Elastic Map Demo

There are several files with elastic map demo. All these files are located in the “tests” folder of repository.

This document contains several examples of Elastic maps usage.

Contents

[1 Density with small number of colours 1](#_Toc5787778)

[2 Densities with “continuous” colours 1](#_Toc5787779)

# Density with small number of colours

Used code is presented below

% Colours to draw are defined now

colours = ['b'; 'g'];

% Download database

load('breastCancer.mat');

if size(d1n, 1) > 286

%We need to transpose matrix

d1n = d1n';

end

% Create map wit 30 rows and 30 columns and initialise along Principal

% components

map = rect2DMap(30, 30);

init(map, d1n, 'pci');

% Draw the firs map to see distribution (map hidden by 'lineWidth', 0)

drawMap(map, d1n, 'classes', col, 'markColour', colours, 'lineWidth', 0);

saveFigures('figures/Distribution.png');

% Map fitting for standard parameters

EM(map, d1n, 'stretch', 0, 'bend', 0.1);

% Draw the density by density function

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', 'density', 'flatColoring', true,...

'ColorMap', parula(7));

saveFigures('figures/flatDensity.png');

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', 'density', 'flatColoring', false,...

'ColorMap', parula(7));

saveFigures('figures/3DDensity.png');

% Draw density by vector of ones

col1 = col;

col1(col1 > 0) = 1;

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', true,...

'ColorMap', parula(7));

saveFigures('figures/flatDensity2.png');

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', false,...

'ColorMap', parula(7));

saveFigures('figures/3DDensity2.png');

% Draw pseudo risk map

col1 = col;

col1(col1 == 2) = -1;

col1 = -col1;

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', true, 'ColorMap', parula(7));

saveFigures('figures/flatRisk.png');

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', false,...

'ColorMap', parula(7));

saveFigures('figures/3DRisk.png');

Let us consider arguments of EM colouring function calls.

% Draw the firs map to see distribution (map hidden by 'lineWidth', 0)

drawMap(map, d1n, 'classes', col, 'markColour', colours, 'lineWidth', 0);

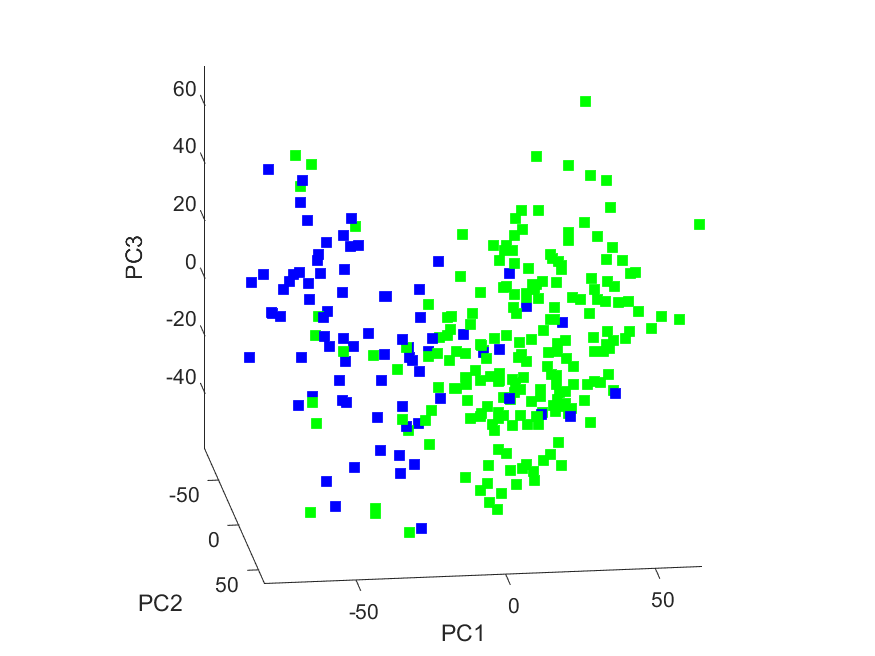
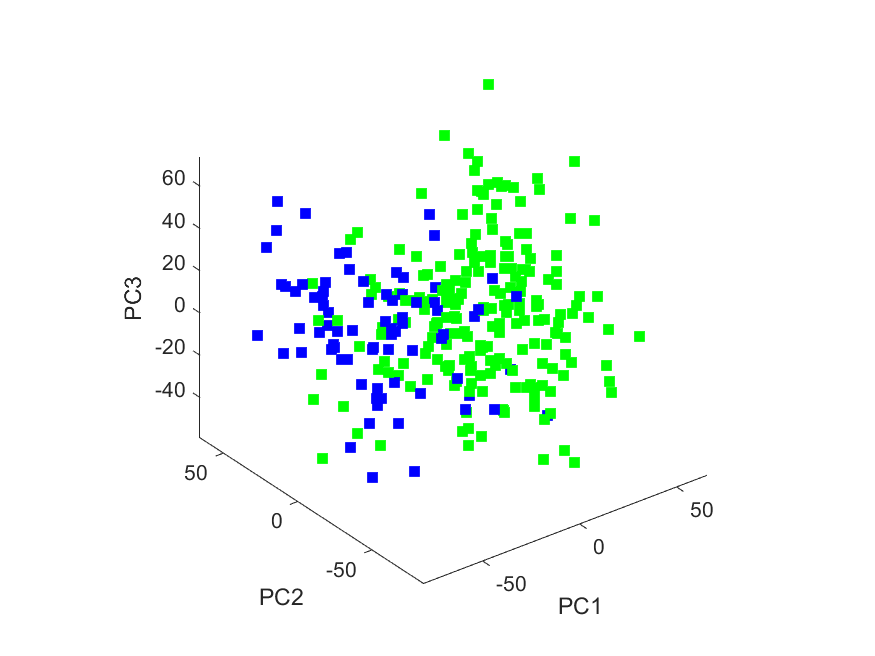
The first map simply presents distribution of data points in the space of the first three principal components. It is not necessary to train map for this. Formed figure ca be rotated by standard Matlab tool (see, for example the right figure below).

Two pairs provide possibility to colour points of different classes by different colours:

Pair 'classes', col defines classes of points;

Pair 'markColour', colours defines colours to use. User also can select different shapes for markers by 'markShape' and size of marker for different classes by 'markSize'.

Pair 'lineWidth', 0 suppress drawing of map.



% Draw the density by density function

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', 'density', 'flatColoring', true,...

'ColorMap', parula(7));

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

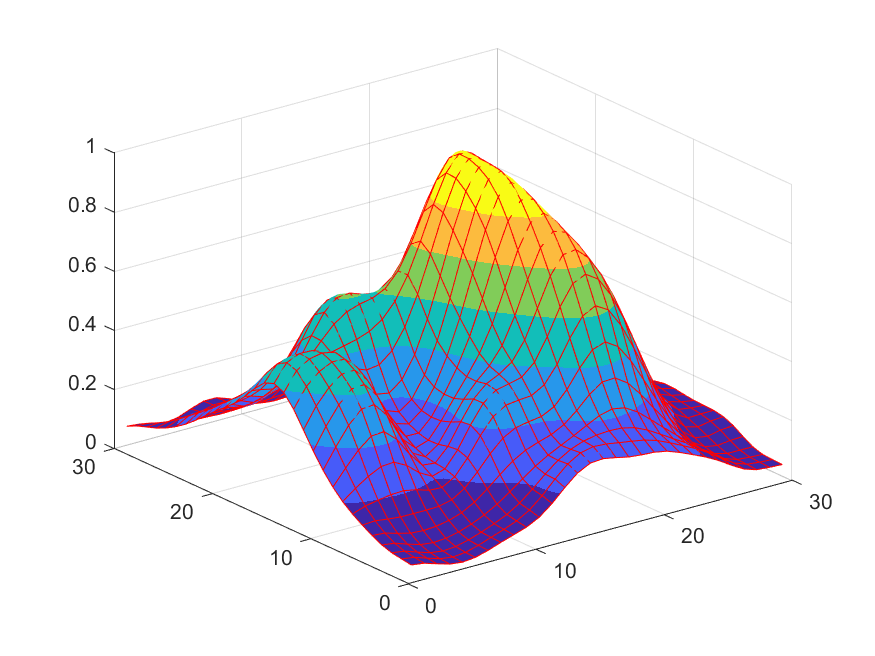
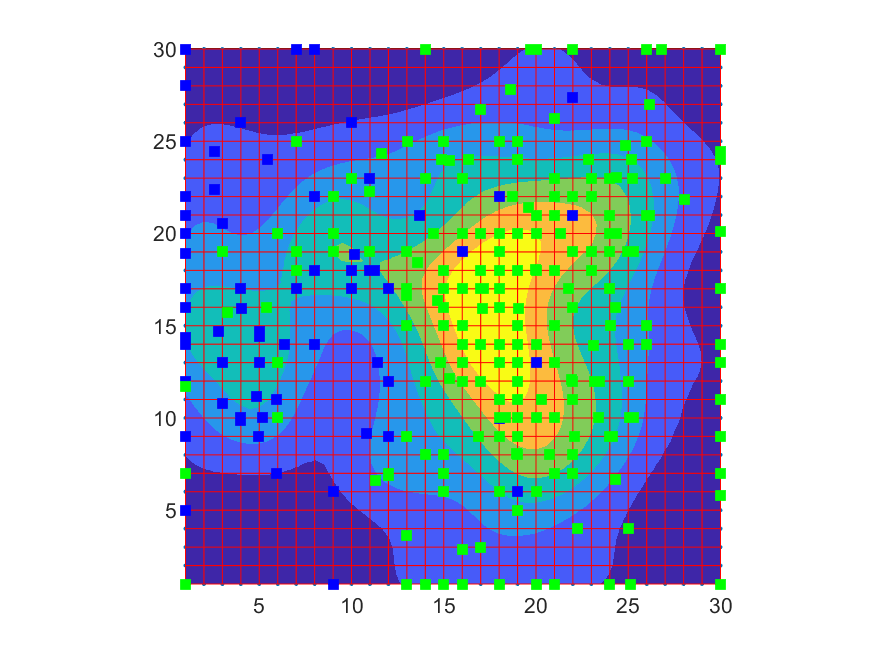
'classes', col, 'markColour', colours, ...

'coloring', 'density', 'flatColoring', false,...

'ColorMap', parula(7));

These commands remove drawing of map nodes ('nodeMarker', 'none'), specified thin lines to draw map ('lineWidth', 0.5), specified classes and colours to depict classes ('classes', col, 'markColour', colours). Both these commands request drawing of density ('coloring', 'density') and usage of reduced colour map wit 7 colours ('ColorMap', parula(7)).

The more important drawing flat ('flatColoring', true) in the first command (left figure below) in comparison with ('flatColoring', false) in the second command (right figure below).



Next fragment shows how to depict arbitrary function defined in data points. Just for simplicity we used constant value 1 which is equivalent to density. First of all we form vector of ones

% Draw density by vector of ones

col1 = col;

col1(col1 > 0) = 1;

Then we use almost the same commands to depict map. The only difference is in argument 'coloring', col1.

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', true, 'ColorMap', parula(7));

drawMapInt(map, d1n, 2, 'nodeMarker', 'none', 'lineWidth', 0.5,...

'classes', col, 'markColour', colours, ...

'coloring', col1, 'flatColoring', false,...

'ColorMap', parula(7));

Graphs are absolutely the same as above.

The next example is pseudo risk map. Let us put value 1 for one class points and -1 for the second class points.

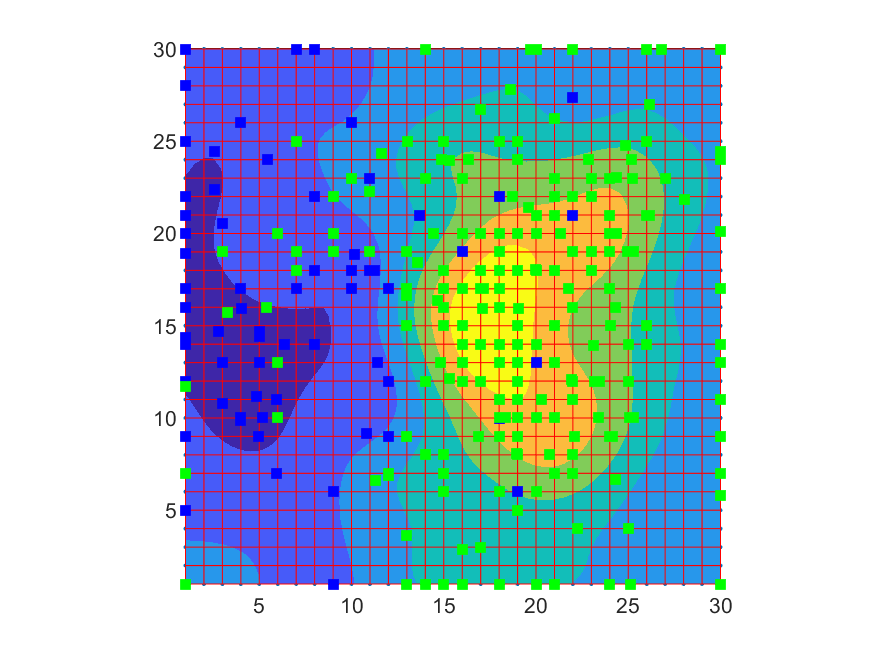
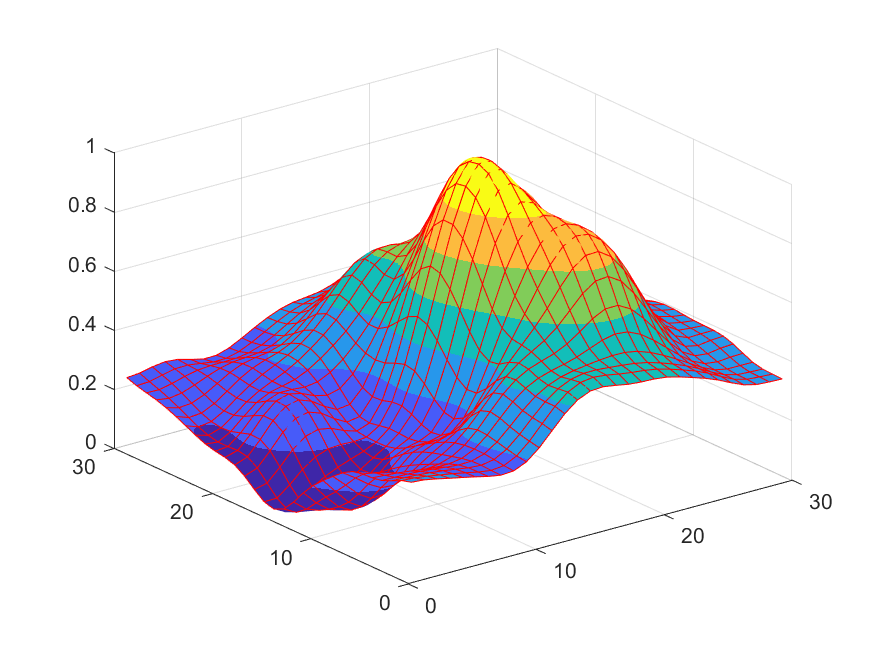
% Draw pseudo risk map

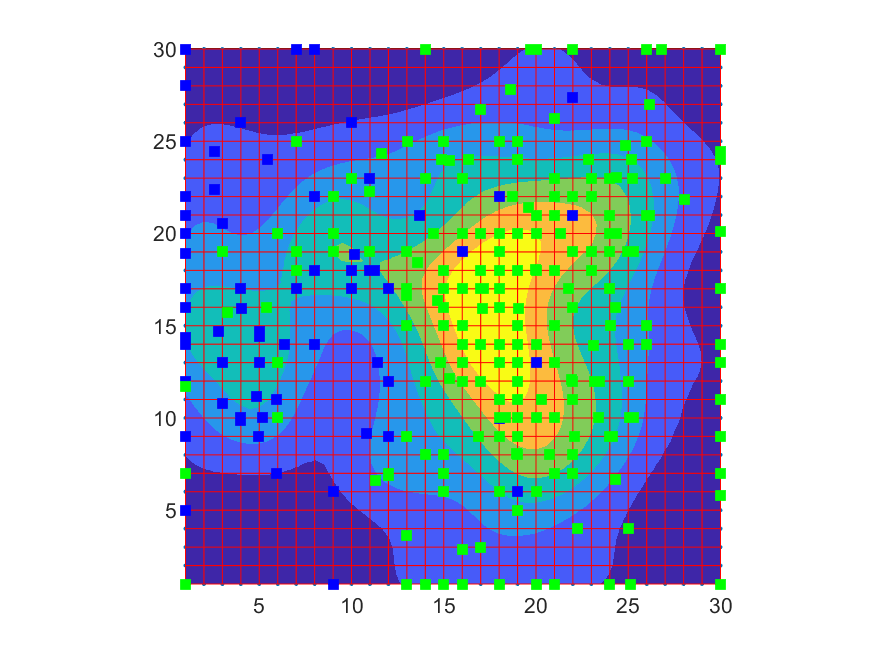
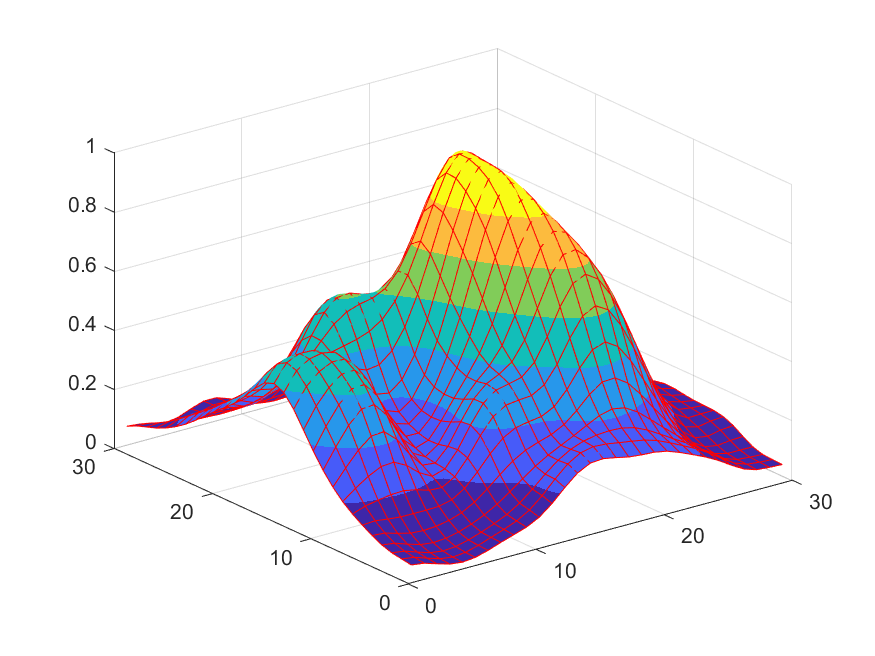
col1 = col;

col1(col1 == 2) = -1;

col1 = -col1;

Then we use completely the same commands to depict map (with pair 'coloring', col1). For simplicity we depict risk maps in the top row and repeat density map in the bottom row. As we can see there is one difference: in the first row dark blue means high concentration of blue points but in the bottom row it means the most empty areas.





# Densities with “continuous” colours

In this case we have one difference only: pair ('ColorMap', parula) instead of pair ('ColorMap', parula(7)).

We consider Risk maps (top row) and density maps (bottom row):

