ECE 271A: Quiz #3

Due on November 27, 2023 at 11:59pm

Professor Vasconcelos

Ray Tsai

A16848188

Parameters

We assume that the class-conditional,

$$P_{\mathbf{x}|\mu,\mathbf{\Sigma}} = \mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma}),$$

where

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\right)\left(x_i - \frac{1}{N}\sum_{i=k}^{N}\mathbf{x}_k\right)^T,$$

is the covariance matrix of the sample and μ is a random variable of the distribution

$$P_\mu(\mu) = \mathcal{G}(\mu, \mu_0, \mathbf{\Sigma}_0),$$

with $\mathbf{\Sigma}_0 = diag(\alpha\mathbf{w})$, for $\mathbf{w}$ and μ_0 given from the dataset.

Predictive Distribution

For each class i, we compute the parameters of

$$P_{\mu|\mathbf{T}}(\mu|\mathcal{D}_i) = \mathcal{G}(\mu, \mu_i, \mathbf{\Sigma}_i).$$

From the textbook, we know

$$\mu_i = \mathbf{\Sigma}_0\left(\mathbf{\Sigma}_0 + \frac{1}{N}\mathbf{\Sigma}\right)^{-1}\mu_{sample} + \frac{1}{N}\mathbf{\Sigma}\left(\mathbf{\Sigma}_0 + \frac{1}{N}\mathbf{\Sigma}\right)^{-1}\mu_0,$$

$$\mathbf{\Sigma}_i = \mathbf{\Sigma}_0\left(\mathbf{\Sigma}_0 + \frac{1}{N}\mathbf{\Sigma}\right)^{-1}\frac{1}{N}\mathbf{\Sigma},$$

where $\mu_{sample} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. This immediately follows that the predictive distribution

$$P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) = \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma} + \mathbf{\Sigma}_i).$$

Maximum Likelihood Estimate

The maximum likelihood estimate of the class conditional distribution is

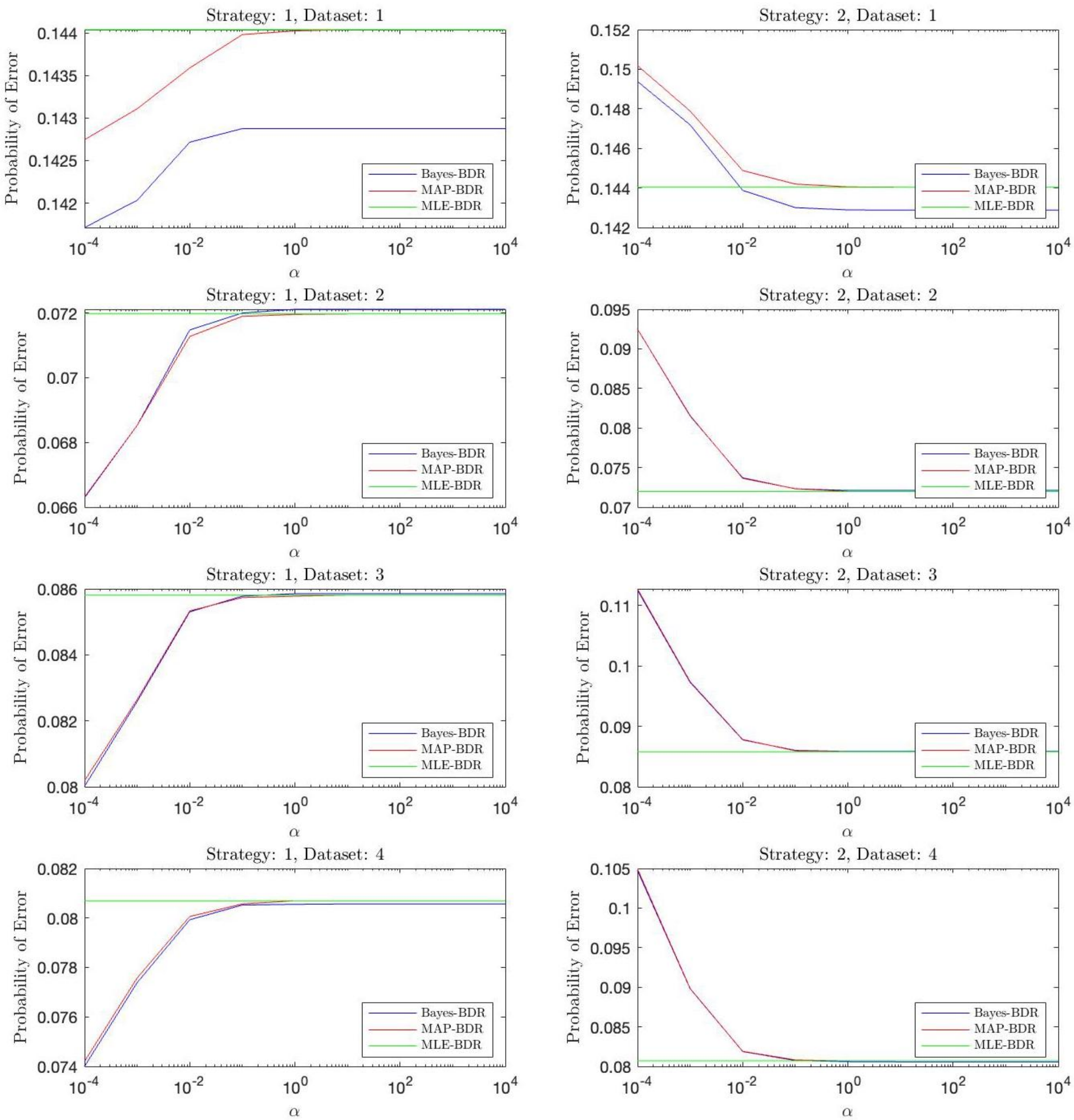
$$P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) = \mathcal{G}(\mathbf{x}, \mu_{sample}, \mathbf{\Sigma}).$$

MAP Approximation

The MAP approximation of the posterior mean is simply the posterior mean we obtained above. Thus,

$$P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) = \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma}).$$

Curves


Strategy: 1, Dataset: 1
Strategy: 2, Dataset: 1
Strategy: 1, Dataset: 2
Strategy: 2, Dataset: 2
Strategy: 1, Dataset: 3
Strategy: 2, Dataset: 3
Strategy: 1, Dataset: 4
Strategy: 2, Dataset: 4
Probability of Error
α
Bayes-BDR
MAP-BDR
MLE-BDR

Interpretations

Part A

We examine the Bayes plot (blue). Notice that as α increases, the probability of error increases and eventually converges to a constant value. This is because α is proportional to the diagonal entries of the prior covariance matrix $\mathbf{\Sigma}_0$. Thus, we know by intuition that the larger α is, the more uncertain μ is, and the less informative the prior distribution of μ is. Since

$$\lim_{\alpha\to\infty} \mathbf{\Sigma}_0 \left(\mathbf{\Sigma}_0 + \frac{1}{N}\mathbf{\Sigma}\right)^{-1} = I,$$

the predictive distribution $P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) \approx \mathcal{G}\left(\mathbf{x}, \mu_{sample}, \left(1+\frac{1}{N}\right)\mathbf{\Sigma}\right)$ as α approaches infinity, and this shows that the probability of error converges to a constant value which depends on the sample size.

Part B

We can see the ML plots (green) are all constant lines. This is because maximum likelihood estimation does not consider the prior distribution of the parameters, and thus the results do not depend on α. Another phenomenon we noticed is that the Bayes-BDR plot gets closer to the ML plot as α increases. From part A, we showed that $P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) \approx \mathcal{G}\left(\mathbf{x}, \mu_{sample}, \left(1+\frac{1}{N}\right)\mathbf{\Sigma}\right)$ as α approaches infinity. This implies that α brings the parameters of the predictive distribution closer to the maximum likelihood solution as it becomes larger and homogenizes the predictive distribution and the resulting distribution of the maximum likelihood solution, which explains the phenomenon we observed.

Part C

We observe in the MAP plot (red) that the probability of error increases as α increases, which eventually converges to the constant line of the ML plot (green). Since $\lim\limits_{\alpha\to\infty} \mu_i = \mu_{sample}$, the MAP-estimated distribution $P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) = \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma}) \approx \mathcal{G}(\mathbf{x}, \mu_{sample}, \mathbf{\Sigma})$ gradually becomes the ML-estimated distribution as α approaches infinity, and this explains our observation. We also notice that the MAP plot (red) is closly related to the Bayes plot (blue). This is because the distribution in both cases share the same mean μ_i.

Part D

We first notice that the predictive distribution performs better when the sample size is small. This is because, for ML and MAP estimation, we only pick one model we think is the best given the dataset, but we utilize all models for the predictive distribution. Thus, given limited data, the ML and MAP estimation would be a lot less accurate than the predictive distribution due to the uncertainty of the actual parameters. As we increase the sample size, the Bayes plot approaches the MAP plot and converges to a constant value extremely close to the ML plot when α is large. This can again be explained by examining μ_i and $\mathbf{\Sigma}_i$. Similar to the α case,

$$\lim_{N\to\infty} \mathbf{\Sigma}_0 \left(\mathbf{\Sigma}_0 + \frac{1}{N}\mathbf{\Sigma}\right)^{-1} = I,$$

which implies that when our sample size grows large, μ_i approaches the sample mean and $\mathbf{\Sigma}_i$ vanishes. Therefore, the predictive distribution $P_{\mathbf{x}|\mathbf{T}}(\mathbf{x}|\mathcal{D}_i) = \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma} + \mathbf{\Sigma}_i) \approx \mathcal{G}(\mathbf{x}, \mu_{sample}, \mathbf{\Sigma})$ when N is large enough, which coincides with the ML-estimated distribution.

Part E

Contrary to strategy 1, strategy 2 exhibits a higher probability of error in the Bayes and MAP solution when α is small, and approach when α gets larger as they did in strategy 1. This phenomenon can be explained by the quality of our priors. Since the prior for both strategy have zeros for all coefficients other than the first, the only difference between two strategies is the choice of the first DCT coefficient. In strategy 1, we have a smaller coefficient for the cheetah class and a larger coefficient for the grass class, and in strategy 2, we have identical coefficients for both class. However, in Quiz 2, we have shown that the distributions of the first DCT coefficient for each class is of great differences, where the values of the cheetah class appears to be smaller those that of the grass class in general. This implies that the priors of strategy 2 fails to reflect the actual situation, and thus it is of worse quality compared to the priors of strategy 1, which explains our observation.

MATLAB Code

main.m

```
load('../dataset/TrainingSamplesDCT_subsets_8.mat');
alpha = load('../dataset/Alpha.mat');
alpha = alpha.alpha;
strat_1 = load('../dataset/Prior_1.mat');
strat_2 = load('../dataset/Prior_2.mat');
zigzag = load('../dataset/Zig-Zag_Pattern.txt');
cheetah = imread('../dataset/cheetah.bmp');
cheetah_mask = imread('../dataset/cheetah_mask.bmp');
target = im2double(cheetah);
mask = im2double(cheetah_mask);

dataset_BG = {D1_BG, D2_BG, D3_BG, D4_BG};
dataset_FG = {D1_FG, D2_FG, D3_FG, D4_FG};
strats = {strat_1, strat_2};

[row_TG, col_TG] = size(target);
[~, alpha_dim] = size(alpha);

zigzag = zigzag + 1;

figure;
for d = 1:4
  training_BG = dataset_BG{d};
  training_FG = dataset_FG{d};

  [row_BG, col_BG] = size(training_BG);
  [row_FG, col_FG] = size(training_FG);

  prior_BG = row_BG / (row_BG + row_FG);
  prior_FG = row_FG / (row_BG + row_FG);

  % pick cheetah if (p(x | grass) / p(x | cheetah)) < threshold
  threshold = prior_FG / prior_BG;

  mean_FG = zeros(1, 64);
  mean_BG = zeros(1, 64);

  cov_FG = zeros(64, 64);
  cov_BG = zeros(64, 64);

  for r = 1:row_FG
      cov_FG = cov_FG + training_FG(r,:)' * training_FG(r,:);
      mean_FG = mean_FG + training_FG(r,:);
  end
```

```
for r = 1:row_BG
    cov_BG = cov_BG + training_BG(r,:)' * training_BG(r,:);
    mean_BG = mean_BG + training_BG(r,:);
end

mean_FG = mean_FG / row_FG;
mean_BG = mean_BG / row_BG;

cov_FG = (cov_FG / row_FG) - mean_FG' * mean_FG;
cov_BG = (cov_BG / row_BG) - mean_BG' * mean_BG;

mle_result = mle(training_BG, training_FG) * ones(alpha_dim);

for s = 1:2
  strat = strats{s};

  bayes_result = zeros(1, alpha_dim);
  map_result = zeros(1, alpha_dim);

  for alp = 1:alpha_dim

      sigma0 = alpha(alp) * diag(strat.W0);

      mu_bayes_FG = sigma0 * inv(sigma0 + cov_FG/row_FG) * mean_FG' + ...
          cov_FG/row_FG * inv(sigma0 + cov_FG/row_FG) * strat.mu0_FG';
      mu_bayes_BG = sigma0 * inv(sigma0 + cov_BG/row_BG) * mean_BG' + ...
          cov_BG/row_BG * inv(sigma0 + cov_BG/row_BG) * strat.mu0_BG';

      sigma_bayes_FG = sigma0 * inv(sigma0 + cov_FG/row_FG) * cov_FG/row_FG;
      sigma_bayes_BG = sigma0 * inv(sigma0 + cov_BG/row_BG) * cov_BG/row_BG;

      A_bayes = zeros(row_TG, col_TG);
      A_map = zeros(row_TG, col_TG);

      for r = 5:row_TG-3
          for c = 5:col_TG-3
              block = target(r - 4:r + 3, c - 4:c + 3);
              dctBlock = dct2(block);
              X = zeros(1, 64);
              for i = 1:8
                  for j = 1:8
                      X(zigzag(i, j)) = dctBlock(i, j);
                  end
              end
              A_bayes(r, c) = int8(mvn(X, mu_bayes_BG', cov_BG + sigma_bayes_BG)/ ...
                  mvn(X, mu_bayes_FG', cov_FG + sigma_bayes_FG) <= threshold);
              A_map(r, c) = int8(mvn(X, mu_bayes_BG', cov_BG)/ ...
                  mvn(X, mu_bayes_FG', cov_FG) <= threshold);
          end
```

```
        end

        error_bayes = 0;
        error_map = 0;
        for r = 1:row_TG
            for c = 1:col_TG
                if (A_bayes(r, c) ~= mask(r, c))
                    error_bayes = error_bayes + 1;
                end

                if (A_map(r, c) ~= mask(r, c))
                    error_map = error_map + 1;
                end
            end
        end

        error_rate_bayes = error_bayes / (row_TG * col_TG);
        error_rate_map = error_map / (row_TG * col_TG);
        bayes_result(alp) = error_rate_bayes;
        map_result(alp) = error_rate_map;
    end

    subplot(4, 2, 2 * d + s - 2);

    plot(alpha, bayes_result, 'b');
    hold on;

    plot(alpha, map_result, 'r');
    hold on;

    plot(alpha, mle_result, 'g');

    set(gca, 'XScale', 'log');

    title("Strategy: " + s + ", Dataset: " + d, 'Interpreter', 'latex');

    xlabel("\alpha");
    ylabel("Probability of Error", 'Interpreter', 'latex');

    legend({"Bayes-BDR", "MAP-BDR", "MLE-BDR"}, 'Location','southeast', ...
        'Interpreter', 'latex', 'FontSize', 6);
  end
end
```

mle.m

```
function error_rate = mle(training_BG, training_FG)
    zigzag = load('../dataset/Zig-Zag Pattern.txt');
    cheetah = imread('../dataset/cheetah.bmp');
    cheetah_mask = imread('../dataset/cheetah_mask.bmp');
    target = im2double(cheetah);
    mask = im2double(cheetah_mask);

    zigzag = zigzag + 1;

    [row_BG, ~] = size(training_BG);
    [row_FG, ~] = size(training_FG);
    [row_TG, col_TG] = size(target);

    prior_BG = row_BG / (row_BG + row_FG);
    prior_FG = row_FG / (row_BG + row_FG);

    % pick cheetah if (p(x | grass) / p(x | cheetah)) < threshold
    threshold = prior_FG / prior_BG;

    mean_FG = zeros(1, 64);
    mean_BG = zeros(1, 64);

    cov_FG = zeros(64, 64);
    cov_BG = zeros(64, 64);

    for r = 1:row_FG
        cov_FG = cov_FG + training_FG(r,:)' * training_FG(r,:);
        mean_FG = mean_FG + training_FG(r,:);
    end

    for r = 1:row_BG
        cov_BG = cov_BG + training_BG(r,:)' * training_BG(r,:);
        mean_BG = mean_BG + training_BG(r,:);
    end

    mean_FG = mean_FG / row_FG;
    mean_BG = mean_BG / row_BG;

    cov_FG = (cov_FG / row_FG) - mean_FG' * mean_FG;
    cov_BG = (cov_BG / row_BG) - mean_BG' * mean_BG;

    A = zeros(row_TG, col_TG);

    for r = 5:row_TG-3
        for c = 5:col_TG-3
            block = target(r - 4:r + 3, c - 4:c + 3);
            dctBlock = dct2(block);
```

```
            X = zeros(1, 64);
            for i = 1:8
                for j = 1:8
                    X(zigzag(i, j)) = dctBlock(i, j);
                end
            end
            A(r, c) = int8(mvn(X, mean_BG, cov_BG)/ ...
                mvn(X, mean_FG, cov_FG) <= threshold);
        end
    end

    error_rate = 0;
    for r = 1:row_TG
        for c = 1:col_TG
            if (A(r, c) ~= mask(r, c))
                error_rate = error_rate + 1;
            end
        end
    end
    error_rate = error_rate / (row_TG * col_TG);
end
```

mvn.m

```
function result = mvn(x, mean, cov)
    [~, dim] = size(mean);
    d = (x - mean) * inv(cov) * (x - mean)';
    c = 1/sqrt((2 * pi)^dim * det(cov));
    result = c * exp(-0.5 * d);
end
```