```
function
 [suspicious_index,lof,normal,outliers]=demo_matlab_cours_Outliers_partie3()
```

LOF

```
k = 10;
threshold = 2;

test = load('TestData2_LOF.mat');
[suspicious_index,lof] = LOF(test.data, k);
outliers = test.data(lof>=threshold, :);
normal = test.data(lof<threshold, :);

figure;
hold on;
scatter(normal(:, 1), normal(:, 2), 'bo');
scatter(outliers(:, 1), outliers(:, 2), 'r*');
xlabel('test.data(:,1)');ylabel('test.data(:,2))');title('LOF')

figure;
hold on;grid;
scatter3(test.data(:,1),test.data(:,2),lof);
xlabel('test.data(:,1)');ylabel('test.data(:,2))');zlabel('LOF')
scatter3(outliers(:,1),outliers(:,2),lof(lof>=threshold),'r*');
view(3);
end

function [suspicious_index,lof] = LOF(A, k)
%
% Local Outlier Factor
% Authors: Markus M. Breunig, Hans-Peter Kriegel,
%          Raymond T. Ng, J?rg Sander
% Original paper :
% LOF: Identifying Density-Based Local Outliers
% e-mail : { breunig | kriegel | sander }
%          @dbs.informatik.uni-muenchen.de
%          rng@cs.ubc.ca
% Programmer: Yi-Ren Yeh(yirenyeh@gmail.com)
% modified by: Zi-Wen Gui(evan176@hotmail.com)
%
%
% Inputs
%   A: the data matrix, each row represents an instance
%   k: the number of nearest neighbors, specified as an integer or
%      as a fraction of the total number of data points
%
% Outputs
%   lof: the local outlier factor for each instance
%   suspicious_index: the ranking of instances according to their
%                     suspicious score
%                     For example, suspicious_index(i)=j means the
%                     ith instance is in jth position in the ranking
%

% if k < 1
%     [numrows,~] = size(A);
%     k = round(k*numrows);
% end

%try
    %Find the nearest neighbors by "KDTree" for each elements
    [k_index, k_dist] = knnsearch(A,A,'k',k
+1,'nsmethod','kdtree','IncludeTies',true);
    %Ignore first element(itself) at nearest neighbors
    k_index = cellfun(@(x) x(2:end),k_index,'UniformOutput',false);
    numneigh = cellfun('length',k_index);

    %Get k-distance
    k_dist1 = cell2mat(cellfun(@(x)
 x(end),k_dist,'UniformOutput',false));
    %Get row length of matrix A
    n = length(A(:,1));
    %Initialize lrd_value vector
    lrd_value = zeros(n,1);
    %Calculate lrd for each elements
    for i = 1:n
        lrd_value(i) = lrd(A, i, k_dist1, k_index, numneigh(i));
    end
    %Initialize lof vector
    lof = zeros(n,1);
    %Calculate LOF
    for i = 1:n
        lof(i) = sum(lrd_value(k_index{i})/lrd_value(i))/numneigh(i);
    end
    %Indices from sorting lof are the suspicious score rankings
    [~,suspicious_index]=sort(lof,'descend');

% catch err
%     if (strcmp(err.message, 'Invalid parameter name: IncludeTies.'))
%         warning('MATLAB:LOF', 'Matlab not newest version? Falling
 back to old version.')
%         [suspicious_index lof] = LOF_old(A, k);
%     else
%         rethrow(err)
%     end
%
% end
end
%=========================================================================
function lrd_value = lrd(A, index_p, k_dist,k_index, numneighbors)
%Calculate the reachability distance for nearest neighbors
Temp = repmat(A(index_p,:), numneighbors, 1) - A(k_index{index_p}, :);
Temp = sqrt(sum(Temp.^2,2));
reach_dist = max([Temp k_dist(k_index{index_p})],[],2);
%Calculate the local reachability density for each elements
```

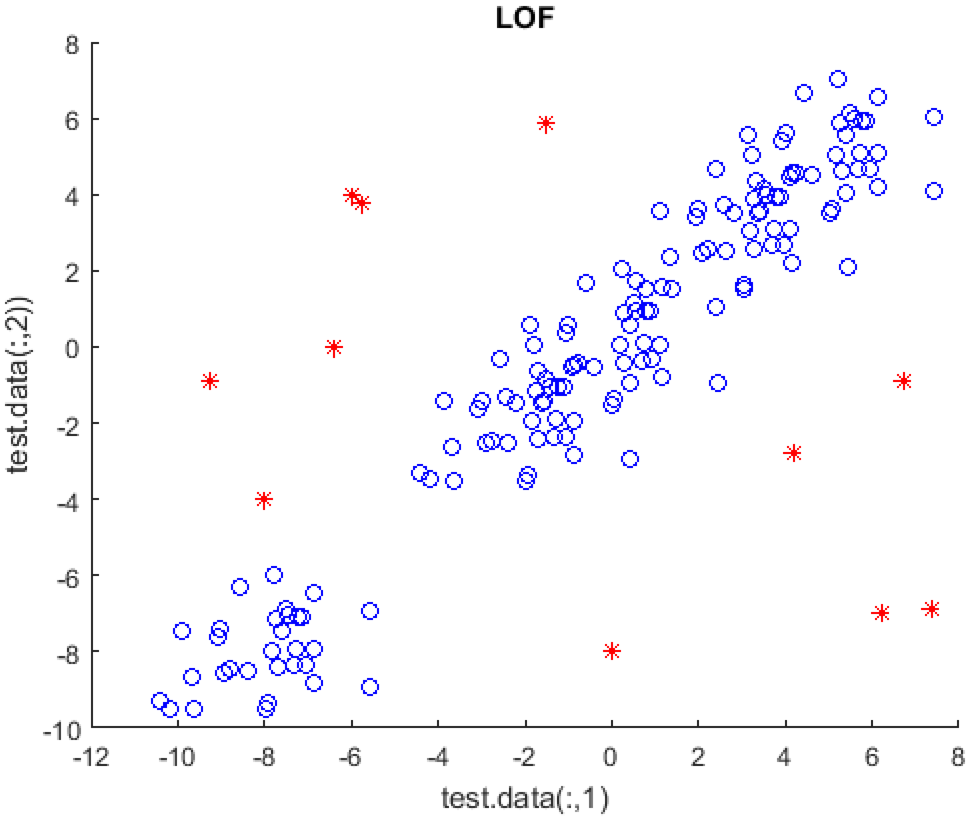
```
lrd_value = numneighbors/sum(reach_dist);
end
```

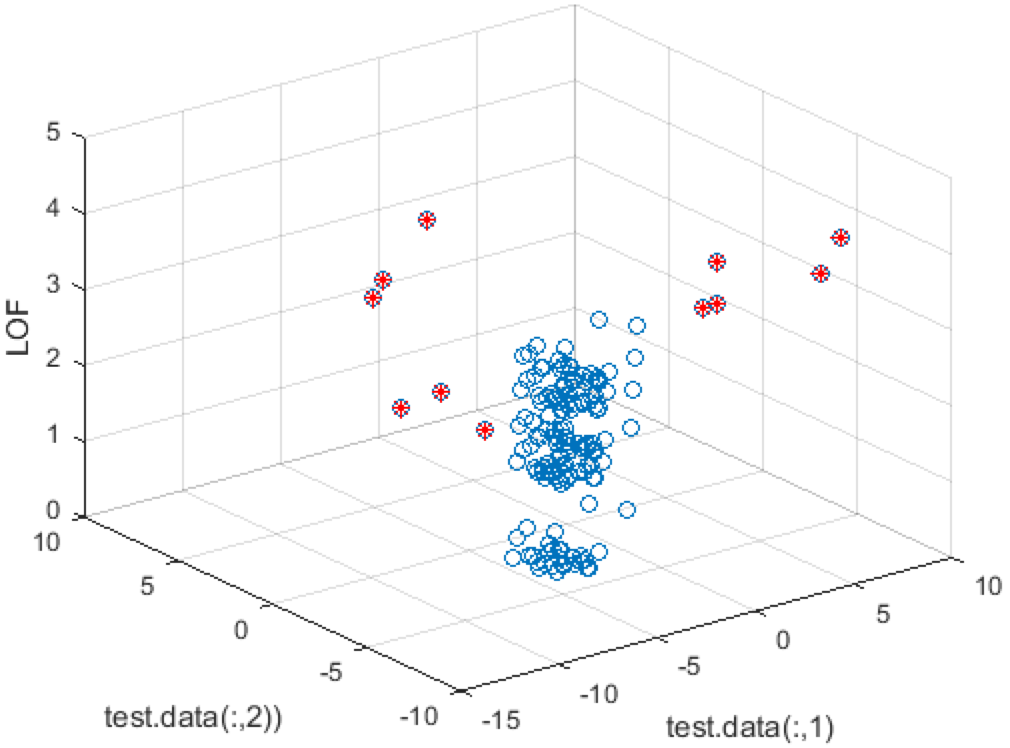
```
ans =

   157
   151
   161
   158
   155
   160
   153
   154
   159
   152
   156
    19
   109
    79
    49
   139
   120
    90
   132
   131
    18
   108
    72
    48
    71
    30
   138
    22
   124
   134
    52
    34
   142
   143
   112
    91
    61
     7
    13
   103
   141
    51
   111
    21
    97
    37
   121
    31
```

```
 82
118
 74
128
125
 39
 64
127
129
  4
  8
 98
 88
 78
 38
 94
 83
  1
 35
 81
 43
 73
136
133
 67
137
116
 46
 53
 68
102
 47
 12
145
 65
 15
105
 45
135
  9
150
 86
 57
 55
147
 77
101
 60
 16
  5
 95
 11
 24
119
```

```
 93
 54
 41
 75
 42
 17
107
 25
144
 58
148
 26
 32
122
110
 89
 44
 28
 36
106
 33
123
126
  6
 76
 63
 27
 80
 56
146
  3
  2
 96
 87
 85
 99
 23
113
 69
117
 92
 66
115
 29
149
 59
114
 62
 10
100
 84
140
 50
 70
 14
104
 40
130
 20
```





Published with MATLAB® R2016a