10 A Taste of Further Applications

Further applications of linear algebra include: (i) ranking sports teams and websites; (ii) binary vector spaces and error-correcting codes; and (iii) principal component analysis. We give here an extremely brief introduction.

10.1 Ranking Sports Teams and Websites

10.1.1 Massey's Original Rating Idea

Consider trying to assign ratings to sports teams. By **rating** we mean a numerical value. (From this one can produce ordinal values, if desired.) The following is the original idea for rating put forward by Massey. Suppose there was a rating such that: *the result (net score) between any teams was always given by the difference in their ratings.* Then one obtains a system of equations of the form

$$r_i - r_j = g_{ij}$$

where r_i and r_j are the (unknown) ratings of teams i and j, and g_{ij} is the actual net score of the game between them.

So the task is, given the actual results g_{ij}, to determine the r_i. There are of course probably many more equations than variables; so this system of linear equations almost surely has no solution (it is "over-determined"). So instead, we try to find the "best" vector $\boldsymbol{r}$. The standard approach is to minimize the sum of the squares of the errors, where here the error is $g_{ij} - (r_i - r_j)$. With a few lines of (vector) calculus, one obtains a system of equations:

$$\text{for each } i\text{:} \quad \text{net rating for team } i = \text{net result for team } i$$

where net rating is the sum of rating differences (the $r_i - r_j$) and net result is the net score (sum of g_{ij}). That is, we need to solve the matrix equation

$$S\boldsymbol{r} = \boldsymbol{N}$$

where S is square matrix such that each diagonal entry is the number of games that team played; and off the diagonal the value is -1 if the corresponding teams played, and 0 otherwise. And vector $\boldsymbol{N}$ is the net score for each team: points-for minus points-against.

However, matrix S is not invertible. There are two ways to see this. The first is that each row of S sums to 0, and thus the all-1 vector is in its null-space. The second way to see this is that the solution for $\boldsymbol{r}$ is not unique: one can add the same constant to all entries and still have a solution. So the way to resolve this is to discard one row of S (since it is redundant), and add a row that normalizes the ratings. For example, add the equation that the ratings sum to 0. Then go ahead and solve, using some software.

10.1.2 Page's Idea

Consider ranking a collection of websites. Suppose there is a bot that does a **random walk** on this collection: at each site it randomly chooses one outgoing link and follows it. (Sometimes called a Markov process.) The idea behind what is called PageRank is that maybe the "importance" of a website is determined by the average proportion of time the bot spends there. Note that one needs to deal with websites with no outgoing links.

Assuming this random-walk proportion converges to some vector π, then you get system of equations saying

$$\text{for each } i\text{:} \quad \pi_i \text{ equals sum of } \tfrac{1}{c_j}\pi_j$$

where the sum is over all j that have a link to i, and c_j is how many outgoing links j has in total. This can be written as the matrix equation $\pi = A\pi$ for some matrix A. So what we want is the eigenvector of A for the eigenvalue 1.

10.2 An Introduction to Error-Correcting Codes

10.2.1 Binary Vector Spaces

For this section the arithmetic will only use the numbers `0` and `1`. That is, arithmetic will be "modulo 2". This is defined as the remainder when the ordinary result is divided by 2. So the arithmetic tables are

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

Define $\mathcal{Z}^n$ as the set of all vectors with n components each of which is either a 0 or a 1. (So there are 2^n vectors in $\mathcal{Z}^n$.) One can view $\mathcal{Z}^n$ as a vector space, where the only allowed scalars are 0 and 1, and all calculations are in binary arithmetic. Most of the definitions we made and theorems we proved about vector spaces are still valid.

10.2.2 Error-Detecting and Error-Correcting Codes

Suppose one has a binary message to send. Then the goal is to "enhance" the message (that is, add redundancy) so that even if the message is somewhat corrupted in transit, the original message can be recovered. (Note that this is different to encryption, where the goal is to send data that is private.)

Example. *One idea is to take the message data and append a **check-bit**: the sum of the message bits in binary arithmetic. For example, if we have two message bits, then what is sent is either 000, 101, 011, 110. The claim is that: the receiver can detect if exactly one bit is changed in transit. But the message has to be resent.*

Example. *Another idea is to repeat oneself. For example, send each message bit three times. So if we have two message bits, then what is sent is either 000000, 000111, 111000, 111111. The claim is that: the receiver can detect and correct a change in one bit.*

The above examples can be generalized. We define a **code** as a collection of binary vectors. For two vectors of the same length, their **Hamming distance** is the number of places the corresponding entries are different. The **distance** of a code is the minimum Hamming distance between any two vectors in it. It can be shown that:

Fact 10.1 *(a) A code can detect k errors if and only if its distance is at least $k+1$.*
(b) A code can correct k errors if and only if its distance is at least $2k+1$.

What one would like is a code that (a) is efficient (the sent message is not much longer than the original), and (b) is easy to decode (if the code has the property that one can correct some number of errors, then these corrections can be done quickly).

10.2.3 Linear Codes

A **linear code** is a subspace of $\mathcal{Z}^n$. The generator matrix M has rows that are the basis of the linear code. Hence, for message data d (treated as a row-vector or matrix with single row), what is sent is

$$s = dM$$

EXAMPLE. *We saw above the codes 000, 101, 011, 110 and 000000, 000111, 111000, 111111. Their generator matrices are:*

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

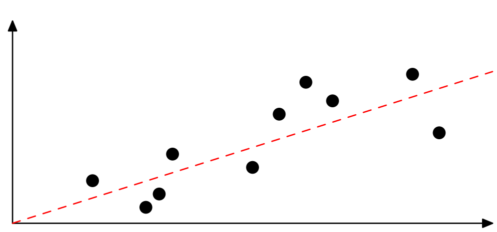
Here is the generator matrix of what is called the **Hamming code**. It can be thought of as adding three check-bits to four data-bits.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

For example, suppose the data is 1101. Then the sent message is 1010101.

10.3 Principal Component Analysis

Consider a typical scatter plot. One can shift it so that regression line goes through origin. Then treat the regression line as new axis. In 2D, the projection onto regression line gives the best dispersion of data.



More generally, suppose data matrix A has rows of observations (where columns represent statistical variables). Consider $V = A^T A$. This matrix is square and symmetric, so its eigenvalues are real. In fact it turns out all its eigenvalues are nonnegative, and eigenvectors from different eigenvalues are orthogonal. The **principal components** of data A are the (unit) eigenvectors of $A^T A$. This is related to what is called the covariance matrix in statistics.

We know these eigenvectors form an eigen-basis of $\mathbb{R}^n$. So one can represent each vector in terms of the coordinates. Because the eigenvectors are orthogonal, the projections are independent. We write the eigenvectors in decreasing order of the eigenvalues. If the eigenvalues decay quickly, then one can approximate each vector by only a few entries in this coordinate system. This transforms a high-dimension system into a low-dimension system.

10.4 Some References

Statistical Models Applied to the Rating of Sports Teams, Kenneth Massey, 1997.

Who's #1? The science of rating and ranking. A.N. Langville and C.D. Meyer, Princeton University Press, 2013.