ELEC 425 – Assignment 1

October 20, 2021

Nicholas Alderman
20060982
16naa5

1.2 Training

Class Conditional Gaussian

Maximum Likelihood Estimation for each class k using the formula:

$$\hat{u}_{ki} = \frac{\sum_{j=1}^{m_k} x_{kji}}{m_k}$$

```
% mu_ki = sum of each vector of D features divided by the number of
% training data points
% mu = average value for each feature in a class
mle = [];

for j = 1:10
    for i = 1:64
        mle(i,j) = sum(digits_train(i,:,j))/700
    end
end
```

Shared σ^2 can be estimated with the equation:

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{K} \sum_{j=1}^{m_k} \sum_{i=1}^{D} \left(x_{kji} - \mu_{ki}\right)^2}{DM}$$

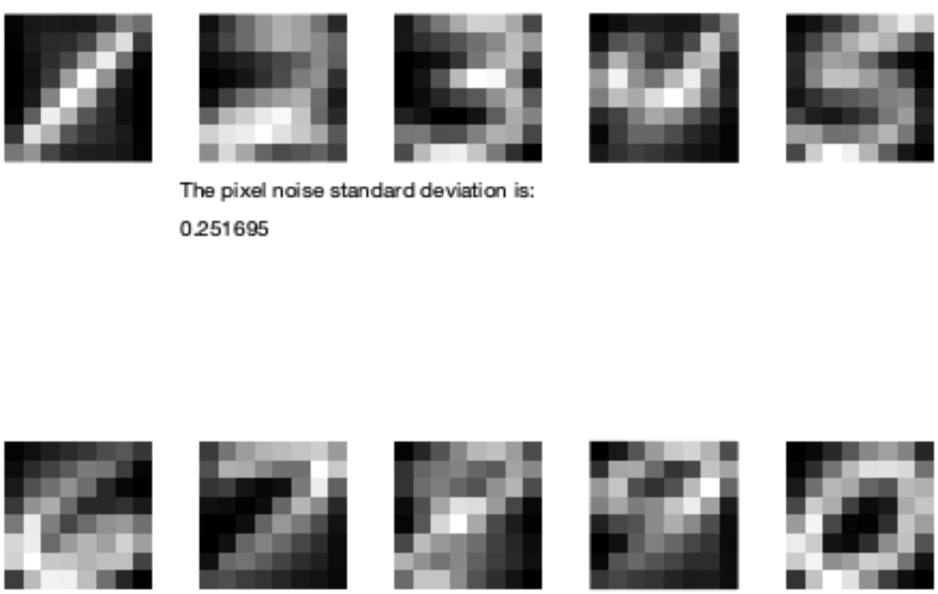
```
% fix shape of mle
mle2 = reshape(mle,64,1,10)
mle2 = repmat(mle2,1,700,1)

% now need to get sigma^2
% sum of difference between feature and mle across all classes across all
% data points across all features
% x = digits_X(i,j,k) -- j = data point #, i = feature number, k = class

s2 = sum((digits_train - mle2).^2, 'all') / (64 * 7000);
```

The value obtained was: $\sigma^2 = 0.0634 \quad \rightarrow \quad \sigma = \sqrt{0.0634} = 0.251695$

Here are the 10 plots of each mean μ_k (maximum likelihood estimation) with the pixel noise standard deviation:



2 Naïve Bayes Classifier

Convert real-valued features **x** into binary features **b** by thresholding:

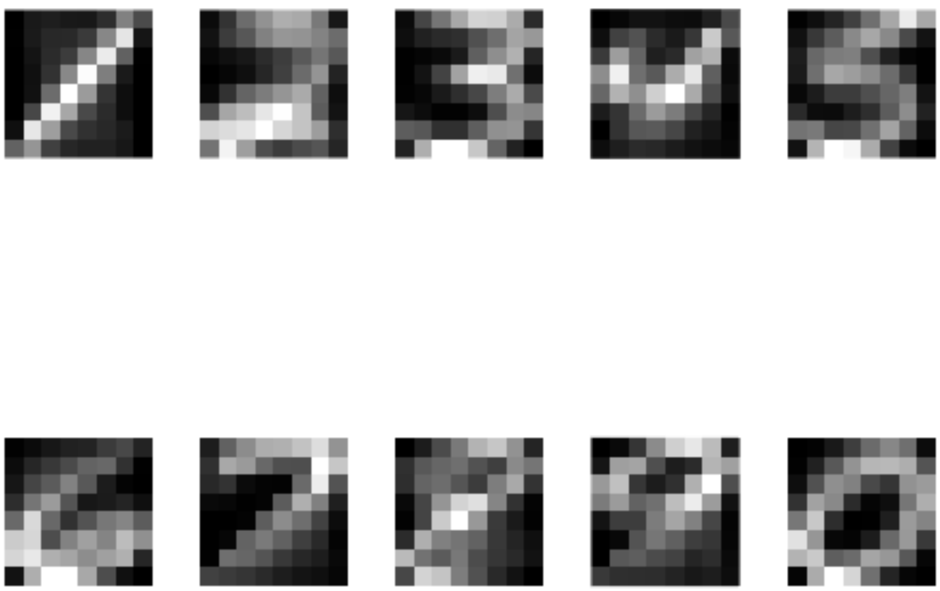
$$b_i = 1 \; if \; x_i > 0.5 \; otherwise \; b_i = 0$$

```
% convert training data to binary values with threshold using fix
digits_train_binary = (digits_train>0.5)
```

Get parameters $\eta_{ki} \equiv p(b_i = 1|C_k)$:

```
% now get eta_ki = p(b_i=1|C_k)
eta = sum(digits_train_binary(:,:,:),2)./ 700
m_eta = 1-eta
```

Plots of each image for each vector η_k



3 Test Performance

Gaussian Classifier

```
% 3 Test Performance
% gaussian test - p(C_k|x) = p(x|C_k)*p(C_k)
t1 = (2*pi*s2)^-32
t2 = (-1/(2*s2))

for i = 1:10
    gaussian_test(i,:,:) = ((t1 .* exp(t2 .* sum((digits_test(:,:,:) - mle(:,i)).^2)
    % bayes theorem
end

% normalize so that each data point sums to 1
gaussian_test(:,:,:) = gaussian_test(:,:,:)./sum(gaussian_test)

% select most likely class for each data point
for i = 1:10
    [mx, idx] = max(gaussian_test(:,:,i), [], 1)
    gaussian_errs(i) = nnz(idx - i)
end
```

Naïve Bayes Classifier

```
% naive bayes p(C_k|x) = p(b|C_k,eta) * p(C_k) = Prod(eta * (1 - eta)) *
% 1/10
digits_test_binary = (digits_test>0.5)

% if feature value is 1, use eta, if zero use 1-eta
for i = 1:10
    % convert data_test_binary matrix to contain only values eta or 1-eta
    eta_combined = m_eta(:,1,i) .* (digits_test_binary==0)
    eta_combined_p = eta(:,1,i) .* (digits_test_binary==1)
    eta_combined(eta_combined == 0) = eta_combined_p(eta_combined==0)
    temp = reshape(prod(eta_combined),1,400,10)
    naive_test(i,:,:) = temp
end

% normalize so that each point sums to 1
naive_test(:,:,:) = naive_test(:,:,:)./sum(naive_test)

for i = 1:10
    [mx, idx] = max(naive_test(:,:,i), [], 1)
    naive_errs(i) = nnz(idx - i)
end
```

Results Table

Code to calculate overall error rate:

```
gaussian_errors_total = sum(gaussian_errs) / 4000;
naive_errors_total = sum(naive_errs) / 4000;
```

Classifier\ Digit	1	2	3	4	5	6	7	8	9	0	Overall Error Rate
Gaussian	69	81	63	61	68	44	63	109	110	53	18.03%
Naive	87	104	91	85	111	60	89	121	133	58	23.48%