Linear Solvers in Communication and Signal Processing

January 12, 2022

Note: I may use n and N interchangeably. Apologies for this inconsistency. When we work with column vectors $\bar{x}$ will be used to underline this, a vector and column. However, I might have missed a few bars. Kindly communicate this back, or any other typos/errors.

1 Introduction

These notes are to be used as an addendum to the slides, with no intention of covering all the ground. The purpose of the lectures is served even if you get interested to look further, or understand how simple principles are engineered to perform higher order functions. In particular, solving linear equations is something each one of you have done many times during your studies elsewhere. We will show some interesting applications. In the past, most of us have dealt with a set of linear equations, written in the matrix form as

$$\underset{n\times 1}{y} = \underset{n\times n}{A}\,\underset{n\times 1}{x},$$

where we have indicated the dimensions below each variable. Our job is to figure out the vector $x := [x_1, \cdots, x_n]^T$ from the observations $y := [y_1, \cdots, y_n]^T$. Notice that our representations assume column vectors, unless otherwise specified. Some kind of elimination method is usually employed in solving these problems using pen and paper. In any case, the unique solution is

$$x = A^{-1}\, y,$$

whenever the matrix A^{-1} exists.

Exercise 1. *Consider a square matrix A having two columns the same. Find the determinant of A. How will your answer change if the columns are all different, but two rows are found to be the same.*

The answers to the above are easily seen by standard elimination methods. For some other matrices you have to compute and see. Here is a very special matrix for which you can calculate the determinant without much effort.

Exercise 2. *Find the determinant (in terms of the entries) of the 4×4 matrix*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a & b & c & d \\ a^2 & b^2 & c^2 & d^2 \\ a^3 & b^3 & c^3 & d^3 \end{bmatrix}.$$

For an easy solution to the above problem, notice that if any two entries of the second row are the same, the condition mentioned in Exercise 1 occurs. Thus $(a - b)$ has to be factor of the determinant (why?). More generally, one can consider a $n \times n$ matrix A such that the entry A_{kl} at position (k, l) is $\alpha_{l-1}^{k-1}, 1 \leq l \leq n, 1 \leq k \leq n$. We will call the case where $\alpha_0, \alpha_1, \cdots, \alpha_{n-1}$ are all different as a **Vandermonde** matrix (Many textbooks may call it Vandermonde even without the condition as mentioned, but we need it here).

Recall that while inverting a matrix, the usual procedures will have the matrix-determinant in the denominator. Not only that these methods fail when $\det(A) = 0$, but also no method can succeed then.

Exercise 3. *Show that $det(A) \neq 0$ is necessary and sufficient for a square matrix A to be invertible.*

Exercise 4. *Show that a Vandermonde matrix is invertible.*

To illustrate the importance of Vandermonde matrices in our further discussions, we give a special notation for the $n-$ dimensional case. The notation F below stands for Fourier (the same guy from the Fourier Transform), and we have denoted $n' = n - 1$ (so that the matrix appears *elegant*).

$$F = \begin{bmatrix} \alpha_0^0 & \alpha_1^0 & \alpha_2^0 & \cdots & \alpha_{n'}^0 \\ \alpha_0^1 & \alpha_1^1 & \alpha_2^1 & \cdots & \alpha_{n'}^1 \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_{n'}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha_0^{n'} & \alpha_1^{n'} & \alpha_2^{n'} & \cdots & \alpha_{n'}^{n'} \end{bmatrix} \tag{1}$$

Clearly F is invertible. Thus, given $y = Fx$, we can easily find $x = F^{-1}y$. Let us connect this to *picking the odd ones* experiments we did in class. Suppose the signal x has at most one non-zero value. Given $y = Fx$, we can find x for sure. However, how about if we are given only the first entry y_1 of y. Observe the matrix F and see that the first entry of Fx will simply give the sum of all the entries in x, and the sum clearly is independent of the position of the non-zero element in the vector. Thus, there is no way one can get back x from y_1. How about from the first two elements $[y_1, y_2]$ of y. There is more hope here. In particular $[y_1, y_2]^T$ is nothing but Ax, where A contains the first two rows of F (verify this by observation). Notice also that no two columns of A are collinear. Thus, one column of A multiplied by a scalar value will never *fall in line* with a ray (or line) through another column. So, one look at the direction of $[y_1, y_2]$ vector is good enough to identify the position of the non-zero input. Once you know the position, then y_1 will tell the value of the non-zero element. That is it.

Exercise 5. *Assume that there is only one non-zero entry in x. Can we use any two rows of F as the matrix A, and then recover x from Ax.*

When the number of non-zero elements is two (or more), we need more sophistication to figure out x from Ax. First of all, vectors not being collinear is not enough. What if one of the columns of A, say $\bar{a}_1$ is a linear combination of two others, $\bar{a}_2$ and $\bar{a}_3$. Then it is possible that the direction of output can be either due to $\bar{a}_1$, or both $\bar{a}_2$ and $\bar{a}_3$ has happened at an appropriate proportion. We have no way of distinguishing, and our recovery procedure may fail. It turns out that in the presence of two odd elements, using the first four rows of F as A suffices to recover x from Ax. Let us prove this last statement.

Claim 1. *Among the set S of all $n \times 1$ vectors having at most two non-zero entries, there exist a unique solution to $y = Ax$, when A is chosen as the first four rows of F.*

Assume that there exists vectors u and v, each having at most two non-zero entries, with $u \neq v$. Our recovery will fail if $y = Au = Av$, as we cannot decidedly choose between u and v. The argument to show this never happens is called a *contradiction* method. To form a contradiction, start with assuming $Au = Av$. This will imply that with $w = u - v$, $Aw = 0$. However w has at most 4 non-zero entries. Take any 4×4 sub-matrix of A, call this B. Observe that B is a square matrix, and we know it to be invertible, see Exercise 2. Thus, there is no non-zero vector z such that

$Bz = 0$. This will simply mean that $Aw \neq 0$. In other words $Au \neq Av$, contradicting our initial assumption. Thus u and v cannot produce the same output without at least one of them having more non-zero entries than two. Our claim on uniqueness now follows.

For both the exercises below, assume that the vector x has real valued entries.

Exercise 6. *Show that in order to recover to any (meaning all such) $n \times 1$ vector x having at most l non-zero entries from the corresponding product Ax, it is sufficient to consider A as the first $2l$ rows of F.*

Exercise 7. *Show that it is not possible to recover to all x, each having at most l non-zero entries, from the product Ax, if A has less than $2l$ rows.*

We have walked through a bit of theory, elementary though. Let us show how it is applied to the two situations at hand, namely *erasure correction* and *error correction*.

1.1 Erasure Correction

In erasure correction, we will take a data vector $\bar{d} = [d_1, \cdots, d_k]^T$. We can now take A as the first k rows of the matrix F. Store the $n \times 1$ vector $\bar{x} = \bar{d}^T A$ into the memory, which is susceptible to possible erasures while being read. Suppose there are at most m erasures for every n symbols stored. From the $n - m$ correctly read values in each block of n, we can form $n - m$ equations to recover the k data values. If $k \leq n - m$, take the first k unerased stored values. Clearly, these values are created as a linear combination of k columns of A. But k columns of A form a square matrix (as we have taken k rows for A), and we know it to be invertible. Thus we can find the data vector $\bar{d}$.

1.2 Error Correction

When there are errors, we assume an additive model where $\bar{y} = \bar{x} + \bar{e}$, where the operations are assumed to take place in the real field. In the notation, $\bar{x}$ *embeds* the data in an appropriate form, and $\bar{e}$ is an unwanted error term, often modeled as additive noise. While the spirit of error correction is similar to the erasure case, there are some key differences. We will describe one such strategy for error correction.

In order to convey/store k data values $\bar{d} = [d_1, \cdots, d_k]^T$, let us first append $n-k$ zeros at the beginning to form an $n \times 1$ vector $\hat{d} = [0, \cdots, 0, d_1, \cdots, d_k]$.

Now we simply compute the product $F^{-1}\hat{d}$, where F is the $n \times n$ Vandermonde matrix introduced earlier. The inverse operation is more of a convenience, and you will learn later that there are easy ways of computing it. Then $\bar{x} = F^{-1}\hat{d}$ is stored in the disk/memory (or even send across a medium). On receiving/reading-out $\bar{y} = \bar{x} + \bar{e}$, we first compute the product

$$F\bar{y} = F(\bar{x} + \bar{e}) = F(F^{-1}\hat{d} + \bar{e}) = \hat{d} + F\bar{e}.$$

If the error vector had all zeros, i.e. no errors, then the first $n - k$ entries of $F\bar{y}$ will be zeros, since $\hat{d}$ has zeros there. This may not be the case if the error is not zero, this can be leveraged to identify if there are errors. Thus, it makes sense to consult the first $n - k$ entries of $F\bar{y}$. We call this the syndrome $\bar{s}$, i.e. $\bar{s} = A\bar{e}$, where A is nothing but the first $n - k$ rows of the Vandermonde matrix F.

The remaining question is whether we can find $\bar{e}$ from $A\bar{e}$. This is clearly the case if $\bar{e}$ has at most $\frac{n-k}{2}$ non-zero entries, the theory of which we already covered, see Exercise 6.

Exercise 8. *Suppose $n = 255$ and $k = 232$, and we have a system with possible additive errors (can be zero value as well) affecting each symbol. How many maximum errors can you successfully correct in a block of n stored symbols. Assume the $n-$ length vectors $\bar{x}$ and $\bar{y}$ to be generated as above.*

Exercise 9. *In the previous question, what is the maximum number of errors that you can detect. Notice that sometimes you may be able to detect, but not correct the errors.*

2 Questions and Answers

Question 1. *In the real world, we cannot be sure of the value of s right? Then do we take something like the powers of primes or something?*

It is true an accurate knowledge of s can be beneficial. However, we are only assuming at most s non-zero values. This will suggest in our schemes that we need $2s$ equations for our decoding to work. So, if you use $2s$ measurements and the number of non-zero values are below s, then the scheme works. Later results also show that from $2s \log_2 n$ measurements, one can nearly get the s biggest values. So an approximate idea of s is good enough.

Question 2. *Please once again explain the blue leds*

The blue LEDs signify positions where the existing value was flipped. A bright spot became dark or vice-versa, can happen due to physical problems, or optical reading issues. These are called errors. I have shown later how to deal with erasures and how to deal with errors. These were done separately, in reality combined techniques are also possible.

Question 3. *Could you please tell the complex form of α.*

The key idea behind all the schemes we used is that a square Vander monde matrix is invertible if any of the rows have distinct values. So if we take

$$[1\alpha_1, \alpha_2, \ldots, \alpha_{N-1}] = [\exp(-j\frac{2\pi}{N}0), \exp(-j\frac{2\pi}{N}1)\exp(-j\frac{2\pi}{N}2), \ldots \exp(-j\frac{2\pi}{N}(N-1))],$$

we still get a unique output vector for each input vector with atmost s non-zero values, provided we take $2s$ rows of F. The choice above will make the transform of the form $\bar{u} = F\bar{x}$, where $\bar{u}$ is called the discrete Fourier Transform of $\bar{x}$.

Question 4. *why are we required to put 0 in the $n-k$ values of k?*

To match the matrix dimensions for multiplication.

Question 5. *What is $\bar{c}$?*

$\bar{x}$ represents a column vector. Sometimes the row form is more convenient to visualize, then I will take transpose or write down the entries. See the equation above.

Question 6. *$m + k = N$ right?*

Yes, $m + k = N$, some times I may use little n, apologies. This has happened since I tailored stuff from different classes.

Question 7. *$k \leq m$ ork $+ m = n$?*

Thanks for following, though I will not ask these things in the exam. Since there are k unknowns, we need just k independent linear equations. After m erasures you will have $n - m$ unerased outputs. If $n - m \geq k$ you can solve any set of k independent equations. Our choice of the multiplying matrix at the encoder ensures that any k unerased values are good enough.

Question 8. *i think we are taking the limiting case*

No, please see the answer above.

Question 9. *will the quality decrease if we compress the information??*

Not always. We can lossless compression with no loss of information. However, in visual perception the quality is subjective, and some loss of data will mostly go unnoticeable, unless we project on a huge screen. So many images use a lossy compression like JPEG. However many high quality cameras have the option for lossless storage. For example, PNG/PPM can store after lossless compression.