Generation of samples from mixture Gaussian distribution; Model-based classification; Data-driven classification using logistic regression

---

ECE 283 Hw1

April 21, 2020

Student: Ivan Arevalo
Email: ifa@ucsb.com

Department of Electrical and Computer Engineering, UCSB

1 Generating 2D synthetic data for binary classification

We'll start by generating mixed Gaussian samples from 2 classes as described in the handout and visualized each one.

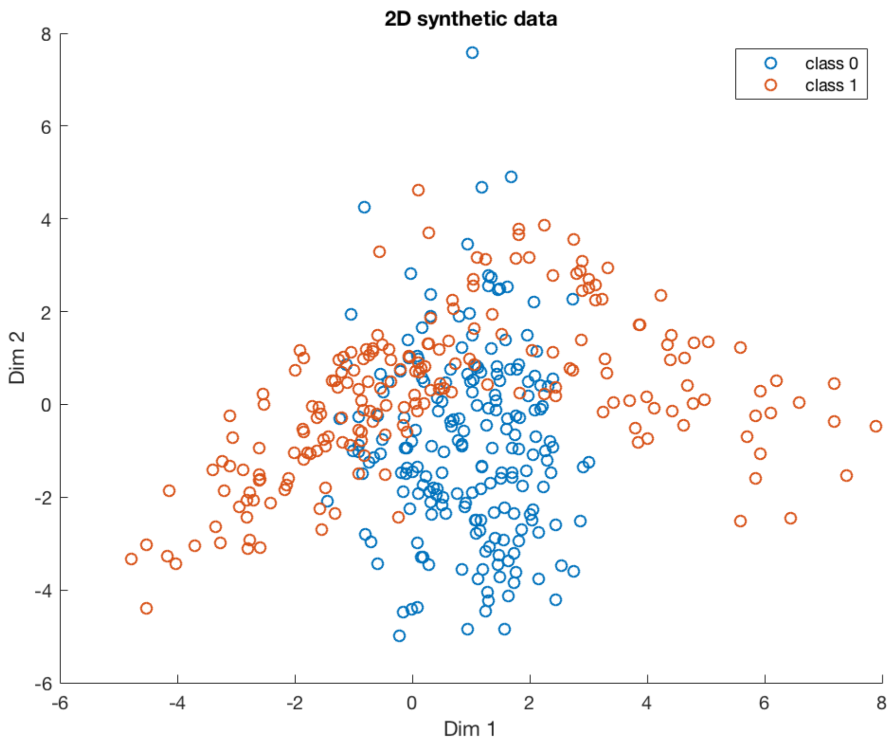

Figure 1: Generated 2D synthetic data

Class 0: Gaussian with mean vector $\mathbf{m} = (1, -1)^T$ and covariance matrix $\mathbf{C}$ with eigenvalue, eigenvector pairs:
$\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = 0$.
Class 1: Gaussian mixture with two components:
Component A: $\pi_A = \frac{2}{3}$, $\mathbf{m}_A = (-1, 0)^T$, $\mathbf{C}_A$ with eigenvalue, eigenvector pairs: $\lambda_1 = 4$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 1/2$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = -\frac{3\pi}{4}$.
Component B: $\pi_B = \frac{1}{3}$, $\mathbf{m}_B = (4, 1)^T$, $\mathbf{C}_B$ with eigenvalue, eigenvector pairs: $\lambda_1 = 1$, $\mathbf{u}_1 = (\cos\theta, \sin\theta)^T$, $\lambda_2 = 4$, $\mathbf{u}_2 = (-\sin\theta, \cos\theta)^T$, with $\theta = \frac{\pi}{4}$.

Figure 2: Characteristics of each class

2 MAP Decision Rule

Assuming equal priors, we'll now implement the MAP decision rule to classify each sample from part 1. Furthermore, we'll find the boundaries and visualize them on the plot.

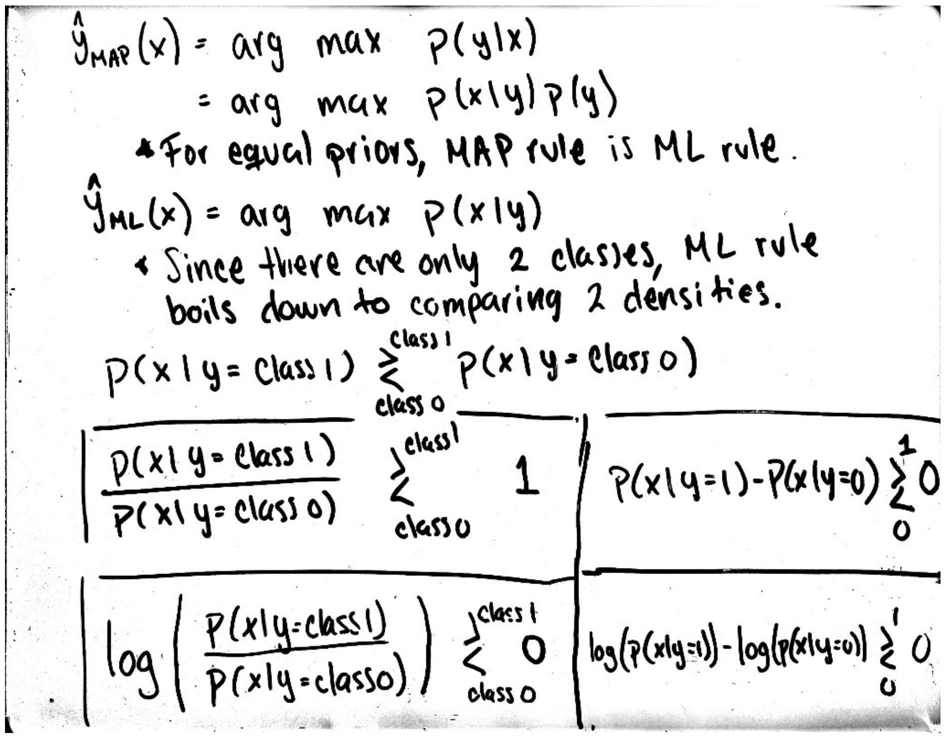

Figure 3: Map Rule derivation with equal priors

As shown in figure 3, the MAP rule boils down to the ML rule for equal priors and we can find the boundaries for this decision rule by finding the points where the probability density of x given y = 1 and 0 are the same.

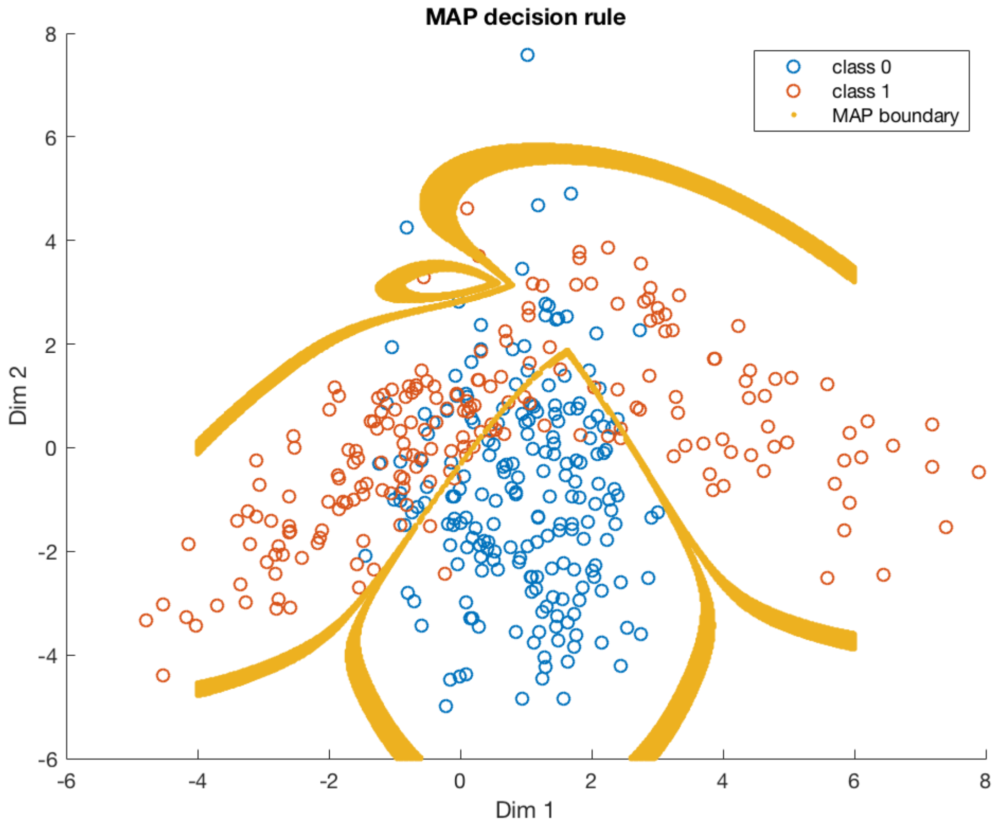

Figure 4: MAP decision rule boundaries

3 Inference performance of MAP rule

Next, we'll estimate the conditional probability of incorrect classification for each class with the MAP decision rule using simulations. We'll generate 2000 samples, 1000 from class 0 and 1000 from class 1. As shown in figure 3, we can find the MAP rule decision by comparing the probability density of x given y = 1 and 0. We can now find the probability of error for each class.

```
prob_error_0 =

    0.2240


prob_error_1 =

    0.0580


prob_error_total =

    0.1410
```

We can see that the probability of error is highest with class 0 at 22.4% followed by class 1 at 5.8%. Finding the probability of error for both classes together, we arrive at a probability of error of 14.1%. Observing figure 4, we can get some intuition why class 0 has such a high probability error given that boundaries favor class 1 whenever samples from each class overlap.

4 Code

```
%% Ivan Arevalo ECE283 Hw1

%% Part 1 2D synthetic data

mean_0 = [1;-1];
U_0 = [1,0;0,1]
lambda_0 = [1,0;0,4]
covariance_0 = U_0 * lambda_0 * U_0.'
size = [200 2]';
class_0 = mvnrnd(mean_0, covariance_0, 200);

priorA = 2/3;
thetaA = -3*pi/4;
mean_1A = [-1;0];
U_1A = [cos(thetaA), -1*sin(thetaA);sin(thetaA), cos(thetaA)]
lambda_1A = [4,0;0,0.5]
covariance_1A = U_1A * lambda_1A * U_1A.'

priorB = 1/3;
thetaB = pi/4;
mean_1B = [4;1];
U_1B = [cos(thetaB), -1*sin(thetaB);sin(thetaB), cos(thetaB)]
lambda_1B = [1,0;0,4]
covariance_1B = U_1B * lambda_1B * U_1B.'

class_1 = zeros(200,2);
for i=1:200
    prior = rand;
    if prior > priorB
        class_1(i, :) = mvnrnd(mean_1A, covariance_1A, 1);
    else
        class_1(i, :) = mvnrnd(mean_1B, covariance_1B, 1);
    end

end

figure('Name', 'Part 2');
scatter(class_0(:,1), class_0(:,2)); hold on
scatter(class_1(:,1), class_1(:,2));
title('2D synthetic data');
legend('class 0', 'class 1');
xlabel('Dim 1'); ylabel('Dim 2');

%% Part 2 MAP decision rule

close all;

figure('Name', 'Part 2');
scatter(class_0(:,1), class_0(:,2)); hold on
scatter(class_1(:,1), class_1(:,2));
```

```
xlabel('Dim 1'); ylabel('Dim 2');
title('MAP decision rule');


%FIND BOUNDARY BY ITERATING THROUGH GRID AND CHECK WHEN PROBABILITY
%DENSITIES FOR EACH CLASS INTERSECT

epsilon = 1*10^-3;
boundary = zeros(10000, 2);
index = 1;
for x=-4:0.02:6
    for y=-6:0.02:6
        point = [x, y];
        pdf_0 = mvnpdf(point, mean_0', covariance_0);
        pdf_1A = mvnpdf(point, mean_1A', covariance_1A);
        pdf_1B = mvnpdf(point, mean_1B', covariance_1B);

        max_A = max(pdf_1A, pdf_1B);
        if abs(max_A - pdf_0) < epsilon && abs(max_A - pdf_0) > epsilon/2
            boundary(index, :) = point;
            index = index + 1;
        end
    end
end

plot(boundary(:,1), boundary(:,2), '.');
legend('class 0', 'class 1', 'MAP boundary');

%% Section 3

mean_0 = [1;-1];
U_0 = [1,0;0,1]
lambda_0 = [1,0;0,4]
covariance_0 = U_0 * lambda_0 * U_0.'
size = [1000 2]';
class_0 = mvnrnd(mean_0, covariance_0, 1000);

priorA = 2/3;
thetaA = -3*pi/4;
mean_1A = [-1;0];
U_1A = [cos(thetaA), -1*sin(thetaA);sin(thetaA), cos(thetaA)]
lambda_1A = [4,0;0,0.5]
covariance_1A = U_1A * lambda_1A * U_1A.'

priorB = 1/3;
thetaB = pi/4;
mean_1B = [4;1];
U_1B = [cos(thetaB), -1*sin(thetaB);sin(thetaB), cos(thetaB)]
lambda_1B = [1,0;0,4]
covariance_1B = U_1B * lambda_1B * U_1B.'
class_1 = zeros(1000,2);

for i=1:1000
    prior = rand;
```

```
    if prior > priorB
        class_1(i, :) = mvnrnd(mean_1A, covariance_1A, 1);
    else
        class_1(i, :) = mvnrnd(mean_1B, covariance_1B, 1);
    end

end

samples = [class_0; class_1];

data_label = zeros(2000, 1);
data_label(1001:end) = 1;

MAP_decision = zeros(2000, 1);

for i=1:length(samples)
    pdf_0 = mvnpdf(samples(i, :), mean_0', covariance_0);
    pdf_1A = mvnpdf(samples(i, :), mean_1A', covariance_1A);
    pdf_1B = mvnpdf(samples(i, :), mean_1B', covariance_1B);

    max_A = max(pdf_1A, pdf_1B);
    if max_A > pdf_0
        MAP_decision(i) = 1; % else already 0
    end

end

prob_error_0 = mean(data_label(1:1000)~=MAP_decision(1:1000))
prob_error_1 = mean(data_label(1001:end)~=MAP_decision(1001:end))
prob_error = mean(data_label~=MAP_decision)
```