AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

Unit 1 - Sem 4 - 22MAT230

Mathematics for Computing 4

Dr Sunil Kumar S and Prof K P Soman

School of Artificial Intelligence

Amrita Vishwa Vidyapeetham

If you find any mistakes or have any comments to share,

I would be grateful to receive them at s_sunilkumar@cb.amrita.edu

https://github.com/mfcpjt/MFC4_22MAT230

```
clearvars
clear all
ready = true;
PUBLISH = ready;
```

Euclidean Distance Matrix

Consider n points from R^m, ie, the data points $X \in R^{m\times n}$

The squared Euclidean Distance Matrix, D is defined as

$$d_{ij}^2 = \|X_i - X_j\|_2^2$$

$$= (X_i - X_j)^T(X_i - X_j)$$

$$= X_i^T X_j - 2X_i^T X_j + X_j^T X_j$$

This can be written as the matrix equation

$$D = \underbrace{\mathbf{1}\cdot diag(G)}_{1} - 2\underbrace{G}_{m} + \underbrace{diag(G)\cdot \mathbf{1}^T}_{1}$$

where $G = X^T X$ and $\mathbf{1}$ is the column vector of n $1'$ s.

The first and the last matrices are rank 1 matrices and the second matrix has rank of m!. (Verify)

Rank of D is at max $m + 2$, ie, rank(D) $\leq m + 2$

Centroid

Centroid of n points from R^m, $X \in R^{m \times n}$

$$C = \frac{1}{n} X \mathbf{1}$$

The points centered about origin are then, $X_c = X - C = X - \frac{1}{n} X \mathbf{1}$

where $\mathbf{1}$ is the vector of n 1's . $\mathbf{1} \in R^n$

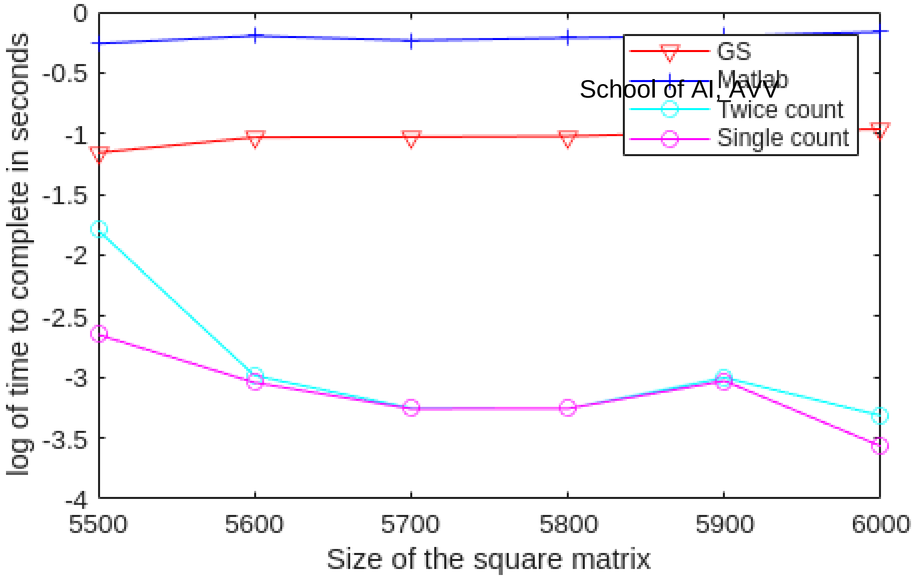
Comparison of distance computation by

1. Prof Gilbert Strang's formula $D = \mathbf{1} \cdot diag(G) - 2G + diag(G) \cdot \mathbf{1}^T$
2. Matlab function **'dist'**
3. Double count - our function - 'full matrix' - **'CreateDistMat'**
4. Single count - our function - 'strictly upper or lower triangular matrix', **'CreateDistMatEcon'**

```
m = 10;
nv = 5500:100:6000;

%Prof Gilbert Strang's method
rng(100)
t = zeros(size(nv));
i = 1;

for n = nv
    X = single(randi([-25,25],m,n));
    tic
    G2 = 2*transpose(X)*X;
    u = ones(n,1);
    T1 = u*(diag(G2))';
    D = T1 - G2 + T1';
    d = sqrt(D);
    t(i) = toc;
    i = i + 1;
end

%Matlab inbuilt function
rng(100)
```

```
t1 = zeros(size(nv));
i = 1;

for n = nv
    X = single(randi([-25,25],m,n));
    tic
    d = dist(X);
    t1(i) = toc;
    i = i + 1;
end

% Count all points twice - Our 1st method
rng(100)
t2 = zeros(size(nv));
i = 1;

for n = nv
    X = single(randi([-25,25],m,n));
    tic
    d = CreateDistMat(X);
    t2(i) = toc;
    i = i + 1;
end

% Count all points once - Our 2nd method
rng(100)
t3 = zeros(size(nv));
i = 1;

for n = nv
    X = single(randi([-25,25],m,n));
    tic
    d = CreateDistMatEcon(X);
    t3(i) = toc;
    i = i + 1;
end
```

```
plot(nv,log10(t),Color="red",Marker="v");hold on
plot(nv,log10(t1),Color="blue",Marker="+");
plot(nv,log10(t2),Color="cyan",Marker="o");
plot(nv,log10(t3),Color="magenta",Marker="o");hold off

xlabel("Size of the square matrix")
ylabel("log of time to complete in seconds")

legend("GS","Matlab","Twice count","Single count")
```



```
function D = CreateDistMat(X)
% Data in Columns
    % X = X';
    [N,~] = size(X);
    D = zeros(N);
    for i = 1:N
        D(i,:) = sqrt(sum((X-X(i,:)).^2,2));
    end
end
```

```
function d = CreateDistMatEcon(X)
    X = X';
    [~,N] = size(X);
    d = zeros(N);
    for i = 1:N-1
        d(i,i+1:N) = sqrt(sum((X(:,i+1:N)-X(:,i)).^2));
    end
    d = d + d';
end
```

```
if(PUBLISH == ready)
    path = '/media/user/DATA4LINUX/new1/Repos/Mine/MFC4_22MAT230/';
    mlxfile = matlab.desktop.editor.getActive().Filename;
    [~, name, ext] = fileparts(mlxfile);
    outfile = [path, name, ext,'.pdf']
    export(matlab.desktop.editor.getActive().Filename, outfile);
    if ispc
        winopen(outfile);
    elseif ismac
        system(['open ' char(outfile)]);
    else
        system("env -u LD_LIBRARY_PATH xdg-open '" + outfile + "' &");
    end
end
```

```
outfile = 
'/media/user/DATA4LINUX/new1/Repos/Mine/MFC4_22MAT230/U1_DistanceMatrix_formula.mlx.pdf'
```