Diney Anand 2022ad05030
Dibyajyoti Sarkar 2022ad05005
Medha Jain 2022ad05083
Shiffin N 2022ad05055

Q-4 As a part of computer application, a sub-routine needs to be written whose input parameter p has to be used in the computation A^p where A is 100×100 symmetric, positive definite matrix A. Note that A is a fixed matrix and it is only p which is the input parameter. What is the most efficient way you can come up with to perform the required computation, if the sub-routine is called millions of times for arbitrary value of p? Your solution needs to be efficient in terms of both time & space taken by the algorithm?

Answers:

There are multiple ways to perform this tasks such as:

* Eigen decomposition
* Spectral theorem
* Conjugate Gradient Method

We will perform the required computation efficiently with conjugate gradient Method.

The conjugate gradient method is an iterative algorithm that can be used to solve systems of linear equations, such as

$Ax = b$ where A is a symmetric positive definite matrix.

The conjugate gradient method has a time complexity of $O(n^2)$ where A is a symmetric matrix and n is the size of the matrix and it requires only $O(n)$ space, making it very efficient solution for this problem.

To use the conjugate gradient matrix method, we will first need to initialize a vector x with an initial guess for the solution

Divey Anand 2022aa05030 | Dibyajyoti 2022aa05005 | Medha Jain 2022aa05083 | Shifbin N 2022aa05055

and set a residual vector $r = b - Ax$, Then, you can iterate the following steps until the residual vector r is sufficiently small:

Steps

1. Choose a search direction p based on the residual vector r.
2. perform a line search along the direction p to find the step size alpha that minimizes the residual vectors.
3. update the solution vector x by $x = x + \text{alpha} * p$
4. Update the residual vector r by $r = r - \text{alpha} * Ap$.

The conjugate gradient method can be improved & implemented in a subroutine that takes the input vector p as an argument and returns the solution vector x. This subroutine can be called multiple times with different values of p to perform the required computation efficiently.

Now using the Diagonal matrix D we can easily find

$$Ap = P D_p P^T$$

In order to find the matrix D, we have to find the $\lambda_1 \dots \lambda_n$ eigen values of matrix A by computing the determinant of Matrix $(A - \lambda I)$.

Then we can write as

$$\begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ & \ddots & & & & \\ 0 & 0 & 0 & 0 & 0 & \lambda_n \end{bmatrix}$$

Once the eigen values are found, we can find matrix P by finding the corresponding $p_1 \dots p_n$ eigen vectors of matrix A. This can be done by solving the linear system of equations

$$(A - \lambda_i I) p_i = 0 \quad \text{where } n \leq i \leq 1$$