23MAT204

Adithya Nair

August 27, 2024

Contents

Part I

Mathematics

Chapter 1

Revision Linear Algebra

Every matrix satisfies its own characteristic matrix.

What this means is

$$\begin{aligned} \lambda^3 - 2\lambda^2 + \lambda - 4 &= 0 \\ A^3 - 2A^2 + A - 4I &= 0 \\ A^2 - 2A + I - 4A^{-1} &= 0 \\ A^{-1} &= \frac{1}{4}[A^2 - 2A + I] \\ X = A^{-1}b &= \frac{1}{4}[A^2 - 2A + I]b \end{aligned}$$

There are some implications behind this. Here's a problem, generate a random 3x3 matrix, find the ranks of $(A - \lambda_1 I)$ and then $(A - \lambda_1 I)(A - \lambda_2 I$

What you will see is that the rank reduces upon multiplying roots of the characteristic equation

1.1 Ways To Calculate Whether A Matrix Is Positive Definite

1. The minors are positive then negative alternating.
2. The eigenvalues are positive.

1.2 Large Eigenvalue Computation

Large eigenvalues are computed by using a matrix multiplication method. This method involves:

1.3 Specral Decomposition

$$S = Q\Lambda Q^T = \lambda_1 q_1 q_1^T + \lambda_2 q_2 q_2^T + \cdots + \lambda_n q_n q_n^T$$

This allows us to represent a matrix by the sum of n rank 1 matrices. This is used in square symmetric matrices.

This is the basis behind Principal Component Analysis.

1.4 Singular Value Decomposition

For rectangular matrices.

$$A = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

This is mainly used for getting useful properties about the matrix without needing to perform significant operations on this matrix, such as the orthogonality.

Singular values are the eigenvalues of a matrix which are **non-zero**

So a rectangular matrix $m \times n$ would have AA^T which is $m \times m$ and A^TA which is $n \times n$. The number of singular values is whether m or n is lower.

The trace of A^TA equal to the sum of all a_{ij}^2 The trace is the sum of all eigenvalues of A^TA, and For $A_{m \times n}$ that's the sum of the eigenvalues squared.

This is known as the **Frobenius Norm**

This method can also be used to generate orthonormal bases for the 4 Fundamental Subspaces

1.5 The Geometry of SVD

Since we have an orthogonal matrix, diagonal and orthogonal matrix... It's a rotation step followed by a stretching step, finally ended by another rotation step.

1.6 Polar Decomposition

This decomposition is a special case of the Singular Value Decomposition. Where you decompose the elements into the polar form with an orthogonal matrix and a positive semi-definite matrix.

For 2D,

$$\vec{x} = r(\cos(\theta) + sin(\theta))$$

$$\begin{aligned} A &= U\Sigma V^T \\ &= U(V^TV)\Sigma V^T \\ &= (UV^T)(V\Sigma V^T) \\ &= Q \times S \end{aligned}$$

Here S is the scaling matrix, and Q is the orthogonal matrix.

Although my assumptions might be wrong, this might be a way to generate rotation matrices for higher-dimension spaces. Might be useful in exploring the semantic spaces of LLMs.

1.7 Principal Component Analysis

Why do we use normalization while working with data? We use normalization to ensure that the units are eliminated while working with that given data. **This normalization is done by subtracting by the average and divide by the standard deviation.**

Matrix where the row and columns sums are equal.

1.8 Norms Of Vectors And Matrices

Norms are a way to measure the size of a vector/matrix. The norm of a vector which calculates the magnitude is known as the L^2 norm or the Euclidean norm $\|v\|_2$. This number gives us the length of the vector.

In general, the $\|u\|_p$ or the p-norm of a vector is $[|u_1|^p + |u_2|^p + \cdots + |u_n|^p]^{\frac{1}{p}}$.

Now, the L_1 norm would be the sum of the components of the vector. This is the sum of the projections to each corresponding axis.

The L_3 norm would be $[|u_1|^3 + |u_2|^3 + \cdots + |u_n|^3]^{\frac{1}{3}}$

The L_∞ norm would be $[|u_1|^\infty + |u_2|^\infty + \cdots + |u_n|^\infty]^{\frac{1}{\infty}}$. This returns the maximum vector component of the vector.

$$x + y\& = 1$$

$$x - y\& = 1$$

Let's take the equation, $\|v\|_1 = 1$ \[

x + y &= 1

-x - y = 1
\]

`./figures/l1eq1`

The S-norm of a vector $\vec{x}$ is $\vec{x}^T S \vec{x}$. When S is a symmetric positive definite matrix, this S-norm is known as the energy of vector $\vec{v}$

There are three types of Matrix norms:

1. Spectral Norm
2. Frobenius Norm $= \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}$
3. Nuclear Norm

1. Spectral Norm We know that the vector norm for a vector $\vec{x}$ is nothing but $\vec{x}^T \vec{x}$. We take this property.

$$\begin{aligned} Max\|A\|_2^2 &= Max\frac{\|Ax\|_2^2}{\|x\|_2^2} \\ &= Max\frac{x^T A^T A x}{x^T x} \\ &= Max\{\lambda_i(S)\} = \lambda_1 = \sigma_1^2 \end{aligned}$$

2. Frobenius Norm The Frobenius norm for a matrix M, $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is the equation $\sqrt{a_{11}^2 + a_{12}^2 + a_{21}^2 + a_{22}^2}$

3. Nuclear Norm The nuclear norm is the sum of the singular values of a matrix A.

   For an identity matrix,

   - The Spectral Norm is 1
   - The Frobenius Norm is $\sqrt{n}$
   - The Nuclear Norm is n

   For an orthogonal matrix,

   - The Spectral Norm is 1
   - The Frobenius Norm is $\sqrt{n}$
   - The Nuclear Norm is n

1.9 Best Low Rank Matrix

We say that a matrix is the best approximation of another matrix, based on the Frobenius Norm.. For a singular value decomposition $A = U\Sigma V^T$, if we assume that the singular values are arranged in descending order… We can select the singular value range where the values are significant contributors to the final matrix.

We can then reduce the size of U, Σ and V^T into a smaller matrix B, based on the number of singular values chosen which would give the best approximation.

Let $A = U\Sigma V^T$ where $\Sigma : \sigma_1 \geq \sigma_2 \geq \cdots \sigma_n$, then B = $U_{m\times m}\Sigma V^T n \times n$ is a best rank-k approx. to A. Where, S is a diagonal matrix of $n \times n$ where $s_i = \sigma_i(i = 1 \cdots k)$ else $s_i = 0$, by best B is a solution to $min_B\|A - B\|F$ where rank(B) = k

Chapter 2

Multi variable Optimization

We minimize $f(x_1, x_2, \cdots, x_3)$.

2.1 Contour Curves

2.2 Multi variable Calculus

The points where $\nabla f = \vec{0}$ are called the stationary points. The Hessian matrix should be positive definite for a minima and negative definite for a maxima. A point where there is no change, is known as a saddle point.

$$
\begin{aligned}
f(x,y) = x^3 + y^3 + 2x^2 + 4y^2 + 6 \\
\frac{\partial f}{\partial x} = 3x^2 + 4x \\
\frac{\partial^2 f}{\partial x} = 6x + 4 \\
\frac{\partial f}{\partial y} = 3y^2 + 8y \\
\frac{\partial f}{\partial y \partial x} = 0 \\
\frac{\partial^2 f}{\partial y} = 6y + 8 \\
x = 0, \frac{-4}{3} \\
y = 0, \frac{-8}{3}
\end{aligned}
$$

2.3 Newton's Method

Newton's Method is a numerical method to find a minimum of a function.

Iterative formula:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

Find the minimum of $f(x) = x^2 + \frac{54}{x}$ using Newton's method.

Newton's method can be practically done for all functions by evaluating the first and second derivatives numerically and then apply the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{to find root of } f(x) = 0$$

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$to find minimum of $f(x)$(root of $f'(x) = 0$)to find minimum of $f(x)$(root of $f'(x) = 0$)to find minim

Find an objective function whose solution is specified as $\vec{x} : A\vec{x} = b$

Take for example, $\frac{1}{2}x^T Ax - b^T x + c$, $x \in R^n, A = A^T$

2.4 Numerical Algorithm Of Gauss-Jacobi Method

Input A = $[a_{ij}]$, X0 = $x^{(0)}$, tolerance TOL, maximum number of iterations,

$Ax = b$

We separate A, A = D - L - U

$$(D - L - U)x = b$$

$$(Dx = b + (L + U)x$$

$$(x = D^{-1}b + D^{-1}(L + U)x$$

$$x^{(k)} = Tx^{x^{(k-1)}} + c$$

The code to implement the method:

```
A = [5,-2,3;-3,9,1;2,-1,-7];
b = [-1;2;3];

n = 100; % No. of iterations
D = diag(diag(A));
L = -tril(A,-1);
U = -triu(A,1);

X = zeros(size(A)(1),n);
T = inv(D)*(L+U);
c = inv(D)*b;

for i = 2:n
  X(:,i) = T*X(:,i-1) + c;
  if(X(:,i) == X(:,i-1))
    i-1
    break;
  end
end
```

2.5 Gauss-Siedel Iteration Method

The iterative method is somewhat similar to <Gauss-Jacobi method>.

The only difference is that the new values are computed by already existing new values.

```
A = [5,-2,3;-3,9,1;2,-1,-7];
b = [-1;2;3];

n = 100; % No. of iterations
D = diag(diag(A));
L = -tril(A,-1);
U = -triu(A,1);

X = zeros(size(A)(1),n);
T = inv(D-L)*U;
c = inv(D-L)*b;

for i = 2:n
  X(:,i) = T*X(:,i-1) + c;
  if(X(:,i) == X(:,i-1))
    display("The iteration converges at:")
    i-1
    break;
  end
end
```

$$(D-L)x^k = Ux^{k-1} + b$$
$$x^k = (D-L)^{-1}\vec{b} + (D-L)^{-1}(U\vec{x})$$

If the matrix is 'diagonally dominant', where the magnitude of the diagonal elements should be greater than the sum of the magnitude of the other elements(absolute value) in the same row.

2.6 Unidirectional/Line search

For a given function,

Find the minima of $f(x) = (x-1)^2 + (y-2)^2$ starting from $(0,0)$ along the direction of x-axis.

How to perform uni-directional search:

- Write the parametric representation of the line search,

$$\vec{s}(t) = \vec{a} + t\vec{b}$$

- Write the function in terms of t, $f(\vec{s}(t))$. this is a single variable function.
- Find the minimum of $f(\vec{s}(t))$, t^*
- Obtain the solution for unidirectional search by substituting t^* in $\vec{s}(t)$

  Find the minimum of the function $f(x) = (x-2)^2 - y$, starting from the point $(-1,0)$ along $(1,1)^T$

2.7 Directions Of Change

A direction given by the vector \$x$^{(k)}$} is a descent direction only if the function value decreases along that direction from the point, $x^{(k)}$, when

$$\nabla f(x^{(k)} \cdot d < 0$$

A direction given by the vector \$x$^{(k)}$} is an ascent direction only if the function value increases along that direction from the point, $x^{(k)}$, when

$$\nabla f(x^{(k)} \cdot d > 0$$

Question: Check whether the given function have a descent direction $x = (2, -1)$ along the given directions d_1 and d_2

$$f = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_2^4$$

2.8 Directions Of Descent/Ascent In Numerical Algorithms.

In any numerical algorithm to find the optimum of an unconstrained problem, the iterative formula used is:

$$x^{k+1} = x^k + \alpha^k d^k$$

Where α^k is the step length in step k and d^k is the direction of descent in step k, if it's a minimization problem or a direction of ascent if it's a maximization problem.

Newton's Method: $x_{k+1} = x_k - (H(x_k)^{-1}\nabla f(x_k))$ Gradient DEscent - $x_{k+1} = x_k + \alpha_k d_k$ where $d_k = -\nabla f(x_k)$ and $\alpha =$ using line search

A direction along which the function increases rapidly from a point is given by the gradient of the function at that point

2.9 Steepest Descent

A direction along which the function increases rapidly from a point is given by the gradient of the function at that point

A direction along which the function decreases rapidly from a point is given by the negative of the gradient of the function at that point

The iterative formula of method of steepest descent is:

$$x^{k+1} = x^k + \alpha^k d^k$$

The descent direction d^k is along the steepest descent direction, $d^k = -\nabla f(x^k$. The step length α^k is obtained by performing a unidirectional search from x^k along the direction d^k, by minimizing,

This method produces successive directions that are perpendicular to each other.

Near the minima, during line search the convergence is very slow.

2.10 Numerical Method

To implement this programmatically, we can write... For $Ax = b$, we can convert this to an equivalent optimization problem.

$$f(x) = \frac{1}{2}x^T Ax + -b^T x + c$$

We have a linear system of equations, which we can be solved by $Ax = b$

We can write $g_0 = Ax_0 - b$

2.11 Conjugate Gradient Descent

The problem with normal gradient descent is that there is no way to predict when the function converges. There's a method that can reliably find the convergence in a predictable fashion. That's the conjugate gradient method.

First we must understnand the conjugate direction and the terms associated with them

1. Krylov Subspace Let $A \in R^{n\times n}, b \in R^n$, the Krylov subspace $K_j(A,b)$ is defined as $K_j(A,b) = span\{\vec{b}, \vec{A}b, A^2b, A^3b, \ldots, A^{j-1}b\}$. Thus $K_j(A,b)$ is subspace of $\mathbb{R}^n$
2. Krylov Matrix The Krylov Matrix is just the matrix consisting of the basis vectors as the column vectors. The Krylov subspace is the column space of such a matrix.
3. Motivation For Krylov Subspaces We know the Cayley Hamilton theorem, and its use in calculating the inverse from the characteristic equation

   Interestingly, the calculation for the inverse leads to a linear combination of the basis vectors of the Krylov subspace when solving linear systems.
4. Conjugate Directions For A real symmetric matrix $n \times n$ with rank n

   The directions $d_0 d_1 \ldots d_{n-1}$ are said to be A-conjugate if,

   $$d_i A d_j = 0$$

   for $i \neq j$

   It's a new definition for orthogonality(not orthonormal)
5. Numerical Method Input A, b, x^0

   Compute $\vec{r_0} = b - Ax^0$

```
A = [1 2 3;2 3 4;3 4 5];

b = [6 9 12]';
x = randi([-9,9], length(b),1);
r = b - A*x;
d = r;
alpha = (r'*r)/(d'*(A*d));
beta = 0;

for i = 1:size(A,1)
  x = x + alpha*d;
  rnew = r - alpha*(A*d);
  beta = (rnew'*rnew)/(r'*r);
  d = rnew + beta*d;
  r = rnew;
  alpha = (r'*r)/(d'*(A*d));
  if sqrt(rnew) < 1e-10
    break;
    end
end

x
r
```