

```
In [5]: #!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 3 20:06:20 2019

@author: jorgeagr
"""

import numpy as np
from sklearn.mixture import GaussianMixture
from requiredFunctions.doubleMoon import doubleMoon
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick

width = 10
height = 10

mpl.rcParams['figure.figsize'] = (width, height)
mpl.rcParams['font.size'] = 18
mpl.rcParams['figure.titlesize'] = 'small'
mpl.rcParams['legend.fontsize'] = 'small'
mpl.rcParams['xtick.major.size'] = 12
mpl.rcParams['xtick.minor.size'] = 8
mpl.rcParams['xtick.labelsize'] = 18
mpl.rcParams['ytick.major.size'] = 12
mpl.rcParams['ytick.minor.size'] = 8
mpl.rcParams['ytick.labelsize'] = 18

cmap = plt.get_cmap('Paired')
cmap_scatter = mpl.colors.ListedColormap(cmap((1, 3, 5)))
cmap_contour = mpl.colors.ListedColormap(cmap((0, 2)))

N = 1000
r = 1
w = 0.6
d = -0.5

# Part B
data = doubleMoon(N, w, r, d, seed=0)
x_train = data[:, :2]
y_train = data[:, -1]

k_components = [1, 2, 3, 5]
```

```

In [6]: fig1, ax1 = plt.subplots(nrows=2, ncols=2, sharex=True, sharey=True)
fig2, ax2 = plt.subplots(nrows=2, ncols=2, sharex=True, sharey=True)
for i, k in enumerate(k_components):
    gmm_up = GaussianMixture(n_components=k)
    gmm_up.fit(x_train[y_train==0])
    gmm_down = GaussianMixture(n_components=k)
    gmm_down.fit(x_train[y_train==1])

    # Part C
    N_draw = 3000
    x_up, _ = gmm_up.sample(N_draw)
    x_down, _ = gmm_down.sample(N_draw)

    #fig, ax = plt.subplots()
    ax1[i//2][i%2].scatter(x_up[:,0], x_up[:,1], 30, c=[cmap_scatter(0)], marker='+')
    ax1[i//2][i%2].scatter(x_down[:,0], x_down[:,1], 30, c=[cmap_scatter(1)], marker='x')
    ax1[i//2][i%2].set_title(str(k) + ' components')
    ax1[1][i//2].set_xlabel(r'$x_0$')
    ax1[i%2][0].set_ylabel(r'$x_1$')
    ax1[i//2][i%2].set_xlim(-2, 3)
    ax1[i//2][i%2].set_ylim(-2, 2)
    ax1[i//2][i%2].xaxis.set_major_locator(mtick.MultipleLocator(1))
    ax1[i//2][i%2].xaxis.set_minor_locator(mtick.MultipleLocator(0.5))
    ax1[i//2][i%2].yaxis.set_major_locator(mtick.MultipleLocator(1))
    ax1[i//2][i%2].yaxis.set_minor_locator(mtick.MultipleLocator(0.5))

    # Part D
    data_test = doubleMoon(N, w, r, d, seed=100)
    x_test = data[:, :2]
    y_test = data[:, -1]

    scores = np.asarray([gmm_up.score_samples(x_test), gmm_down.score_samples(x_test)]).T
    test_pred = np.argmax(scores, axis=1)
    error = np.abs(test_pred - y_test).mean()
    accuracy = 1 - error

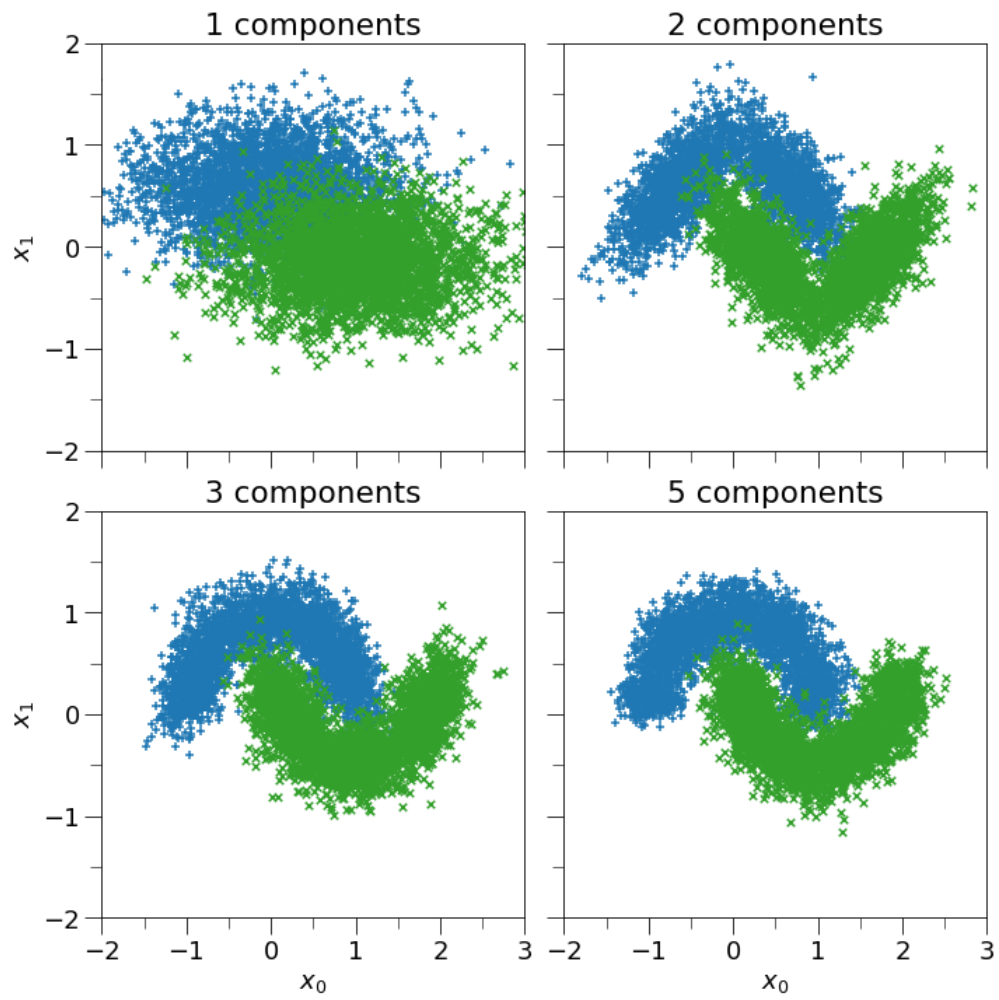
    y_right = np.where(test_pred == y_test)[0]
    y_wrong = np.where(test_pred != y_test)[0]
    blue_ind = y_right[np.where(y_test[y_right]==0)]
    green_ind = y_right[np.where(y_test[y_right]==1)]

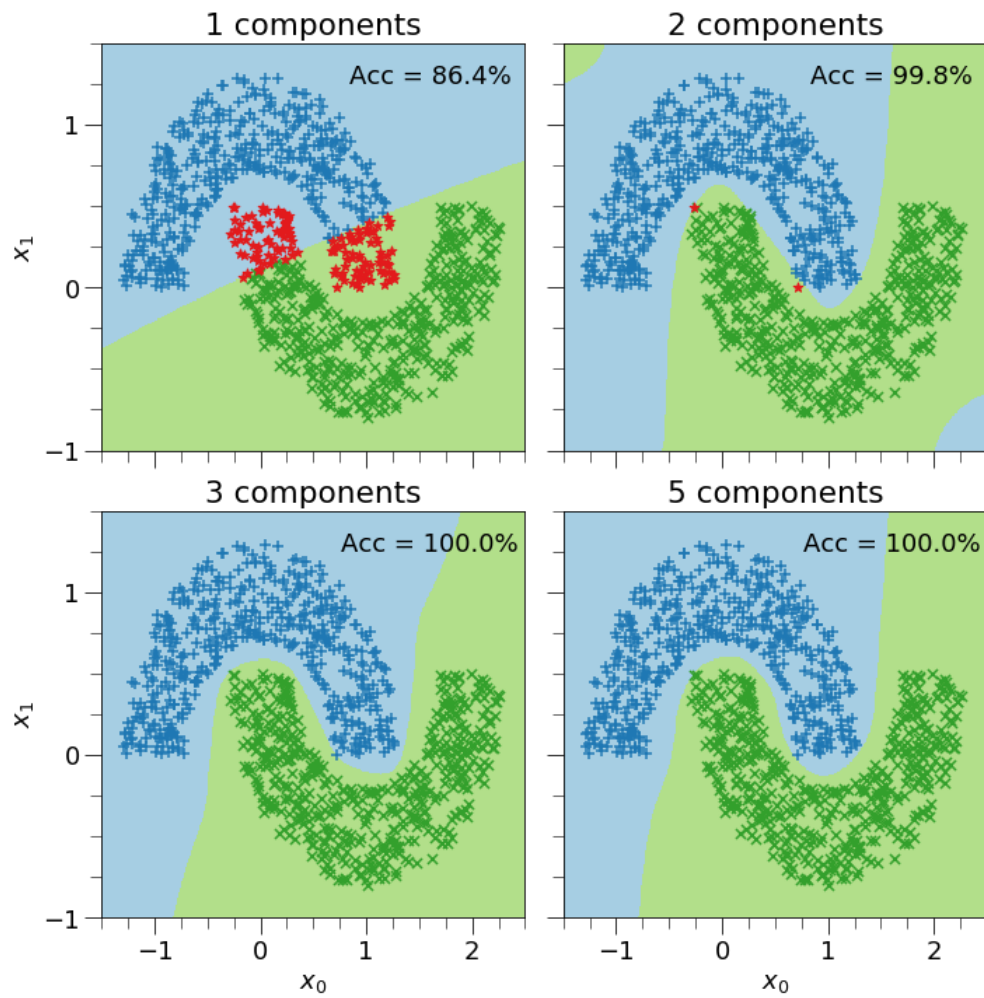
    x0_min, x0_max = -1.5, 2.5
    x1_min, x1_max = -1, 1.5
    xx0, xx1 = np.meshgrid(np.arange(x0_min, x0_max, 0.01),
                           np.arange(x1_min, x1_max, 0.01))
    cc = np.argmax(np.asarray([gmm_up.score_samples(np.c_[xx0.ravel(), xx1.ravel()]),
                              gmm_down.score_samples(np.c_[xx0.ravel(), xx1.ravel()])]).T,
                  axis=1)
    cc = cc.reshape(xx0.shape)

    #fig, ax = plt.subplots()
    ax2[i//2][i%2].contourf(xx0, xx1, cc, cmap=cmap_contour)
    ax2[i//2][i%2].scatter(x_test[:,0][blue_ind], x_test[:,1][blue_ind], 50,
                          c=[cmap_scatter(0)], marker='+')
    ax2[i//2][i%2].scatter(x_test[:,0][green_ind], x_test[:,1][green_ind], 50,
                          c=[cmap_scatter(1)], marker='x')
    ax2[i//2][i%2].scatter(x_test[:,0][y_wrong], x_test[:,1][y_wrong], 50,
                          c=[cmap_scatter(2)], marker='*')
    ax2[i//2][i%2].text(1.60, 1.25, 'Acc = {:.1f}%'.format(accuracy*100),
                      horizontalalignment='center')
    ax2[i//2][i%2].set_xlim(x0_min, x0_max)
    ax2[i//2][i%2].set_ylim(x1_min, x1_max)
    ax2[i//2][i%2].xaxis.set_major_locator(mtick.MultipleLocator(1))
    ax2[i//2][i%2].xaxis.set_minor_locator(mtick.MultipleLocator(0.25))
    ax2[i//2][i%2].yaxis.set_major_locator(mtick.MultipleLocator(1))
    ax2[i//2][i%2].yaxis.set_minor_locator(mtick.MultipleLocator(0.25))
    ax2[1][i//2].set_xlabel(r'$x_0$')
    ax2[i%2][0].set_ylabel(r'$x_1$')
    ax2[i//2][i%2].set_title(str(k) + ' components')

fig1.tight_layout(pad=0.5)
fig2.tight_layout(pad=0.5)
#fig1.savefig('../prob4c.eps', dpi=500)
#fig2.savefig('../prob4d.eps', dpi=500)

```





In []: