Rank of a matrix

The rank of A is the number of pivot columns in A.

Rank of a matrix

Determine the rank of the matrices

$$\begin{bmatrix} 2 & 5 & -3 & -4 & 8 \\ 4 & 7 & -4 & -3 & 9 \\ 6 & 9 & -5 & 2 & 4 \\ 0 & -9 & 6 & 5 & -6 \end{bmatrix}$$

$$A \sim \begin{bmatrix} 2 & 5 & -3 & -4 & 8 \\ 0 & -3 & 2 & 5 & -7 \\ 0 & -6 & 4 & 14 & -20 \\ 0 & -9 & 6 & 5 & -6 \end{bmatrix} \sim \cdots \sim \begin{bmatrix} 2 & 5 & -3 & -4 & 8 \\ 0 & -3 & 2 & 5 & -7 \\ 0 & 0 & 0 & 4 & -6 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Pivot columns

The matrix A has 3 pivot columns, so rank $A = 3$.

The Invertible Matrix Theorem

Let A be a square n×n matrix. Then the following statements are equivalent. That is, for a given A, the statements are either all true or all false.

- A is an invertible matrix.
- A has n pivot positions.
- There is an n×n matrix C such that CA = I.
- There is an n×n matrix D such that AD = I
- A^T is an invertible matrix.
- Rank A=n

Inverse

Singular or non-invertible matrix	**Non-singular or invertible matrix**
It has no inverse, A^{-1} does not exists	It has an inverse, A^{-1} exists
Its determinant is zero	The determinant is nonzero
The rank is less than n	The rank equals n

Application of Matrices

Computer Graphics

- One important use of matrices is in the digital representation of images.

  - A digital camera or a scanner converts an image into a matrix by dividing the image into a rectangular array of elements called pixels.

  - Each pixel is assigned a value that represents the color, brightness, or some other feature of that location.

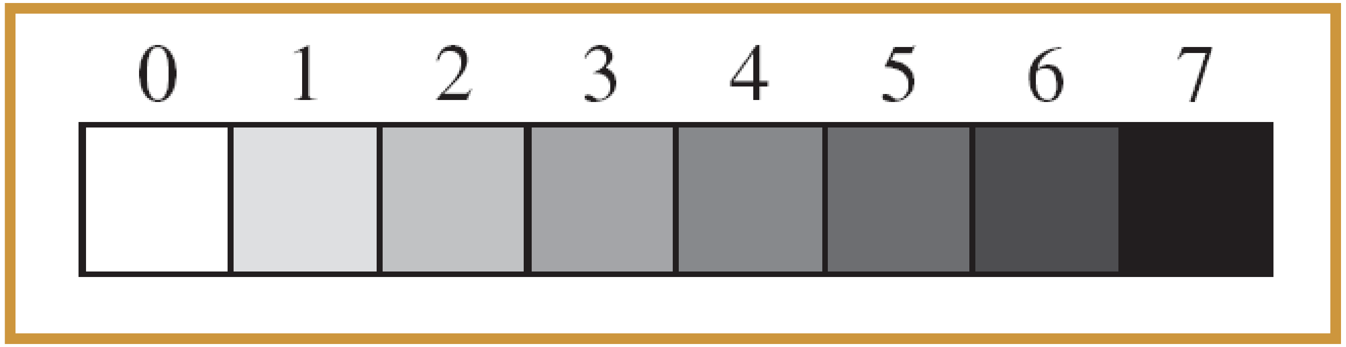
Pixels

- For example, in a 256-level gray-scale image, each pixel is assigned a value between 0 and 255.
  - 0 represents white.
  - 255 represents black.
  - The numbers in between represent increasing gradations of gray.

Gray Scale

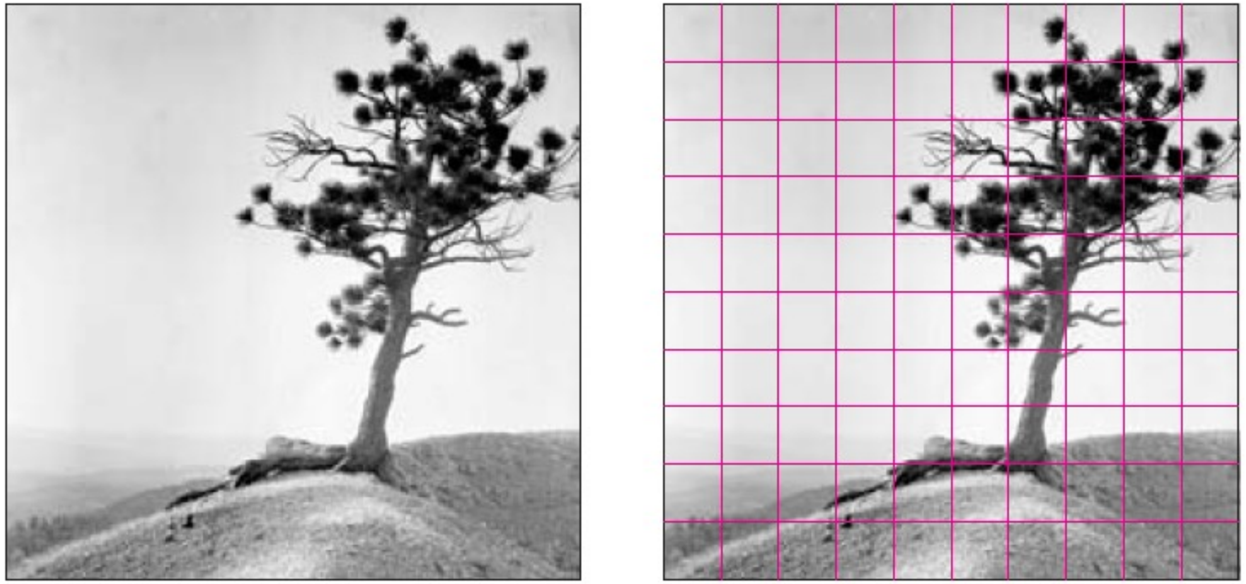
- The gradations of a much simpler 8-level gray scale are shown.

  - We use this 8-level gray scale to illustrate the process.



Digitizing Images

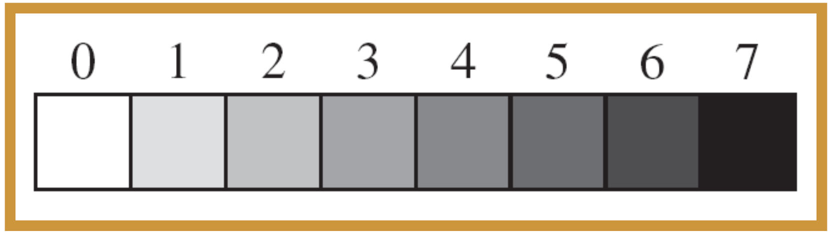
- To digitize the black and white image shown, we place a grid over the picture.

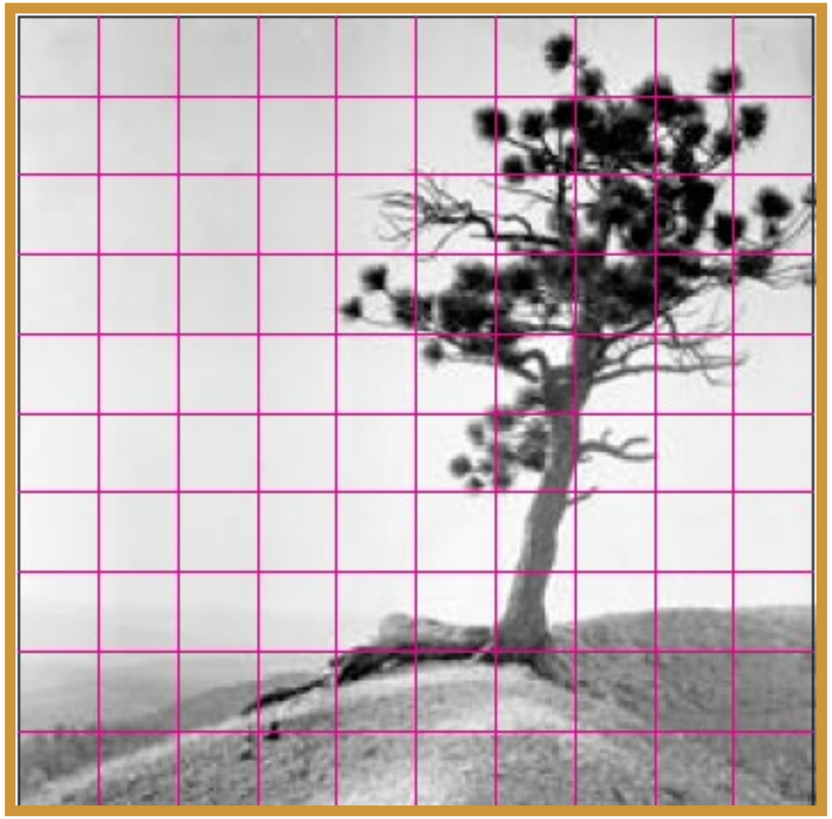


(a) Original image (b) 10×10 grid

Digitizing Images

- Each cell in the grid is compared to the gray scale.
  - It is assigned a value between 0 and 7, depending on which gray square in the scale most closely matches the "darkness" of the cell.
  - If the cell is not uniformly gray, an average value is assigned.



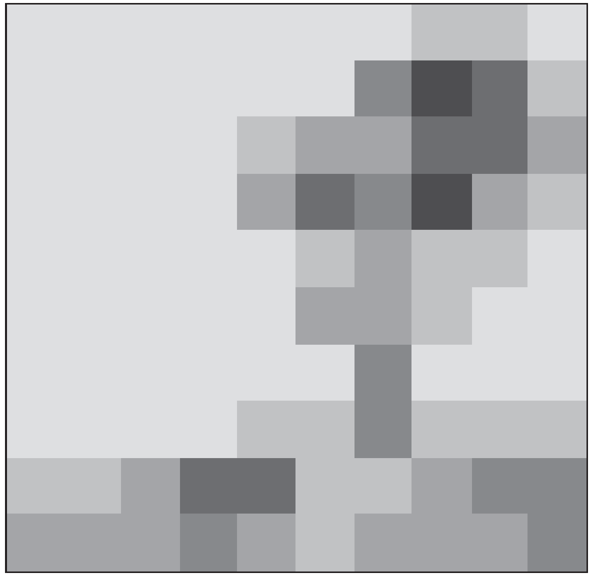


Digitizing Images

- The values are stored in the matrix shown.
  - The digital image corresponding to this matrix is shown in the accompanying image.

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\
1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\
1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\
1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\
1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\
2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\
3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4
\end{bmatrix}
$$

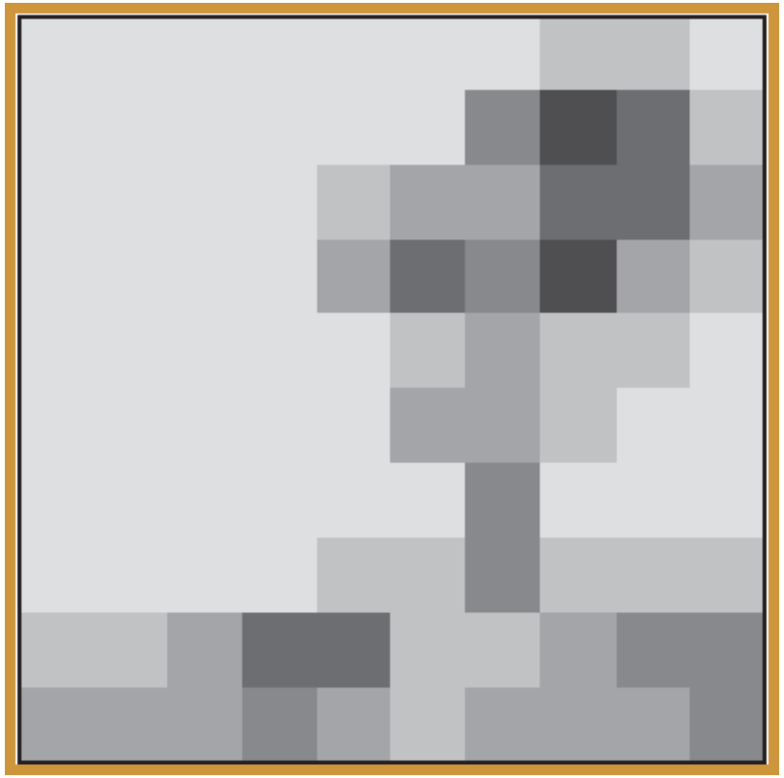
(c) Matrix representation



(d) Digital image

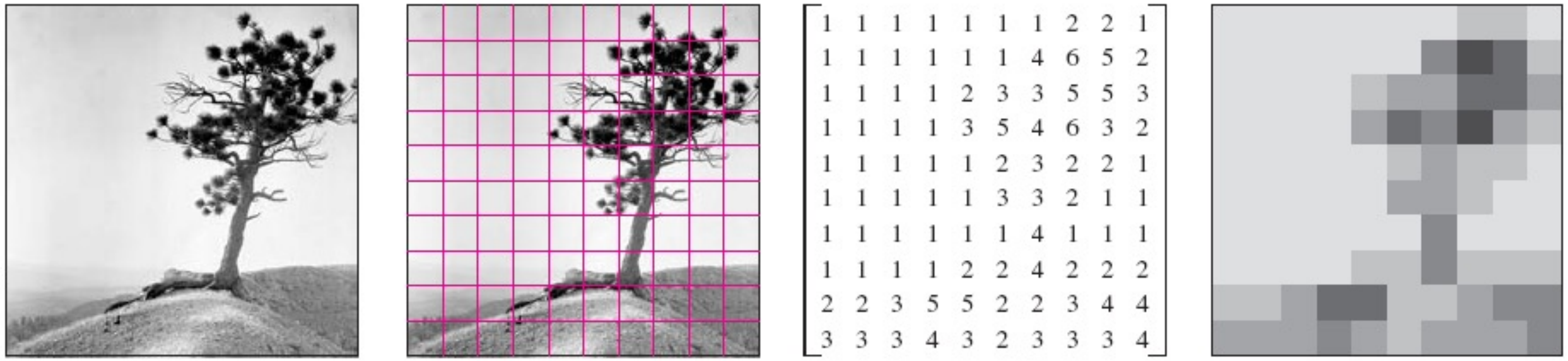
Digitizing Images

- Obviously, the grid we have used is far too coarse to provide good image resolution.
  - In practice, currently available high-resolution digital cameras use matrices with larger dimensions



Digitizing Images

- Here, we summarize the process.



(a) Original image (b) 10 × 10 grid (c) Matrix representation (d) Digital image

Manipulating Images

- Once the image is stored as a matrix, it can be manipulated using matrix operations.

  - To darken the image, we add a constant to each entry in the matrix.

  - To lighten the image, we subtract.

Manipulating Images

- To increase the contrast, we darken the darker areas and lighten the lighter areas.

  - So, we could add 1 to each entry that is 4, 5, or 6 and subtract 1 from each entry that is 1, 2, or 3.
  - Note that we cannot darken an entry of 7 or lighten a 0.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}$$

Modifying Matrices

- Applying this process to the earlier matrix produces a new matrix.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 6 & 5 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 & 5 & 4 & 6 & 3 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 \\ 2 & 2 & 3 & 5 & 5 & 2 & 2 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}$$

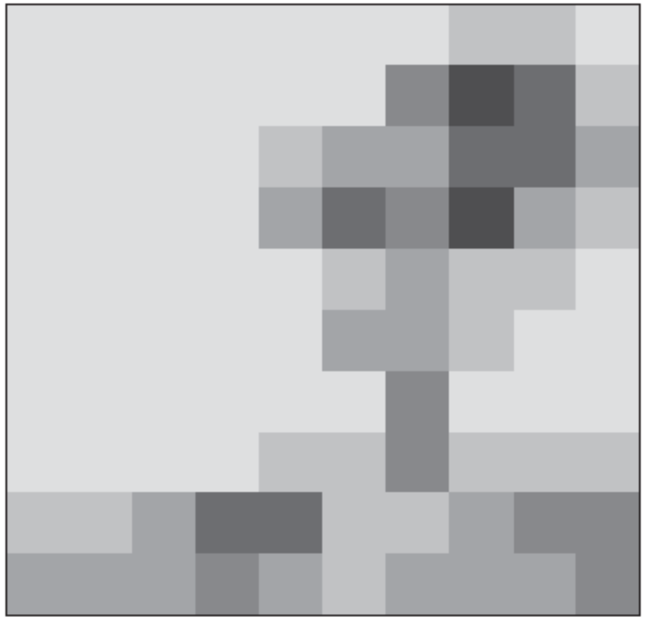
(c) Matrix representation

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 7 & 6 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 & 6 & 6 & 2 \\ 0 & 0 & 0 & 0 & 2 & 6 & 5 & 7 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 5 & 1 & 1 & 1 \\ 1 & 1 & 2 & 6 & 6 & 1 & 1 & 2 & 5 & 5 \\ 2 & 2 & 2 & 5 & 2 & 1 & 2 & 2 & 2 & 5 \end{bmatrix}$$

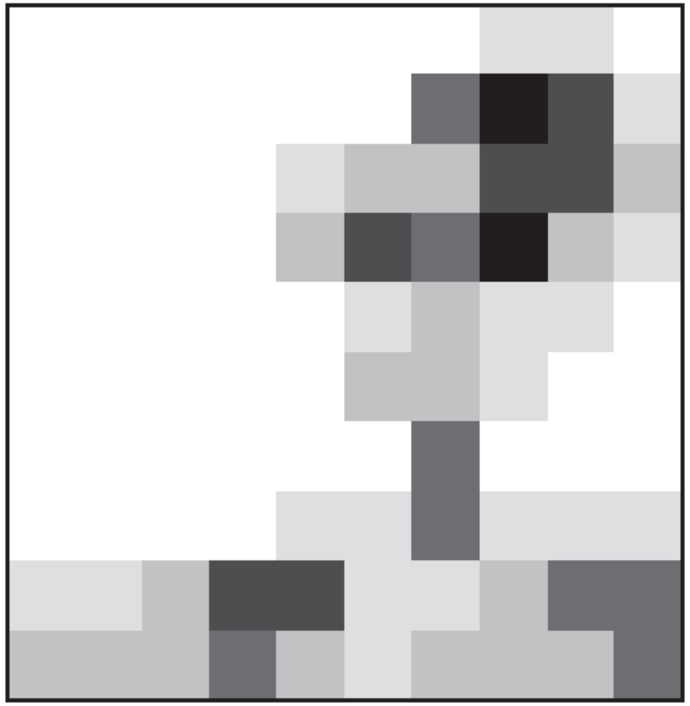
(a) Matrix modified to increase contrast

Modifying Images

- That generates the high-contrast image in the figure



(d) Digital image



(b) High-contrast image

Sparse Matrix

- A sparse matrix is a matrix that is comprised of mostly zero values. Sparse matrices are distinct from matrices with mostly non-zero values, which are referred to as dense matrices.
- A matrix is sparse if many of its coefficients are zero. The interest in sparsity arises because its exploitation can lead to enormous computational savings and because many large matrix problems that occur in practice are sparse.

- The sparsity of a matrix can be quantified with a score, which is the number of zero values in the matrix divided by the total number of elements in the matrix.

$$\text{Sparsity} = \frac{\text{count of zero elements}}{\text{total elements}}$$

- Below is an example of a small 3 × 6 sparse matrix.
- $A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}$
- The example has 13 zero values of the 18 elements in the matrix, giving this matrix a sparsity score of 0.722 or about 72%.

Space Complexity

- In practice, most large matrices are sparse — almost all entries are zeros.
- An example of a very large matrix that is too large to be stored in memory is a link matrix that shows the links from one website to another.
- An example of a smaller sparse matrix might be a word or term occurrence matrix for words in one book against all known words in English.
- In both cases, the matrix contained is sparse with many more zero values than data values. The problem with representing these sparse matrices as dense matrices is that memory is required and must be allocated for each 32-bit or even 64-bit zero value in the matrix. This is clearly a waste of memory resources as those zero values do not contain any information.

Time Complexity

- Assuming a very large sparse matrix can be fit into memory, we will want to perform operations on this matrix. Simply, if the matrix contains mostly zero-values, i.e. no data, then performing operations across this matrix may take a long time where the bulk of the computation performed will involve adding or multiplying zero values together.
- This is a problem of increased time complexity of matrix operations that increases with the size of the matrix. This problem is compounded when we consider that even trivial machine learning methods may require many operations on each row, column, or even across the entire matrix, resulting in vastly longer execution times.

Sparse Matrices in Machine Learning

- Sparse matrices turn up a lot in applied machine learning. Some common examples to motivate you to be aware of the issues of sparsity.

Data

- Sparse matrices come up in some specific types of data, most notably observations that record the occurrence or count of an activity. Three examples include:
- Whether or not a user has watched a movie in a movie catalogue.
- Whether or not a user has purchased a product in a product catalogue.
- Count of the number of listens of a song in a song catalogue.

Data Preparation

- Sparse matrices come up in encoding schemes used in the preparation of data. Three common examples include:
- One hot encoding, used to represent categorical data as sparse binary vectors.
- Count encoding, used to represent the frequency of words in a vocabulary for a document
- TF-IDF encoding, used to represent normalized word frequency scores in a vocabulary.

Areas of Study

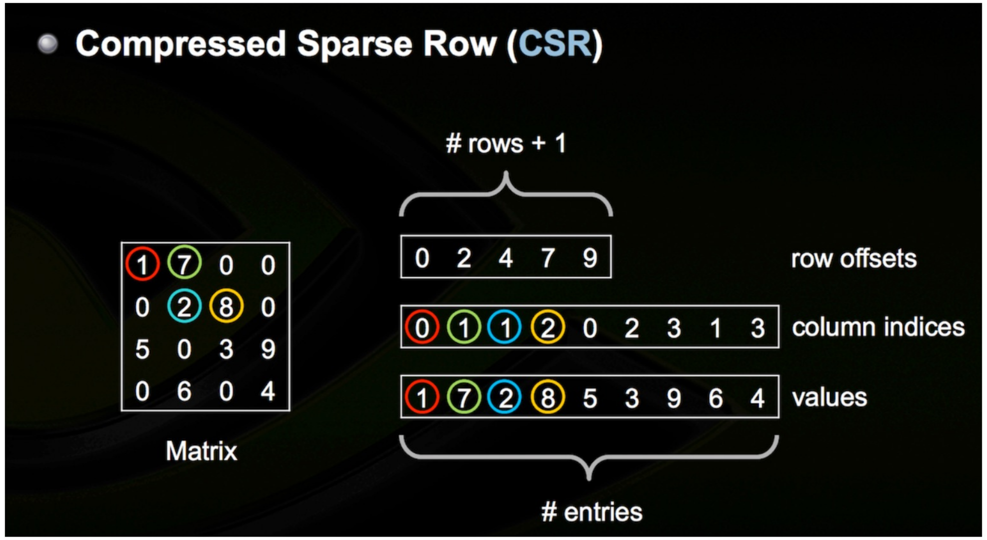
- Some areas of study within machine learning must develop specialized methods to address sparsity directly as the input data is almost always sparse. Three examples include:
- Natural language processing for working with documents of text.
- Recommender systems for working with product usage within a catalog.
- Computer vision when working with images that contain lots of black pixels.
- If there are 100,000 words in the language model, then the feature vector has length 100,000, but for a short email message almost all the features will have count zero.

Working with Sparse Matrices

- The solution to representing and working with sparse matrices is to use an alternate data structure to represent the sparse data. The zero values can be ignored and only the data or non-zero values in the sparse matrix need to be stored or acted upon. There are multiple data structures that can be used to efficiently construct a sparse matrix; three common examples are listed below.
- Dictionary of Keys. A dictionary is used where a row and column index is mapped to a value.
- List of Lists. Each row of the matrix is stored as a list, with each sublist containing the column index and the value.
- Coordinate List. A list of tuples is stored with each tuple containing the row index, column index, and the value.

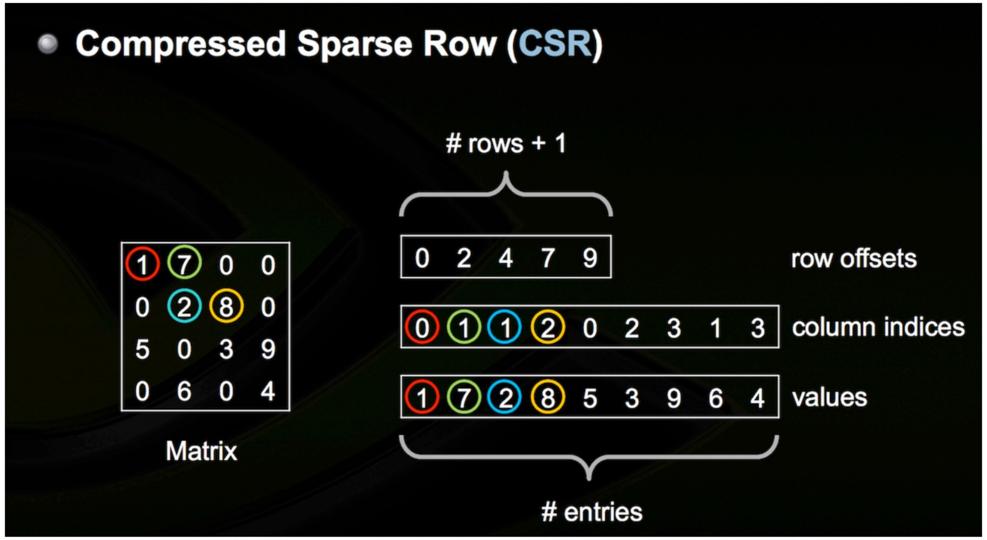
- There are also data structures that are more suitable for performing efficient operations; two commonly used examples are listed below.
  - Compressed Sparse Row. The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.
  - Compressed Sparse Column. The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.
  - The Compressed Sparse Row, also called CSR for short, is often used to represent sparse matrices in machine learning given the efficient access and matrix multiplication that it supports.

Compressed Sparse Row



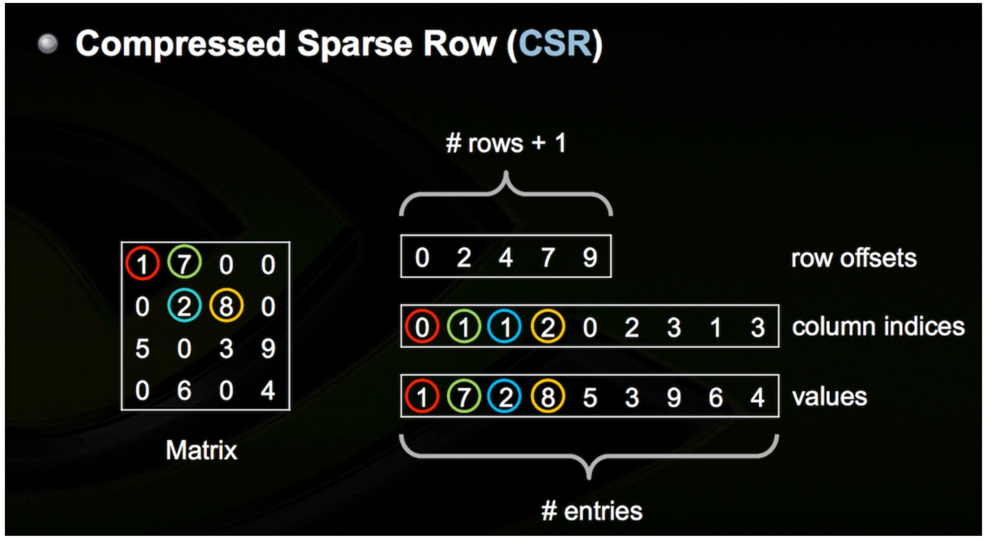
- The first step is to populate the first array. It always starts with 0. Since there are two nonzero values in row 1, we update our array like so [0 2]. There are 2 nonzero values in row 2, so we update our array to [0 2 4]. Doing that for the remaining rows yields [0 2 4 7 9]. By the way, the length of this array should always be the number of rows + 1.

Compressed Sparse Row



- Step two is populating the second array of column indices. Note that the columns are zero-indexed. The first value, 1, is in column 0. The second value, 7, is in column 1. The third value, 2, is in column 1. And so on. The result is the array [0 1 1 2 0 2 3 1 3].

Compressed Sparse Row

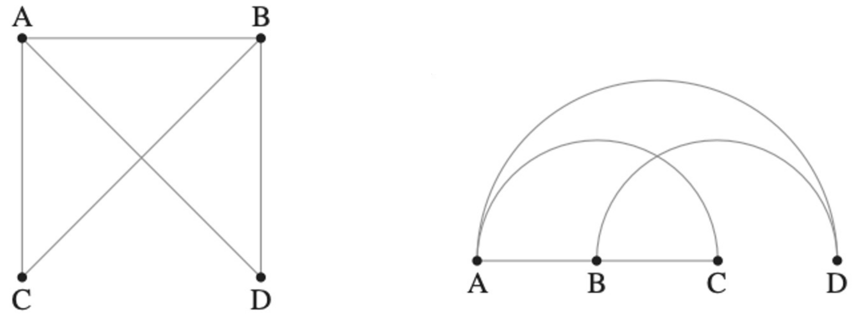


- Finally, we populate the third array which looks like this [1 7 2 8 5 3 9 6 4]. Again, we are only storing nonzero values.

Matrix Application in Graph Theory

Graphs

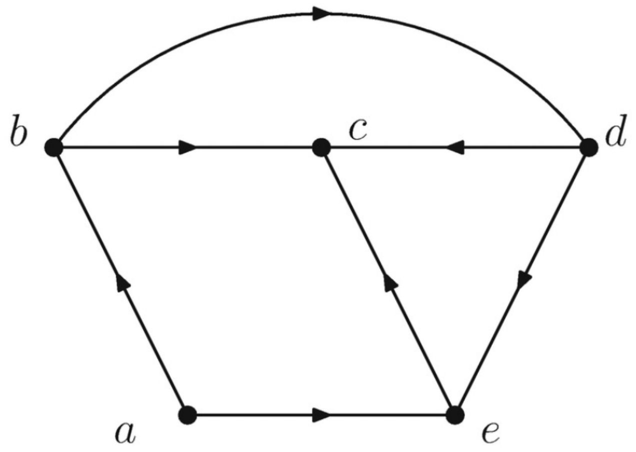
- A graph consists of a finite set of points (called vertices) and a finite set of edges, each of which connects two (not necessarily distinct) vertices.
- Graph is a pair G = (V, E)   , where V is the set of vertices , while E is the set of edges , connecting some pairs of vertices.



Two representations of the same graph

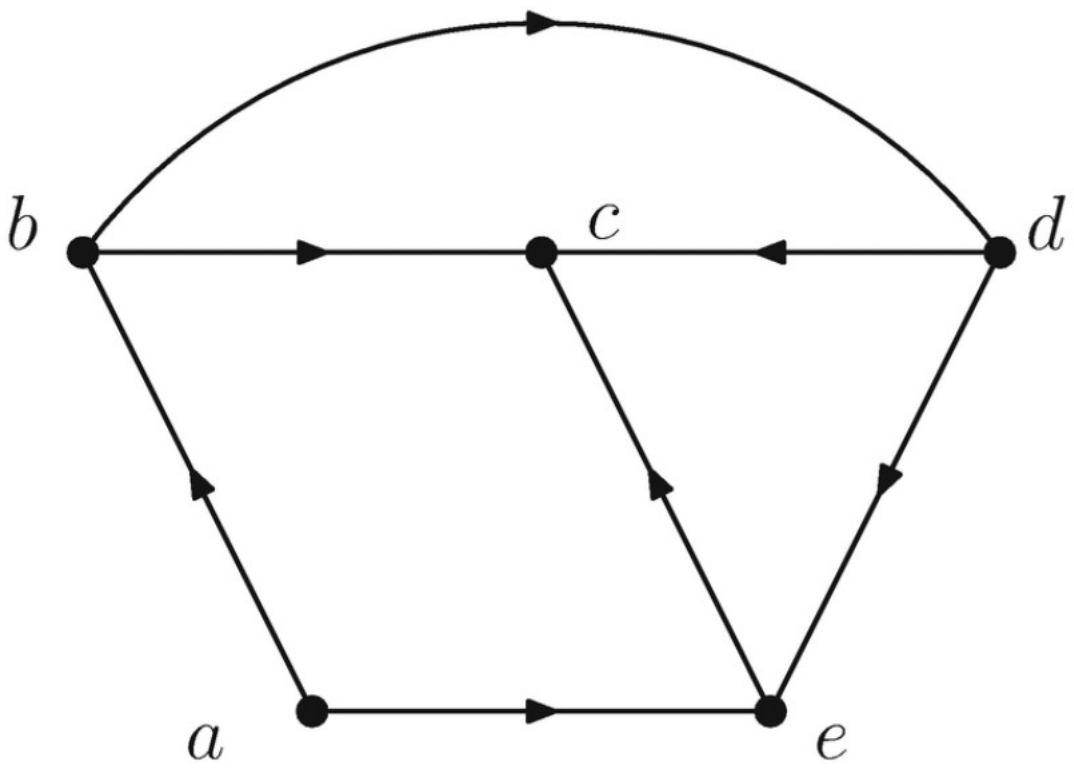
Digraphs

- In directed graphs , the edges are the ordered pair of vertices, i.e. it is of importance which vertex is the beginning of the edge and which one is the end.
- Directed graphs are also referred to as digraphs .



Graph Diagram

- A drawing where the graph vertex is shown as points and the edges are shown as segments or arcs is called a **graph diagram**
- Consider a digraph *D*(*V*, *E*), the set of vertices *V* and the set of edges *E* of which are specified as follows:
- V = {a,b,c,d,e}
- E= {ab,ae,bc,bd,dc,de,ec}



Matrix in the Graph Theory

- Two vertices v_i and v_j of the graph are **adjacent**, if they are connected by the edge *r* = $v_i\ v_j$.
- In this case it is said that the vertices v_i and v_j are the **endpoints** of the edge *r*.
- If the vertex *v* is the endpoint of the edge *r*, then *v* and *r* are considered to be **incident**
- The number of elements (**cardinality**) of any set, for example *V* , is denoted as $|V|$.

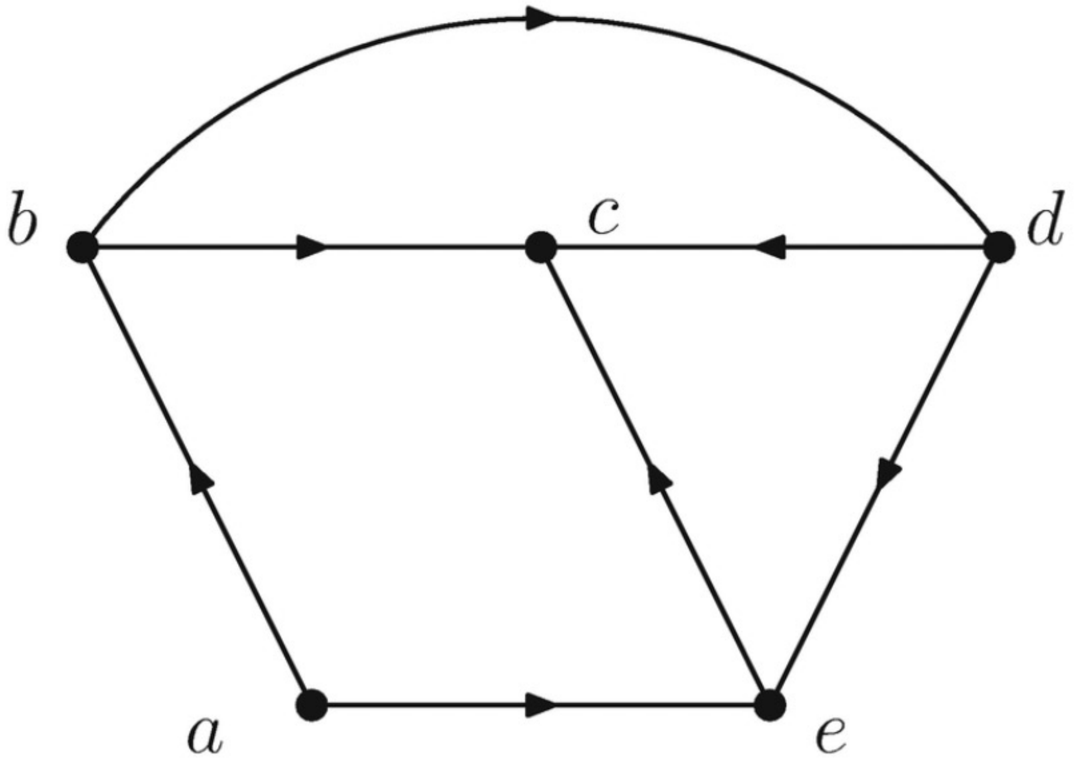
Adjacency Matrix

- **Adjacency matrix *A*** is a binary matrix of a relation over the set of vertices of the graph $G(V, E)$, which is specifies by its edges. The adjacency matrix as the size $|V| \times |V|$, and its elements are determined in accordance with the rule
- $a(i,j) = \begin{cases} 1, \text{if there is an edge between vertices } i \text{ and } j \\ 0, otherwise. \end{cases}$

Adjacency matrix example

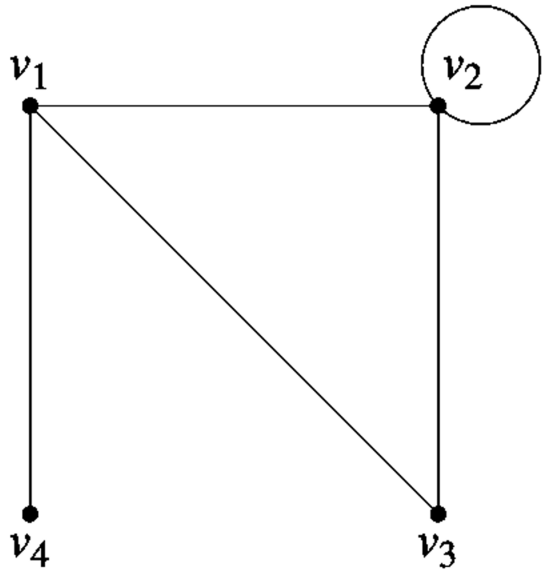
- Consider a digraph $D(V, E)$, the set of vertices V and the set of edges E of which are specified as follows:
- The adjacency matrix A of the digraph D has the form:

$$A = \text{From}\ \begin{array}{c c} & \overbrace{\begin{array}{ccccc} a & b & c & d & e \end{array}}^{\text{To}} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$



Adjacency matrix example

- A diagonal entry a_{ij} of A is zero unless there is a loop at vertex *i*.
- Example, there is a loop at v_2 , hence a_{22} = 1.
- In some situations, a graph may have more than one edge between a pair of vertices. In such cases, it may make sense to modify the definition of the adjacency matrix so that a_{ij} equals the *number* of edges between vertices *i* and *j*.



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

Paths

- A **path** of length *k* in the graph *G* is a sequence of vertices

$v_0, v_1, \ldots v_k$ such that $\forall$ $i = 1, \ldots . k$ the vertices v_{i-1} and v_i are adjacent.

For undirected graphs , paths are also called **routes** .

The **length** of the path is the number of edges in it, taking into account the iterations.

K-paths

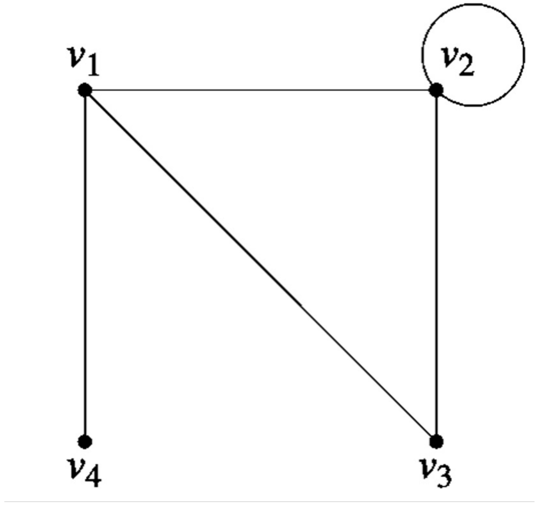
- We will refer to a path with k edges as a k-path
- If A is the adjacency matrix of a graph G, then the (i, j) entry of A^k is equal to the number of k-paths between vertices i and j.

K-paths

- What do the entries of A^2 represent? Look at the (2, 3) entry. From the definition of matrix multiplication, we know that

$$(A^2)_{23} = a_{21}a_{13} + a_{22}a_{23} + a_{23}a_{33} + a_{24}a_{43}$$

- The only way this expression can result in a nonzero number is if at least one of the products $a_{2k}\, a_{k3}$ that make up the sum is nonzero. But $a_{2k}a_{k3}$ is nonzero if and only if both a_{2k} and a_{k3} are nonzero, which means that there is an edge between v_2 and v_k as well as an edge between v_k and v_3. Thus, there will be a 2-path between vertices 2 and 3 (via vertex *k*).



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

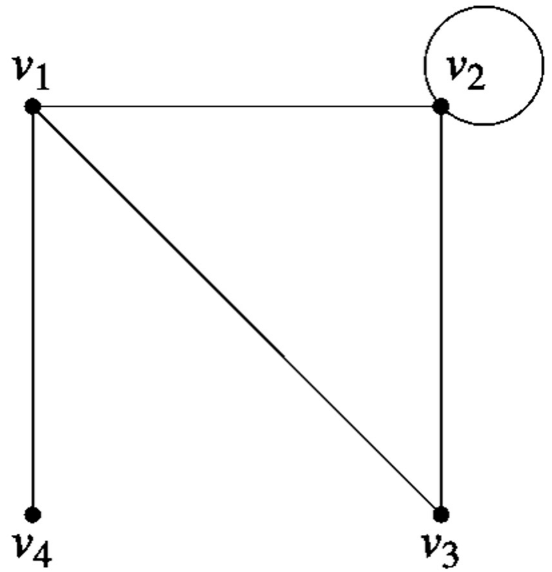
$$A^2 = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

K-paths

$$
\begin{aligned}
(A^2)_{23} &= a_{21}a_{13} + a_{22}a_{23} + a_{23}a_{33} + a_{24}a_{43} \\
&= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 \\
&= 2
\end{aligned}
$$

which tells us that there are two 2-paths between vertices 2 and 3.

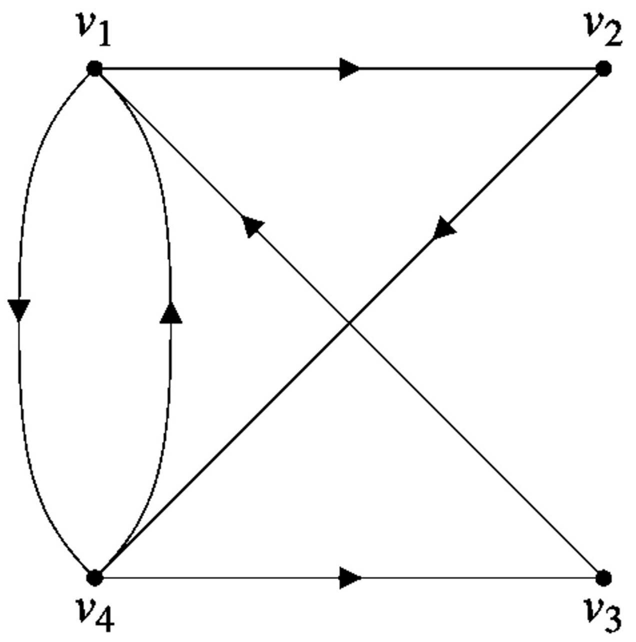
- $v_2 \to v_2 \to v_3$
- $v_2 \to v_1 \to v_3$



K-paths example 1

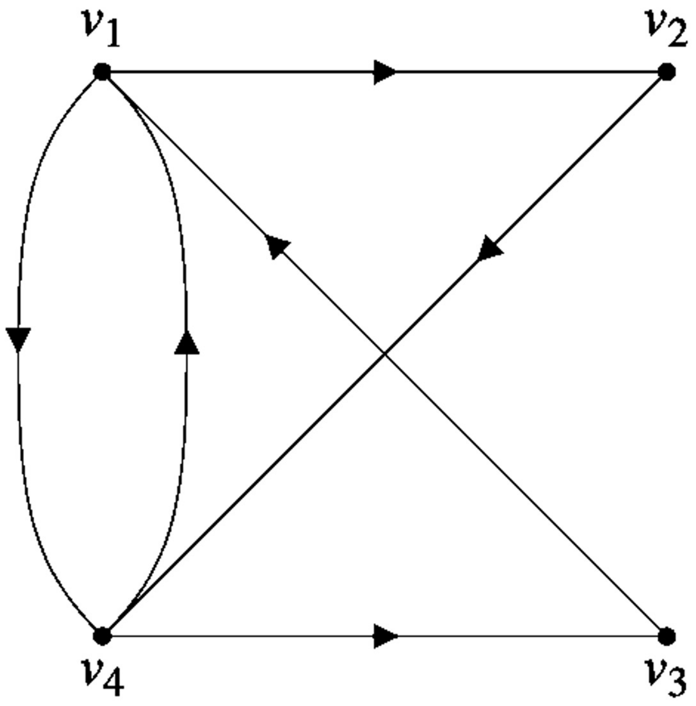
- How many 3-paths are there between v_1 and v_2

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



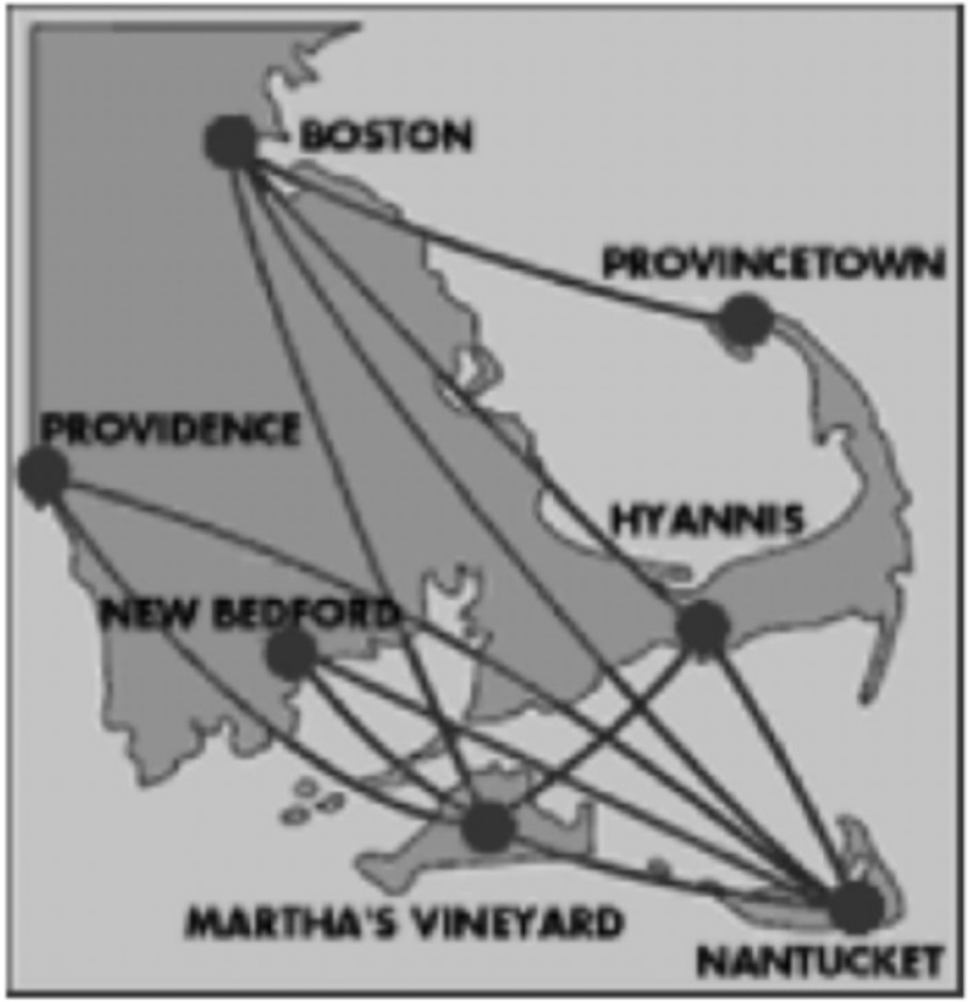
K-paths solution 1

- We need the (1, 2) entry of A^3, which is the dot product of row 1 of A^2 and column 2 of A.
- $A^2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$
- $A^3 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$
- There is one path between v_1 and v_2 of length 3



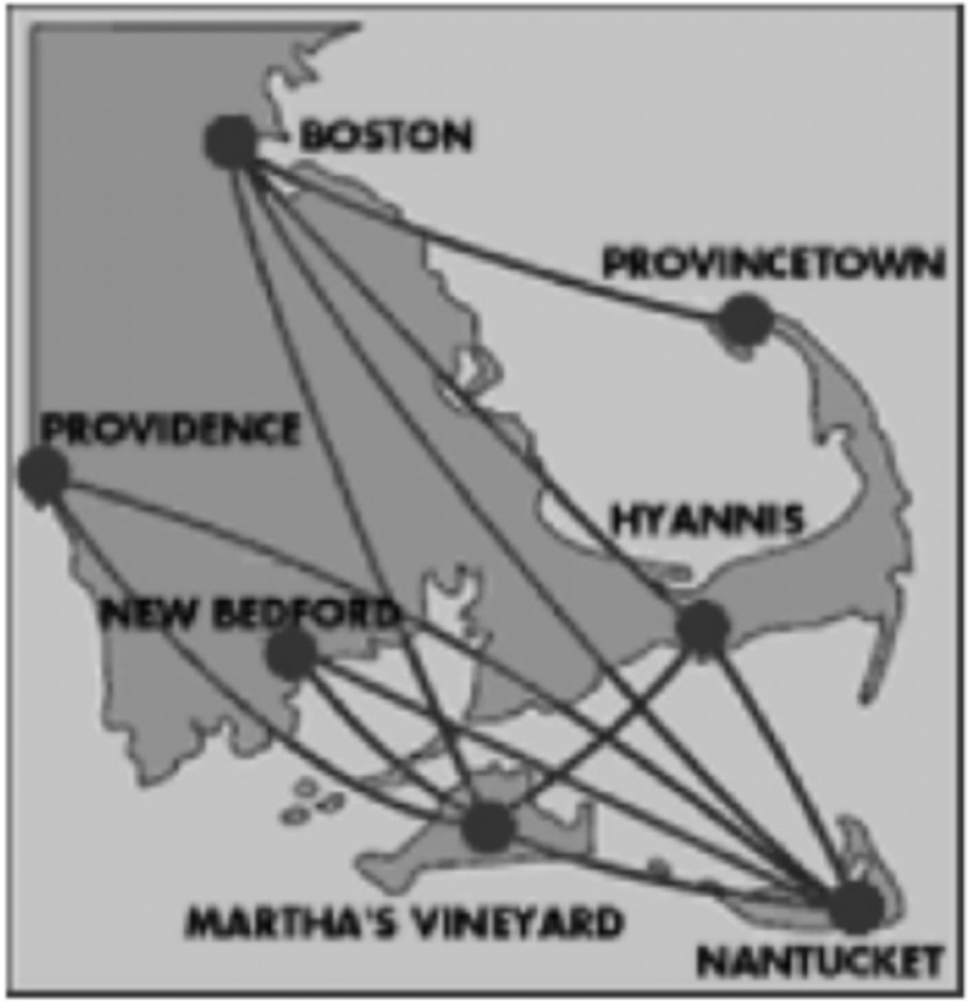
Application of adjacency matrix

- An example of a graph is the route map that most airlines (or railways) produce.
- A copy of the northern route map for Cape Air from May 2001 is Figure. Here the vertices are the cities to which Cape Air flies, and two vertices are connected if a direct flight flies between them.



Application of adjacency matrix

- In the route map, Provincetown and Hyannis are connected by a two-edge sequence, meaning that a passenger would have to stop in Boston while flying between those cities on Cape Air. It might be important to know if it is possible to get from a vertex to another vertex.



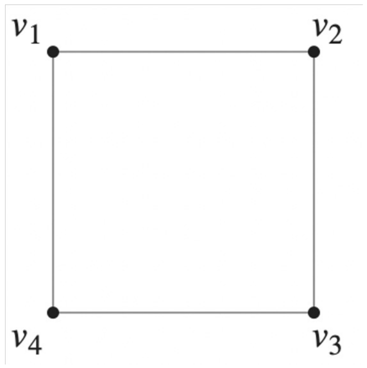
Application of adjacency matrix

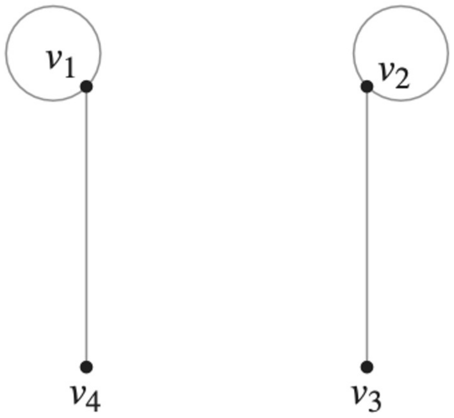
- If the vertices in the Cape Air graph respectively correspond to Boston,
- Hyannis, Martha’s Vineyard, Nantucket, New Bedford, Providence, and Provincetown, then the adjacency matrix for Cape Air is

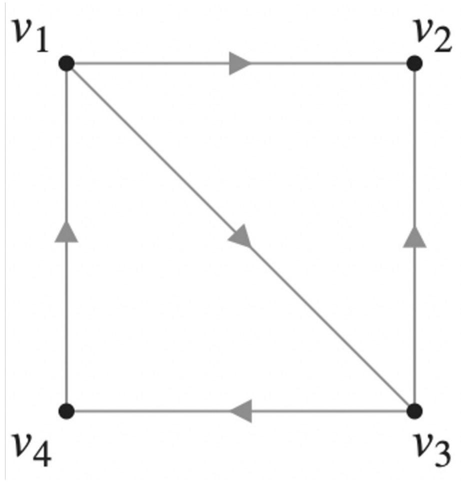
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

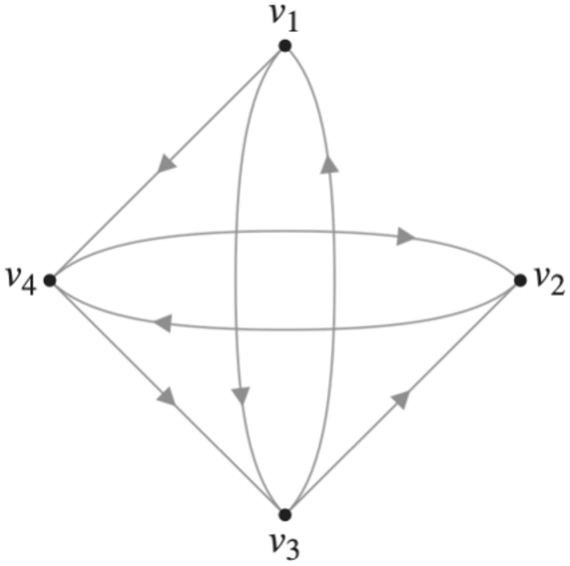
Example

Determine the adjacency matrix of the given graph.



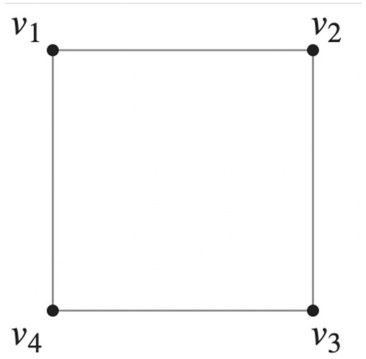


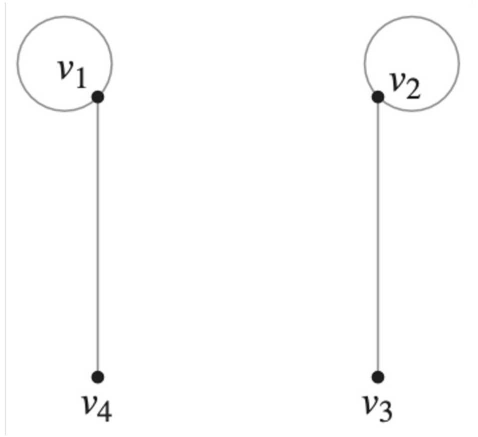


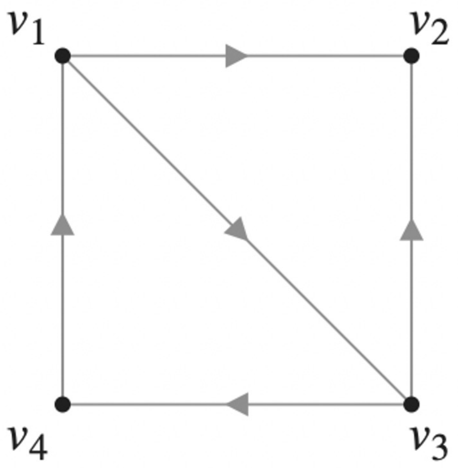


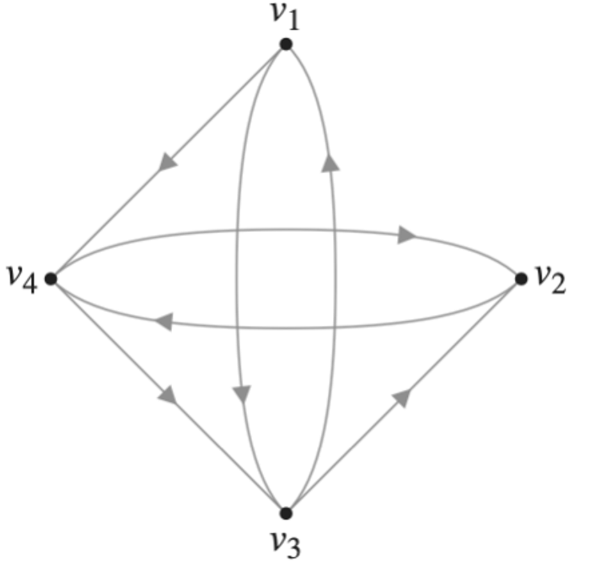
Solution

Determine the adjacency matrix of the given graph.









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

Example

*Draw a **graph** that has the given adjacency matrix.*

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

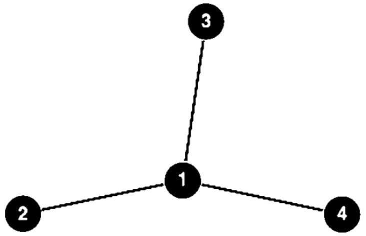
*Draw a **digraph** that has the given adjacency matrix.*

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

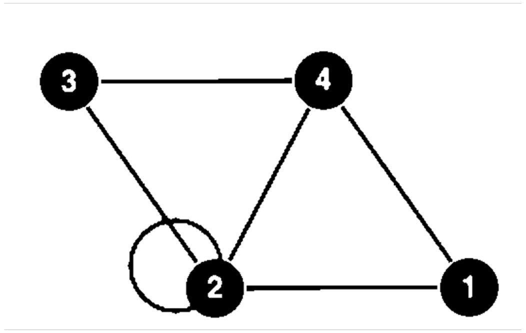
Solution

*Draw a **graph** that has the given adjacency matrix.*

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

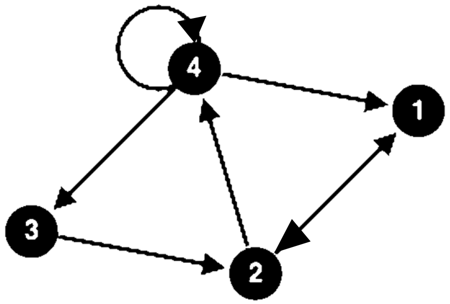


$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$



*Draw a **digraph** that has the given adjacency matrix.*

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



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