

ELEC 425 - Assignment 1

Ali Bekheet

February 13, 2025

Note

This document contains answers for questions 2, 3, and 4 of the assignment with the complete code for the assignment in the `.zip` file. Also attached is a README detailing how to run the code but you can just skip all that and run the `main.m` script. The code is refactored for the assignment submission just so it can run modularly better.

Question 2 - Gaussian Classifier

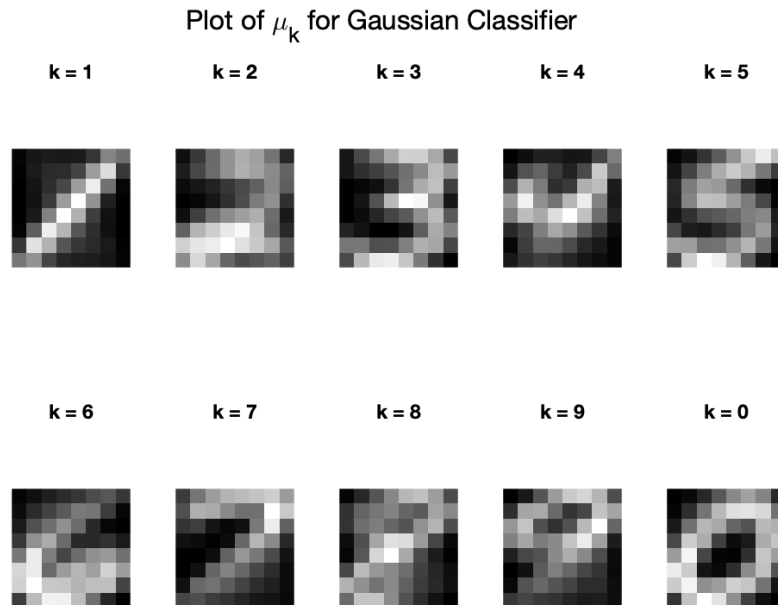


Figure 1: Visualization of μ_k (Mean Estimates) for the Gaussian Classifier

The value for the variance $\sigma^2 = 0.063351$.

Listing 1: Gaussian Classifier Training - Matlab Code

```
%% Training Gaussian Classifier
u_ki = zeros(features , classes);
var = 0;
gaussianClassifier = zeros(samples , classes);

for k = 1:classes
    u_ki(:, k) = mean(digits_train(:, :, k), 2);
    var = var + sum(sum((digits_train(:, :, k) - u_ki(:, k)).^2));
end

var = var / (features * samples * classes);
```

```

prefactor = (2 * pi * var)^(-features / 2);

%% Question 2 Deliverable
figure;
for k = 1:classes
    subplot(ceil(classes / 5), 5, k);
    imagesc(reshape(u_ki(:,k), 8, 8)');
    axis equal; axis off; colormap gray;
    title(['k=' num2str(mod(k,10))]);
end
sgtitle('Plot of  $\mu_k$  for Gaussian Classifier');
saveas(gcf, 'Gaussian_Classifier_Means.png');

```

Question 3 - Naive Bayesian Classifier

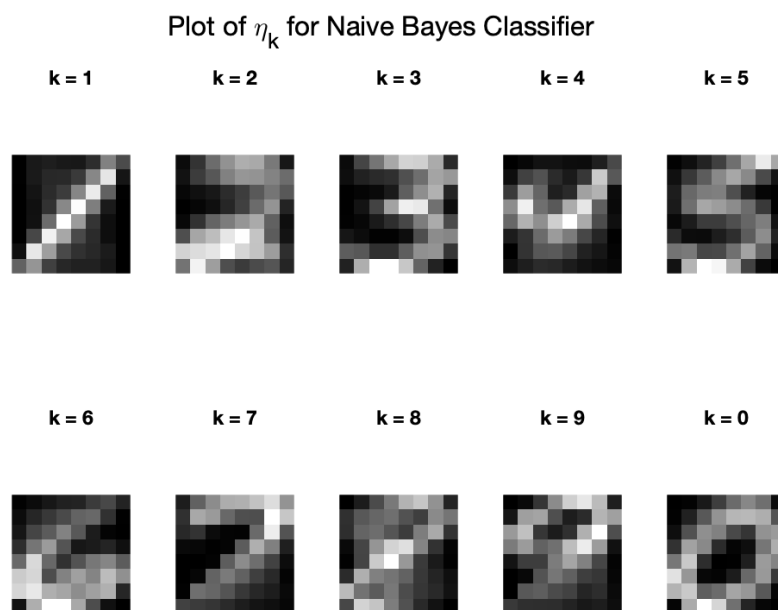


Figure 2: Visualization of η_k (Probability Estimates) for the Naive Bayes Classifier

Listing 2: Naive Bayes Classifier Training - Matlab Code

```

%% Training Naive Bayes
log_digits_train = digits_train > 0.5;
n_ki = zeros(features, classes);
naiveBayesClassifier = zeros(samples, classes);

for k = 1:classes
    n_ki(:, k) = mean(log_digits_train(:, :, k), 2);
end

for k = 1:classes
    naiveBayesClassifier(:, k) = prod((n_ki(:, k) .* (1 - n_ki(:, k))), 1);
end

%% Question 3 Deliverable
figure;

```

```

for k = 1:classes
    subplot(ceil(classes / 5), 5, k);
    imagesc(reshape(n_ki(:,k), 8, 8)');
    axis equal; axis off; colormap gray;
    title(['k=' num2str(mod(k,10))]);
end
sgtitle('Plot of \eta_k for Naive Bayes Classifier');
saveas(gcf, 'NaiveBayesian_Classifier_Means.png');

```

Question 4 - Testing

Classifier	1	2	3	4	5	6	7	8	9	10	Total Errors
Gaussian Classifier	69	81	63	61	68	44	63	109	110	53	721
Naive Bayes Classifier	87	104	91	85	111	60	89	121	133	58	939

Table 1: Classification Errors per Class and Total Errors for Gaussian and Naive Bayes Classifiers, with 400 samples per class.

Overall error rates:
Gaussian Classifier: 0.1802
Naive Bayes Classifier: 0.2347

Listing 3: Classifier Testing and Error Rates - Matlab Code

```

%% Testing
test_samples = size(digits_test, 2); % 400

digits_test_resaped = reshape(digits_test, features, test_samples * classes); % 64x4000
bdigits_test_resaped = digits_test_resaped > 0.5;

gaussianClassifier_test = zeros(test_samples * classes, classes);
naiveBayesClassifier_test = zeros(test_samples * classes, classes);
gaussian_errors = zeros(classes, 1);
bayesian_errors = zeros(classes, 1);

for k = 1:classes % Gaussian Test
    gaussianClassifier_test(:, k) =
        prefactor * exp(-(sum((digits_test_resaped(:, :) - u_ki(:,k)).^2, 1)) / (2 * var)));
    naiveBayesClassifier_test(:, k) =
        prod((n_ki(:,k).^ bdigits_test_resaped) .* (1-n_ki(:,k)).^(~ bdigits_test_resaped));
end

[~, gaussian_test_results] = max(gaussianClassifier_test, [], 2);
[~, bayesian_test_results] = max(naiveBayesClassifier_test, [], 2);
gaussian_test_results = reshape(gaussian_test_results, test_samples, classes);
bayesian_test_results = reshape(bayesian_test_results, test_samples, classes);

for k = 1:classes % Gaussian Error Count
    gaussian_errors(k,1) = sum(gaussian_test_results(:,k) ~= k);
    bayesian_errors(k,1) = sum(bayesian_test_results(:,k) ~= k);
end

%% Question 4 Deliverable
ks = 1:classes;
disp("Gaussian Classifier Errors per Class");
disp(ks);
disp(gaussian_errors');
disp("Naive Bayes Classifier Errors per Class");
disp(ks);
disp(bayesian_errors');

```