DATA SCIENCE

UNIT –IV Computational Methods

Objective:

- To know the importance of statistical learning.

Syllabus:

UNIT - IV: Computational Methods

Programming for basic computational methods such as Eigen values and Eigen vectors, sparse matrices, QR and SVD.

Learning Outcomes:

The student will be able to

- describe how to experimentally obtain and evaluate sequence information
- CO1 Apply statistical methods to data for inferences.
- CO2 analyze data using Classification, Graphical and computational methods.

Learning Material

https://cran.r-project.org/web/packages/matlib/vignettes/eigen-ex1.html

UNIT - IV: Computational Methods

1. Eigen values and Eigen vectors,

A linear equation about scalar λ and vector **x** in the form of

$$(\mathbf{A}-\lambda\mathbf{I})\mathbf{x}=0 \quad \text{or} \quad \mathbf{Ax}=\lambda\mathbf{x} \qquad (1)$$

apparently has a zero solution of **x**=0. This zero solution is called a trivial solution, for it is obvious and needs no mathematical effort. Only when $(\mathbf{A}-\lambda\mathbf{I})$ is a singular matrix can the above equation has a non-zero (non-trivial) solution $\mathbf{x}\neq 0$.

For $(\mathbf{A}-\lambda\mathbf{I})$ to be singular, that is, $\det(\mathbf{A}-\lambda\mathbf{I})=0$, λ cannot be arbitrary. A value of λ which permits a non-trivial solution $\mathbf{x}\neq 0$ of

$(\mathbf{A}-\lambda\mathbf{I})\mathbf{x}=0$ is called an eigenvalue of matrix **A**. The corresponding solution **x**≠0 is called an eigenvector. Together a pair of (λ, **x**) is called an eigen-solutions of **A**. There can be as many eigenvalues (eigenvectors) as the order of a square matrix.

Let A be an nxn matrix over a field F . We recall that a scalar is said to be an *eigenvalue* (*characteristic value*, or a *latent root*) of A, if there exists a nonzero vector x such that $(\mathbf{A}-\lambda\mathbf{I})\mathbf{x}=0$ and that such an x is called an *eigen-vector* (*characteristic vector*, or a *latent vector*) of A corresponding to the eigenvalue and that the **pair (λ , x)** is called an *eigen-pair* of A. If is an eigenvalue of A, the equation: (λ I-A)x = 0, has a non-trivial (non-zero) solution and conversely. Thus, this being a homogeneous equation, it follows that **λ** is an eigenvalue of A iff |λ I-A| = 0.

This vignette uses an example of a 3×3 matrix to illustrate some properties of eigenvalues and eigenvectors. We could consider this to be the variance-covariance matrix of three variables, but the main thing is that the matrix is **square** and **symmetric**, which guarantees that the eigenvalues, λ_i are real numbers. Covariance matrices are also **positive semi-definite**, meaning that their eigenvalues are non-negative, $\lambda_i \geq 0$ $\lambda_i \geq 0$.

```
A <- matrix(c(13, -4, 2, -4, 11, -2, 2, -2, 8), 3, 3, byrow=TRUE)
A

##      [,1] [,2] [,3]
## [1,]   13   -4    2
## [2,]   -4   11   -2
## [3,]    2   -2    8
```

Get the eigenvalues and eigenvectors using eigen(); this returns a named list, with eigenvalues named valuesand eigenvectors named vectors.

```
ev <- eigen(A)
# extract components
(values <- ev$values)

## [1] 17  8  7

(vectors <- ev$vectors)

##         [,1]    [,2]   [,3]
## [1,]  0.7454  0.6667 0.0000
## [2,] -0.5963  0.6667 0.4472
```

```
## [3,]  0.2981 -0.3333 0.8944
```

Prove that the vectors a = {1; 2} and b = {2; -1} are orthogonal.

Solution:

Calculate the dot product of these vectors:

$a \cdot b = 1 \cdot 2 + 2 \cdot (-1) = 2 - 2 = 0$

Answer: since the dot product is zero, the vectors a and b are orthogonal.

The eigenvalues are always returned in decreasing order, and each column of vectors corresponds to the elements in values.

$$W = \begin{bmatrix} 20 & 14 & 0 & 0 \\ 14 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Sparse Matrix Representations

http://btechsmartclass.com/data_structures/sparse-matrix.html

When a sparse matrix is represented with a 2-dimensional array, we waste a lot of space to represent that matrix. For example, consider a matrix of size 100 X 100 containing only 10 non-zero elements. In this matrix, only 10 spaces are filled with non-zero values and remaining spaces of the matrix are filled with zero. That means, totally we allocate 100 X 100 X 2 = 20000 bytes of space to store this integer matrix. And to access these 10 non-zero elements we have to make scanning for 10000 times. To make it simple we use the following sparse matrix representation.

http://www.johnmyleswhite.com/notebook/2011/10/31/using-sparse-matrices-in-r/

A sparse matrix can be represented by using TWO representations, those are as follows...

1. Triplet Representation (Array Representation)
2. Linked Representation

2.1 Triplet Representation (Array Representation)

In this representation, we consider only non-zero values along with their row and column index values. In this representation, the 0^{th}row stores the total number of rows, total number of columns and the total number of non-zero values in the sparse matrix.

For example, consider a matrix of size 5 X 6 containing 6 number of non-zero values. This matrix can be represented as shown in the image...

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \Rightarrow$$

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

2.2 Linked Representation:

In linked representation, we use a linked list data structure to represent a sparse matrix.

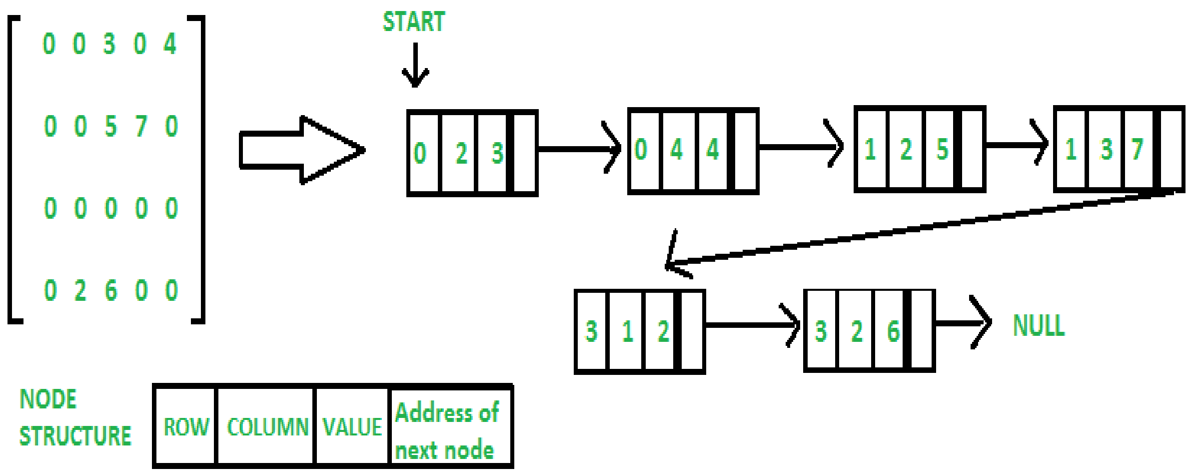
In linked list, each node has four fields. These four fields are defined as:

Row: Index of row, where non-zero element is located

Column: Index of column, where non-zero element is located

Value: Value of the non zero element located at index – (row,column)

Next node: Address of the next node



Coordinate List Representation

A *coordinate list representation*, also known as *COO*, or *triplet represantation* is simply a list of the non zero entries. Each element in the list is a triplet of the row, column, and value, of each non-zero entry in the matrix.

3. QR factorization

1. Any **n x n** real matrix can be written as ***A=QR***, where ***Q*** is orthogonal and ***R*** is upper triangular.
2. Every n × m matrix A have a matrix decomposition A = QR . where R is a n × m upper triangular matrix. Q is a n × n unitary matrix.
3. QR decomposition of a matrix is also called as factorization, a decomposition of matrix A into a product of Q and R. i.e A= Q R an orthogonal matrix *Q* and an upper triangular matrix *R*.
4. QR decomposition is often used to solve the linear least squares problem and is the basis for a particular eigenvalue algorithm, the QR algorithm.
5. where *Q* is an orthogonal matrix (its columns are orthogonal unit vectors meaning Q $QQ' = Q'Q = I$ and *R* is an upper triangular matrix (also called right triangular matrix). If *A* is invertible, then the factorization is unique if we require the diagonal elements of *R* to be positive.
6. If instead *A* is a complex square matrix, then there is a decomposition *A* = *QR* where *Q* is a unitary matrix.

1 Gram-Schmidt process

Consider the GramSchmidt procedure, with the vectors to be considered in the process as columns of the matrix A. That is,

$$A = \left[\mathbf{a}_1 \mid \mathbf{a}_2 \mid \cdots \mid \mathbf{a}_n \right].$$

Then,

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{a}_1, \qquad \mathbf{e}_1 = \frac{\mathbf{u}_1}{||\mathbf{u}_1||}, \\
\mathbf{u}_2 &= \mathbf{a}_2 - (\mathbf{a}_2 \cdot \mathbf{e}_1)\mathbf{e}_1, \qquad \mathbf{e}_2 = \frac{\mathbf{u}_2}{||\mathbf{u}_2||}. \\
\mathbf{u}_{k+1} &= \mathbf{a}_{k+1} - (\mathbf{a}_{k+1} \cdot \mathbf{e}_1)\mathbf{e}_1 - \cdots - (\mathbf{a}_{k+1} \cdot \mathbf{e}_k)\mathbf{e}_k, \qquad \mathbf{e}_{k+1} = \frac{\mathbf{u}_{k+1}}{||\mathbf{u}_{k+1}||}.
\end{aligned}$$

Note that $||\cdot||$ is the L_2 norm.

1.1 QR Factorization

The resulting QR factorization is

$$A = \left[\mathbf{a}_1 \mid \mathbf{a}_2 \mid \cdots \mid \mathbf{a}_n \right] = \left[\mathbf{e}_1 \mid \mathbf{e}_2 \mid \cdots \mid \mathbf{e}_n \right] \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{e}_1 & \mathbf{a}_2 \cdot \mathbf{e}_1 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_1 \\ 0 & \mathbf{a}_2 \cdot \mathbf{e}_2 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_n \end{bmatrix} = QR.$$

Note that once we find $\mathbf{e}_1, \ldots, \mathbf{e}_n$, it is not hard to write the QR factorization.

4. Singular Value Decomposition

1. Transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items.
2. Identifying and ordering the dimensions along which data points exhibit the most variation.
3. Find the best approximation of the original data points using fewer dimensions→Data Reduction
Consider the 2-dimensional data points shown below.

Basic ideas behind SVD Taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from most variation to the least.

4.1 Example of Full Singular Value Decomposition

SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix Σ, and the transpose of an orthogonal matrix V.

$$\boldsymbol{A_{mn} = U_{mm}\Sigma_{mn}V_{nn}^T}$$

where $\boldsymbol{U^TU = I;\ V^TV = I}$.

Consider, let us start with

$$\boldsymbol{AA^T} = \begin{bmatrix} \mathbf{3} & \mathbf{1} & \mathbf{1} \\ \mathbf{-1} & \mathbf{3} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{3} & \mathbf{-1} \\ \mathbf{1} & \mathbf{3} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{11} & \mathbf{1} \\ \mathbf{1} & \mathbf{11} \end{bmatrix}.$$

Find the eigenvalues and corresponding eigenvectors of $\boldsymbol{AA^T}$, *i.e.*,

$$\boldsymbol{AA^T\vec{v} = \lambda\vec{v}} \quad \rightarrow \quad \begin{bmatrix} \mathbf{11} & \mathbf{1} \\ \mathbf{1} & \mathbf{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

The set of equations becomes

$$(11-\lambda)x_1 + x_2 = 0$$

$x_1 + (11-\lambda)x_2 = 0$. A nontrivial solution occurs when

$\begin{vmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{vmatrix} = 0$, which results in $\boldsymbol{\lambda = 10,\ 12}$.

For $\boldsymbol{\lambda = 12}$, we have $\boldsymbol{x_1 = x_2}$, *i.e.*, $\boldsymbol{\vec{v}_1} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$. For $\boldsymbol{\lambda = 10}$, we have $\boldsymbol{x_1 = -x_2}$, *i.e.*,

$\boldsymbol{\vec{v}_2} = \begin{bmatrix} \mathbf{1} \\ \mathbf{-1} \end{bmatrix}$.

Finally, we have to convert this matrix into an orthogonal matrix which we do by applying the **Gram-Schmidt** orthonormalization process to the column vectors.

Begin by normalizing $\boldsymbol{\vec{v}_1}$:

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = \frac{[1,1]}{\sqrt{1^2+1^2}} = \frac{[1,1]}{\sqrt{2}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

We then compute $\vec{w}_2 = \vec{v}_2 - \{\vec{u}_1 \cdot \vec{v}_2\}\vec{u}_1 = [1,-1] - \{[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \cdot [1,-1]\}[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$

$$= [1,-1] - 0 * [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = [1,-1].$$

Normalize $\vec{w}_2$, we achieve $\vec{u}_2 = \frac{\vec{w}_2}{|\vec{w}_2|} = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$ and then obtain

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}.$$

The calculation of $\boldsymbol{V}$ is similar by starting to find the eigenvalues and corresponding eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$

$$A^T A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}.$$

$$\boldsymbol{A}^T\boldsymbol{A}\vec{v} = \lambda\vec{v} \quad \rightarrow \quad \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

$$(10 - \lambda)x_1 + 2x_3 = 0$$
$$(10 - \lambda)x_2 + 4x_3 = 0$$
$$2x_1 + 4x_2 + (2 - \lambda)x_3 = 0$$

The system of equations which has a nontrivial solution when

$$\begin{vmatrix} (10-\lambda) & 0 & 2 \\ 0 & (10-\lambda) & 4 \\ 2 & 4 & (2-\lambda) \end{vmatrix} = 0$$

The eigenvalues are $\lambda = 0, \lambda = 10, \lambda = 12$.

For $\lambda = 12$, $\vec{v}_1 = [1,2,1]$. For $\lambda = 10$, $\vec{v}_1 = [2,-1,0]$. For $\lambda = 0$, $\vec{v}_1 = [1,2,-5]$.

Use the **Gram-Schmidt orthonormalization process** to convert the eigenvectors into orthonormal set

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = [\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$$
$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} = [2, -1, 0]$$
$$\vec{u_2} = \frac{\vec{w_2}}{|\vec{w_2}|} = [\frac{2}{\sqrt{5}}, \frac{-1}{\sqrt{5}}, 0]$$
$$\vec{w_3} = \vec{v_3} - \vec{u_1} \cdot \vec{v_3} * \vec{u_1} - \vec{u_2} \cdot \vec{v_3} * \vec{u_2} = [\frac{-2}{3}, \frac{-4}{3}, \frac{10}{3}]$$
$$\vec{u_3} = \frac{\vec{w_3}}{|\vec{w_3}|} = [\frac{1}{\sqrt{30}}, \frac{2}{\sqrt{30}}, \frac{-5}{\sqrt{30}}]$$

This gives us

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-5}{\sqrt{30}} \end{bmatrix}.$$

For Σ we take the square roots of the non-zero eigenvalues and populate the diagonal with them, putting the largest in Σ_{11}, the next largest in Σ_{22} and so on until the smallest value ends up in Σ_{mm}. The non-zero eigenvalues of ***U*** and ***V*** are always the same, so that's why it doesn't matter which one we take them from. The diagonal entries in Σ are the singular values of ***A***, the columns in ***U*** are called left singular vectors, and the columns in ***V*** are called right singular vectors.

$$\Sigma = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}.$$ Finally

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} =$$
$$\begin{bmatrix} \frac{\sqrt{12}}{\sqrt{2}} & \frac{\sqrt{10}}{\sqrt{2}} & 0 \\ \frac{\sqrt{12}}{\sqrt{2}} & \frac{-\sqrt{10}}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

4.2 **Singular Value Decomposition** (SVD) of a matrix:

This is a powerful matrix operation useful for generalized least square regressions, collinearity corrections and better understanding of the data.

SVD gives 3 matrics from one matrix A defined by the relation: $A=u\ d\ v^T$

One application of the SVD is to find the square root matrix. Square root of the matrix A is found by replacing the middle diagonal Matrix d by its squre root and multiplying out as follows.

help(svd) #singular value decomposition help
asvd=svd(A) #place the output of svd to place named asvd
asvd

svd gives 3 matrices from the matrix A= udv'

```
#check if we get the original matrix back
# the middle d is a vector, must make a diag matrix from it first.

(asvd$u) %*% diag(asvd$d) %*% t(asvd$v)
```

https://fenix.tecnico.ulisboa.pt/downloadFile/3779576344458/singular-value-decomposition-fast-track-tutorial.pdf

Step 1. Compute its transpose A^T and A^TA.

Step 2. Determine the eigenvalues of $\mathbf{A^TA}$ and sort these in descending order, in the absolute sense. Square roots these to obtain the singular values of **A**.

Step 3. Construct diagonal matrix S by placing singular values in descending order along its diagonal. Compute its inverse, S^{-1}.

Step 4. Use the ordered eigenvalues from step 2 and compute the eigenvectors of $\mathbf{A^TA}$. Place these eigenvectors along the columns of **V** and compute its transpose, $\mathbf{V^T}$

Step 5. Compute **U** as **U = AVS-1**. To complete the proof, compute the full SVD using **A = USVT**.

Assignment –Cum –Tutorial Questions

Unit-IV

SECTION A

1. The ________ nature of the **V** and **U** matrices is evident in SVD by inspecting their eigenvectors.

   a) orthogonal

   b) Kappa

   c) RMSE

   d) All of the Mentioned

Answer: orthogonal.

2. Which of the following are True in SVD?

i) Determine the eigenvalues of $\mathbf{A}^T\mathbf{A}$ and sort these in descending order, in the absolute sense. Square roots these to obtain the singular values of **A**.

ii) Compute **U** as $\mathbf{U} = \mathbf{AVS}^{-1}$.

iii) To complete the proof, compute the full SVD using $\mathbf{A} = \mathbf{USV}^T$

A) i only B) ii only C) ii and iii D) All

4. To compute the determinant of a matrix the QR decomposition is much more efficient than using Eigen values. [True/false]

5. The ___________of a matrix is a decomposition of the matrix into an orthogonal matrix and a triangular matrix.

   A) SVD B) QR C) Both D) NONE

6. Which of the following are True?

1. The science of **statistics** deals with the collection, analysis, interpretation, and presentation of **data**.

2. SVD is based on a theorem from linear algebra which says that a rectangular matrix ***A*** can be broken down into the product of three matrices - an orthogonal matrix ***U***, a diagonal matrix Σ, and transpose of an orthogonal matrix ***V***

A) 1 only B) 2 only C) None D) All

Descriptive Questions

1. Prove that the vectors a = {1; 2} and b = {2; -1} are orthogonal.
2. Programming for basic computational methods such as Eigen values and Eigen Vectors.
3. Discuss about QR
4. Explain the procedure for SVD.
5. Find the eigenvalues and eigenvectors of the matrix

$$A = \begin{pmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{pmatrix}.$$

6) Explain the representation of sparse matrices