THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

Advanced Predictive Modeling

Lecture 1 cont: recap

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Types of matrices

- Rectangular: number of rows and columns are different
- Square: number of rows and columns are the same

Types of matrices

- Rectangular: number of rows and columns are different
- Square: number of rows and columns are the same
- Transpose of a matrix is denoted by A^T and $A^T(i,j) = A(j,i)$.

Types of matrices

- Rectangular: number of rows and columns are different
- Square: number of rows and columns are the same
- Transpose of a matrix is denoted by A^T and $A^T(i,j) = A(j,i)$.
- Asymmetric: $A \neq A^T$
- Symmetric: $A = A^T$

Types of matrices

- Rectangular: number of rows and columns are different
- Square: number of rows and columns are the same
- Transpose of a matrix is denoted by A^T and $A^T(i,j) = A(j,i)$.
- Asymmetric: $A \neq A^T$
- Symmetric: $A = A^T$
- $A_{ij} \in \mathbb{R}$
- Binary $A_{ij} \in \{0,1\}$

Types of matrices - dense and sparse

- A dense $m \times n$ matrix is one which has $O(mn)$ nonzero elements.
- A sparse $m \times n$ matrix is one which has $o(mn)$ nonzero elements, i.e. most of the elements are zeros.
  - The user-item matrix in recommender systems is sparse
  - Adjacency matrix of Facebook is sparse
  - The term-document matrix is sparse

Types of matrices - dense and sparse

- If you know your matrix is sparse, you should use data structures for sparse matrices.
- This way, you just need to store the (row, col, val) triplets for the nonzero elements.

Types of matrices - dense and sparse

- If you know your matrix is sparse, you should use data structures for sparse matrices.
- This way, you just need to store the (row, col, val) triplets for the nonzero elements.
- Say I gave you a map which shows different paths between N locations in a city.
- I asked you to compute all pairs of shortest paths between these locations.
- Will this $N \times N$ matrix be sparse, or dense?

Types of matrices - dense and sparse

- Now, say I told you to "cap" the distances at some number. So all distances larger than X is capped at X
- Will this be sparse, or dense? Will you need $O(N^2)$ storage?
- Lets pause for 5 minutes.

Types of matrices - dense and sparse

- Now, say I told you to “cap” the distances at some number. So all distances larger than X are set to X
- Well, technically, sure, it will be a dense matrix.
- But can you still store it as a sparse matrix?

Types of matrices - dense and sparse

- Now, say I told you to "cap" the distances at some number. So all distances larger than X are set to X
- Well, technically, sure, it will be a dense matrix.
- But can you still store it as a sparse matrix?
- Yes, all you have to do is store the $location, location, distance$ triplets for values less than X, and then store X because every other pair has value X.

Types of matrices - dense and sparse

- ```
  import time
  import numpy as np
  ```
- `A1 = np.random.rand(5000,5000)`
  ```

Types of matrices - dense and sparse

- ```
  import time
  import numpy as np
  ```
- ```
  A1 = np.random.rand(5000,5000)
  ```
- ```
  start = time.time()
  np.sum(A1)
  end = time.time()
  print(end - start)
  ```
  ```

Types of matrices - dense and sparse

- ```
  import time
  import numpy as np
  ```
- ```
  A1 = np.random.rand(5000,5000)
  ```
- ```
  start = time.time()
  np.sum(A1)
  end = time.time()
  print(end - start)
  ```
- 0.02
  ```

Types of matrices - dense and sparse

- ```
  import time
  import numpy as np
  ```
- `A1 = np.random.rand(5000,5000)`
- ```
  start = time.time()
  np.sum(A1)
  end = time.time()
  print(end - start)
  ```
- `0.02`
- Now we will make a sparse matrix by using scipy.sparse.csr_matrix.

Types of matrices - dense and sparse

- `A2=A1`
- `low_values_flags = A1 >.1`
  `A2[low_values_flags] = 0`
- ```
  start = time.time()
  np.sum(A2)
  end = time.time()
  print(end - start)
  ```
- 0.02
  ```

Types of matrices - dense and sparse

- Now we will make a sparse matrix by using scipy.sparse.csr_matrix.
- `from scipy.sparse import csr_matrix`
- `A2sparse = csr_matrix(A2)`
- ```
  start = time.time()
  np.sum(A2sparse)
  end = time.time()
  print(end - start)
  ```
- `0.005`
  ```

Eigenvalues and eigenvectors

- A pair (λ, v) is called the eigenvalue, eigenvector pair of a *square* matrix A if the following holds:

$$Av = \lambda v$$

- For a symmetric matrix, all eigenvalues are real.
- A real symmetric matrix can be written as:

$$A = USU^T,$$

where columns of U are linearly independent eigenvectors, and S is a diagonal matrix of eigenvalues.

Singular values and singular vectors

- For real $m \times n$ matrices, we can always do a singular value decomposition.

$$A = U\Sigma V^T$$

  - Σ is a diagonal $r \times r$ matrix with the non-zero singular values on the diagonal
  - Where U is $m \times r$ matrix and V is a $n \times r$ matrix
  - $U^T U = I_r$ and $V^T V = I_r$
  - The rank is given by $r \leq \min(m, n)$

Useful notations and definitions

- Trace of a square matrix is $\mathrm{trace}(A) = \sum_i A_{ii}$

Useful notations and definitions

- Trace of a square matrix is $\text{trace}(A) = \sum_i A_{ii}$
- Define the inner product between two $m \times n$ matrices A and B as

$$\langle A, B \rangle = \text{trace}(A^T B) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij}$$

Useful notations and definitions

- Trace of a square matrix is $\text{trace}(A) = \sum_i A_{ii}$
- Define the inner product between two $m \times n$ matrices A and B as

$$\langle A, B \rangle = \text{trace}(A^T B) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij}$$

- The Frobenius norm is:

$$\|A\|_F^2 = \sum_{ij} A_{ij}^2 = \langle A, A \rangle = \text{trace}(A^T A)$$

Useful notations and definitions

- The operator norm of a matrix A ($\|A\|_{\mathrm{op}}$) is its largest singular value.
- The nuclear norm is given by:

$$\|A\|_* = \sum_i \sigma_i$$

where σ_i are the singular values

Useful notations and definitions

- The operator norm of a matrix A ($\|A\|_{\text{op}}$) is its largest singular value.
- The nuclear norm is given by:

$$\|A\|_* = \sum_i \sigma_i$$

  where σ_i are the singular values
- The Frobenius norm is the ℓ_2 norm of the singular values.
- The operator norm is the ℓ_∞ norm of the singular values.
- The nuclear norm is the ℓ_1 norm of the singular values.