



IIT Madras
ONLINE DEGREE

Mathematics for Data Science 1
Professor. Madhavan Mukund
Department of Computer Science
Indian Institute of Technology, Madras
Lecture No. 71
Matrix Multiplication

(Refer Slide Time: 00:14)

Matrix Multiplication

Madhavan Mukund
<https://www.cmi.ac.in/~madhavan>

Mathematics for Data Science 1
Week 11

We have used matrices to represent graphs. So, we have this adjacency matrix. And when we looked at the transitive closure problem, we suggested that it might be useful to describe the transitive closure computation, using some operations directly on matrices. So, let us look at matrix operations. In particular, we want to lead up to this notion of matrix multiplication.

(Refer Slide Time: 00:36)

Matrices

- A matrix is a two dimensional table
 - $r \times c$ matrix — r rows, c columns
 - Assume rows are numbered $\{0, 1, \dots, r-1\}$, columns are numbered $\{0, 1, \dots, c-1\}$
 - Graph with n nodes, $n \times n$ adjacency matrix, entries are from $\{0, 1\}$
- Example: Freight traffic by rail between major cities: Bangalore, Chennai, Delhi, Hyderabad, Kolkata, Mumbai
 - Number the cities:
0-Bangalore, 1-Chennai, 2-Delhi,
3-Hyderabad, 4-Kolkata,
5-Mumbai
 - Represent freight volume for a month as a 6×6 matrix

	0	1	2	3	4	5
0	0	694	828	384	247	479
1	642	0	919	575	402	673
2	768	734	0	231	595	540
3	731	606	156	0	351	804
4	825	607	316	490	0	998
5	196	580	339	588	394	0

So, a matrix in general is a 2-dimensional table of values, usually numbers, and it has a certain number of rows, because it is 2 dimensions. So, it has rows and it has columns. So, usually, it has some r rows and some c columns. r and c could be different numbers. And we write this as $r \times c$, or we call it an $r \times c$ matrix. So, r is the number of rows c is the number of columns. And as usual, we will number the rows starting from 0 and the column starting from 0. So, the rows are 0,1,2, up to $r-1$, and the columns are 0,1,2, up to $c-1$.

So, in the example that we have been looking at the adjacency matrix, $r=c=n$, which is the number of vertices in our graph. So, we have 1 row and 1 column for every vertex in the graph. So, we number our columns and rows is 0 to $n-1$. In this particular case, the entries in our matrix are 0 and 1, but in general, they could be any numbers, even other values in general, but we will look mainly at matrices where the entries are numbers.

So, for a completely different example, let us consider that we want to record some information about freight traffic by rail between some cities. So, let us take say the 6 biggest cities in India, Bangalore, Chennai, Delhi, Hyderabad, Kolkata, and Mumbai. And suppose we have information about the volume of traffic, say in terms of 1000s of tonnes of millions of tonnes, whatever is the appropriate metric for this. So, we have some information. So, we want to represent this metric.

So, the first thing is that every city will correspond to an entry in the matrix in a row and the column. So, because we are normally numbering our rows and columns from 0 to $n-1$, or 0 to $r-1$, and 0 to $c-1$ in this case, again, notice that it is going to be a square matrix. So, it is going to have 6 rows and 6 columns. Let us just number them in alphabetical order. So, 0 is Bangalore, 1 is Chennai, 2 is Delhi, 3 is Hyderabad, 4 is Kolkata and 5 is Mumbai, it does not really matter in what order we put it.

So, this could be the matrix. So, this is not a symmetric matrix, as they say. So, there is a certain amount of traffic say, from Bangalore to Chennai, but there is a different amount of traffic from Chennai to Bangalore, because in general, there is no reason why the amount of freight traffic should be the same in both directions. Notice that along the diagonal, there is no freight traffic recorded from a city to itself, because that does not make any sense for us. So, this is another way of using a matrix where we are representing some information about freight between pairs of cities.

(Refer Slide Time: 03:02)

Adding matrices



- Suppose we have freight volumes for first and second half of financial year.

April–September							October–March						
	0	1	2	3	4	5		0	1	2	3	4	5
0	0	694	828	384	247	479	0	0	851	626	280	399	365
1	642	0	919	575	402	673	1	544	0	479	269	432	933
2	768	734	0	231	595	540	2	867	804	0	681	326	398
3	731	606	156	0	351	804	3	727	976	418	0	667	294
4	825	607	316	490	0	998	4	894	390	247	547	0	314
5	196	580	339	588	394	0	5	914	147	574	859	524	0

- How do we compute the freight volumes for the entire financial year?

- Add the corresponding entries in the two tables

- Total freight volume from 2 (Delhi) to 4 (Kolkata) is $595 + 326 = 921$

Navigation icons

Madhavan Mukund

Matrix Multiplication

Mathematics for Data Science



So, now, suppose we have this information, and it is counted over a certain period of time. So, in India, you may know that the financial year starts in April. So, April 1st is the beginning of the financial year and March 31st is the end. So, maybe for the first half of the financial year, from April 1 to September 30th, we have a certain number of amounts of information about these freight volumes, and that is the matrix on the left. So, this is the first 1, so, this is the first half year and then we have a corresponding thing for the second half year. So, the numbers are different.

So, for instance, here, there is only 247 whereas here in the same entry from Bangalore to whatever was 4, you have Kolkata I think, there you have 399. So, now, a natural thing would be to compute the freight volume for the full year given this information about the freight volume in the first half of the year, and the freight volume in the second half of the year. And quite clearly, the freight volume for any pair of cities is going to be the sum of the 2.

So, if you want to know how much traffic there has been from Delhi, which is city 2 to Calcutta, which is city 4, you look up the entry for the first half of the year, it says okay, there were 595 million tonnes or whatever it is from Delhi to Calcutta in the first half. And then the second half, there are 326. So, clearly put together it is $595 + 326$, which is 921. And this is how we would compute this information.

(Refer Slide Time: 04:22)

Adding matrices

- For a matrix M , $M[i, j]$ is the entry in row i , column j
- Let A and B represent the volumes in the two half-years
- Let C represent the annual volume
- For each i, j , $C[i, j] = A[i, j] + B[i, j]$
- More concisely, $C = A + B$ — matrix addition

	0	1	2	3	4	5
0	0	694	828	384	247	479
1	642	0	919	579	402	673
2	768	734	0	231	595	540
3	731	606	156	0	351	804
4	825	607	316	490	0	998
5	196	580	339	588	394	0

	0	1	2	3	4	5
0	0	851	626	280	399	365
1	544	0	479	209	432	933
2	867	804	0	661	326	398
3	727	976	418	0	667	294
4	894	390	247	547	0	314
5	914	147	574	859	524	0

	0	1	2	3	4	5
0	0	1545	1454	664	646	844
1	1186	0	1398	844	834	1606
2	1635	1538	0	912	921	938
3	1458	1582	574	0	1018	1098
4	1719	997	563	1037	0	1312
5	1110	727	913	1447	918	0

So, some notation. So, if we have a matrix, then remember our rows are numbered 0 to $r-1$ and our columns are 0 to $c-1$. So, we will look at the I 'th row in the J 'th column, and that will give us 1 entry in the matrix, where the row intersects the column and we will refer to that as $M_{i,j}$. So, $M_{i,j}$ is the entry in the matrix at row i and column j .

So, now here is our information about the 2 half years again, presented on the right-hand side. So, these are the 2 matrices that were given. And we want to compute the aggregate volume for the whole year into a new matrix, call it C . So, I have used A for the first half year, B for the second half year, and now we want to compute a matrix C , which represents the total volume for the whole year. And as we said before, the ij 'th entry, the entry in row I column, J of C is going to be the sum of the ij 'th entries in A and B . It is quite obvious.

So, if we want to look at a pair of cities and find out the total traffic between the 2 cities, you add up what traffic was there in the first half of the year to what traffic was there in the second half of the year. So, when we do this, we are doing this for every element. So, we are taking this element adding this element and getting this element, we are taking this element, this element and getting this element. So, we are doing it in the same way for every element of the matrix. And so, we can just concisely say that we are adding these 2 matrices. So, this is called matrix addition.

So, when we have a matrix A and a matrix B , and they both have the same size, the same number of rows and the same number of columns, then for every position in A there is a corresponding position in B . If I look at row ij , in column J A , there is a row in column J and B because they have the same size. And So, I can now add these entries and put it into a third

matrix of the same size. And I will just write $C = A+B$. So, $C = A+B$ captures this fact that for every row and every column, I am adding the entries in A and B to put them into C . So, remember that we were not so interested in adding matrices but multiplying them. So, what should we do for multiplying matrices?

(Refer Slide Time: 06:25)

Multiplying matrices

- Can we multiply matrices the same way?
- For each i, j , $C[i, j] = A[i, j] \times B[i, j]$

	0	1	2	3
A	0	0	6	8
1	4	0	9	7
2	6	3	0	1
3	1	0	5	0

	0	1	2	3
B	0	0	1	2
1	5	0	4	2
2	6	0	0	1
3	2	6	4	0

	0	1	2	3
C	0	0	6	16
1	20	0	36	14
2	36	0	0	1
3	2	0	20	0

Madhavan Mukund Matrix Multiplication Mathematics for Data

So, can we multiply matrices the same way? So, let us take simpler numbers. So, here are 2 matrices A and B . And suppose we do the same rule, the same rule would say instead of adding A_{ij} , into B_{ij} multiply A_{ij} , by B_{ij} . So, here, for instance, if I look at C 1, 0, it is A 1, 0, $4 \times B$ 1, 0, $5, 4 \times 5$ is 20. Similarly, if I look at, say for instance, A 3, 2, it is 5, B 3, 2 is 4. So, again, it is 20. If I look at this entry, it is 9×4 is 36. So, I have just multiplied entries, one at a time.

So, this could be a candidate for the operation of matrix multiplication is the obvious way in which we can take addition and replace addition by multiplication.

(Refer Slide Time: 7:14)

Multiplying matrices



- Can we multiply matrices the same way?

- For each i, j , $C[i, j] = A[i, j] \times B[i, j]$

- This turns out to be not very useful

- Instead, we compute $C[i, j]$ in a more complicated way

- Assume $r = c = n$. Let

$$C[i, j] = A[i, 0] \cdot B[0, j] + A[i, 1] \cdot B[1, j] + \dots + A[i, n-1] \cdot B[n-1, j]$$



	0	1	2	3
0	0	6	8	3
1	4	0	9	7
2	6	3	0	1
3	1	0	5	0

	0	1	2	3
0	0	1	2	8
1	5	0	4	2
2	6	0	0	1
3	2	6	4	0



$C[i, j]$

Navigation icons

Madhavan Mukund

Matrix Multiplication

Mathematics for Data Science



Now, it turns out that this particular way of manipulating matrices, there is nothing wrong with it, it is just that it does not turn out to be very useful in practice, addition made sense for us. Multiplication, like this usually does not give us any useful number. So, there is a different operation, which is given the name matrix multiplication, which is a little more complicated than this.

So, we want to compute the multiplication $A \times B$, where A and B are matrices. So, again, we have to tell us a rule for computing the ij 'th entry of the product. So, the product of A and B has, again, entries i comma j . So, what is the value that goes into C ij ? So, here is the complicated rule. The complicated rule is that you look at row i . So, you look at row. So, this is my matrix A and my matrix B . So, if I want to compute the ij 'th entry in C , I look at row i in A and I look at column j in B .

So, I look at the first entry here, what is the first entry here it is i comma 0 . And look at the first entry there. What is the first entry there it is row 0 , column j . So, I multiply those 2 numbers. Now I move to the right, and I look at the next number here. That is $1, i$ comma 1 . And I look at the next number in that column, that is 1 comma j . And I multiply those 2 numbers. So, I keep walking down the row and A , A 'th row at the same time I walked down the j 'th column and in B , and every point I stop and I multiply the 2 numbers that I get, I take all these numbers and add them up. It seems really complicated, but this is how matrix multiplication works.

(Refer Slide Time: 8:56)

Multiplying matrices



- Can we multiply matrices the same way?

- For each i, j , $C[i, j] = A[i, j] \times B[i, j]$

- This turns out to be not very useful

- Instead, we compute $C[i, j]$ in a more complicated way

- Assume $r = c = n$. Let

$$C[i, j] = A[i, 0] \cdot B[0, j] + A[i, 1] \cdot B[1, j] + \dots + A[i, n-1] \cdot B[n-1, j]$$

- For instance, $C[1, 3] = 4 \cdot 8 + 0 \cdot 2 + 9 \cdot 1 + 7 \cdot 0 = 41$

	0	1	2	3
0	0	6	8	3
1	4	0	9	7
2	6	5	0	1
3	1	0	5	0

	0	1	2	3
0	0	1	2	3
1	5	0	4	6
2	6	0	0	0
3	2	6	4	0



Madhavan Mukund

Matrix Multiplication


Mathematics for Data Science

So, let us do an example. So, supposing I take these 2 matrices, and I want to find out, what is the final entry going to be for $C[1, 3]$, that is row 1, column 3 of the final matrix, what is it going to be? Well, I have to look at row 1 in A, and I have to look at row 3 in B. And then I have to start one by one. So, first, I multiply 4 by 8. So, that is my first component, then I have 0, which is nice, 0×2 is 0, no problem, then I have 9×1 is 9. And then I have again, 7×0 , another 0, so, I get 32. And I get 9, so, $32+9$, so, I get 41.

So, this is how I compute 1 entry. And this is just 1 entry. And I have to do this for every position. So, for every position in the output, I have to scan a row of the input and a column of the second matrix or row of the first matrix and add up those that many, I have to multiply each individual pair and add them all up. So, I have to do a lot of work to compute 1 entry.

(Refer Slide Time: 9:55)

Multiplying matrices



IIT Madras
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

- Can we multiply matrices the same way?
- For each i, j , $C[i, j] = A[i, j] \times B[i, j]$
- This turns out to be not very useful

$$A = \begin{bmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 6 & 8 & 3 \\ 1 & 4 & \sqrt{0} & 9 & \sqrt{7} \\ 2 & 6 & 3 & 0 & 1 \\ 3 & 1 & 0 & 5 & 0 \end{bmatrix}$$

- Instead, we compute $C[i, j]$ in a more complicated way
- Assume $r = c = n$. Let

$$C[i, j] = A[i, 0] \cdot B[0, j] + A[i, 1] \cdot B[1, j] + \dots + A[i, n-1] \cdot B[n-1, j]$$

$C[i, j]$

$$B = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 4 & 2 & 8 \\ 1 & 5 & 0 & 4 & 2 \\ 2 & 6 & 0 & 0 & 1 \\ 3 & 2 & 6 & 4 & 0 \end{bmatrix}$$

$4+0$
 $+0$
 $+42$
 $\hline 46$

- For instance,
 $C[1, 3] = 4 \cdot 8 + 0 \cdot 2 + 9 \cdot 1 + 7 \cdot 0 = 41$

$$C = \begin{bmatrix} 0 & 84 & 18 & 36 & 20 \\ 1 & 68 & 46 & 36 & 41 \\ 2 & 17 & 12 & 28 & 54 \\ 3 & 30 & 1 & 2 & 13 \end{bmatrix}$$

Navigation icons: back, forward, search, etc.

Madhavan Mukund

Matrix Multiplication

Mathematics for Data Science

And if you do this, then you get this matrix, let us look at some other entries. And see, how did we get this 46? How did we get C_{11} ? Well, C_{11} means I have to look at row 1. And I have to look at column 1. So, I take 4×1 is 4, then I get 0×0 , then I get 9×0 is 0. And then I get 7×6 is 42. So, this is how I computed it. And then I finally got 46. So, by any entry, I can do that, so, I can take this 54 and do the same thing.

(Refer Slide Time: 10:50)

Multiplying matrices

- Can we multiply matrices the same way?
- For each i, j , $C[i, j] = A[i, j] \times B[i, j]$
- This turns out to be not very useful
- Instead, we compute $C[i, j]$ in a more complicated way
- Assume $r = c = n$. Let

$$C[i, j] = A[i, 0] \cdot B[0, j] + A[i, 1] \cdot B[1, j] + \dots + A[i, n-1] \cdot B[n-1, j]$$
- For instance,

$$C[1, 3] = 4 \cdot 8 + 0 \cdot 2 + 9 \cdot 1 + 7 \cdot 0 = 41$$

A

	0	1	2	3
0	0	6	8	3
1	4	0	9	7
2	6	3	0	1
3	1	0	5	0

B

	0	1	2	3
0	0	1	2	8
1	5	0	4	2
2	6	0	0	1
3	2	6	4	0

48
 $+0$
 $+9$

 57

C

	0	1	2	3
0	84	18	36	20
1	68	46	36	41
2	17	12	28	54
3	30	1	2	13

So, I want now, if I want this 1, then I want to go to the, I want to go to 2 comma 3. So, I want to go to row 2, column 3, and now I have 8×6 is 48, $+3 \times 2$ is 6, $+0 \times 1$ is 0, $+1 \times 0$ is 0, so, I get 54. So, this is how I compute the product.

(Refer Slide Time: 11:05)

Multiplying matrices

- Matrix product: $C = A \times B$
- For $r = c = n$,

$$C[i, j] = A[i, 0] \cdot B[0, j] + A[i, 1] \cdot B[1, j] + \dots + A[i, n-1] \cdot B[n-1, j]$$
- More concisely, $C[i, j] = \sum_{k=0}^{n-1} A[i, k] \cdot B[k, j]$
- Don't require both A and B to be $n \times n$
 - Each row entry of A must have a matching column entry in B
 - If A is $m \times n$ and B is $n \times p$, $A \times B$ is $m \times p$


A $r_1 \times c_1$

B $r_2 \times c_2$

	0	1	2	3
0	0	6	8	3
1	4	0	9	7
2	6	3	0	1
3	1	0	5	0

	0	1	2	3
0	0	1	2	8
1	5	0	4	2
2	6	0	0	1
3	2	6	4	0

	0	1	2	3
0	84	18	36	20
1	68	46	36	41
2	17	12	28	54
3	30	1	2	13



So, what we have described is called the matrix product. So, this thing on the right, by computing for every. So, what we have done in particular is a square matrix. So, r is n , c is n . So, we say C_{ij} is $A_{i0} \cdot B_{0j} + A_{i1} \cdot B_{1j}$ up to $A_{i,n-1} \cdot B_{n-1,j}$, the last column, and $B_{n-1,j}$, the last row in the column J . So, there is a simple way to write this in mathematics notation, which is that you replace this entire summation by the summation sign. So, we say, let this second position range from 0 to $N-1$. K . So, we take the summation over all k ranging from 0 to $N-1$ of $A_{ik} \times B_{kj}$. So, this is just a shortcut for writing that long sum without having to specify every term and writing dot dot dot.

So, in general, we do not need these matrices to be squared, what we want is? That this thing is well defined, for this thing to be well defined. When I am walking down row i and A , I must be able to take the same number of steps in column j and B . So, therefore, if I look at the length of a row in A , if I process some number of elements in that row, then I must be able to find a matching element for each of them to multiply and then add it. So, the length of a row in A must be the same as the length of a column and B .

So, remember that we write these things as r by c . So, if I look at A . So, it has some $r_1 \times c_1$, and look at B , it has some $r_2 \times c_2$. So, what is the length of a row in a matrix? The length of a row is a number of columns. The number of rows is how many rows there are, but I want to know how far can I go in a row. So, that is the length of a row in A is c_1 . What is the height of a column in B ? It is the number of rows, I go from the top row to the bottom row. So, that is r_2 . So, what I want is that $r_2 = c_1$.

So, what we are saying is that if A is $m \times n$, then this n must be the number of rows in B and B can have any number of columns. So, we can take 2 matrices, which have different shapes, provided the number of columns in A = the number of rows in B . Only then I can do this the summation correctly.

(Refer Slide Time: 13:26)

Multiplying matrices

- Matrix product: $C = A \times B$
- For $r = c = n$,

$$C[i,j] = A[i,0] \cdot B[0,j] + A[i,1] \cdot B[1,j] + \dots + A[i,n-1] \cdot B[n-1,j]$$
- More concisely, $C[i,j] = \sum_{k=0}^{n-1} A[i,k] \cdot B[k,j]$
- Don't require both A and B to be $n \times n$
 - Each row entry of A must have a matching column entry in B
 - If A is $m \times n$ and B is $n \times p$, $A \times B$ is $m \times p$

Handwritten Examples:

Matrix A (3x4):

0	1	2	3
0	0	6	8
1	4	0	9
2	6	3	0

Matrix B (4x3):

0	1	2
0	1	2
1	5	0
2	6	0
3	2	6

Calculation for $C[1,1]$:

0	1	2
0	18	36
1	66	46
2	17	12

Handwritten calculation: $30 + 48 + 6 = 84$

Video Inset: A man in a blue shirt is speaking.

Footer: Madhavan Mukund, Matrix Multiplication, Mathematics for Data Science

So, let us look at an example. So, what happens here is supposing now I have something which is 3×4 , I have 3 rows and 4 columns. And here I have 4 rows and 3 columns. So, now when I take an entry here, and I multiply it by an entry here, I get 1 entry here. So, I get $6 \times 530 + 6 \times 848 + 3 \times 2,6$, which is 84. So, I have an entry in C for every row and column, which is there as a row of A and a column of B .

So, finally, I end up with a matrix which has as many rows as A and as many columns as B . So, if I start with m by n , and multiply it by an n by p matrix and up with an m by p matrix, in our context, remember that we are dealing with adjacency matrix. So, now very limited context in which we are using it for graphs. This does not matter because we are always going to do square matrices, but matrix multiplication in general does not require the matrices to be square. It only requires this correspondence between the columns of A and the rows of B .

(Refer Slide Time: 14:33)

Transitive closure

- A is an adjacency matrix
 - $A[i, j] = 1$ if and only if there is a direct edge (path of length 1) from i to j
- A^2 represents paths of length 2
 - $A^2[i, j] = 1$ if, for some k , $A[i, k] = 1$ and $A[k, j] = 1$
- Algebra of boolean values

	0	1	2	3	4	5
0	0	0	1	0	0	1
1	0	0	1	0	0	0
2	1	0	0	0	0	0
3	0	0	0	0	1	0
4	1	0	0	1	0	0
5	0	0	0	1	0	0

Handwritten notes: "True" with an arrow pointing to the cell (0,5) and "False" with an arrow pointing to the cell (1,5).

Madhavan Mukund Matrix Multiplication Mathematics for Data Science

So, now let us get back to the problem which motivated all this, the problem of transitive closure. So, remember that transitive closure was trying to capture the fact that 2 vertices in a graph. So, it started off with a relation, we said that we have a relation and we want to find out whether 2 objects in our set are related by a sequence of pairs in the relation. And then we said we will model it as a graph. And now every relational pair in our relation modelled as an edge in the graph.

So, finding a sequence of relational pairs, and our relation is the same as finding a sequence of edges in our graph. So, it is the same as the reachability problem. So, can we find out which all pairs are reachable from each other? So, we started with an adjacency matrix. So, this is just some arbitrary adjacency matrix, we do not really need to worry about what this graph represents, because once we have the adjacency matrix, we said we can compute directly with it.

So, the property that we know is that adjacency matrix captures paths of length 1, in other words, direct relationships, which are already given to us and the R relation we started with it. So, the edge relation is the relation R that we are trying to visualize. So, we know all things which are directly related because that is given to us. Now we want to go from that to say things which are related indirectly in 1 step.

So, we look at what we call A^2 , which is all pairs, which are connected by a path of length 2, by length 2, we mean there are 2 edges, I need to traverse 2 relationships to go from i to j . And we said that if you want to go from i to j in 2 steps, that means there must be an intermediate

vertex, and we call that k . So, there must be some k , such that I go from i to k , and then I go from k to j . So, that was our A^2 .

So, now, how do we put this in the framework of matrix multiplication, that is what we are trying to do. So, we are not doing multiplication addition over our normal numbers. We are doing multiplication and addition, over the values true and false. So, we are doing something which is called Boolean algebra. So, let us just write it down and understand what we are doing. So, remember that these things indicate that there is an edge, I can write this as true or there is no edge, this is false.

So, this is saying that there is an edge from 0 to 4, there is not an edge from 1 to 5. So, the 0 is interpreted as the nonexistence of an edge. So, if this is the answer to the question is, is there an edge from 1 to 5? The answer is no, false there is no such edge, therefore, the answer is 0. And is there an edge from say 3 to 4. So, the entry a_{34} is 1, and therefore the answer is true. So, we are working with these 2 values only. So, this is a 0 1 matrix. So, 0 represents false 1 represent true. So, this is our starting point.

And now the operations that we are interested in over Boolean values are AND and OR. So, if I want to say OR, OR of 2 Boolean values is true provided at least 1 of them is true. So, only if both of them are false is the answer false. So, if I now represent this OR by +, so, this is where now this matrix multiplication idea is going to come. So, this is the algebra. So, we are now representing the values by 0 1. So, first of all, we have removed true and false from our Boolean algebra, from our Boolean set and we have replaced it by 1 for true and 0 for false.

(Refer Slide Time: 18:08)

Transitive closure



- A is an adjacency matrix
 - $A[i, j] = 1$ if and only if there is a direct edge (path of length 1) from i to j
- A^2 represents paths of length 2
 - $A^2[i, j] = 1$ if, for some k , $A[i, k] = 1$ and $A[k, j] = 1$
- Algebra of boolean values

	0	1	2	3	4	5
0	0	1	0	0	1	0
1	0	0	1	0	0	0
2	1	0	0	0	0	0
3	0	0	0	0	1	0
4	1	0	0	1	0	0
5	0	0	0	1	0	0

- True is 1, False is 0

- Logical or is represented by +:

$$0+0=0, 0+1=1+0=1+1=1$$

$$\begin{matrix} 1 & + & 1 \\ \text{True} & \text{or} & \text{True} \end{matrix} = \text{True}$$



Madhavan Mukund

Matrix Multiplication

Mathematics for Data Science

Now we are taking operations and replacing them by what we think of as arithmetic operations. So, we are taking the logical OR which says false or false is false and false or anything any other pair gives true by saying that, if + is or and 0 is false, then $0+0$ is clearly 0. $0+1$ is 1 and $1+1$, $1+0$ is, is 1. This might even follow from the fact that there are numbers, obviously, even as integers, $1+0$ is 1 and $0+1$ is 1.

What is perhaps surprising from a numeric point of view is that $1+1$ is also 1. So, clearly, we are thinking of these as integers, $1+1$ would be 2, but 2 is not really a value that we are dealing with, we are dealing with only true and false. So, what this is saying is that true or true, =true. So, we are taking this and making it 1 taking this and making it 1 taking this and making it 1 and saying this is 1. So, this is a kind of, strange kind of arithmetic we are doing but it is justified because the underlying interpretation is in terms of logical values and logical operations.

The other interesting operation for us is AND. So, AND is kind of symmetric to OR in the sense that in AND we need both to be true. So, the only interesting case is 1×1 . So, 1×1 is 1, that means true and true is true. And if any of them is 0, then the answer is false. So, false in anything is false. So, both are false, or 1 is false, it does not matter. So, now this is our setup. So, we are working with the 0 1 matrix. And now we have these operations multiplication which represents AND, and OR, which is represented by a +.

(Refer Slide Time: 19:34)

Transitive closure

- A is an adjacency matrix
 - $A[i, j] = 1$ if and only if there is a direct edge (path of length 1) from i to j
- A^2 represents paths of length 2
 - $A^2[i, j] = 1$ if, for some k , $A[i, k] = 1$ and $A[k, j] = 1$
- Algebra of boolean values
 - True is 1, False is 0
 - Logical or is represented by +:
 $0 + 0 = 0, 0 + 1 = 1 + 0 = 1 + 1 = 1$
 - Logical and is represented by \times :
 $1 \times 1 = 1, 0 \times 0 = 0 \times 1 = 1 \times 0 = 0$

	0	1	2	3	4	5
0	0	0	1	0	0	1
1	0	0	0	1	0	0
2	1	0	0	0	0	0
3	0	0	0	0	0	1
4	1	0	0	0	1	0
5	0	0	0	0	1	0

- $A^2[i, j] = (A[i, 0] \text{ and } A[0, j]) \text{ or } (A[i, 1] \text{ and } A[1, j]) \text{ or } \dots (A[i, n-1] \text{ and } A[n-1, j])$
- $A^2[i, j] = A[i, 0] \times A[0, j] + A[i, 1] \times A[1, j] + \dots + A[i, n-1] \times A[n-1, j]$

Madhavan Mukund Matrix Multiplication Mathematics for Data Science I

So, now let us look at what this says? It says for some k . So, A^2 ij is 1. If for some k , I can find $A_{ik} = 1$ and $A_{kj} = 1$. So, for some case, so, what are the possible values of k ? k could be 0, k could be 1. So, k has to be one of the vertices. So, k could be 0, k could be 1, k could be 2, anything up to $n-1$.

So, this expression, this sentence here translates to saying either k is 0, in which case I have A_{i0} is true and A_{0j} is true, or I have $k=1$ in case A_{i1} is true and A_{1j} is true and so, on. Or finally, the last possibility is that it is $A_{i, n-1}$ and $A_{n-1, j}$. So, if any of these pairs A_{ik} , A_{kj} is true simultaneously true, then there is an edge from i to j . If more than 1 is true, it is still true.

If I have multiple ways of going from i to j in 2 steps, it is still okay, I need at least 1. So, this is just the expression for this left-hand side definition, written out using the logical operations and an order and interpreting the entries in the original adjacency matrix A as true and false. But notice that this is the expression if we write it using this algebra here, we replace every AND by \times and we replace every OR by $+$. So, this is an algebraic way of writing out this ANDs and ORs.

(Refer Slide Time: 21:04)

Transitive closure

- $A^2[i, j] = (A[i, 0] \text{ and } A[0, j]) \text{ or } (A[i, 1] \text{ and } A[1, j]) \text{ or } \dots (A[i, n-1] \text{ and } A[n-1, j])$
- $A^2[i, j] = A[i, 0] \times A[0, j] + A[i, 1] \times A[1, j] + \dots + A[i, n-1] \times A[n-1, j]$
- So $A^2[i, j] = \sum_{k=0}^{n-1} A[i, k] \times A[k, j]$
- In other words, $A^2 = A \times A$

- Likewise, $A^3[i, j] = 1$ if, for some k , $A^2[i, k] = 1$ and $A[k, j] = 1$
- Paths of length 3
- $A^3 = A^2 \times A = A \times A^2$
- In general, $A^{\ell+1} = A^\ell \times A = A \times A^\ell$
- Paths of length $\ell+1$
- Finally, $A^+ = A + A^2 + \dots + A^{n-1}$
- $A^+[i, j] = 1$ if i, j connected by path of length 1 or path of length 2 or ... or path of length $n-1$

Madhavan Mukund
Matrix Multiplication
Mathematics for Data Science

So, once we have written this out using ANDs and ORs, it is very clear that we have written out a matrix multiplication entry in terms of this new interpretation of ANDs and ORs. So, A^2_{ij} is a summation of all k from 0 to $n-1$, $A_{ik} \times A_{kj}$. This is exactly the equation we wrote for an arbitrary matrix product, we said C_{IJ} is a summation from $K=0$ to $n-1$ of $A_{ik} \times B_{kj}$.

So, that is the reason why A square is just in matrix terms, $A \times A$ provided we think of A as having Boolean values 0 1. And we think of OR as $+$, and, AND as multiplication. So, using that interpretation, and those rules for $+$ and \times , we get precisely $A \times A$. Now proceed, next step in transitive closure, we look at paths of length 3, and we said a path of length 3, for instance, could break up as a path of length 2, followed by a path of length 1. So, I need to go from i to k in 2 steps, and then from k to j in 1 step.

So, then, in the same logic, we can compute that this is the all over all k of this. So, it is a summation over all k of $A^2_{ik} \times A_{kj}$, because A^2 captures the length 2 paths, A captures the length 1 paths. Now of course, we could also decompose this length 3 paths as a length 1 path from i to k and a length 2 parts from k to j is the same thing, I just have to reduce that path of length 3 to 2 things.

I already know either $2+1$, or $1+2$. So, it could be $A^2 \times A$, it could also be $A \times A^2$, the intermediate k could be first finding an entry in A and then finding an entry from k to j in a square, it does not matter. So, A^3 is either $A^2 \times A$ or a 10 A^2 . Notice that here, when I did A square, I did not have an option, because both are A . So, it does not, there is only 1 way to decompose, a 2-length path is $1+1$, because that is the only information I have. But for a 3-length path, I can do $1+2$, or I can do $2+1$.

Now, in general, we said if I want to, if I already knew paths of length l , then I can extend to $l+1$ by again multiplying by A . So, I can take all paths, which are Connect pairs, which are connected by a path of length l , extended by 1 more thing, and then I get all paths of length $l+1$. So, each successive length path can be got by matrix multiplication. Again, we could also invert it, we can first take 1 step and then take l steps. And of course, you could do other things also, supposing l is 7, you could first take 3 steps, and then take 4 steps. But it is convenient to write it in a uniform way. So, that we know how to do it systematically from 1, 2, 3, onwards.

So, this is how we get paths of different lengths by consistently multiplying 1 more time by A . So, we take the matrix, we already have for L steps, multiply 1 more time by A , So, each time we multiply by A , we take the matrix we have computed in the previous step and multiply 1 more time base. So, it is $A \times A \times A \times A$. So, implicitly, this thing contains a lot of $A \times A \times A$ in it $L \times$. So, there are $l \times$ that, and then 1 more time.

So, this now gives us paths of specifically of length 1, length 2, and 3 and all that but what we said is we wanted to know whether there was a path of any length. And then we observed that of course, because it is a graph. If there is a path at all, there must be a part of at most length $n-1$ because if it is longer than that, then there will be a loop which you can remove, because there are only n vertices in the whole thing.

So, if I take $n-1$ steps, starting from a vertex, and each time I go to a new vertex, there are only $n-1$ new vertices I can go to, if I take 1 more step, then the next vertex must be 1 of the n vertices I have already seen before, and that would be useless. So, therefore, it is sufficient to look for this matrix of length, path lengths up to $N-1$. So, what we want is, is there a path of length 1 or link 2 or length 3 or this but each of them is represented by a matrix and OR is $+$.

And so, now we can go back to matrix addition, we want for each ij , we want to check whether A_{ij} is 1, or A^2_{ij} is 1, or A^3_{ij} is 1, or A^4_{ij} is 1 dot dot dot upto A^{n-1}_{ij} is 1, we want to check if any of them is 1. And that is the same as doing $++++$. So, this is why we get the transitive closure, which we wrote as A to the power $+$, the transitive closure is just the sum the matrix sum of these $n-1$ matrix is the A , A , which is this is really A to the power 1 implicitly.

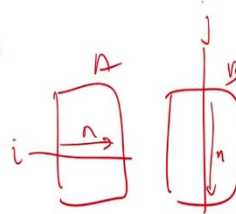
So, A which is the 1 length path, A^2 which is the 2 length path, and so, on. And, that is how we got from transitive closure to matrix multiplication, by using the entries in A as Boolean values and interpreting the 2 operations $+$ and \times as logical OR, and logical AND.

(Refer Slide Time: 25:49)

Summary



- Matrix addition is defined elementwise
 - $C[i,j] = A[i,j] + B[i,j]$
 - A, B, C have same dimensions $r \times c$
- Matrix multiplication is a more complicated operation
 - Let A be $m \times n$ and B be $n \times p$
 - $C[i,j] = \sum_{k=0}^{n-1} A[i,k] \times B[k,j]$
 - C has dimensions $m \times p$



So, to summarize, what we have seen is that we can take matrices and operate them as a whole. And depending on what operation we are doing, the operation is defined differently. So, matrix addition is very simple. It is just defined element wise. And it tells us that the ij 'th entry of the final sum is the sum of the 2 ij 'th entries in the initial matrices. And for this, we need that A and B are compatible in the sense that they have the same number of rows and the same number of columns. So, then we get $C=A+B$.

On the other hand, if we want to do a matrix multiplication, then we have to take for each entry ij . In the final thing, we have to go through row i and column j of A and B , and then pairwise, multiply all the terms and then add them up. So, that is the summation. And for this, we need that the number of rows a number of columns in A must be = the number of rows.

So, if I walk right, some $n \times$, then I must be able to walk down some $n \times$. So, n is the number of columns in A , it must be = the number of rows and B . So, we have this constraint that A and B must agree on this component, the number of columns in A and the number of rows and B . And finally, we end up with a matrix, which has as many rows as A and as many columns as B .

(Refer Slide Time: 27:05)

Summary



- Matrix addition is defined elementwise
 - $C[i,j] = A[i,j] + B[i,j]$
 - A, B, C have same dimensions $r \times c$
- Matrix multiplication is a more complicated operation
 - Let A be $m \times n$ and B be $n \times p$
 - $C[i,j] = \sum_{k=0}^{n-1} A[i,k] \times B[k,j]$
 - C has dimensions $m \times p$
- Using Boolean algebra, describe transitive closure using matrix multiplication
 - A , adjacency matrix, paths of length 1
 - $A^{\ell+1} = A^{\ell} \times A$, paths of length ℓ
 - Transitive closure, $A^+ = A + A^2 + \dots + A^{n-1}$

Navigation icons



Madhavan Mukund

Matrix Multiplication

Mathematics for Data Science

And then we saw that, if we interpret the adjacency matrix as a Boolean matrix, the values 0 and 1 as being representing true and false, and we think of the operations AND and OR as being represented by multiplication and addition, then we can think of our transitive closure computation as initially doing a bunch of matrix products. So, we get A to the power $l+1$ as $A^l \times A$.

And finally, we add up all the matrices we have computed from A^1 the original 1 up to $A^{(n-1)}$, where n is the number of vertices in our graph.