

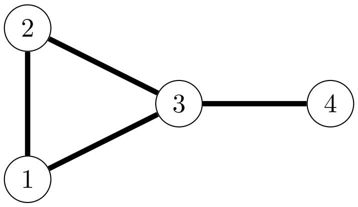
(b) Find the number of walks of length 10 from vertex 2 to vertex 4.

(c) Examine the number of walks of length k from vertex 3 to vertex 5 for various even k. What do you notice? Give an intutive explanation for this.

(d) Examine the number of walks of length k from vertex 2 to vertex 4 for various odd k. What do you notice? Give an intutive explanation for this.

Exercise 13.3. A small theorem in the book shows that the number of triangles in a graph G equals $\frac{1}{6}\text{tr}(A^3)$, where A is the adjacency matrix for G. Why does this not work for squares, etc.? In other words why does the number of squares not equal some multiple of $\text{tr}(A^4)$, why does the number of pentagons not equal some multiple of $\text{tr}(A^5)$, and so on?

Exercise 13.4. Consider the following graph:

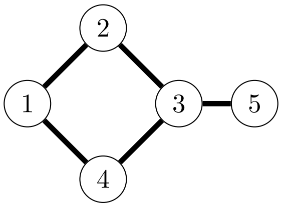


(a) Intuitively how would you partition the graph into two subgraphs in a reasonable manner?

(b) Apply the Fiedler Method to partition the graph.

(c) Do the results match?

Exercise 13.5. Consider the following graph:

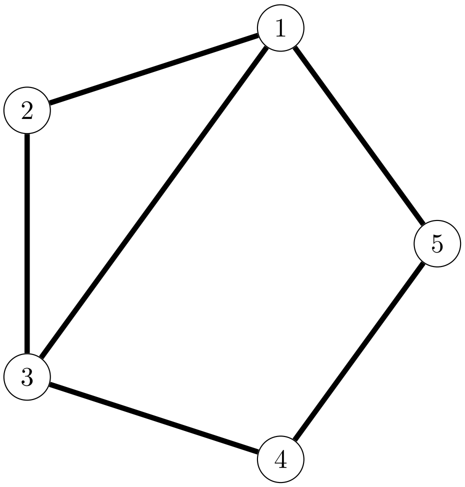


(a) Intuitively how would you partition the graph into two subgraphs in a reasonable manner?

(b) Apply the Fiedler Method to partition the graph.

(c) Do the results match?

Exercise 13.6. Consider the following graph:



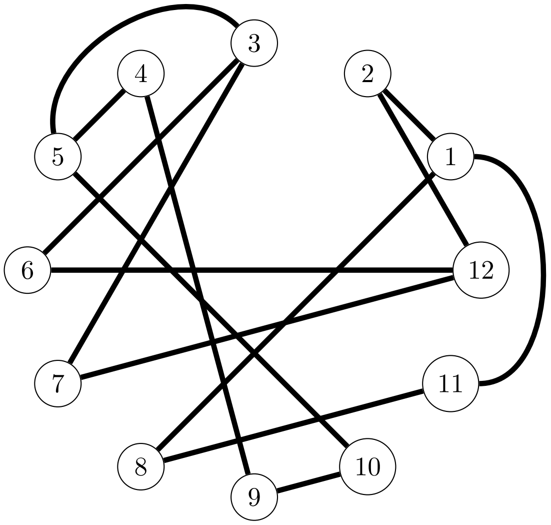
(a) Write down the Laplacian matrix for this graph.

(b) This matrix has eigenvalues 0, 1.3820, 2.3820, 3.6180, 4.6180 with corresponding eigenvectors

$$\begin{bmatrix} -0.447 \\ -0.447 \\ -0.447 \\ -0.447 \\ -0.447 \end{bmatrix}, \begin{bmatrix} -0.195 \\ -0.632 \\ -0.195 \\ 0.512 \\ 0.512 \end{bmatrix}, \begin{bmatrix} 0.372 \\ 0 \\ -0.372 \\ -0.602 \\ 0.602 \end{bmatrix}, \begin{bmatrix} 0.512 \\ -0.632 \\ 0.512 \\ -0.195 \\ -0.195 \end{bmatrix}, \begin{bmatrix} 0.602 \\ 0 \\ -0.602 \\ 0.372 \\ -0.372 \end{bmatrix}$$

Using this, partition the graph with the Fiedler method.

Exercise 13.7. Consider the following graph:



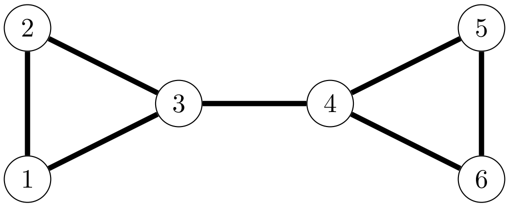
The Fiedler vector is:

$$[0.37, 0.23, -0.12, -0.34, -0.28, -0.04, -0.03, 0.42, -0.36, -0.34, 0.42, 0.06]^T$$

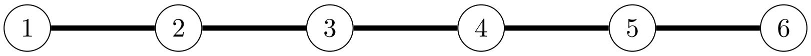
Use this vector to partition the graph into three components and then use this to draw a more understandable picture of the graph.

Exercise 13.8. Without doing any calculation match the following graphs with their Fiedler Vectors. Explain your decision.

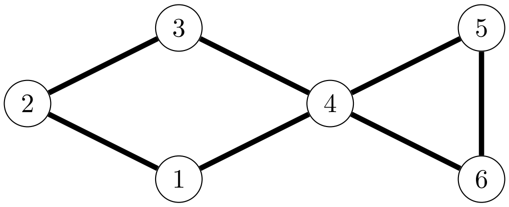
$G1$ shown here:



$G2$ shown here:



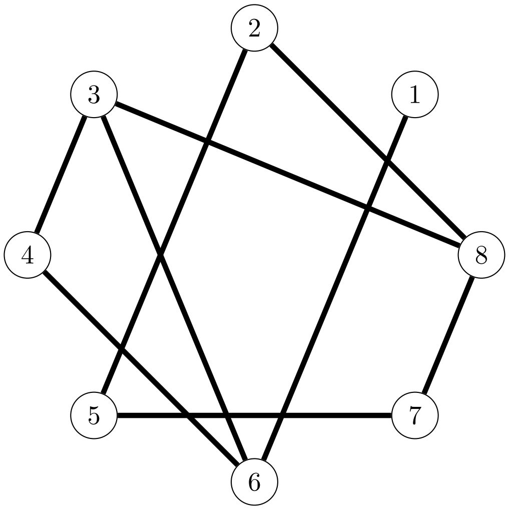
$G3$ shown here:



with:

$$\bar{v} = \begin{bmatrix} -0.56 \\ -0.41 \\ -0.15 \\ 0.15 \\ 0.41 \\ 0.56 \end{bmatrix} \text{ and } \bar{w} = \begin{bmatrix} 0.46 \\ 0.46 \\ 0.26 \\ -0.26 \\ -0.46 \\ -0.46 \end{bmatrix} \text{ and } \bar{x} = \begin{bmatrix} 0.31 \\ 0.39 \\ 0.31 \\ 0.09 \\ -0.39 \\ -0.70 \end{bmatrix}$$

Exercise 13.9. Consider the following graph:

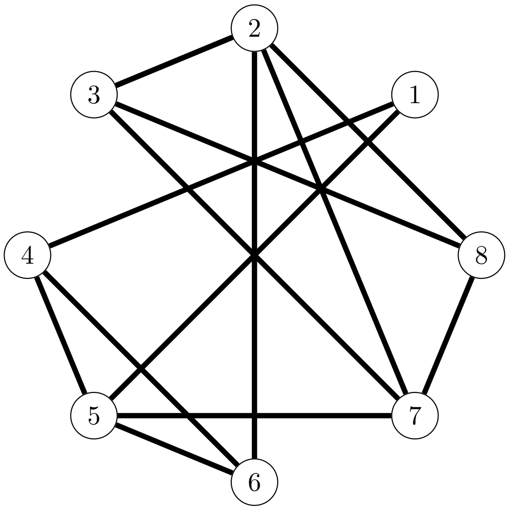


(a) Use the Fiedler Method to partition the graph.

(b) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(c) From the previous step are there any insights you gain about the structure of the graph?

Exercise 13.10. Consider the following graph:

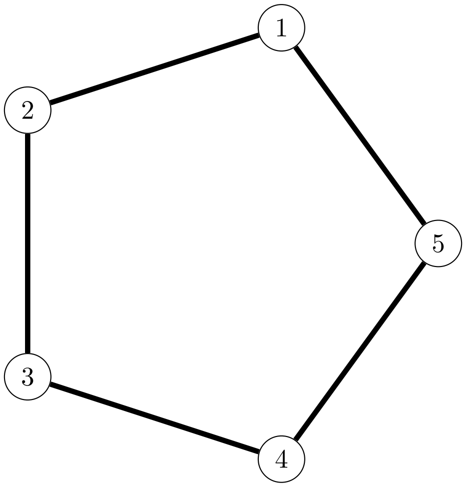


(a) Use the Fiedler Method to partition the graph.

(b) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(c) From the previous step are there any insights you gain about the structure of the graph?

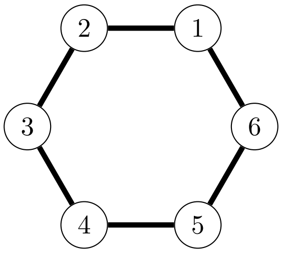
Exercise 13.11. Consider the following graph:



(a) Write down the Laplacian Matrix for the graph.

(b) The Fiedler Vectors span a two-dimensional subspace. Analyze all possible partitions which result. Be methodical.

Exercise 13.12. Consider the following graph:



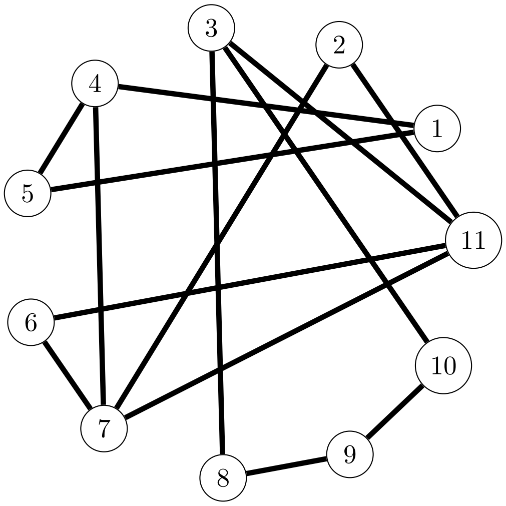
(a) The Fiedler Value has a two-dimensional corresponding eigenspace. Find a basis $\{\bar{v}_1, \bar{v}_2\}$ for the set of all Fiedler Vectors

(b) Any nonzeo linear combination of $\bar{v}_1$ and $\bar{v}_2$ will give a reasonable partition using the Fiedler Method. Experiment to see how many different partitions you can find.

(c) Suppose $\bar{v} = c_1\bar{v}_1 + c_2\bar{v}_2$ for constants c_1, c_2. Assuming neither of the 1-entry and 4-entry of v are 0 explain why vertices 1 and 4 will be in different subgraphs. Repeat for the 2-entry and 5-entry and for the 3-entry and 6-entry.

Exercise 13.13. Let L_n be the Laplacian Matrix for the complete graph K_n (the graph with n vertices with edges between all pairs). By testing various values of n make an educated guess about the eigenvalues of L_n for any n.

Exercise 13.14. Consider the following graph:



(a) Find a Fielder vector.

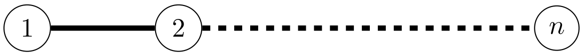
(b) The values in this Fiedler Vector, when sorted, can be grouped into three separated subsets. Do so.

(c) Use this grouping to partition the graph into three subgraphs.

(d) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(e) From the previous step are there any insights you gain about the structure of the graph?

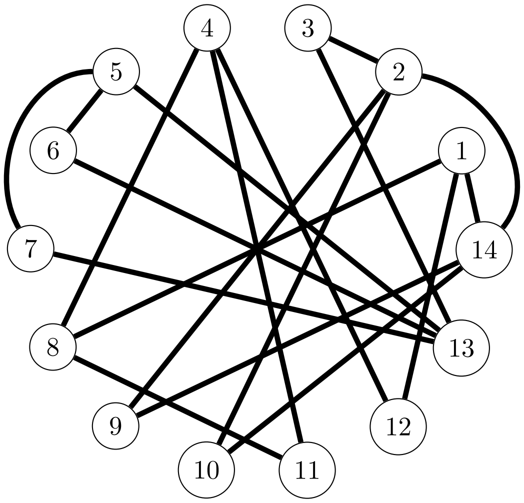
Exercise 13.15. Consider the graph with n vertices:



(a) Before doing any calculation what do you think the outcome of the Fiedler Method might be? You may need cases. Justify informally.

(b) Check your hypothesis with a few values of n.

Exercise 13.16. Consider the following graph:



(a) Find a Fiedler vector.

(b) Separate the values into three groups.

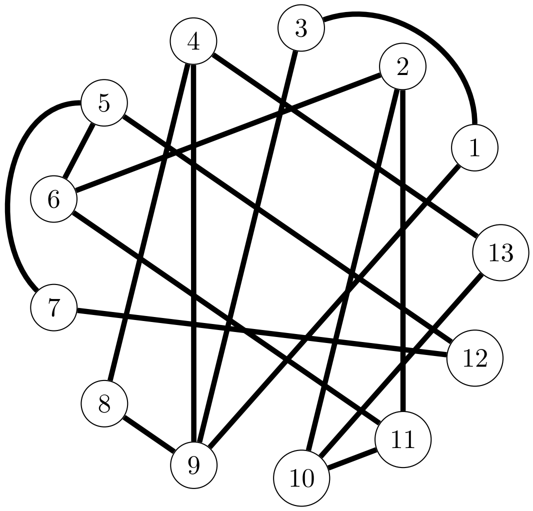
(c) Use these groups to partition the graph into three subgraphs.

(d) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(e) Which vertex seems like the most critical and why?

(f) From the previous step are there any insights you gain about the structure of the graph?

Exercise 13.17. Suppose the following graph shows all of the people in a small part of a social network. An edge connecting two people indicates that they are friends.



(a) Use the Fiedler Method to identify the group which is most strongly connected to Person 10.

(b) Why is this method ineffective in terms of providing a reasonable answer to (a)? Hint: Is anyone missing from your answer to (a) that is probably important to Person 10?

Exercise 13.18. Suppose a small LAN (local area network) consists of ten computers connected as follows:

- C1 is connected to C8, C9, C10
- C2 is connected to C5, C6.
- C3 is connected to C4, C9.
- C4 is connected to C3, C9.
- C5 is connected to C2, C7.
- C6 is connected to C2, C7, C8.
- C7 is connected to C5, C6.
- C8 is connected to C1, C6, C10.
- C9 is connected to C1, C3, C4.
- C10 is connected to C1, C8.

(a) Write down the Laplacian Matrix for this graph. You don't need to draw the graph!

(b) Find the Fiedler Vector and re-order the entries in increasing order.

(c) Partition the graph into some obvious number of subgraphs.

(d) Draw each of the subgraphs neatly and then use dashed lines to represent the edges that go between them.

(e) From the previous step are there any insights you gain about the structure of the graph?

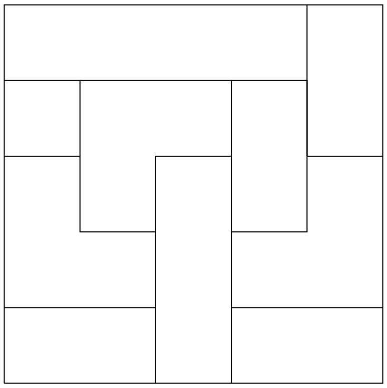
Exercise 13.19. Suppose a small LAN (local area network) consists of ten computers connected as follows:

- C1 is connected to C6.
- C2 is connected to C3, C7, C8 and C9.
- C3 is connected to C2, C4 and C7.
- C4 is connected to C3, C5, C6 and C7.
- C5 is connected to C4 and C6.
- C6 is connected to C1, C4 and C5.
- C7 is connected to C2, C3, C4 and C9.
- C8 is connected to C2, C9 and C10.
- C9 is connected to C2, C7 and C8.
- C10 is connected to C8.

(a) If a network technician wishes to divided these into two groups in order to connect two backup power supplies how should this be done using the Fiedler Method?

(b) If we define the most important links as those that would be removed using the Fiedler Method what are the most important links in this network?

Exercise 13.20. The following is a simplified map of some countries. What we'd like to do is divide the countries into two subsets in a way that tries to balance the number of countries in each subset while minimizing the number of border crossings between subsets.



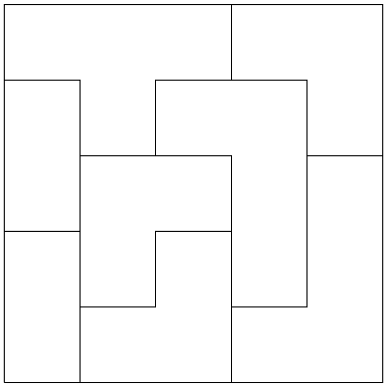
(a) Create a graph from this map by assigning a vertex for each country and connecting two vertices by an edge if the two countries share a border.

(b) Use the Fiedler Method to partition the graph.

(c) Explain in terms of the map what the Fiedler Method has attempted to do.

(d) Shade one subset of the countries in accordance with the result.

Exercise 13.21. The following is a simplified map of some countries. What we'd like to do is divide the countries into two subsets in a way that tries to balance the number of countries in each subset while minimizing the number of border crossings between subsets.



(a) Create a graph from this map by assigning a vertex for each country and connecting two vertices by an edge if the two countries share a border.

(b) Use the Fiedler Method to partition the graph.

(c) Explain in terms of the map what the Fiedler Method has attempted to do.

(d) Shade one subset of the countries in accordance with the result.

Exercise 13.22. A class of ten students needs to be split up. The goal is to get two groups of size as close as possible while minimizing the number of friendships that must be broken up. If the friendships are given in the following table use the Fiedler method to split up the class. How many friendships must be broken up? Draw the two resulting friend networks and indicate with dotted lines where the broken friendships are.

	Austin	Beth	Charlie	Dana	Erik	Fiona	Greg	Helen	Ian	Justin
Austin		✓	✓							✓
Beth	✓				✓					✓
Charlie	✓			✓			✓		✓	
Dana			✓		✓				✓	
Erik		✓		✓		✓		✓		✓
Fiona					✓			✓		
Greg			✓					✓		
Helen					✓	✓	✓		✓	
Ian			✓	✓				✓		✓
Justin	✓	✓			✓				✓	

Exercise 13.23. Pick an area of the world which is geographically divided into at least ten areas. These could be countries, states, counties, anything. Use the Fiedler Method to partition the area. Show the graph, relevant calculations, and a resulting map with the regions colored in two separate colors.

Exercise 13.24. The Fiedler method attempts to do two things with regards to the way it partitions the graph. What are those two things?

Exercise 13.25. What is happening when the Fiedler value turns out to have two or more linearly independent vectors associated to it? Give an example of a graph for which you believe that this would be the case and provide a basic and intutive explanation of why you believe this would be the case.

Chapter 14

Cryptography

Contents

14.1 Introduction

The goal of this chapter is to present a method of encryption which forms the basis of that used in many applications and show how linear algebra can be used to break this encryption. The basic method we present is not used as-is because it is fairly easily broken but it forms a building block for more sophisticated methods.

14.2 Background

Imagine we have a stream of bits, `0` and `1`, and we wish to encrypt them dynamically, meaning as each new bit comes along we have to encrypt it and send it along. It would seem reasonable to replace some `0` by `1` and vice versa in a way that the recepient would know how to undo the process.

Places where encryption like this may be useful would be things that are real-time critical like voice conversations, dynamical exchange of data, etc. Variations on the method we'll discuss are used in the Bluetooth protocol, various protocols used with GSM phones and various protocols used by the cable and other communication industries to scramble digital signals.

14.3 Preliminary Notes

A few things for this chapter:

(a) Modulo 2 arithmetic.

Definition 14.3.0.1. *Modulo 2 arithmetic* is arithmetic defined in such a way that all even numbers are considered equivalent to 0 and all odd numbers are considered equivalent to 1.

We write $a = b\,(mod\,2)$ if a and b are equivalent to one another and typically when we do operations we will write the result as either 0 or 1.

Example 14.1. For example we have the following:

$$\begin{aligned} 3 &= 1\,(mod\,2) \\ 1+1 &= 0\,(mod\,2) \\ (3)(5) &= 1\,(mod\,2) \\ -1 &= 1\,(mod\,2) \end{aligned}$$

and so on.

This also extends to matrices. For example:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} (mod\,2)$$

(b) We'll use the notation $[a; b; c]$ often to denote the vector $[a\,b\,c]^T$ because it's a bit neater.

(c) We'll often have vectors whose entries are bits, either 0 or 1, so we might write $\bar{c} \in \{0, 1\}^5$ for example to indicate that $\bar{c}$ is a vector with five entries either 0 or 1.

14.4 Basic Encryption Technique

14.4.1 How to Encrypt and Decrypt

Suppose Alice wishes to send the following binary stream to Bob without Eve intercepting it and understanding it. This is the *plaintext* and could go on indefinitely:

`10100011001111101011010111101010010001...`

In order to encrypt this so that Eve may not understand it if she intercepts it what Alice and Bob can do is the following: First they create and share a *key* consisting of a binary string such as `11010`. For now let's not worry about how that key is created or shared.

First a quick observation:

Fact 14.4.1.1. If we start with 0 or 1 then doing addition mod 2 twice with the same value (0 or 1) cancels out. In other words we see:

$$\begin{array}{c}
0+0=0\,(mod\,2) \text{ then } 0+0=0\,(mod\,2)\\
0+1=1\,(mod\,2) \text{ then } 1+1=0\,(mod\,2)\\
1+0=1\,(mod\,2) \text{ then } 1+0=1\,(mod\,2)\\
1+1=0\,(mod\,2) \text{ then } 0+1=1\,(mod\,2)
\end{array}$$

Alice takes her binary stream and does addition bit-by-bit with the key $(mod\,2)$. When she runs out of key she just repeats the key from the beginning. In this way we think of her key as infinitely long and having period 5. We'll use the term "key" to refer to both the repeated fragment and the infinitely long repetition. She then sends the bits one by one.

We can view this as follows:

Plaintext	`10100011001111101011010111101010010001...`	
+ Key	`11010110101101011010110101101011010110...`	bit-by-bit
= Ciphertext	`01110101100010110001100010000001000111...`	$(mod\,2)$

She sends this final string, the *ciphertext*, to Bob who decrypts it by doing +m2 again just like Alice did, but this undoes the encryption as noted above.

Ciphertext	`01110101100010110001100010000001000111...`	
+ Key	`11010110101101011010110101101011010110...`	bit-by-bit
= Plaintext	`10100011001111101011010111101010010001...`	$(mod\, 2)$

If Eve intercepts the ciphertext in the middle of the process she has no way of knowing what either the key or the plaintext are.

Keep in mind that in reality both Alice and Bob have the key at their disposal and the message is dealt with bit-by-bit. On a per-bit basis they cycle through the key and add $(mod\, 2)$ as they go. When they get to the end of the key they simply start again at the beginning.

14.4.2 Practical Note

This method is nice primarily for two reasons. First, it is very fast. Second, doing $(mod\, 2)$ is the same as doing an XOR (exclusive or) and can easily built into hardware circuits.

14.5 Key Creation and Sharing

The major issue with this is that the key needs to be created and shared between Alice and Bob before any communication can take place.

There are various ways that this can happen. When you rent a cable box the key might be hard-coded into a chip in the cable box, when you use a GSM phone the key might be hard-coded in the same way, and when you connect two Bluetooth devices by typing a code given by one machine into the other machine you are effectively sharing a key.

In both of these cases one problem is that a key with a larger period is more secure (keys with smaller periods can be brute-force guessed, for example there are only 32 possible keys of period 5 and yet a key with a larger period takes more memory to store (in the hardcoded case) and more time to manually transfer (in the Bluetooth case). In light of this we might ask if it is possible to create a key with a longer period using less data than the period? In other words, for example, could six bits of data be used to create a key with a period of more than six?

The answer is that we could create the key recursively in the following sense: We give some initial number of bits and then we define successive bits in terms of previous ones.

Example 14.2. We could assign a key $x_1x_2x_3...$ with $x_i \in \{0,1\}$ by assigning $x_1 = 0$, $x_2 = 1$, $x_3 = 1$ and for $n \geq 4$ we set $x_n = x_{n-3} + x_{n-1}\, (mod\, 2)$. So then $x_4 = x_1 + x_3 = 0 + 1 = 1\, (mod\, 2)$, $x_5 = x_2 + x_4 = 1 + 1 = 0\, (mod\, 2)$ and so on. If we do this repeatedly we get:

`01110100111010...`

which we notice is repeating after 7 bits, meaing we've created the key `0111010` having period 7. If we think of the key's recursive part as

$$x_n = 1x_{n-3} + 0x_{n-2} + 1x_{n-1}\,(mod\,2) \text{ for } n \geq 4$$

then by giving $\bar{s} = [0;1;1]$ (initial bits) and $\bar{c} = [1;0;1]$ (recursive coefficients) we generate a key with period 7 using only 6 bits.

Definition 14.5.0.1. A *linear recursively defined key of length* i can be defined by two vectors $\bar{s} = [x_1;...;x_i] \in \{0,1\}^i$ and $\bar{c} = [c_1;...;c_i] \in \{0,1\}^i$. The key then starts with the bits $x_1...x_i$ and for $n \geq i+1$ we have

$$x_n = c_1x_{n-i} + c_2x_{n-i+1} + ... + c_{i-1}x_{n-2} + c_ix_{n-1}\,(mod\,2)$$

That is, the first vector gives the starting bits of the key and the second vector gives the coefficients of the linear combination which tells us how to build each subsequent bit of the key as a linear combination of the previous i bits.

In addition we insist that $c_1 = 1$ because otherwise any given x_n depends only on the previous $i-1$ bits instead of the previous i and so technically it would not have generating length i.

We write the generating pair as:

$$K = \{\bar{s}, \bar{c}\}$$

Theorem 14.5.0.1. A linear recursively defined key of length i is reversible, meaning if we know any i bits we can recover any previous bits.

Proof. Observe that for any $k \geq 1$, letting $n = k+i \geq i+1$ we have the following, all $(mod\,2)$:

$$\begin{aligned}
x_n &= c_1x_{n-i} + c_2x_{n-i+1} + ... + c_ix_{n-1}\\
x_n &= c_1x_{n-i+0} + c_2x_{n-i+1} + ... + c_ix_{n-i+(i-1)}\\
x_{k+i} &= c_1x_{(k+i)-i+0} + c_2x_{(k+i)-i+1} + ... + c_ix_{(k+i)-i+(i-1)}\\
x_{k+i} &= c_1x_k + c_2x_{k+1} + ... + c_ix_{k+i-1}\\
x_{k+i} &= 1x_k + c_2x_{k+1} + ... + c_ix_{k+i-1}\\
x_k &= c_2x_{k+1} + ... + c_ix_{k+i-1} + x_{k+i}
\end{aligned}$$

□

Definition 14.5.0.2. The *period* of the key is the shortest number of bits after which the key repeats.

Theorem 14.5.0.2. A linearly recursively defined key of length i has period less than or equal to $2^i - 1$.

Proof. There are only 2^i arrangements of strings of i bits. If the string `0...0` ever appeared then it would be preceeded and followed by all zeros by Definition 14.5.0.1 and Theorem 14.5.0.1 and would therefore have length $i = 1$. If the string `0...0` never appears then some string of i bits must appear again after $2^i - 1$ bits. □

Example 14.2 Revisited. The pair $K = \{[0;1;1],[1;0;1]\}$ has length 3 and generates a key with period 7.

Okay, our example is not particularly impressive. Here are two more:

Example 14.3. The pair $K = \{[0;1;0;0;0],[1;0;1;0;0]\}$ has length 5 and generates the key with period 31 shown here to 62 bits:

`01000010010110011111000110111010100001001011001111100011011101...`

Example 14.4. For any $\bar{s} \in \{0,1\}^{31}$ if $\bar{c} \in \{0,1\}^{31}$ with $\bar{c} = [1;0;0;1;0;...;0]$ then the pair $K = \{\bar{s},\bar{c}\}$ generates a key with period $2^{31} - 1 = 2147483647$, not shown.

14.6 Breaking the Key

14.6.1 Circumstances

The approach we take is to assume that we have obtained some fragment $x_1x_2x_3...$ of the key. This could happen if we obtain both some ciphertext and some matching plaintext (maybe by snooping) since we can add the two bit-by-bit ($mod\, 2$) in order to get the correspoding part of the key.

The goal is to figure out the recursion relation and in doing so figure out the entire key.

It does not matter which part of the key we start with since the key repeats so any particular bit can be considered the "start" of the key. If we find out the length is, say, seven, then any consecutive seven bits could be considered x_1 through x_7.

14.6.2 Brute Force

One option could be a brute-force approach. Since the recursion relation has each bit being a linear combination of some number of previous bits we could take the portion of the key and take a trial-and-error approach.

What this means is we first check if a linearly recursively defined key of length 2 works. If not, we check if a linearly recursively defined key of length 3 works, then if a linearly recursively defined key of length 4 works, and so on until we either find the recursion relation or run out of key.

This is best illustrated with an example:

Example 14.5. Suppose we obtain the following portion of the key:

```
0110101111000100...
```

We could proceed by asking progressively as follows:

Question: Could a linearly recursively defined key of length 2 work?

If so then we would have $x_1 = 0$, $x_2 = 1$, and $x_n = c_1 x_{n-2} + c_2 x_{n-1}$ for $n \geq 3$. Applying this to x_3 and x_4 we get:

$$\begin{aligned} x_3 &= c_1 x_1 + c_2 x_2 \, (mod\, 2) \\ x_4 &= c_1 x_2 + c_2 x_3 \, (mod\, 2) \end{aligned}$$

which fills in to:

$$\begin{aligned} 1 &= c_1(0) + c_2(1) \, (mod\, 2) \\ 0 &= c_1(1) + c_2(1) \, (mod\, 2) \end{aligned}$$

or as a matrix equation:

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} (mod\, 2)$$

This matrix equation has the solution $c_1 = 1$, $c_2 = 1$ so we might guess that we have

$$x_n = x_{n-2} + x_{n-1} \, (mod\, 2) \text{ for } n \geq 3$$

If we test this on the key fragment (for $n \geq 5$) we find that $x_5 = x_3 + x_4 \, (mod\, 2)$ but $x_6 \neq x_4 + x_5 \, (mod\, 2)$ so clearly that didn't work.

Question: Could a linearly recursively defined key of length 3 work?

If we proceed as above we get the matrix equation:

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} (mod\, 2)$$

This matrix equation has no solution. This can be seen easily because the columns of the matrix each add to 0 mod 2 but the target column does not and so it is not a linear combination of the columns of the matrix.

Question: Could a linearly recursively defined key of length 4 work?

If we proceed as above we get the matrix equation:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \ (mod\,2)$$

This matrix equation has the solution $c_1 = 1$, $c_2 = 1$, $c_3 = 0$, $c_4 = 0$ so we might guess that we have

$$x_n = x_{n-4} + x_{n-3} \ (mod\,2) \text{ for } n \geq 5$$

If we test this on the remaning bits of the key fragment we find it works, so we believe, with the information we have, that this is it.

Before proceeding there are a few things that we should note:

- The matrix equation we're solving at each step is not as confusing as it may look. When testing whether a linearly recursively defined key of length m could work we simply look at:

$$\begin{bmatrix} x_1 & x_2 & \dots & x_m \\ x_2 & x_3 & \dots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \dots & x_{2m-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{2m} \end{bmatrix} \ (mod\,2)$$

  Notice the first column is the m bits of the key starting at x_1, the second column is the m bits of the key starting at x_2, and so on, finishing with the column vector on the right being the m bits of the key starting at x_{m+1}.

- This is computationally intensive.

14.6.3 Refining Brute Force

Luckily there is a theorem which comes to our rescue. Note that everything is mod 2, meaning a determinant of a matrix is either 0 or 1.

Theorem 14.6.3.1. For any given m define

$$M_m = \begin{bmatrix} x_1 & x_2 & ... & x_m \\ x_2 & x_3 & ... & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & ... & x_{2m-1} \end{bmatrix}$$

Then:

(i) If $\det(M_m) = 1\,(mod\,2)$ then no linear recursion of length less than m will satisfy the sequence $x_1, x_2, ..., x_{2m-1}$.

(ii) If $\det(M_m) = 0\,(mod\,2)$ and if there is a linear recursion of length m which does satisfy the sequence $x_1, x_2, ..., x_{2m-1}$. then there is a linear recursion of length less than m which will also satisfy that sequence.

Proof. Proof of (i) by contrapositive:

Suppose for a given m some linear recursion of length $i < m$ will work. Consider the matrix:

$$M_m = \begin{bmatrix} x_1 & x_2 & ... & x_m \\ x_2 & x_3 & ... & x_{1+m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i+1} & x_{i+2} & ... & x_{i+m} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{1+m} & ... & x_{2m-1} \end{bmatrix}$$

(Note that it's possible that $i + 1 = m$ in which case those are the same row.)

The fact that a linear recursion of length $i < m$ works means for all $n > i$ we have:

$$x_n = c_1 x_{n-i} + c_2 x_{n-i+1} + ... + c_i x_{n-1}\,(mod\,2)$$

Specifically this tells us that we have, all $(mod\,2)$:

$$\begin{aligned} x_{i+1} &= c_1 x_1 + c_2 x_2 + ... + c_i x_i \\ x_{i+2} &= c_1 x_2 + c_2 x_3 + ... + c_i x_{1+i} \\ \vdots &= \quad \vdots \\ x_{i+m} &= c_1 x_m + c_2 x_{m+1} + ... + c_i x_{i+m} \end{aligned}$$

As far as M_m this system is simply saying that:

$$\text{Row } i+1 = c_1(\text{Row } 1) + c_2(\text{Row } 2) + \ldots + c_i(\text{Row } i)\,(mod\,2)$$

So that the $(i+1)^{\text{th}}$ row of the matrix is a linear combination of the previous rows. Hence the rows are linearly dependent and $det(M_m) = 0$.

The proof of (ii) is more technical and is omitted.

□

The ramifications of this proof are important.

Theorem 14.6.3.2. The length of the shortest linear recursion relation will be the largest m for which $\det(M_m) = 1\,(mod\,2)$.

Proof. There is definitely a shortest linear recursion relation that works because there is definitely a linear recursion relation that works. Suppose its length is m, then $\det(M_m) = 1\,(mod\,2)$ because otherwise $\det(M_m) = 0\,(mod\,2)$ and a shorter one would work by (ii). Moreover for $k > m$ we must have $\det(M_k) = 0\,(mod\,2)$ otherwise m would not work by (i). □

Consequently, assuming we can obtain such an m the solution will be given by solving the matrix equation corresponding to, all $(mod\,2)$:

$$\begin{aligned} c_1x_1 + c_2x_2 + \ldots + c_mx_m &= x_{m+1} \\ c_1x_2 + c_2x_3 + \ldots + c_mx_{m+1} &= x_{m+2} \\ \ldots &= \ldots \\ c_1x_m + c_2x_{m+1} + \ldots + c_mx_{2m-1} &= x_{2m} \end{aligned}$$

which is

$$\begin{bmatrix} x_1 & x_2 & \ldots & x_m \\ x_2 & x_3 & \ldots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \ldots & x_{2m-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{2m} \end{bmatrix} (mod\,2)$$

Of course there is still an issue - we can test determinants all day but we'll never know if we have the largest value m for which $\det(M_m) = 1$ since how could we?

The approach we take is therefore as follows:

Refined Brute Force Method

(a) Calculate $det(M_m)\,(mod\,2)$ for $m = 1, 2, 3, ...$ until we encounter a 1 followed by some reasonable number of 0s. The term "reasonable" is ambiguous and might depend upon the technology being used.

(b) Solve the matrix equation:

$$\begin{bmatrix} x_1 & x_2 & ... & x_m \\ x_2 & x_3 & ... & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & ... & x_{2m-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{2m} \end{bmatrix} (mod\,2)$$

(c) Check whether the recursion relation given by the solution works for the entire key fragment. If so, stop and conclude we have found the recursion relation. If not, proceed with higher values of m.

(d) If we run out of key fragment before encountering a solution then we simply do not have enough key fragment to find the recursion relation.

Example 14.6. Suppose we obtain the following key fragment consisting of fifty bits:

```
10011011001001010001110100010001111100111110111101
```

We calculate determinants of M_m for $m = 1, 2, 3, ...$ until we see a few 0s in a row. All are $(mod\,2)$:

$$\begin{aligned} det(M_1) &= 1 \\ det(M_2) &= 0 \\ det(M_3) &= 1 \\ det(M_4) &= 1 \\ det(M_5) &= 0 \\ det(M_6) &= 1 \\ det(M_7) &= 0 \\ det(M_8) &= 0 \\ det(M_9) &= 1 \\ det(M_{10}) &= 0 \\ det(M_{11}) &= 0 \\ det(M_{12}) &= 0 \\ det(M_{13}) &= 0 \end{aligned}$$

We notice that we've run into a string of 0s after $m = 9$ so we suggest that $m = 9$ might give us the solution since it gave the seemingly final determinant of 1.

We therefore examine the matrix equation for $m = 9$:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \ (mod\, 2)$$

This has the solution $c_1 = 1$, $c_2 = 0$, $c_3 = 1$, $c_4 = 1$, $c_5 = 0$, $c_6 = 1$, $c_7 = 0$, $c_8 = 0$, $c_9 = 0$ which suggests that $x_n = x_{n-9} + x_{n-7} + x_{n-6} + x_{n-4}\, (mod\, 2)$ for $n \geq 9$.

If we test this we find that it works on all of our key fragment so we accept it as a solution.

If this had failed to work we would have needed to calculate further determinants knowing another 1 followed by 0s on the horizon.

Example 14.7. Suppose we obtain the following key fragment consisting of fifty bits:

`01110111110111110101111110001000011110110001111011`

We calculate determinants of M_m for $m = 1, 2, 3, ...$ until we see a few 0s in a row. All are $(mod\, 2)$:

$$\begin{aligned}
det(M_1) &= 0\\
det(M_2) &= 1\\
det(M_3) &= 1\\
det(M_4) &= 1\\
det(M_5) &= 1\\
det(M_6) &= 0\\
det(M_7) &= 1\\
det(M_8) &= 0\\
det(M_9) &= 0\\
det(M_{10}) &= 0
\end{aligned}$$

We notice that we've run into a string of 0s after $m = 7$ so we suggest that $m = 7$ might give us the solution since it gave the seemingly final determinant of 1.

We therefore examine the matrix equation for $m = 7$:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 1 & 1 & 1 & 1\\ 0 & 1 & 1 & 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 1 & 1 & 0 & 1\\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1\\ c_2\\ c_3\\ c_4\\ c_5\\ c_6\\ c_7 \end{bmatrix} = \begin{bmatrix} 1\\ 1\\ 1\\ 0\\ 1\\ 1\\ 1 \end{bmatrix} \ (mod\, 2)$$

This has the solution $c_1 = 0$, $c_2 = 1$, $c_3 = 0$, $c_4 = 0$, $c_5 = 0$, $c_6 = 0$, $c_7 = 0$ which suggests that

$$x_n = x_{n-6}\,(mod\, 2) \text{ for } n \geq 8$$

However this has a problem in that $c_1 \neq 1$. Moreover even if we overlook that and test it for our key fragment we find it fails because $x_{19} \neq x_{13}\,(mod\, 2)$.

So then we calculate further determinants of M_m. All are $(mod\, 2)$:

$$\begin{aligned}
det(M_{11}) &= 0\\
det(M_{12}) &= 1\\
det(M_{13}) &= 0\\
det(M_{14}) &= 0\\
det(M_{15}) &= 0
\end{aligned}$$

Aha, a new 1 showed up at $m = 12$, followed by some 0s.

We therefore examine the matrix equation for $m = 12$:

$$\begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1\\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1\\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1\\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1\\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0\\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1\\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0\\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1\\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1\\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix} c_1\\ c_2\\ c_3\\ c_4\\ c_5\\ c_6\\ c_7\\ c_8\\ c_9\\ c_{10}\\ c_{11}\\ c_{12} \end{bmatrix}
=
\begin{bmatrix} 1\\ 1\\ 1\\ 1\\ 0\\ 1\\ 0\\ 1\\ 1\\ 1\\ 1\\ 1 \end{bmatrix}
(mod\,2)$$

This has the solution $c_1 = 1$, $c_2 = 1$, $c_3 = 0$, $c_4 = 1$, $c_5 = 0$, $c_6 = 1$, $c_7 = 1$, $c_8 = 0$, $c_9 = 1$, $c_{10} = 0$, $c_{11} = 0$, $c_{12} = 0$ which suggests that

$$x_n = x_{n-12} + x_{n-11} + x_{n-9} + x_7 + x_6 + x_4\,(mod\,2) \text{ for } n \geq 13$$

If we test this we find that it works on all of our key fragment so we accept it as a solution.

14.7 System of Equations Mod 2

In this chapter we needed to solve certain matrix equations mod 2. To do this we should be aware of a few points.

Theorem 14.7.0.1. A square matrix A with entries in $\{0, 1\}$ is invertible mod 2 if and only if $det(A) = 1\,(mod\,2)$.

Note that "is invertible $(mod\,2)$" means there is another matrix B, also having entries in $\{0, 1\}$, such that $AB = BA = I\,(mod\,2)$.

Proof. (Partial Proof) If A is invertible $(mod\,2)$ then there is a matrix B with entries in $\{0,1\}$ and with $AB = I\,(mod\,2)$. Taking the determinant of both sides yields:

$$det(A)det(B) = det(I)\,(mod\,2)$$

which is the same as:

$$det(A)det(B) = 1\,(mod\,2)$$

So that both $det(A) = det(B) = 1\,(mod\,2)$.

The converse, that if $det(A) = 1\,(mod\,2)$ then A has an inverse mod 2, is much more challenging and is omitted. □

The basic gist of the omitted proof is relevant from the standpoint of finding the inverse, however, so we'll mention it.

The general idea is that if A is a matrix with entries in $\mathbb{R}$ then

$$A^{-1} = \frac{1}{det(A)}adj(A)$$

where $adj(A)$ is the adjugate of A. Therefore we have:

$$A\left[\frac{1}{det(A)}adj(A)\right] = \left[\frac{1}{det(A)}adj(A)\right]A = I$$

If we multiply through by $det(A)$ we get:

$$A\left[det(A)\frac{1}{det(A)}adj(A)\right] = \left[det(A)\frac{1}{det(A)}adj(A)\right]A = det(A)I$$

Now then since $det(A) = 1\,(mod\,2)$ the right side is just I so the expression:

$$det(A)\frac{1}{det(A)}adj(A)$$

is the inverse of $A\,(mod\,2)$ provided the entries are in $\mathbb{Z}$ and hence can be taken $(mod\,2)$. However since the entries of $adj(A)$ are in $\mathbb{Z}$ and the fraction is cancelled we know that is the case.

Since we have:

$$det(A)\frac{1}{det(A)}adj(A) = det(A)A^{-1}$$

It follows that we can calculate the inverse of $A\,(mod, 2)$ by finding A^{-1} in the traditional sense, multiplying by $det(A)$ which clears out the fractions, and then taking this expression $(mod\,2)$.

In other words:

$$A^{-1} = det(A)A^{-1}\,(mod\,2)$$

Example 14.8. Consider the matrix:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

We have $det(A) = -1$ and

$$A^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Therefore:

$$det(A)A^{-1} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

And taken mod 2 we get:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

So this is the inverse of A mod 2.

This is useful when we are solving matrix equations mod 2.

Example 14.9. Consider the matrix equation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \bar{c} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Since the determinant of the matrix equals 1 mod 2 we can find the solution mod 2 by multipying both sides by the inverse mod 2:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \bar{c} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \bar{c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\bar{c} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

14.8 Matlab

The following Matlab m-file will generate n digits of the key with initial vector `s` and coefficient vector `c`:

```
function x = genkey(s,c,n)
  % Generates n bits of the recursively defined key
  % using initial string s and vector c.
  % The result is returned as a vector
  % so it's usually good to wrap it in
  % transpose or ' it to look at it.
  % Usage:
  % >> genkey([1;0;1;1],[1;1;0;0],10)'
  % ans =
  %   1     0     1     1     1     1     0     0     0     1
  x = s;
  l = length(s);
  for j = [l+1:n]
    x = [x;mod(transpose(x(j-l:j-1))*c,2)];
  end
end
```

Usage as per the help:

```
>> genkey([1;0;1;1],[1;1;0;0],10)'
ans =
  1     0     1     1     1     1     0     0     0     1
```

The following Matlab m-file will generate the matrix M_m for a specific m for a specific key fragment.

```
function M = genmatrixfromfragment(v,m)
  % Generates the matrix M_m from the key fragment vector v.
  % Note that the length of f must be >= 2m-1.
  % Usage:
  % >> genmatrixfromfragment([1;0;1;1;0;1;1;1;1;0;1;1;0],3)
  % ans =
  %      1     0     1
  %      0     1     1
  %      1     1     0
  M = [];for i=1:m;M=[M v(i:i+m-1)];end;
end
```

Usage as per the help:

```
>> genmatrixfromfragment([1;0;1;1;0;1;1;1;1;0;1;1;0],3)
ans =
     1     0     1
     0     1     1
     1     1     0
```

If you don't give it enough key fragment it will error:

```
>> genmatrixfromfragment([1;0;1;1;0;1;1;1;1;0;1;1;0],10)
Index exceeds matrix dimensions.
Error in genmatrixfromfragment (line 10)
  M = [];for i=1:m;M=[M v(i:i+m-1)];end;
```

Notice that we can generate the matrix from the key vectors by combining the two commands:

```
>> v=genkey([1;0;1;1;1],[1;1;0;0;1],20);
>> genmatrixfromfragment(v,4)
ans =
     1     0     1     1
     0     1     1     1
     1     1     1     0
     1     1     0     1
```

Or in one fell swoop:

```
>> genmatrixfromfragment(genkey([1;0;1;1;1],[1;1;0;0;1],20),4)
ans =
     1     0     1     1
     0     1     1     1
     1     1     1     0
     1     1     0     1
```

Here is Example 14.6 worked out via Matlab. First we define the vector containing the key fragment. Here it's broken over several lines to fit on the page but of course it doesn't have to be entered this way.

```
 x=[
1;0;0;1;1;0;1;1;0;0;1;0;0;1;0;1;0;0;0;
1;1;1;0;1;0;0;0;1;0;0;0;1;1;1;1;1;0;0;
1;1;1;1;1;0;1;1;1;1;0;1];
```

Then we check determinants of M_m until we hit a 1 followed by a bunch of 0s. We can do this with a `for` loop. The choice of going to $m = 15$ is just experimenting. Here we've also used `round` to round the determinant before taking the `mod`. The reason for this is that the precision of the determinant can be slightly off and so rounding it makes sure that we get the integer that we know we should get:

```
>> for m=1:15
mod(round(det(genmatrixfromfragment(x,m))),2)
end
ans =
     1
ans =
     0
ans =
     1
ans =
     1
ans =
     0
ans =
     1
ans =
     0
ans =
     0
ans =
     1
ans =
     0
ans =
     0
ans =
     0
ans =
     0
ans =
     0
ans =
     0
```

So now we see that $m = 9$ might be our goal. Thus we solve $M_m\bar{c} = [x_{m+1}; ...; x_{2m}]$ using the inverse of M_m mod 2, again using `round` in there:

```
>> M = genmatrixfromfragment(x,9);
>> c = mod(round(det(M)*inv(M)*x(10:18)),2)
c =
     1
     0
     1
     1
     0
     1
     0
     0
     0
```

Finally we need to check that the key that this generates matches the key fragment at the start.

If we take the first nine bits from the key fragment and these nine bits we found and we use them to generate a key with the same length as the original fragment we can compare this generated key with the key fragment to see if this actually works. To be really fancy we can just take the difference between the vectors:

```
>> norm(genkey(x(1:9),c,length(x)) - x)
ans =
     0
```

14.9 Exercises

Exercise 14.1. Encrypt the stream 10110100010100101 using the key 10111.

Exercise 14.2. Encrypt the stream 11011010010000010110010 1 using the key 110101.

Exercise 14.3. Write down the first 30 digits of the key defined by

$$\{[1;1;0;0],[1;0;0;1]\}$$

Can you see what the key length is?

Exercise 14.4. Write down the first 30 digits of the key defined by

$$\{[1;0;1;1],[1;0;0;1]\}$$

Can you see what the key length is?

Exercise 14.5. Use brute force to find the recursion relation for the key fragment 1011100101 and use it to find the period.

Exercise 14.6. Use brute force to find the recursion relation for the key fragment 0011101001 and use it to find the period.

Exercise 14.7. Use brute force to find the recursion relation for the key fragment 010001101000110 and use it to find the period.

Exercise 14.8. Use brute force to find the recursion relation for the key fragment 010110010001111 and use it to find the period.

Exercise 14.9. Use refined brute force (look for a determinant of 1 followed by at least three determinants of 0) to find the recursion relation for the key fragment 01010010001011111011 and use it to find the next three bits of the key.

Exercise 14.10. Use refined brute force (look for a determinant of 1 followed by at least three determinants of 0) to find the recursion relation for the key fragment 01111000010111010101 and use it to find the next three bits of the key.

Exercise 14.11. Use refined brute force (look for a determinant of 1 followed by at least three determinants of 0) to find the recursion relation for the key fragment 10101111000001001110110110101 0 and use it to find the next three bits of the key.

Exercise 14.12. Use refined brute force (look for a determinant of 1 followed by at least three determinants of 0) to find the recursion relation for the key fragment `01001011100010011100111000001` and use it to find the next three bits of the key.

Exercise 14.13. Suppose you intercept the following ciphertext along with a fragment of the plaintext:

Ciphertext:	`1110100101100110110101100101101111111000000110001`
Plaintext:	`0011101111011110010101`

(a) Obtain as large a key fragment as you can.

(b) Use refined brute force to find the recursion relation for the key.

(c) Continue building the key until you see it repeat. At this point you know the full key.

(d) Use the key to decrypt the full ciphertext.

(e) If groups of five bits, converted to decimal, indicate letters with $1 = A$, $2 = B$, etc., what does the message say?

Exercise 14.14. Suppose you intercept the following ciphertext along with a fragment of the plaintext:

Ciphertext:	`0010001111010010111100011110111100011100110011011011011`
Plaintext:	`1001101101000011001010100`

(a) Obtain as large a key fragment as you can.

(b) Use refined brute force to find the recursive relation for the key.

(c) Continue building the key until you see it repeat. At this point you know the full key.

(d) Use the key to decrypt the full ciphertext.

(e) If groups of five bits, converted to decimal, indicate letters with $1 = A$, $2 = B$, etc., what does the message say?

Exercise 14.15. This method of producing a key can also be used to produce pseudorandom numbers. These are numbers that appear random but are not (generating random numbers and even defining what it means to be random is a difficult thing).

Create the first ten digits of a pseudorandom string of numbers between 0 and 15 as follows:

(a) Consider the key defined by the vectors $[1; 0; 1; 1; 1; 0; 0]$ and $[1; 0; 1; 1; 0; 0; 0]$. Write down 40 bits of this key.

(b) Break the key into 4-bit chunks and convert each chunk from binary to decimal.

Exercise 14.16. Suppose you encounter the pseudorandom number sequence:

```
9,9,10,0,12,7,15,12,5
```

Assuming these came from 4-bit chunks of a key generated using recursion, find the next three digits of the sequence.

Exercise 14.17. Suppose you encounter the pseudorandom number sequence:

```
12,11,4,8,12,11,13,1,15,13
```

Assuming these came from 4-bit chunks of a key generated using recursion, find the next three digits of the sequence.

Exercise 14.18. Suppose when attempting to break a recursively defined key (from a key fragment you have) you apply the Theorem and find for $m = 1$, 2, ... that:

$$\det(M_m) = 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0$$

at which point you can no longer calculate determinants because you run out of key fragment.

(a) What is your guess for the length of the recursive relation?

(b) Suppose you solve for the corresponding c_i but they don't actually work when applied to the key fragment. What does this mean?
Hint: If this is confusing, think about what would happen if you'd only had enough key fragment to generate determinants $0, 1, 0, 0, 1, 1, 0$.

Exercise 14.19. Consider the following (repeating key):

```
110010111001011110010
```

(a) What is the key length?

(b) Find the recursion relation.
Note: The recursion relation is short and the systems of equations you need to solve are easy by hand.

Exercise 14.20. Consider the key recursively defined using the vectors $[1; 1; 0; 1]$ and $[1; 0; 1; 1]$. This recursive relation has length 4 but show that the key it generates can in fact be genereated using a recursive relation of length 2.

Chapter 15

Portfolio Optimization

Contents

15.1 Introduction

Suppose we are investing in the stock market and we are examining several companies. Each company has stock which we can buy and sell. Suppose for each stock we know both the average weekly rate of return for the stock, as well as the variance in that rate of return.

At this point if we had to invest we might pick the stock with the highest return (if we were willing to take the corresponding risk) or with the lowest variance (if we were risk-averse). Or we might invest some of our money in one stock and the rest in another.

However to complicate things suppose the performance of the companies affect one another. Perhaps when one company's stock goes up, another goes down, and perhaps the third goes up but not by as much.

We will address two questions:

(I) Suppose we have no interest in the return but simply want to invest our money to minimize the risk. How should we do this?

(II) Suppose we want a particular rate of return. How can we achieve this while minimizing the risk?

15.2 A Brief Review of Statistics

15.2.1 Random Variables

Definition 15.2.1.1. A *random variable* is a variable whose outcomes follow some unknown (perhaps random) pattern. Typically random variables are denoted by capital letters such as X, Y, X_1, etc.

Example 15.1. If a fair die is rolled and the result is assigned to the random variable X. Then $X \in \{1, 2, 3, 4, 5, 6\}$.

The word "random" is slightly inaccurate.

Example 15.2. The daily percentage return on a stock, can be treated as a random variable. It's not really random (it's affected by company performance, investor behavior, etc.) but it is unpredictable and we can ask questions about its value.

15.2.2 Expected Value and Variance

Definition 15.2.2.1. The *expected value* of a random variable X, denoted $E(X)$ (or sometimes $\mu(X)$ or μ_X or just μ when it's clear), is the long-term average of that variable, meaning if the variable took on values over and over again forever what the average would be.

If each outcome is equally likely then the expected value is simply the average.

Example 15.3. A die is rolled and the result is assigned to the random variable X. Then $E(X) = (1+2+3+4+5+6)/6 = 3.5$.

The expected value of the daily return of a stock can be approximated simply be averaging the returns over a wide sample of days, perhaps the last 30 or 60 or 90 days. Of course this isn't perfect since the stock might change in behavior but it gives us something to work with.

The expected value isn't really a time-related thing since we can imagine rolling infinitely many dice all at once instead of rolling one die infinitely many times but often it is time-related or can be thought of that way.

Fact 15.2.2.1. It is fairly clear intutively that E is a linear operator, meaning for random variables X_1, ... , X_n and for constants a_1, ..., a_n we have

$$E(a_1X_1 + ... + a_nX_n) = a_1E(X_1) + ... + a_nE(X_n)$$

Definition 15.2.2.2. The *variance* of a random variable X, denoted $Var(X)$, is the long-term average of the square of the difference between the variable and its long-term average. In other words:

$$Var(X) = E\left((X - E(X))^2\right)$$

Note: The square root of variance is the *standard deviation* and is denoted $\sigma(X)$ or σ_X or just σ. Consequently sometimes the variance is denoted $\sigma(X)^2$ or σ_X^2 or just σ^2 when it's clear.

If each outcome is equally likely then so is each $X - E(X)$ so then the variance is simply the average of $X - E(X)$ taken over all possible X.

Example 15.4. The variance for the die we rolled is:

$$\begin{aligned}Var(X) &= \frac{(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (4-3.5)^2 + (5-3.5)^2 + (6-3.5)^2}{6}\\ &= \frac{17.5}{6}\\ &= \frac{35}{12}\\ &\approx 2.9167\end{aligned}$$

Basically if a random variable X has a high variance this means that the values it takes on can be more spread out away from the average.

As far as stocks are concerned this can be understood as a measurement of risk.

15.2.3 Covariance

Suppose we have two random variables which take on values together so intead of looking at what each does independently we look at what they do in pairs. We might want to measure the connection between them.

Example 15.5. Suppose $X \in \{1,2,3\}$ and $Y \in \{1,2,3\}$ and we find that in real world data we see the three pairs $(1,1)$, $(2,2)$ and $(3,3)$. We notice that smaller X values correspond to smaller Y values and larger X values correspond to larger Y values.

Definition 15.2.3.1. The *covariance* of two random variables X and Y, denoted $Cov(X,Y)$, is the long-term average of the product of the differences between the two variables and their long-term averages. In other words:

$$Cov(X,Y) = E\left((X - E(X))(Y - E(Y))\right)$$

Note: Sometimes the covariance is denoted $\sigma(X,Y)^2$ or σ_{XY}^2 in order to match the notation of variance, even though neither $\sigma(X,Y)$ or σ_{XY} is a meaningful value other than square root of covariance.

If all outcomes of the pairs that appear are equally likely then the covariance is just the average of $(X - E(X))(Y - E(Y))$ over all possible pairs.

Example 15.6. In our example above we have $E(X) = 2$ and $E(Y) = 2$ and so we average $(X - 2)(Y - 2)$ over our three pairs:

$$Cov(X,Y) = \frac{(1-2)(1-2) + (2-2)(2-2) + (3-2)(3-2)}{3} = \frac{2}{3}$$

Example 15.7. If the three pairs that we observed had been $(1,3)$, $(2,2)$ and $(3,1)$ then we find:

$$Cov(X,Y) = \frac{(1-2)(3-2) + (2-2)(2-2) + (1-2)(3-2)}{3} = -\frac{2}{3}$$

which makes sense because now larger X values correspond to smaller Y values and vice-versa.

A positive covariance means that as one of X and Y increases, so does the other, whereas a negative covariance means that as one of X and Y increases, the other decreases. A covariance of zero means that a change in one has no impact on the other.

Notice that this does not imply that either of them directly influences the other but that they tend to act together, for whatever reason.

This is clear from the formula since if X and Y tend to be both larger or smaller than their individual expected values at the same time then $(X - E(X))(Y - E(Y))$ will tend to be positive, giving a positive expected value for that product, whereas if one tends to be larger while the other is smaller then that product will tend to be negative, giving an negative expected value for that product.

Theorem 15.2.3.1. We can also calculate the covariance via the formula:

$$Cov(X,Y) = E(XY) - E(X)E(Y)$$

Proof. We have:

$$\begin{aligned} Cov(X,Y) &= E\left((X - E(X))(Y - E(Y))\right) \\ &= E\left(XY - XE(Y) - YE(X) - E(X)E(Y)\right) \\ &= E(XY) - E(X)E(Y) - E(Y)E(X) + E(X)E(Y) \\ &= E(XY) - E(X)E(Y) \end{aligned}$$

Notice that line 3 follow from line 2 by the linearity of E. □

Notice that the covariance between a random variable and itself is simply its variance. In other words $Cov(X,X) = Var(X)$, consequently we have the following corollary:

Corollary 15.2.3.1. We can calculate the variance via the formula:

$$Var(X) = E(X^2) - E(X)^2$$

Proof.

$$Var(X) = Cov(X,X) = E(XX) - E(X)E(X) = E(X^2) - E(X)^2$$

□

Notice that we can pull constants out of covariance in the following manner:

Theorem 15.2.3.2. We have:

$$Cov(aX, bY) = ab\,Cov(X,Y)$$

Proof. We have:

$$\begin{aligned} Cov(aX, bY) &= E(aXbY) - E(aX)E(bY) \\ &= abE(XY) - abE(X)E(Y) \\ &= ab\,Cov(X, Y) \end{aligned}$$

□

Corollary 15.2.3.2. It follows immediately that:

$$Var(aX) = aVar(X)$$

The last fact we need to have involves the covariance of the sum of a collection of random variables.

Theorem 15.2.3.3. Given random variables X_1, ..., X_n we have:

$$Var\left(\sum_{i=1}^{n} X_i\right) = \sum_{\substack{1\le i\le n\\1\le j\le n}} Cov(X_i, X_j)$$

Proof. We have:

$$\begin{aligned} Var\left(\sum_{i=1}^{n} X_i\right) &= E\left(\left(\sum_{i=1}^{n} X_i\right)^2\right) - E\left(\sum_{i=1}^{n} X_i\right)^2 \\ &= E\left(\sum_{\substack{1\le i\le n\\1\le j\le n}} X_iX_j\right) - \left(\sum_{i=1}^{n} E(X_i)\right)^2 \\ &= \sum_{\substack{1\le i\le n\\1\le j\le n}} E(X_iX_j) - \sum_{\substack{1\le i\le n\\1\le j\le n}} E(X_i)E(X_j) \\ &= \sum_{\substack{1\le i\le n\\1\le j\le n}} [E(X_iX_j) - E(X_i)E(X_j)] \\ &= \sum_{\substack{1\le i\le n\\1\le j\le n}} Cov(X_i, X_j) \end{aligned}$$

□

One final comment that will be useful later. Note that the variances and covariances can all be placed in a handy matrix:

Definition 15.2.3.2. Given random variables X_1, ..., X_n we define the *covariance matrix* :

$$\Sigma = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_n) \\ Cov(X_1, X_2) & Var(X_2) & \dots & Cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_1, X_n) & Cov(X_2, X_n) & \dots & Var(X_n) \end{bmatrix}$$

15.3 A Brief Review of Lagrange Multipliers

Typically Lagrange Multipliers are approached this way for the two variable case:

We have a function $f(x, y)$ for which we wish to find a minimum or maximum (which we know exists) subject to a constraint $g(x, y) = 0$.

We solve the system of equations:

$$\begin{aligned} f_x &= \lambda g_x \\ f_y &= \lambda g_y \\ g(x, y) &= 0 \end{aligned}$$

Then we test f at each resulting (x, y) and choose whichever (minimum or maximum) we wanted.

Another classic way to approach this is to define the *Lagrange Function*:

$$L(x, y, \lambda) = f(x, y) + \lambda g(x, y)$$

And then solving the original system is exactly the same as solving:

$$\begin{aligned} L_x &= 0 \\ L_y &= 0 \\ L_\lambda &= 0 \end{aligned}$$

The only difference being that we use $-\lambda$ instead of λ but that doesn't matter.

The advantage of this second approach is that it generalizes easily not just to more variables but to additional constraints.

Theorem 15.3.0.1. General Method of Lagrange Mulipliers: Suppose we wish to maximize or minimize the function $f(\bar{x})$ subject to the set of constraints $g_1(\bar{x}) = ... = g_k(\bar{x}) = 0$. If we define the Lagrange function:

$$L(\bar{x}, \lambda_1, ..., \lambda_k) = f(\bar{x}) + \lambda_1 g_1(\bar{x}) + ... + \lambda_k g_k(\bar{x})$$

then the maximum and minimum will occur at a solution to the system

$$\begin{aligned} L_{x_1} &= 0 \\ \vdots\ &\ \vdots \\ L_{x_n} &= 0 \\ L_{\lambda_1} &= 0 \\ \vdots\ &\ \vdots \\ L_{\lambda_k} &= 0 \end{aligned}$$

Proof. Omitted. □

Notice that the system can be succinctly written as $\nabla L = \bar{0}$ where ∇L is the gradient.

Example 15.8. To find the assumed minimum value of the function $f(x, y, z) = x^2 + x + y^2 + z^2 + 2z$ subject to the constraints $x + y - z = 2$ and $2x - y + z = 0$ we set $g_1(x, y, z) = x + y - z - 2$ and $g_2(x, y, z) = 2x - y + z$ and then we set:

$$L(x, y, z, \lambda_1, \lambda_2) = x^2 + x + y^2 + z^2 + 2z + \lambda_1(x + y - z - 2) + \lambda_2(2x - y + z)$$

and solve the system $\nabla L = \bar{0}$ which is:

$$\begin{aligned} 2x + 1 + \lambda_1 + 2\lambda_2 &= 0 \\ 2y + \lambda_1 - \lambda_2 &= 0 \\ 2z + 2 - \lambda_1 + \lambda_2 &= 0 \\ x + y - z - 2 &= 0 \\ 2x - y + z &= 0 \end{aligned}$$

If we solve these we get $(x, y, z, \lambda_1, \lambda_2) = \left(\frac{2}{3}, \frac{1}{6}, -\frac{7}{6}, -1, -\frac{2}{3}\right)$

Since there's only one answer it must be our minimum and so the minimum occurs at $\left(\frac{2}{3}, \frac{1}{6}, -\frac{7}{6}\right)$ and equals

$$f\left(\frac{2}{3}, \frac{1}{6}, -\frac{7}{6}\right) = \frac{1}{6}$$

15.4 Portfolio Optimization

15.4.1 Introduction

Let's begin with a simple examle which will help us see why linear algebra is useful.

We are assuming the *Constant Expected Return (CER) Model*. This is a standard model used in asset return theory which makes many assumptions about the behavior of the returns for the sake of simplicity.

For our purposes the only assumptions we need to note is that the expected value and variance of a return is constant over time and therefore we can get reasonable values by collecting a large sample.

So now let's suppose we have three companies each with a stock. Each stock has a return which is a random variable so we will denote these R_1, R_2 and R_3.

Suppose then we collect data over a long period of time and we calculate the monthly expected values which we denote $\mu_1 = E(R_1)$, $\mu_2 = E(R_2)$, $\mu_3 = E(R_3)$, we calculate the monthly variances which we denote σ_1^2, σ_2^2, σ_3^2, and we calculate the monthly covariances which we denote σ_{12}^2, σ_{13}^2, σ_{23}^2.

Note we can also put the monthly expected values in a vector $\bar{\mu}$.

Suppose we have a total amount to invest and we wish to split this amount between the three stocks. Rather than deal with a specific total amount we will just look at the proportion of our amount which should be invested in each stock.

Call those proportions x_1, x_2, x_3 and so we have $x_1 + x_2 + x_3 = 1$. Note we can also put these proportions in a vector $\bar{x}$.

The return for our portfolio will then have random variable:

$$R = R_1 x_1 + R_2 x_2 + R_3 x_3$$

The expected return for our portfolio, will then be:

$$E(R) = \mu_1 x_1 + \mu_2 x_2 + \mu_3 x_3 = \bar{\mu}^T \bar{x} = \bar{x}^T \bar{\mu}$$

The variance for our portfolio will then be:

$$\begin{aligned}
Var(R) &= Var(R_1x_1 + R_2x_2 + R_3x_3) \\
&= \sum_{\substack{i=1,2,3 \\ j=1,2,3}} Cov(R_ix_i, R_jx_j) \\
&= \sum_{\substack{i=1,2,3 \\ j=1,2,3}} Cov(R_i, R_j)x_ix_j \\
&= \sigma_1^2x_1^2 + \sigma_2^2x_2^2 + \sigma_3^2x_3^2 + 2\sigma_{12}^2x_1x_2 + 2\sigma_{13}^2x_1x_3 + 2\sigma_{23}^2x_2x_3
\end{aligned}$$

A simple and useful formula for the variance is that if Σ is the covariance matrix then:

$$Var(R) = \bar{x}^T\Sigma\bar{x}$$

15.4.2 Global Minimum Variance Portfolio

Suppose we wish to invest our portfolio in a way that minimizes the variance without regard for the return. Basically you've got to put your money somewhere but you want as little risk as possible.

What this means is that we wish to choose x_1, x_2, x_3 which minimizes:

$$Var(R) = \sigma_1^2x_1^2 + \sigma_2^2x_2^2 + \sigma_3^2x_3^2 + 2\sigma_{12}^2x_1x_2 + 2\sigma_{13}^2x_1x_3 + 2\sigma_{23}^2x_2x_3$$

subject to the constraint:

$$x_1 + x_2 + x_3 = 1$$

We define the Lagrange function:

$$\begin{aligned}
L &= Var(X) + \lambda(x_1 + x_2 + x_3 - 1) \\
&= \sigma_1^2x_1^2 + \sigma_2^2x_2^2 + \sigma_3^2x_3^2 + 2\sigma_{12}^2x_1x_2 + 2\sigma_{13}^2x_1x_3 + 2\sigma_{23}^2x_2x_3 \\
&+ \lambda(x_1 + x_2 + x_3 - 1)
\end{aligned}$$

and solve the system $\nabla L = \bar{0}$ which is:

$$\begin{aligned}
2\sigma_1^2 x_1 + 2\sigma_{12}^2 x_2 + 2\sigma_{13}^2 x_3 + \lambda &= 0\\
2\sigma_{12}^2 x_1 + 2\sigma_2^2 x_2 + 2\sigma_{23}^2 x_3 + \lambda &= 0\\
2\sigma_{13}^2 x_1 + 2\sigma_{23}^2 x_2 + 2\sigma_3^2 x_3 + \lambda &= 0\\
x_1 + x_2 + x_3 - 1 &= 0
\end{aligned}$$

Notice that this can be rewritten as a matrix equation:

$$\begin{bmatrix} 2\sigma_1^2 & 2\sigma_{12}^2 & 2\sigma_{13}^2 & 1 \\ 2\sigma_{12}^2 & 2\sigma_2^2 & 2\sigma_{23}^2 & 1 \\ 2\sigma_{13}^2 & 2\sigma_{23}^2 & 2\sigma_3^2 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Which can be rewritten much more simply using the covariance matrix as:

$$\begin{bmatrix} 2\Sigma & \bar{1} \\ \bar{1}^T & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{0} \\ 1 \end{bmatrix}$$

which is especially nice since it generalizes to more than three options and can be solved via a matrix inverse:

$$\begin{bmatrix} \bar{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\Sigma & \bar{1} \\ \bar{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ 1 \end{bmatrix}$$

The questions as to why this matrix is invertible and why this yields a solution of minimum variance will be left for later.

Observe that it's not necessary to know μ to find the Global Minimum Variance Portfolio but it is necessary if we wish to know the expected return.

Example 15.9. Suppose three companies have expected monthly earnings:

$$\bar{\mu} = \begin{bmatrix} 0.0385 \\ 0.0021 \\ 0.0202 \end{bmatrix}$$

And they have variances and covariances given by:

$$\Sigma = \begin{bmatrix} 0.0110 & 0.0015 & 0.0008 \\ 0.0015 & 0.0121 & 0.0016 \\ 0.0008 & 0.0016 & 0.0218 \end{bmatrix}$$

Then the global minimum variance portfolio can be found via:

$$\begin{aligned}
\begin{bmatrix} \bar{x} \\ \lambda \end{bmatrix} &= \begin{bmatrix} 2\Sigma & \bar{1} \\ \bar{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.0220 & 0.0030 & 0.0016 & 1.0000 \\ 0.0030 & 0.0242 & 0.0032 & 1.0000 \\ 0.0016 & 0.0032 & 0.0436 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.4271 \\ 0.3672 \\ 0.2057 \\ 0.9892 \end{bmatrix}
\end{aligned}$$

Thus to minimize our risk we should set:

$$\bar{x} = \begin{bmatrix} 0.4271 \\ 0.3672 \\ 0.2057 \end{bmatrix}$$

meaning proportionally speaking we should invest 0.4271 of our portfolio in Company 1, 0.3672 of our portfolio in Company 2, and 0.2057 of our portfolio in Company 3.

The variance in this case would be:

$$Var(R) = \bar{x}^T \Sigma \bar{x} = 0.0054$$

and this is the minimum variance we can ever achieve. Notice that it is significantly lower than any of the individual company variances (the diagonal entries in Σ) because by spreading out our portfolio we have taken advantage of the covariances.

The expected return would be:

$$E(R) = \bar{x}^T \bar{\mu} = 0.0214$$

15.4.3 Minimum Variance Portfolio

Suppose now we wish to invest our portfolio by specifying a desired return and then minimizing the variance. The only change that occurs is that we gain a new condition.

Now we wish to choose x_1, x_2, x_3 which minimizes:

$$Var(R) = \sigma_1^2 x_1^2 + \sigma_2^2 x_2^2 + \sigma_3^2 x_3^2 + 2\sigma_{12}^2 x_1 x_2 + 2\sigma_{13}^2 x_1 x_3 + 2\sigma_{23}^2 x_2 x_3$$

subject to the two constraints:

$$x_1 + x_2 + x_3 = 1$$

and:

$$E(R) = \mu_1 x_1 + \mu_2 x_2 + \mu_3 x_3 = \mu_0$$

where μ_0 is the desired return.

We define the Lagrange function:

$$\begin{aligned} L = {} & \sigma_1^2 x_1^2 + \sigma_2^2 x_2^2 + \sigma_3^2 x_3^2 + 2\sigma_{12}^2 x_1 x_2 + 2\sigma_{13}^2 x_1 x_3 + 2\sigma_{23}^2 x_2 x_3 \\ & + \lambda_1 (x_1\mu_1 + x_2\mu_2 + x_3\mu_3 - \mu_0) \\ & + \lambda_2 (x_1 + x_2 + x_3 - 1) \end{aligned}$$

and solve the system $\nabla L = \bar{0}$ which is:

$$\begin{aligned} 2\sigma_1^2 x_1 + 2\sigma_{12}^2 x_2 + 2\sigma_{13}^2 x_3 + \mu_1\lambda_1 + \lambda_2 &= 0 \\ 2\sigma_{12}^2 x_1 + 2\sigma_2^2 x_2 + 2\sigma_{23}^2 x_3 + \mu_2\lambda_1 + \lambda_2 &= 0 \\ 2\sigma_{13}^2 x_1 + 2\sigma_{23}^2 x_2 + 2\sigma_3^2 x_3 + \mu_3\lambda_1 + \lambda_2 &= 0 \\ \mu_1 x_2 + \mu_2 x_2 + \mu_3 x_3 - \mu_0 &= 0 \\ x_1 + x_2 + x_3 - 1 &= 0 \end{aligned}$$

Notice that this can be rewritten as a matrix equation:

$$\begin{bmatrix} 2\sigma_1^2 & 2\sigma_{12}^2 & 2\sigma_{13}^2 & \mu_1 & 1 \\ 2\sigma_{12}^2 & 2\sigma_2^2 & 2\sigma_{23}^2 & \mu_2 & 1 \\ 2\sigma_{13}^2 & 2\sigma_{23}^2 & 2\sigma_3^2 & \mu_3 & 1 \\ \mu_1 & \mu_2 & \mu_3 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda_1 \end{bmatrix} \lambda_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu_0 \\ 1 \end{bmatrix}$$

Which can be rewritten much more simply using the covariance matrix as:

$$\begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \bar{0} \\ \mu_0 \\ 1 \end{bmatrix}$$

which is again especially nice since it generalizes to more than three options and can be solved via a matrix inverse:

$$\begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ \mu_0 \\ 1 \end{bmatrix}$$

Example 15.9 Revisited.

Again, the questions as to why this matrix is invertible and why this yields a solution of minimum variance will be left for later.

To continue our example above suppose that we wish for a return of $\mu_0 = 0.0305$. Then the corresponding minimum variance portfolio can be found via:

$$\begin{aligned} \begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} &= \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ \mu_0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.0220 & 0.0030 & 0.0016 & 0.0385 & 1.0000 \\ 0.0030 & 0.0242 & 0.0032 & 0.0021 & 1.0000 \\ 0.0016 & 0.0032 & 0.0436 & 0.0202 & 1.0000 \\ 0.0385 & 0.0021 & 0.0202 & 0 & 0 \\ 1.0000 & 1.0000 & 1.0000 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.0305 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.6770 \\ 0.1154 \\ 0.2076 \\ -0.2770 \\ 0.9951 \end{bmatrix} \end{aligned}$$

Thus to minimize our risk we should set:

$$\bar{x} = \begin{bmatrix} 0.6770 \\ 0.1154 \\ 0.2076 \end{bmatrix}$$

meaning proportionally speaking we should invest 0.6770 of our portfolio in Company 1, 0.1154 of our portfolio in Company 2, and 0.2076 of our portfolio in Company 3.

Notice that the variance in this case equals:

$$Var(R) = \bar{x}^T \Sigma \bar{x} = 0.0067$$

Any other $\bar{x}$ which achieves the same return would have a larger variance.

Notice also that this variance is higher than the global minimimum variance. In this case we are getting a higher return , our desired 0.0305 rather than the 0.0214 return from the global minimum variance portfolio return, in exchange for a higher variance, 0.0067 rather than 0.0054.

15.4.4 Extreme Examples and Interpretations

The example in the previous section was fairly straightforward but it's worth adjusting the example a couple of different ways to see the results:

Example 15.9 Revisited.

Suppose we want a return of 0.035. Let's see what happens.

$$\begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ 0.035 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.8002 \\ -0.0087 \\ 0.2085 \\ -0.4136 \\ 0.9980 \end{bmatrix}$$

This suggests that we should allocate our portfolio according to:

$$\bar{x} = \begin{bmatrix} 0.8002 \\ -0.0087 \\ 0.2085 \end{bmatrix}$$

What's going on here, how can we have a negative amount of Company 2 and how can our two positive amounts add to more than 1?

The answer to this first question is initially that the mathematics doesn't know that, it simply finds the $\bar{x}$ which does the job. However this is not totally unreasonable. It's possible when investing to hold a *short position* on a stock. This works as follows:

Example 15.10. Suppose a stock costs \$10 per share. You believe it's going to go down soon so clearly you would not buy it. However because you believe it's going to go down you borrow 100 shares from your broker and sell them, earning a quick \$1000. Suppose later when the stock drops to \$0.50 per share you buy 100 shares for \$500 and give them back to the broker. You have earned \$500. This is called closing the short position.

Before this final sale you are said to have a short (negative) position on the stock since you technically purchased -100 shares when you borrowed and sold them.

When the stock goes down you are happy because you are gaining money relative to if you closed the position and when the stock goes up you are unhappy because you are losing money relative to if you closed the position.

A negative value in $\bar{x}$ can be interpreted simply as that, holding a short position on the stock.

The answer to this second question is that since you earned money from your short position you can allocate more money to the other two. The net total portfolio value is still 1.

Not all brokers allow short positions as they are very risky.

Example 15.9 Revisited.

Suppose we get greedy and we want a return of 0.1, a seriously large 10% return! Let's see what happens:

$$\begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ 0.1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.5792 \\ -1.8011 \\ 0.2219 \\ -2.3855 \\ 1.0401 \end{bmatrix}$$

This suggests that we should allocate our portfolio according to:

$$\bar{x} = \begin{bmatrix} 2.5792 \\ -1.8011 \\ 0.2219 \end{bmatrix}$$

So it seems possible even with the strange numbers, so why wouldn't we do it? The answer is in the risk (the variance), and this may have occurred to us when we saw the graph earlier. The variance corresponding to this return is:

$$Var(R) = \bar{x}^T \Sigma \bar{x} = 0.0992$$

which is quite high.

This becomes even more pronounced if we get even greedier and demand, for example 100% return.

We find:

$$\begin{bmatrix} \bar{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ 1.00 \\ 1 \end{bmatrix} = \begin{bmatrix} 27.2121 \\ -26.6198 \\ 0.4077 \\ -29.6900 \\ 1.6236 \end{bmatrix}$$

This suggests that we should allocate our portfolio according to:

$$\bar{x} = \begin{bmatrix} 27.2121 \\ -26.6198 \\ 0.4077 \end{bmatrix}$$

So we'd have to massively short Company 2 and invest the money from this short primarily in Company 1.

However the variance corresponding to this return is:

$$Var(R) = \bar{x}^T \Sigma \bar{x} = 14.5332$$

which implies that this would be an extremely risky (stupidly so) portfolio.

15.4.5 Questions to Answer

A few closing notes:

(I) Why does the Method of Lagrange Multipliers find a minimum?

Since variance $Var(R) = \bar{x}^T \Sigma \bar{x}$ is always nonnegative when we are looking at the global minimum variance portfolio if we examine the set:

$$\left\{ \bar{x}^T \Sigma \bar{x} \,\middle|\, \bar{x}^T \bar{1} = 1 \right\}$$

or when we are looking the minimum variance portfolio if we examine the set:

$$\left\{ \bar{x}^T \Sigma \bar{x} \,\middle|\, \bar{x}^T \bar{1} = 1, \bar{x}^T \bar{\mu} = \mu_0 \right\}$$

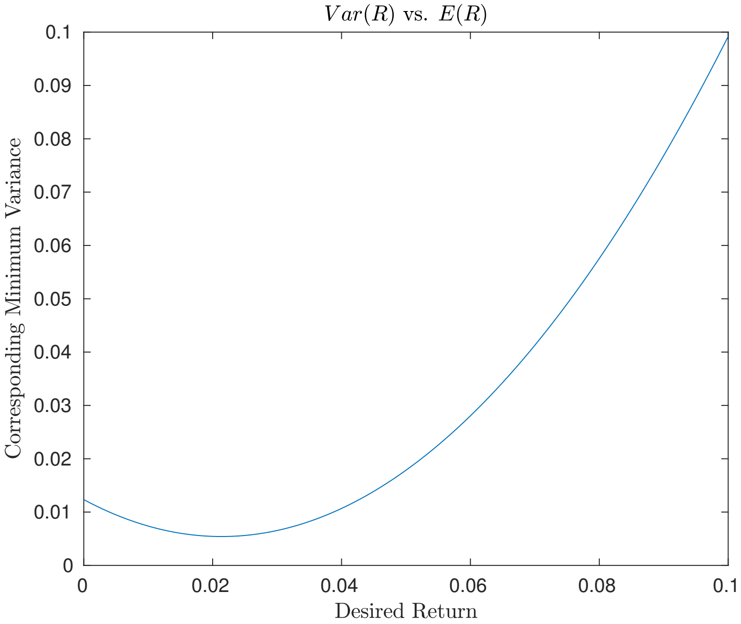
we know the set has an infimum which is in fact a minimum by the continuity of all functions involved. This guarantees that the method of Lagrange Multipliers will in fact find that minimum.

(II) The second such set actually leads to a nice graph which shows what's going on.

If we plot a set of desired returns μ_0 along with their corresponding variance values we see the pattern:

Example 15.9 Revisited.

Here is a plot of the minimum variance values coresponding to desired return values between 0 and 0.1:



From this graph we see that there is a minimum variance of about 0.005 corresponding to a return of slightly more than 0.02. The exact values are 0.0054 and 0.0214 which we found as our global minimum variance portfolio.

(III) Why does the matrix inverse exist when we used it?

Omitted for now since a simple explanation isn't clear to me.

15.5 Matlab

If we have the covariance matrix we can create the appropriate necessary matrix very easily for the global minimum variance portfolio easily.

Here is an example where we start with the covariance matrix Σ, construct the matrix:

$$M = \begin{bmatrix} 2\Sigma & \bar{1} \\ \bar{1}^T & 0 \end{bmatrix}$$

and use it to find the global minimum variance portfolio using the inverse as per the text:

```
>> S = [
0.0110    0.0015    0.0008
0.0015    0.0121    0.0016
0.0008    0.0016    0.0218];
>> M = [
2*S ones(3,1)
ones(3,1)’ 0
]
M =
    0.0220    0.0030    0.0016    1.0000
    0.0030    0.0242    0.0032    1.0000
    0.0016    0.0032    0.0436    1.0000
    1.0000    1.0000    1.0000         0
>> a = inv(M)*[zeros(3,1);1]
a =
    0.4271
    0.3672
    0.2057
   -0.0108
```

Note: We could do `ones(1,3)` instead of `ones(3,1)’` but it’s been left this way to be consistent with the mathematical notation in the text.

Then we can also find the associated variance:

```
>> x = a(1:3);
>> x’*S*x
ans =
    0.0054
```

If in addition we have the expected value vector and a specified desired return then we can find the minimum variance portfolio.

Here is how we construct and use the matrix:

$$M = \begin{bmatrix} 2\Sigma & \bar{\mu} & \bar{1} \\ \bar{\mu}^T & 0 & 0 \\ \bar{1}^T & 0 & 0 \end{bmatrix}$$

```
>> mu = [0.0385;0.0021;0.0202]
mu =
    0.0385
    0.0021
    0.0202
>> M = [
2*S mu ones(3,1)
mu' 0 0
ones(3,1)' 0 0
]
M =
    0.0220    0.0030    0.0016    0.0385    1.0000
    0.0030    0.0242    0.0032    0.0021    1.0000
    0.0016    0.0032    0.0436    0.0202    1.0000
    0.0385    0.0021    0.0202         0         0
    1.0000    1.0000    1.0000         0         0
>> inv(M)*[zeros(3,1);0.0305;1]
ans =
    0.6770
    0.1154
    0.2076
   -0.2770
   -0.0049
```

If we'd like to check the variance that's easy:

```
>> a = inv(M)*[zeros(3,1);0.0305;1];
>> x = a(1:3);
>> x'*S*x
ans =
    0.0067
```

The graph of variance versus return can be drawn easily. Here is the example which appears in the text. We plot all return values from 0 to 0.01 in steps of 0.001. The titles and labels are just there to be pretty. The picture is omitted because it's included in the text.

```
>> X = [0:0.001:0.10];
>> Y = [];
>> for r=X;a=inv(M)*[ones(3,1);r;1];x=a(1:3);Y=[Y,x’*S*x];end;
>> plot(X,Y)
>> title(’$Var(R)$ vs. $E(R)$’,’interpreter’,’latex’);
>> xlabel(’Desired Return’,’interpreter’,’latex’);
>> ylabel(’Corresponding Minimum Variance’,’interpreter’,’latex’);
```

15.6 Exercises

Exercise 15.1. Suppose the covariance matrix for the stocks of three companies 1, 2 and 3 is given by:

$$\Sigma = \begin{bmatrix} 0.0250 & 0.0018 & -0.0021 \\ 0.0018 & 0.0301 & 0.0010 \\ -0.0021 & 0.0010 & 0.0111 \end{bmatrix}$$

(a) How should a portfolio be allocated in order to minimize the variance and what would the associated variance be?

(b) If the average returns of the three stocks are given by $\mu = [0.0100; 0.0200; 0.0182]$ and the desired return is 0.0150 how should a portfolio be allocated in order to minimize the variance and what would the associated variance be?

(c) Answer the previous questions in the context of a portfolio with total value \$84230.

Exercise 15.2. Suppose the covariance matrix for the stocks of three companies 1, 2 and 3 is given by:

$$\Sigma = \begin{bmatrix} 0.0200 & -0.0005 & -0.0009 \\ -0.0005 & 0.0210 & -0.0007 \\ -0.0009 & -0.0007 & 0.0180 \end{bmatrix}$$

(a) How should a portfolio be allocated in order to minimize the variance and what would the associated variance be?

(b) If the average returns of the three stocks are given by $\mu = [0.0200; 0.0210; 0.0112]$ and the desired return is 0.0250 how should a portfolio be allocated in order to minimize the variance and what would the associated variance be?

(c) Answer the previous questions in the context of a portfolio with total value \$8.4M.

Exercise 15.3. At the instant this problem is written the 30-day historical data for Bitcoin and Ethereum has approximately the following covariance matrix where index 1 corresponds to Bitcoin and index 2 corresponds to Ethereum:

$$\Sigma = \begin{bmatrix} 0.024 & 0.012 \\ 0.012 & 0.019 \end{bmatrix}$$

How should a portfolio be allocated in order to minimize the variance and what would the associated variance be?

Exercise 15.4. Suppose the covariance matrix for the stocks of three companies 1, 2 and 3 is given by:

$$\Sigma = \begin{bmatrix} 0.0250 & 0.0018 & -0.0021 \\ 0.0018 & 0.0301 & 0.0010 \\ -0.0021 & 0.0010 & 0.0111 \end{bmatrix}$$

Suppose the average returns of the three stocks are given by $\mu = [0.0100; 0.0200; 0.0182]$.

(a) If the desired return is r how should a portfolio be allocated in order to minimize the variance? Note that your answer will have r in it.

(b) What would the associated variance be?

(c) Use the answer to the previous question to calculate the return r which would minimize the variance and calculate that minimum variance. This latter answer should agree with the answer to Exercise 15.1(a).

(d) For which values of r can you avoid shorting any of the stocks?

Index