College of Engineering & Applied Sciences

CSPB 2820

Linear Algebra With Computer Science Applications

Study Guide 6 - Matrices

Taylor Larrechea

2023

Sections

Study Guide 6

Matrices

Study Guide Instructions

- Submit your work in Gradescope as a PDF - you will identify where your "questions are."
- Identify the question number as you submit. Since we grade "blind" if the questions are NOT identified, the work WILL NOT BE GRADED and a 0 will be recorded. Always leave enough time to identify the questions when submitting.
- One section per page (if a page or less) - We prefer to grade the main solution in a single page, extra work can be included on the following page.
- Long instructions may be removed to fit on a single page.
- **Do not start a new question in the middle of a page.**
- Solutions to book questions are provided for reference.
- You may NOT submit given solutions - this includes minor modifications - as your own.
- Solutions that do not show individual engagement with the solutions will be marked as no credit and can be considered a violation of honor code.
- If you use the given solutions you must reference or explain how you used them, in particular...

Method Selection

For full credit, EACH book exercise in the Study Guides must use one or more of the following methods and FOR EACH QUESTION. Identify the number the method by number to ensure full credit.

- **Method 1** - Provide original examples which demonstrate the ideas of the exercise in addition to your solution.
- **Method 2** - Include and discuss the specific topics needed from the chapter and how they relate to the question.
- **Method 3** - Include original Python code, of reasonable length (as screenshot or text) to show how the topic or concept was explored.
- **Method 4** - Expand the given solution in a significant way, with additional steps and comments. All steps are justified. This is a good method for a proof for which you are only given a basic outline.
- **Method 5** - Attempt the exercise without looking at the solution and then the solution is used to check work. Words are used to describe the results.
- **Method 6** - Provide an analysis of the strategies used to understand the exercise, describing in detail what was challenging, who helped you or what resources were used. The process of understanding is described.

Problem 1

Problem Statement

Pick a section of Chapter 6 to annotate.

For annotations this week, I have chosen to annotate section 6.4 of VMLS. The annotations for this problem can be seen on the following page.

This agrees with our definition for vectors when A is a vector, *i.e.*, $n = 1$. The norm of an $m \times n$ matrix is the norm of an mn-vector formed from the entries of the matrix (in any order). Like the vector norm, the matrix norm is a quantitative measure of the magnitude of a matrix. In some applications it is more natural to use the RMS values of the matrix entries, $\|A\|/\sqrt{mn}$, as a measure of matrix size. The RMS value of the matrix entries tells us the typical size of the entries, independent of the matrix dimensions.

The matrix norm (6.3) satisfies the properties of any norm, given on page 46. For any $m \times n$ matrix A, we have $\|A\| \geq 0$ (*i.e.*, the norm is nonnegative), and $\|A\| = 0$ only if $A = 0$ (definiteness). The matrix norm is nonnegative homogeneous: For any scalar γ and $m \times n$ matrix A, we have $\|\gamma A\| = |\gamma|\|A\|$. Finally, for any two $m \times n$ matrices A and B, we have the triangle inequality,

$$\|A + B\| \leq \|A\| + \|B\|.$$

(The plus symbol on the left-hand side is matrix addition, and the plus symbol on the right-hand side is addition of numbers.)

The matrix norm allows us to define the distance between two matrices as $\|A - B\|$. As with vectors, we can say that one matrix is close to, or near, another one if their distance is small. (What qualifies as small depends on the application.)

In this book we will only use the matrix norm (6.3). Several other norms of a matrix are commonly used, but are beyond the scope of this book. In contexts where other norms of a matrix are used, the norm (6.3) is called the *Frobenius norm*, after the mathematician Ferdinand Georg Frobenius, and is usually denoted with a subscript, as $\|A\|_F$.

One simple property of the matrix norm is $\|A\| = \|A^T\|$, *i.e.*, the norm of a matrix is the same as the norm of its transpose. Another one is

$$\|A\|^2 = \|a_1\|^2 + \cdots + \|a_n\|^2,$$

where $a_1, \ldots, a_n$ are the columns of A. In other words: The squared norm of a matrix is the sum of the squared norms of its columns.

6.4 Matrix-vector multiplication

If A is an $m \times n$ matrix and x is an n-vector, then the *matrix-vector product* $y = Ax$ is the m-vector y with elements

$$y_i = \sum_{k=1}^{n} A_{ik} x_k = A_{i1} x_1 + \cdots + A_{in} x_n, \quad i = 1, \ldots, m. \tag{6.4}$$

As a simple example, we have

$$\begin{bmatrix} 0 & 2 & -1 \\ -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} (0)(2) + (2)(1) + (-1)(-1) \\ (-2)(2) + (1)(1) + (1)(-1) \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}.$$

3-columns length 3 $n=3$

Matrix must have same number of columns as vector has length

Matrix $A \doteq m \times n$ → Rows, → columns $x \doteq n$ → length $n = n$

Row and column interpretations. We can express the matrix-vector product in terms of the rows or columns of the matrix. From (6.4) we see that y_i is the inner product of x with the ith row of A:

$$y_i = b_i^T x, \quad i = 1, \ldots, m,$$

Inner product of row w/ Column

where b_i^T is the row i of A. The matrix-vector product can also be interpreted in terms of the columns of A. If a_k is the kth column of A, then $y = Ax$ can be written

$$y = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n.$$

This shows that $y = Ax$ is a linear combination of the columns of A; the coefficients in the linear combination are the elements of x.

General examples. In the examples below, A is an $m \times n$ matrix and x is an n-vector.

- *Zero matrix.* When $A = 0$, we have $Ax = 0$. In other words, $0x = 0$. (The left-hand 0 is an $m \times n$ matrix, and the right-hand zero is an m-vector.)
- *Identity.* We have $Ix = x$ for any vector x. (The identity matrix here has dimension $n \times n$.) In other words, multiplying a vector by the identity matrix gives the same vector.
- *Picking out columns and rows.* An important identity is $Ae_j = a_j$, the jth column of A. Multiplying a unit vector by a matrix 'picks out' one of the columns of the matrix. $A^T e_i$, which is an n-vector, is the ith row of A, transposed. (In other words, $(A^T e_i)^T$ is the ith row of A.)
- *Summing or averaging columns or rows.* The m-vector $A\mathbf{1}$ is the sum of the columns of A; its ith entry is the sum of the entries in the ith row of A. The m-vector $A(\mathbf{1}/n)$ is the average of the columns of A; its ith entry is the average of the entries in the ith row of A. In a similar way, $A^T\mathbf{1}$ is an n-vector, whose jth entry is the sum of the entries in the jth column of A.
- *Difference matrix.* The $(n-1) \times n$ matrix

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \tag{6.5}$$

(where entries not shown are zero, and entries with diagonal dots are 1 or -1, continuing the pattern) is called the *difference matrix*. The vector Dx is the $(n-1)$-vector of differences of consecutive entries of x:

$$Dx = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{bmatrix}.$$

Identity matrix has to be a Square matrix (n×n)

Zero matrix can be any dimension (m×n)

Difference matrix is a matrix of differences between consecutive entries

- *Running sum matrix.* The $n \times n$ matrix

Similar to difference matrix but is the sum instead

$$S = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \qquad (6.6)$$

is called the *running sum matrix*. The ith entry of the n-vector Sx is the sum of the first i entries of x:

Constantly summing values

$$Sx = \begin{bmatrix} x_1 \\ x_1 + x_2 \\ x_1 + x_2 + x_3 \\ \vdots \\ x_1 + \cdots + x_n \end{bmatrix}.$$

Application examples.

See the weight of values

- *Feature matrix and weight vector.* Suppose X is a feature matrix, where its N columns $x_1, \ldots, x_N$ are feature n-vectors for N objects or examples. Let the n-vector w be a *weight vector*, and let $s_i = x_i^T w$ be the score associated with object i using the weight vector w. Then we can write $s = X^T w$, where s is the N-vector of scores of the objects.

Similar to feature matrix

- *Portfolio return time series.* Suppose that R is a $T \times n$ asset return matrix, that gives the returns of n assets over T periods. A common trading strategy maintains constant investment weights given by the n-vector w over the T periods. For example, $w_4 = 0.15$ means that 15% of the total portfolio value is held in asset 4. (Short positions are denoted by negative entries in w.) Then Rw, which is a T-vector, is the time series of the portfolio returns over the periods $1, \ldots, T$.

  As an example, consider a portfolio of the 4 assets in table 6.1, with weights $w = (0.4, 0.3, -0.2, 0.5)$. The product $Rw = (0.00213, -0.00201, 0.00241)$ gives the portfolio returns over the three periods in the example.

- *Polynomial evaluation at multiple points.* Suppose the entries of the n-vector c are the coefficients of a polynomial p of degree $n-1$ or less:

$$p(t) = c_1 + c_2 t + \cdots + c_{n-1} t^{n-2} + c_n t^{n-1}.$$

  Let $t_1, \ldots, t_m$ be m numbers, and define the m-vector y as $y_i = p(t_i)$. Then we have $y = Ac$, where A is the $m \times n$ matrix

$$A = \begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-2} & t_1^{n-1} \\ 1 & t_2 & \cdots & t_2^{n-2} & t_2^{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & t_m & \cdots & t_m^{n-2} & t_m^{n-1} \end{bmatrix}. \qquad (6.7)$$

Each row of matrix is multiplied with coefficient vector to get polynomial

So multiplying a vector c by the matrix A is the same as evaluating a polynomial with coefficients c at m points. The matrix A in (6.7) comes up often, and is called a *Vandermonde matrix* (of degree $n-1$, at the points $t_1, \ldots, t_m$), named for the mathematician Alexandre-Théophile Vandermonde.

All generate Vectors

- *Total price from multiple suppliers.* Suppose the $m \times n$ matrix P gives the prices of n goods from m suppliers (or in m different locations). If q is an n-vector of quantities of the n goods (sometimes called a *basket* of goods), then $c = Pq$ is an N-vector that gives the total cost of the goods, from each of the N suppliers.
- *Document scoring.* Suppose A in an $N \times n$ document-term matrix, which gives the word counts of a corpus of N documents using a dictionary of n words, so the rows of A are the word count vectors for the documents. Suppose that w in an n-vector that gives a set of weights for the words in the dictionary. Then $s = Aw$ is an N-vector that gives the scores of the documents, using the weights and the word counts. A search engine, for example, might choose w (based on the search query) so that the scores are predictions of relevance of the documents (to the search).
- *Audio mixing.* Suppose the k columns of A are vectors representing audio signals or tracks of length T, and w is a k-vector. Then $b = Aw$ is a T-vector representing the mix of the audio signals, with track weights given by the vector w.

Inner product. When a and b are n-vectors, $a^T b$ is exactly the inner product of a and b, obtained from the rules for transposing matrices and forming a matrix-vector product. We start with the (column) n-vector a, consider it as an $n \times 1$ matrix, and transpose it to obtain the n-row-vector a^T. Now we multiply this $1 \times n$ matrix by the n-vector b, to obtain the 1-vector $a^T b$, which we also consider a scalar. So the notation $a^T b$ for the inner product is just a special case of matrix-vector multiplication.

Similar to Vectors $\rightarrow$

Linear dependence of columns. We can express the concepts of linear dependence and independence in a compact form using matrix-vector multiplication. The columns of a matrix A are linearly dependent if $Ax = 0$ for some $x \neq 0$. The columns of a matrix A are linearly independent if $Ax = 0$ implies $x = 0$.

Expansion in a basis. If the columns of A are a basis, which means A is square with linearly independent columns $a_1, \ldots, a_n$, then for any n-vector b there is a unique n-vector x that satisfies $Ax = b$. In this case the vector x gives the coefficients in the expansion of b in the basis $a_1, \ldots, a_n$.

Properties of matrix-vector multiplication. Matrix-vector multiplication satisfies several properties that are readily verified. First, it distributes across the vector argument: For any $m \times n$ matrix A and any n-vectors u and v, we have

$$A(u + v) = Au + Av.$$

$\uparrow$

Distributes across vector addition

Matrix-vector multiplication, like ordinary multiplication of numbers, has higher precedence than addition, which means that when there are no parentheses to force the order of evaluation, multiplications are to be carried out before additions. This means that the right-hand side above is to be interpreted as $(Au) + (Av)$. The equation above looks innocent and natural, but must be read carefully. On the left-hand side, we first add the vectors u and v, which is the addition of n-vectors. We then multiply the resulting n-vector by the matrix A. On the right-hand side, we first multiply each of n-vectors by the matrix A (this is two matrix-vector multiplies); and then add the two resulting m-vectors together. The left- and right-hand sides of the equation above involve very different steps and operations, but the final result of each is the same m-vector.

Matrix-vector multiplication also distributes across the matrix argument: For any $m \times n$ matrices A and B, and any n-vector u, we have

Also distributes across matrix addition

$$(A + B)u = Au + Bu.$$

On the left-hand side the plus symbol is matrix addition; on the right-hand side it is vector addition.

Another basic property is, for any $m \times n$ matrix A, any n-vector u, and any scalar α, we have

Commutative with multiplication $\longrightarrow$

$$(\alpha A)u = \alpha(Au)$$

(and so we can write this as αAu). On the left-hand side, we have scalar-matrix multiplication, followed by matrix-vector multiplication; on the right-hand side, we start with matrix-vector multiplication, and then perform scalar-vector multiplication. (Note that we also have $\alpha Au = A(\alpha u)$.)

Input-output interpretation. We can interpret the relation $y = Ax$, with A an $m \times n$ matrix, as a mapping from the n-vector x to the m-vector y. In this context we might think of x as an input, and y as the corresponding output. From equation (6.4), we can interpret A_{ij} as the factor by which y_i depends on x_j. Some examples of conclusions we can draw are given below.

- If A_{23} is positive and large, then y_2 depends strongly on x_3, and increases as x_3 increases.
- If A_{32} is much larger than the other entries in the third row of A, then y_3 depends much more on x_2 than the other inputs.
- If A is square and lower triangular, then y_i only depends on $x_1, \ldots, x_i$.

6.5 Complexity

Computer representation of matrices. An $m \times n$ matrix is usually represented on a computer as an $m \times n$ array of floating point numbers, which requires $8mn$ bytes. In some software systems symmetric matrices are represented in a more efficient way, by only storing the upper triangular elements in the matrix, in some

Problem 1 Summary

Procedure

- Annotate a chapter from VMLS and highlight important concepts from the section.

Key Concepts

- This section highlights the aspects involving matrix vector multiplication.
- In order to perform matrix / vector multiplication the number of columns in the matrix must match the number of rows in the vector.
  - Namely, n must be equal in $(m \times n) \times (n \times m)$. First value is the number of rows and the second is columns.

Variations

- Since this problem involves annotating a section, the only variation of this problem could involve annotating another section.
  - We would then comment on the sections just like we did in the original problem.

Problem 2

Problem Statement

Solve and explain the solution to 6.3 here in your own words. (Since you are given a solution, you will be graded on your ability to explain).

Original Question

Block matrix. Assuming the matrix

$$K = \begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}$$

makes sense, which of the following statements must be true? ('Must be true' means that it follows with no additional assumptions.)

(a) K is square.

(b) A is square or wide.

(c) K is symmetric, *i.e.*, $K^T = K$.

(d) The identity and zero submatrices in K have the same dimensions.

(e) The zero submatrix is square.

Solution - Part (a)

For this problem I will be using **Method 4**.

VMLS Solution:

(a) Must be true. K is $(m+n)\times(m+n)$.

Explanation:

(a) We know from the properties of matrices we can write the dimension of a matrix as $m \times n$. We know that identity matrices are square $(m \times m)$. Zero matrices are matrices where every entry is a 0 so it can be of an arbitrary size, such as $(m \times n)$. As for the matrix A it can be of an arbitrary size such as $(m \times n)$. Expanding this for matrix K we can then see that the matrix is consisted of submatrices with the following dimensions

$$K = \begin{bmatrix} n \times n & n \times m \\ m \times n & m \times m \end{bmatrix}. \tag{1}$$

For a matrix to be square, it must have the same number as rows as columns. From the refined definition of K in (1), we see that the dimension of K is then

$$(n+m) \times (n+m). \tag{2}$$

Since, from (2) we see that K indeed has the same number of rows as columns, we know that K is indeed **square**. This statement **must be true**.

Solution - Part (b)

For this problem I will be using **Method 4**.

VMLS Solution:

(b) Might not be true. A could be tall.

Explanation:

(b) Since A is of an arbitrary size, it is completely possible that A could have a larger value for m than it does for n. Because of this, any of the possible matrix descriptions are valid:

- **Square:** $m \times n$ where $n = m$
- **Tall:** $m \times n$ where $m > n$
- **Wide:** $m \times n$ where $m < n$

Because of this reality, the statement **might not be true**.

Solution - Part (c)

For this problem I will be using **Method 4**.

VMLS Solution:

(c) Must be true. Using the block transpose formula on K, you get K back again.

Explanation:

(c) For a matrix to be considered symmetric, it must follow that $A = A^T$ where $A_{ij} = A_{ji}$ for all values of i and j. If we transpose the block matrix K we see

$$K^T = \begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}^T = \begin{bmatrix} I^T & A^T \\ A & 0^T \end{bmatrix} = \begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} = K. \quad (3)$$

In order for K to be indeed square, we require that $0^T = 0$. Because of this we can see that the transposes are indeed equal and thus K is symmetric. Therefore the statement **must be true**.

Solution - Part (d)

For this problem I will be using **Method 4**.

VMLS Solution:

(d) Might not be true. They are $n \times n$ and $m \times m$, respectively, and we don't need to have $m = n$.

Explanation:

(d) We know from part (a) that in this example, both the identity and zero matrices are square. This of course means that $I = n \times n$ and $0 = m \times m$. This follows because in order for K to be square, 0 must have the same number of rows A, so this means that $0 = m \times m$. As for the identity, I must have the same number of rows as A^T, so $I = n \times n$. This then implies that it is not a necessary condition that $n = m$ and thus I and 0 don't necessarily have to be of the same dimension. Therefore the statement **might not be true**.

Solution - Part (e)

For this problem I will be using **Method 4**.

VMLS Solution:

(e) Must be true. The zero submatrix has dimensions $m \times m$.

Explanation:

(e) We have pretty much shown this in the previous parts. In order for K to be square, it must follow that the 0 submatrix is square in order for it to be symmetric. Because of this we know that $0 = m \times m$ and is indeed square. Therefore the statement **must be true**.

Problem 2 Summary

Procedure

- For part (a), add up the number of rows and columns to determine the size of the matrix.
- For part (b), reason that because the matrix is of an arbitrary size it is possible that it could be tall.
- For part (c), because of the nature of K being square, we can reason that it is also symmetric.
- For part (d), reason that because the identity matrix is square, it is not necessarily true that the submatrix A is going to be of the same dimension.
- For part (e), reason that because of the dimensions of the identity and A matrix and the shape of K that the zero matrix must be square.

Key Concepts

- **Squareness of Matrix K:** The matrix K is determined to be square based on the arrangement and dimensions of its submatrices.
- **Matrix A Can Be Square or Wide:** The matrix A can have arbitrary dimensions, meaning it could be square, tall, or wide.
- **Symmetry of Matrix K:** The symmetry of the matrix K is deduced by applying the block transpose formula, leading to the conclusion that $K^T = K$.
- **Dimensions of Identity and Zero Matrices:** The identity and zero submatrices in K do not necessarily have the same dimensions.
- **Zero Submatrix Being Square:** The zero submatrix in K is shown to be square to maintain the overall square structure of K.
- **Block Matrix Analysis:**
  - The problem involves analyzing a block matrix K and determining the truth of several statements about its properties.
  - The matrix K is assumed to be in a block form involving an identity matrix I, a zero matrix, and a matrix A along with its transpose A^T.

Variations

- We could be given a different matrix with submatrices to examine.
  - We would then use the same logic of determining what the sizes and shapes of the matrices are.

Problem 3

Problem Statement

Solve and Explain the solution to 6.9 here in your own words. (Since you are given a solution, you will be graded on your ability to explain).

Original Question

Multiple channel marketing campaign. Potential customers are divided into m market segments, which are groups of customers with similar demographics, e.g., college educated women aged 25–29. A company markets its products by purchasing advertising in a set of n channels, i.e., specific TV or radio shows, magazines, web sites, blogs, direct mail, and so on. The ability of each channel to deliver impressions or views by potential customers is characterized by the reach matrix, the $m \times n$ matrix R, where R_{ij} is the number of views of customers in segment i for each dollar spent on channel j. (We assume that the total number of views in each market segment is the sum of the views from each channel, and that the views from each channel scale linearly with spending.) The n-vector c will denote the company's purchases of advertising, in dollars, in the n channels. The m-vector v gives the total number of impressions in the m market segments due to the advertising in all channels. Finally, we introduce the m-vector a, where a_i gives the profit in dollars per impression in market segment i. The entries of R, c, v, and a are all nonnegative.

(a) Express the total amount of money the company spends on advertising using vector/matrix notation.

(b) Express v using vector/matrix notation, in terms of the other vectors and matrices.

(c) Express the total profit from all market segments using vector/matrix notation.

(d) How would you find the single channel most effective at reaching market segment 3, in terms of impressions per dollar spent?

(e) What does it mean if R_{35} is very small (compared to other entries of R)?

Solution - Part (a)

For this problem I will be using **Method 4**.

VMLS Solution:

(a) $\mathbf{1}^T c$. The company's purchases in advertising for the n channels is given by c, so summing up the entries of c gives the total amount of money the company spends across all channels.

Explanation:

(a) In order to correctly represent the amount of money that the company spends on advertising for these channels we need a scalar value. In the context of vector notation, to generate a scalar value we would use an inner product. Since we do not want to alter the cost of advertising for a specific channel, we would want to multiply that cost for the given channel with 1. In the context of vectors, this would mean an inner product of length n for the 1 vector with the cost vector c. Namely,

$$C = \mathbf{1}^T c \tag{1}$$

where C is the total cost of advertising. We are effectively summing up the cost of advertising the company spends for all channels.

Solution - Part (b)

For this problem I will be using **Method 4**.

VMLS Solution:

(b) $v = Rc$. To find the total number of impressions v_i in the ith market segment, we need to sum up the impressions from each channel, which is given by an inner product of the ith row of R and the amounts c spent on advertising per channel.

Explanation:

(b) Because we want a vector representation for v, this most likely means that we need to multiply a matrix with a vector. In this context, we know that R is a matrix and c is a vector. Since R represents the number of views of customers on a segment i for each dollar spent on channel j, we would want to multiply the row of R with the vector c that represents the cost the company spent on advertising for a given channel. Piecing this together we have the expression

$$v = Rc. \tag{2}$$

Because of the nature of matrix/ vector multiplication, each entry in v will denote the number of impressions for each segment across all channels.

Solution - Part (c)

For this problem I will be using **Method 4**.

VMLS Solution:

(c) $a^T v = a^T Rc$. The total profit is the sum of the profits from each market segment, which is the product of the number of impressions for that segment v_i and the profit per impression for that segment a_i. The total profit is the sum $\sum_i a_i v_i = a^T v$. Substituting $v = Rc$ gives $a^T v = a^T Rc$.

Explanation:

(c) Since we know that v represents the total number of impressions across all segments from all channels, we would want to sum the total profit from each segment to get the final profit across all segments. We would then need to multiply the profit per impression for each segment with the total number of impressions for the given segment. This is screaming for a inner product of two vectors. So, we then take the inner product of a (the profit in dollars per impressions for a given segment) with that of the v (the total number of impressions per segment) to get

$$A = a^T v \tag{3}$$

where A represents the total profit. Since we know that $v = Rc$ from (2), we can also write (3) as

$$A = a^T Rc. \tag{4}$$

(3) and (4) accurately depict the total profit from all market segments.

Solution - Part (d)

For this problem I will be using **Method 4**.

VMLS Solution:

(d) $\text{argmax}_j(R_{3j})$, i.e., the column index of the largest entry in the third row of R. The number of impressions made on the third market segment, for each dollar spent, is given by the third row of R. The index of the greatest element in this row is then the channel that gives the highest number of impressions per dollar spent.

Explanation:

(d) Since we are concerned with segment 3 in the problem statement, we would want to find the column in row 3 of R that is the greatest value. So, we would write some algorithm that could find the largest element in row 3 of R and this in turn would be the single channel most effective at reaching market segment 3.

Solution - Part (e)

For this problem I will be using **Method 4**.

VMLS Solution:

(e) The fifth channel makes relatively few impressions on the third market segment per dollar spent, compared to the other channels.

Explanation:

(e) This means that for the third segment, channel 5 is making significantly smaller impressions per dollar spent compared to the other channels. Keep in mind that the entries in R are the number of views on segment i for every dollar spent on channel j. In English, this channel for the given segment is not getting a good return on investment.

Problem Summary

Procedure

- For part (a), use an inner product representation with a **1** vector and the cost vector c.
- For part (b), perform matrix / vector multiplication with the matrix R and c to find the vector v.
- For part (c), perform an inner product of dollars per impression vector a with the vector v.
- For part (d), write an algorithm for finding the max element the third row of R.
- For part (e), comment on what the element means if the element is very small.

Key Concepts

- **Reach Matrix R:**
  - An $m \times n$ matrix where R_{ij} represents the number of views in market segment i per dollar spent on channel j.
  - This matrix is central to understanding the effectiveness of different advertising channels.
- **Advertising Cost Vector c:**
  - An n-vector representing the company's spending on each advertising channel.
- **Impressions Vector v:**
  - An m-vector indicating the total number of impressions in each market segment.
- **Profit Vector a:**
  - An m-vector where each entry a_i gives the profit in dollars per impression for each market segment.
- **Calculating Total Advertising Spend:**
  - Using the inner product of a vector of ones with the cost vector c to find the total spending.
- **Impressions and Profit Calculation:**
  - The total number of impressions v is calculated using the matrix-vector product Rc.
  - Total profit from all market segments is calculated using the dot product $a^T v$ or equivalently $a^T Rc$.
- **Evaluating Channel Effectiveness:**
  - Determining the most effective channel for a specific market segment in terms of impressions per dollar spent.
- **Interpretation of Reach Matrix Entries:**
  - Understanding the implications of small entries in the reach matrix, indicating lower impressions per dollar spent in specific segments.

Variations

- We could be given a different matrix indicating different qualities.
  - We would then use the same procedure as depicted in this problem.

Problem 4

Problem Statement

Solve and Explain the solution to 6.13 here in your own words. (Since you are given a solution, you will be graded on your ability to explain).

Original Question

Polynomial differentiation. Suppose p is a polynomial of degree $n-1$ or less, given by $p(t) = c_1 + c_2 t + \cdots + c_n t^{n-1}$. Its derivative (with respect to t) $p'(t)$ is a polynomial of degree $n-2$ or less, given by $p'(t) = d_1 + d_2 t + \cdots + d_{n-1} t^{n-2}$. Find a matrix D for which $d = Dc$. (Give the entries of D, and be sure to specify its dimensions.)

Solution

For this problem I will be using **Method 4**.

VMLS Solution:

Solution. The derivative is the polynomial

$$p'(t) = c_2 + 2c_3 t + \cdots + c_n(n-1)t^{n-2},$$

so

$$d_1 = c_2, \quad d_2 = 2c_3, \quad d_3 = 3c_4, \quad \ldots \quad d_{n-1} = (n-1)c_n.$$

Since c is an n-vector and d is an $(n-1)$-vector, the matrix D must be $(n-1)\times n$. It is given by

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 2 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & n-1 \end{bmatrix} = \begin{bmatrix} 0_{(n-1)\times 1} & \mathbf{diag}(1, 2, \ldots, n-1) \end{bmatrix}.$$

Explanation:

We first take the derivative of the polynomial $p(t)$

$$p(t) = c_1 + c_2 t + \cdots + c_n t^{n-1}$$

with respect to t to get

$$p'(t) = 0c_1 + c_2 + 2c_3 t + \cdots + c_n(n-1)t^{n-2} = c_2 + 2c_3 t + \cdots + c_n(n-1)t^{n-2}. \tag{1}$$

We can see that there is a constant that is being multiplied in front of each c term in (1). From this we can see the relationship

$$d_{n-1} = (n-1)c_n \tag{2}$$

for all values of d where n indexes from 1. We wish to represent d as a vector. To do this, we need to have matrix multiplication involving a matrix and a vector. Namely

$$d = Dc \tag{3}$$

where

$$d = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \quad D = \begin{bmatrix} D_{11} & \ldots & D_{1j} \\ \vdots & \ddots & \vdots \\ D_{i1} & \ldots & D_{ij} \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}. \tag{4}$$

To follow the results from (2), we constitute that only one entry of each row is non zero for D. For a matrix with n columns, we will have $n-1$ rows. This means the dimension of D is then $(n-1)\times n$. This then means for any row i, the $j^{\text{th}}+i$ column of that row will be the only non zero element. It then follows that

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & n-1 \end{bmatrix}. \tag{5}$$

Piecing this all together we then find the relationship for (3) for each element of d to be

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & n-1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix} \tag{6}$$

where n is the number of columns in D.

Problem 4 Summary

Procedure

- Take the derivative of the polynomial.
- Construct a matrix D for all the constants that are in front of the terms in the derivative.
- Create a vector c for the coefficients that show up in the derivative.
- Show that the linear equations can be constructed with the expression $d = Dc$.

Key Concepts

- **Polynomial Differentiation:**
  - The problem deals with polynomial differentiation, specifically the differentiation of a polynomial $p(t)$ of degree n expressed as $p(t) = c_1 + c_2t + \cdots + c_nt^{n-1}$.
  - The derivative of $p(t)$, denoted as $p'(t)$, is a polynomial of degree $n-2$ or less.
- **Matrix Representation of Differentiation:**
  - The goal is to find a matrix D such that $d = Dc$, where d represents the coefficients of the derivative polynomial $p'(t)$, and c represents the coefficients of the original polynomial $p(t)$.
  - The matrix D captures the operation of differentiation on the polynomial's coefficients.
- **Derivation of Coefficients:**
  - The coefficients d_i of the derivative polynomial are related to the coefficients c_i of the original polynomial by a constant multiplier, which is the degree of the term in $p(t)$ corresponding to c_i.
  - The coefficients d_i are expressed as $d_1 = c_2, d_2 = 2c_3, \ldots, d_{n-1} = (n-1)c_n$.
- **Matrix D Structure and Dimensions:**
  - The matrix D is an $(n-1) \times n$ matrix.
  - The structure of D is characterized by its diagonal and upper-diagonal elements, where the i-th row and $(i+1)$-th column element of D is i, and all other elements are zero.
- **Matrix D as a Differentiation Operator:**
  - The matrix D effectively serves as an operator that transforms the vector of coefficients c of the polynomial $p(t)$ into the vector of coefficients d of its derivative $p'(t)$.
  - This representation provides a linear algebraic approach to polynomial differentiation.

Variations

- We could be given a different polynomial.
  - We would then go about the same process of determining the matrix.

Problem 5

Problem Statement

Solve and Explain the solution to 6.17 here in your own words. (Since you are given a solution, you will be graded on your ability to explain).

Original Question

Stacked matrix. Let A be an $m \times n$ matrix, and consider the stacked matrix S defined by

$$S = \begin{bmatrix} A \\ I \end{bmatrix}$$

(a) When does S have linearly independent columns?

(b) When does S have linearly independent rows?

Your answer can depend on m, n, or whether or not A has linearly independent columns or rows.

Solution - Part (a)

For this problem I will be using **Method 4**.

VMLS Solution:

(a) S always has linearly independent columns. To see this, suppose that $Sx = 0$. Since $Sx = (Ax, x)$, this means that $x = 0$.

Explanation:

(a) The identity matrix always has linearly independent columns. So the focus is on whether or not A has linearly independent columns. Since the columns of a matrix can be considered as vectors we are concerned with whether or not the following property holds

$$\beta_1 a_1 + \cdots + \beta_n a_n = 0 \tag{1}$$

where the only possibility of (1) being true is if all β's are zero. If we consider

$$\beta S = 0 \tag{2}$$

this means that β is being carried through to both A and the identity matrix. Since the columns of the identity matrix are always linearly independent, this means that $\beta = 0$ and thus in turn means that the columns of A are also linearly independent. Moreover

$$\beta S = \begin{bmatrix} \beta A \\ \beta I \end{bmatrix} \rightarrow \beta = 0. \tag{3}$$

S **always** has linearly independent columns.

Solution - Part (b)

For this problem I will be using **Method 4**.

VMLS Solution:

(b) S never has linearly independent rows. S has $m + n$ rows, and each row has dimension n, so by the independence-dimension inequality, the rows are dependent.

Explanation:

(b) To answer this question, we need to take into account how many rows S has. We know that the identity matrix is a square matrix and thus has the dimensions of $n \times n$. As for the matrix A, in order for S to be a valid matrix it must have the same number of columns as the identity matrix. So this means that the dimension of A is then $m \times n$ where n is the same value that is found in the identity matrix.

Now, the independence-dimension inequality states

Any collection of $n + 1$ or more n-vectors is linearly dependent.

Since each row will have the same number of elements that S has columns, this means that S has $m + n$ rows with dimension n. By the independence-dimension inequality this of course means that the rows of S must be linearly dependent. S **never** has linearly independent rows.

Problem 5 Summary

Procedure

- For part (a), reason that the identity matrix always has linearly independent columns and this constitutes that the columns of A must also be linearly independent.
- For part (b), reason with the independence-dimension inequality that the rows will always be linearly dependent.

Key Concepts

- **Stacked Matrix Analysis:**
  - The problem introduces a stacked matrix S, defined by stacking an $m \times n$ matrix A on top of an identity matrix I.
- **Linear Independence of Columns:**
  - The problem discusses the conditions under which the columns of S are linearly independent.
  - It asserts that S always has linearly independent columns, demonstrated by considering a vector x such that $Sx = 0$. This leads to the conclusion that $x = 0$, indicating linear independence.
  - The identity matrix's linear independence of columns plays a key role in this determination.
- **Linear Independence of Rows:**
  - The problem addresses the linear independence of the rows of S.
  - It concludes that S never has linearly independent rows due to the independence-dimension inequality.
  - The inequality states that any collection of $n+1$ or more n-vectors is linearly dependent. Since S has m+n rows and each row has dimension n, the rows of S must be linearly dependent.
- **Dimension and Row Analysis:**
  - The identity matrix I is square, with dimensions $n \times n$.
  - For S to be valid, A must have the same number of columns as I, leading to its dimension being $m \times n$.
  - Applying the independence-dimension inequality reveals that the rows of S are linearly dependent.

Variations

- We could be given a different set of submatrices.
  - We would then determine the linear independence of each submatrix to determine the overall linear independence.

Problem 6

Problem Statement

Solve and Explain the solution to 6.22 here in your own words. (Since you are given a solution, you will be graded on your ability to explain).

Original Question

Distribute or not? Suppose you need to compute $z = (A+B)(x+y)$, where A and B are $m \times n$ matrices and x and y are n-vectors.

(a) What is the approximate flop count if you evaluate z as expressed, i.e., by adding A and B, adding x and y, and then carrying out the matrix-vector multiplication?

(b) What is the approximate flop count if you evaluate z as $z = Ax + Ay + Bx + By$, i.e., with four matrix-vector multiplies and three vector additions?

(c) Which method requires fewer flops? Your answer can depend on m and n. *Remark.* When comparing two computation methods, we usually do not consider a factor of 2 or 3 in flop counts to be significant, but in this exercise you can.

Solution - Part (a)

For this problem I will be using **Method 4**.

VMLS Solution:

(a) Adding A and B costs mn flops, adding x and y costs n, and multiplying $(A+B)(x+y)$ costs $2mn$ flops, so the total is $3mn + n$ flops. This is approximately $3mn$ flops.

Explanation:

(a) When adding matrices, we have to add each element of one matrix with the corresponding element in the other matrix. For example, adding two 2×2 matrices will result in two FLOPS for the first row, and then two for the second row. This means a 2×2 matrix has four FLOPS. Conversely, addition of a 3×3 matrix will have three FLOPS for one row, and since it has three total rows, the total number of FLOPS is then 9. We can generalize this to say, for any $m \times n$ matrix the total number of flops for that operation is then

$$\alpha_{m\times n} + \beta_{m\times n} = mn \text{ (FLOPS)}. \tag{1}$$

Adding two vectors together is easier to understand. Since the vectors must have the same length, the total number of flops for adding two vectors of length n is then

$$\alpha_n + \beta_n = n \text{ (FLOPS)}. \tag{2}$$

When multiplying a matrix with a vector, we know that each row of the matrix is being multiplied with the vector, and all multiplications are then summed to find the element of the new vector. Since the matrix in question must have the same number of columns as the vector has length, this means that each new element will correspond to n FLOPS. Furthermore, since the matrix has m columns, this means that this operation of calculating a new element for the new vector will occur m times. Summing up the multiplications between the matrix and vector will correspond with mn FLOPS. Putting this together we then have

$$\alpha_{m\times n}\beta_n = 2mn \text{ (FLOPS)}. \tag{3}$$

Piecing together (1), (2), and (3) we now have

$$mn \text{ (FLOPS)} + n \text{ (FLOPS)} + 2mn \text{ (FLOPS)} = 3mn + n \text{ (FLOPS)}. \tag{4}$$

Solution - Part (b)

For this problem I will be using **Method 4**.

VMLS Solution:

(b) Each matrix-vector multiply costs $2mn$ flops, so the total is $8mn$ flops. The three vector adds cost $3m$ flops. The total is $8mn + 3m$ flops, but we can approximate this as $8mn$ flops.

Explanation:

(b) As previously derived in part (a), we know that a matrix/ vector multiplication will correspond to $2mn$ FLOPS. Adding two vectors together will correspond to n FLOPS. Four matrix/ vector multiplications will correspond to

$$4 \cdot 2mn \text{ (FLOPS)} = 8mn \text{ (FLOPS)}. \tag{5}$$

Summing four vectors together will then correspond to

$$3 \cdot n \text{ (FLOPS)} = 3n \text{ (FLOPS)}. \tag{6}$$

Piecing (5) and (6) together we then have

$$8mn \text{ (FLOPS)} + 3n \text{ (FLOPS)} = 8mn + 3n \text{ (FLOPS)}. \tag{7}$$

Solution - Part (c)

For this problem I will be using **Method 4**.

VMLS Solution:

(c) The first method always requires fewer flops.

Explanation:

(c) Because the first operation resulted in $3mn+n$ (FLOPS), and the second resulted in $8mn+3n$ (FLOPS), the first method requires less FLOPS.

Problem 6 Summary

Procedure

- For part (a), count the number of flops in the operation.
- For part (b), count the number of flops in the operation.
- For part (c), count the number of flops in the operation.

Key Concepts

- **Context and Problem Statement:**
  - The task is to analyze two methods for computing $z = (A + B)(x + y)$ where A and B are $m \times n$ matrices, and x and y are n-vectors.
- **Computational Complexity Analysis:**
  - The problem involves comparing the computational complexity (flop count) of two different methods for evaluating z.
- **Method 1 (Direct Evaluation):**
  - **Process:** Add matrices A and B, add vectors x and y, then perform matrix-vector multiplication.
  - **Flop Count:** Calculated as $3mn + n$ flops, approximately $3mn$ flops.
- **Method 2 (Distributed Evaluation):**
  - **Process:** Perform four matrix-vector multiplications (Ax, Ay, Bx, By) and three vector additions.
  - **Flop Count:** Calculated as $8mn + 3m$ flops, approximated to $8mn$ flops.
- **Comparison and Conclusion:**
  - The first method (direct evaluation) always requires fewer flops compared to the second method (distributed evaluation).
  - The complexity of each operation, such as matrix addition, vector addition, and matrix-vector multiplication, is carefully analyzed and quantified in flops.
- **Linear Algebra Operations in Computation:**
  - The problem illustrates the importance of understanding computational complexity in linear algebra operations, particularly in the context of computer science and algorithm optimization.

Variations

- We could be given a different set of matrices.
  - We would then go through the same process of counting the flops for each operation.