Gaussian Discriminant Analysis

Red Barrel Image Segmentation

Michael Dimitrov
Computer Science
CS 458 - Intro to Robotics
Purdue University - West Lafayette
dimitrm@purdue.edu

I. Introduction

Detecting and segmenting objects in images is a fundamental problem in robotics and computer vision. For autonomous robots, the ability to identify specific targets, in this case red barrels, is essential for navigation, obstacle avoidance, and interaction with the environment. A robot needs to reliably distinguish the object of interest from a potentially cluttered background under varying lighting conditions and in the presence of distracting colors or partial occlusions.

In this project, I approached barrel segmentation as a pixel-wise classification problem using Gaussian Discriminant Analysis (GDA). Each pixel in an image is represented by its RGB color values, and the goal is to classify it as either barrel (class 1) or background (class 0). I implemented two versions of GDA:

- **Linear Discriminant Analysis (LDA):** assumes both classes share the same covariance (common variance).
- **Quadratic Discriminant Analysis (QDA):** allows each class to have its own covariance (variable variance).

The overall pipeline follows the assignment specifications:

1) **Manual labeling** of positive regions in both training and testing images and negative regions in training images to create ground-truth masks for training and performance evaluation.
2) **Data loading** to extract pixel features (RGB) and labels from the labeled regions.
3) **Model training** to estimate class means, covariances, and priors.
4) **Prediction** on testing images using the learned decision boundary functions.
5) **Evaluation** using precision and recall for both classes, along with qualitative visualization via segmented images.

The rest of the report formulates the problem mathematically, describes the technical approach and implementation details, presents both quantitative and qualitative results, and concludes with a discussion of what worked well, what did not, and why.

II. Problem Formulation

The goal of this project is to perform pixel-wise binary classification for barrel segmentation. Each pixel in an image is represented by its RGB color values $x \in \mathbb{R}^3$, and the objective is to predict whether it belongs to the *barrel* class ($y = 1$) or the *background* class ($y = 0$).

We assume that the color distribution of each class follows a multivariate Gaussian:

$$x|y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

The discriminant function used for classification is:

$$f_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k) + \log \pi_k$$

Where:

- $\mu_k \in \mathbb{R}^3$ is the mean color vector of class k.
- $\Sigma_k \in \mathbb{R}^{3\times 3}$ is the covariance matrix of class k.
- $\pi_k = P(y = k)$ is the prior probability of class k, estimated from the proportion of pixels labeled as k in the training set.

Two versions of the algorithm are implemented:

- **Linear Discriminant Analysis (LDA):** assumes $\Sigma_0 = \Sigma_1 = \Sigma$, i.e., both classes share the same covariance matrix (common variance).
- **Quadratic Discriminant Analysis (QDA):** uses separate covariance matrices Σ_0 and Σ_1 for each class (variable variance).

Both models are trained on the labeled training pixels and then applied to all pixels in the test images to evaluate segmentation performance.

III. Technical Approach

A. Hand Labeling

Training and testing datasets were first annotated to create ground-truth masks. In the training set, each image was opened twice with Roipoly: once to mark a positive (barrel) polygon and once to mark a negative (background) polygon of roughly equal area. In the testing set, each image was opened once to annotate a positive region.

The interface allowed the user to draw polygons by left-clicking to add vertices and right-clicking to close the polygon, and close the window. Each labeled mask was stored as a boolean array inside a .npz file in either the train_region or test_region folder.

For training, both positive and negative masks were bundled (under pos_mask and neg_mask), whereas for testing only positive masks were included in the .npz.

B. Data Loading

Pixel features and labels were extracted from the labeled masks using the functions *import_pre_labeled_training* and *import_pre_labeled_testing*.

For the **training set**, only pixels inside the annotated polygons were used, since all other pixels were unlabeled. Each pixel's RGB values were normalized to [0,1] and stored as rows of

$$X_{\text{train}} \in \mathbb{R}^{N_{\text{train}} \times 3}$$

with corresponding labels

$$y_{\text{train}} \in \{0,1\}^{N_{\text{train}}}$$

where N_{train} was the number of pixels in each positive mask plus the number of pixels in each negative mask. Positive pixels were labeled as $y = 1$, negative pixels as $y = 0$.

For the **testing set**, all pixels in each image were included:

$$X_{\text{test}} \in \mathbb{R}^{N_{\text{test}} \times 3}$$

with ground-truth labels

$$y_{\text{test}} \in \{0,1\}^{N_{\text{train}}}$$

where N_{test} was the total pixels in all test images. Pixels inside the positive mask were labeled as 1, and all others as 0.

This separation allowed training on a limited set of labeled pixels while evaluating on the entire image.

C. Model Training

The function class_stats(features, labels) computed the class priors, means, and covariances:

$$\pi_k = \frac{N_k}{N}$$

$$\mu_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top.$$

Two versions of Gaussian Discriminant Analysis were implemented:

- **Linear Discriminant Analysis (LDA):** train_GDA_common_variance(features, labels) assumes a shared covariance:

  $$\Sigma = \frac{N_0\Sigma_0 + N_1\Sigma_1}{N_0 + N_1}$$

  In code, this pooled covariance Σ was returned for both classes.
- **Quadratic Discriminant Analysis (QDA):** train_GDA_variable_variance(features, labels) used the class-specific Σ_0 and Σ_1 without pooling.

Both methods returned $\{\pi_k, \mu_k, \Sigma_k\}$ parameters for use in prediction.

D. Prediction

The function predict(X, prior, mu, cov) computed the discriminant

$$f_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k) + \log \pi_k$$

for each pixel and assigned the label $\hat{y} = \arg\max_k f_k(x)$.

For **LDA** the same Σ was used for both classes, whereas for **QDA**, each class had its own covariance. In implementation, row-wise np.sum computed $(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)$ for each pixel.

E. Accuracy Analysis

The function accuracy_analysis computed per-class precision and recall:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}, \quad \text{Recall}_k = \frac{TP_k}{TP_k + FN_k}$$

Here TP_k = true positives, FP_k = false positives, and FN_k = false negatives for class k.

- $TP_k = \{\hat{y} = k, y = k\}$
- $FP_k = \{\hat{y} = k, y =\neq k\}$
- $TN_k = \{\hat{y} \neq k, y = k\}$

The function printed results in the required format for both the common-variance and variable-variance models without modifying main().

F. Segmentation

For each test image, the trained GDA model was applied at the pixel level using the predict function. The image was first reshaped so each pixel's RGB values formed one row of the input feature matrix. The prediction returned a 0/1 array indicating whether each pixel was classified as barrel (1) or background (0).

This prediction array was then reshaped back to the original image dimensions, and a new output image was created by coloring pixels labeled as 1 in red [1,0,0] and all others in black [0,0,0]. The resulting segmented images were saved to the appropriate output folders for the common-variance and variable-variance models.

IV. Results

A. Quantitative Results

The performance of both the common-variance (LDA) and variable-variance (QDA) GDA models was evaluated on the labeled testing set using precision and recall for each class. The final metrics are:

Model	Label	Precision (%)	Recall (%)
LDA	0 (bg)	99.91	96.06
	1 (barrel)	39.58	96.68
QDA	0 (bg)	99.93	94.90
	1 (barrel)	33.79	97.60

Both models achieved very high recall for the barrel class, with moderate precision, while background performance remained strong across both precision and recall.

B. *Qualitative Results*

The following subsection presents three representative examples to illustrate segmentation quality. Each example includes the original image, the LDA segmentation, and the QDA segmentation so that both methods can be compared directly.

Training Example

Fig. 1. Training Image (Original)

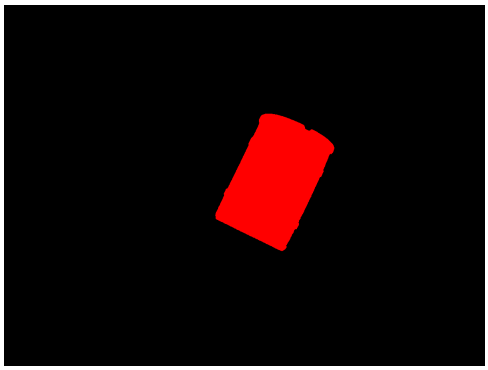

Fig. 2. Training Image (LDA Segmentation)

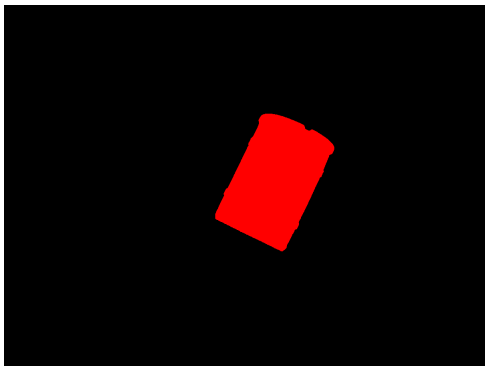

Fig. 3. Training Image (QDA Segmentation)

The training example shows nearly perfect segmentation with crisp barrel boundaries for both models. Because these images were used to estimate the model parameters, the segmentations are expected to be clean and accurate.

Testing Example 1

Fig. 4. Testing Image 1 (Original)

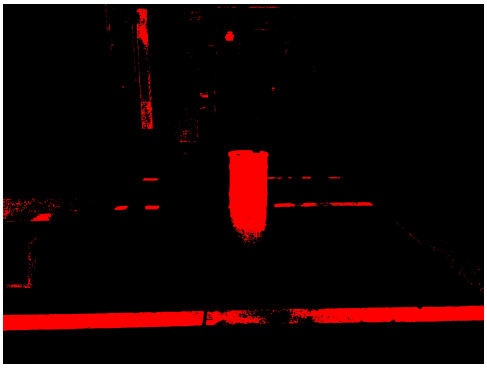

Fig. 5. Testing Image 1 (LDA Segmentation)

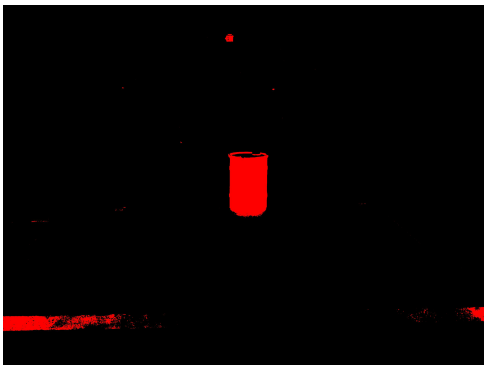

Fig. 6. Training Image 1 (QDA Segmentation)

In this testing example, both models capture the entire barrel with some background noise. Barrels are fully detected, with QDA producing slightly cleaner edges and less noise overall.

Testing Example 2

Fig. 7. Testing Image 2 (Original)

Fig. 8. Testing Image 2 (LDA Segmentation)

Fig. 9. Training Image 2 (QDA Segmentation)

In this example the red Coca-Cola machine is selected. Due to the nature of the models, the pixels of the machine closely match the barrel color and uniformity, leading to the entire machine being selected. QDA shows less background noise, suggesting better adaptation to the barrel's tighter color distribution.

C. Discussion

The quantitative results show very high recall for the barrel class (>96%) across both models, meaning almost all barrel pixels were detected. However, barrel precision was relatively low (40% for LDA, 34% for QDA) because the classifier lacks explicit object detection: it labels individual red pixels as barrels, so many red background objects can become false positives.

For the background class, both precision and recall remained high (>94%), but the recall was noticeably lower than the precision. This indicates that some background pixels were incorrectly labeled as barrel, which is consistent with the low barrel precision scores.

Comparing the two models, LDA achieved higher barrel precision than QDA, suggesting the pooled covariance assumption produced tighter decision boundaries for distinguishing barrel from background pixels. However, recall was similar for both methods, meaning both models found nearly all barrel pixels. One possible explanation is that background pixels exhibit higher color variance than the relatively uniform red barrels. Thus, pooling in LDA may weight the smoother barrel class more heavily, improving precision but limiting flexibility. QDA, with separate covariances, might better adapt to the barrel class but risks overfitting to background noise.

The qualitative results reinforce this. Training images show very crisp segmentations, possibly due to overfitting since the models were trained on these exact pixel statistics. On testing images, QDA often selected fewer random red regions than LDA, producing cleaner barrels. This suggests that separate covariance matrices helped QDA adapt to the tight color distribution of barrels while allowing more flexibility for background pixels.

D. Potential Improvements

Several factors could further improve the segmentation accuracy:

1) **More precise labeling:** Avoid including background pixels inside barrel polygons, especially thin occlusions (table legs or chair arms) crossing the barrel. These dilute the true barrel color distribution.
2) **Better negative selection:** Background masks should either consistently exclude red objects or deliberately include many red distractors so the model can learn their distribution separately from barrels. Mixing strategies may have confused the classifier.
3) **Multiple barrels per image:** Labeling all barrels rather than only the most visible one could increase training diversity and help the model generalize to different barrel appearances.