ECE 271A - Assignment 5

Srinidhi Bharadwaj Kalgundi Srinivas

A59010581

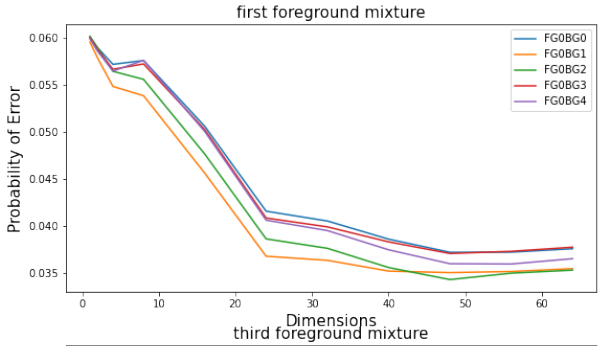
a) For each class, learn 5 mixtures of C = 8 components, using a random initialization (recall that the mixture weights must add up to one). Plot the probability of error vs. dimension for each of the 25 classifiers obtained with all possible mixture pairs. Comment the dependence of the probability of error on the initialization.

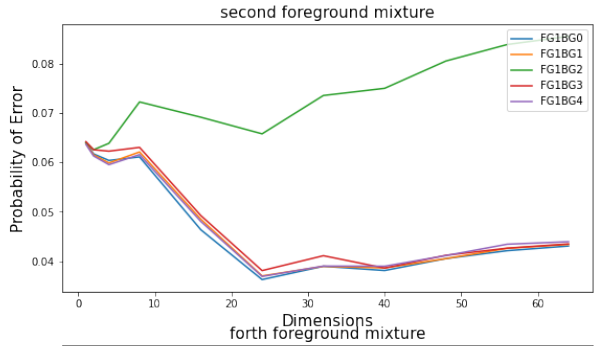
Solution:

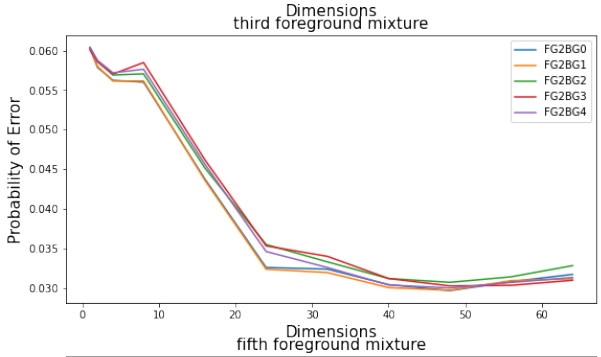
Below is the plot for probability of error for 25 classifiers. THe curves varies slightly everytime the code is run because of random initialization.

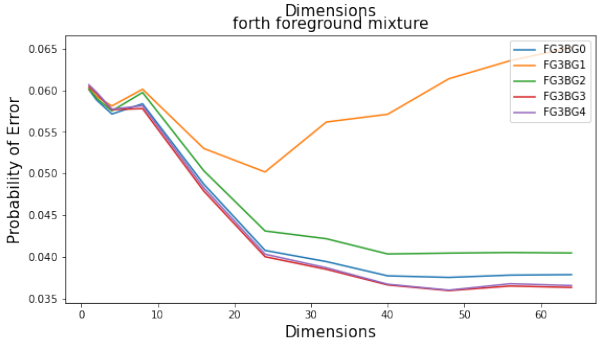
As the number of dimensions are increased, the probability of error starts to decrease. It can observed from the plot that the probability of error is at its minimum when the dimension is between 40 and 50. This is corroborated with what was observed in homework 2 where selecting best 8 features yielded better results than when 64 features were used.

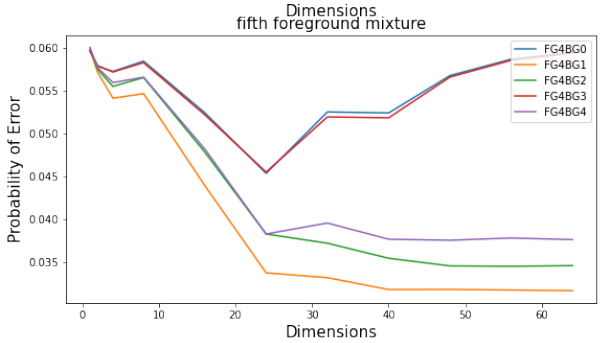
The effect of random initialization is that the probabilities of error are different for different combinations of foreground and background mixtures











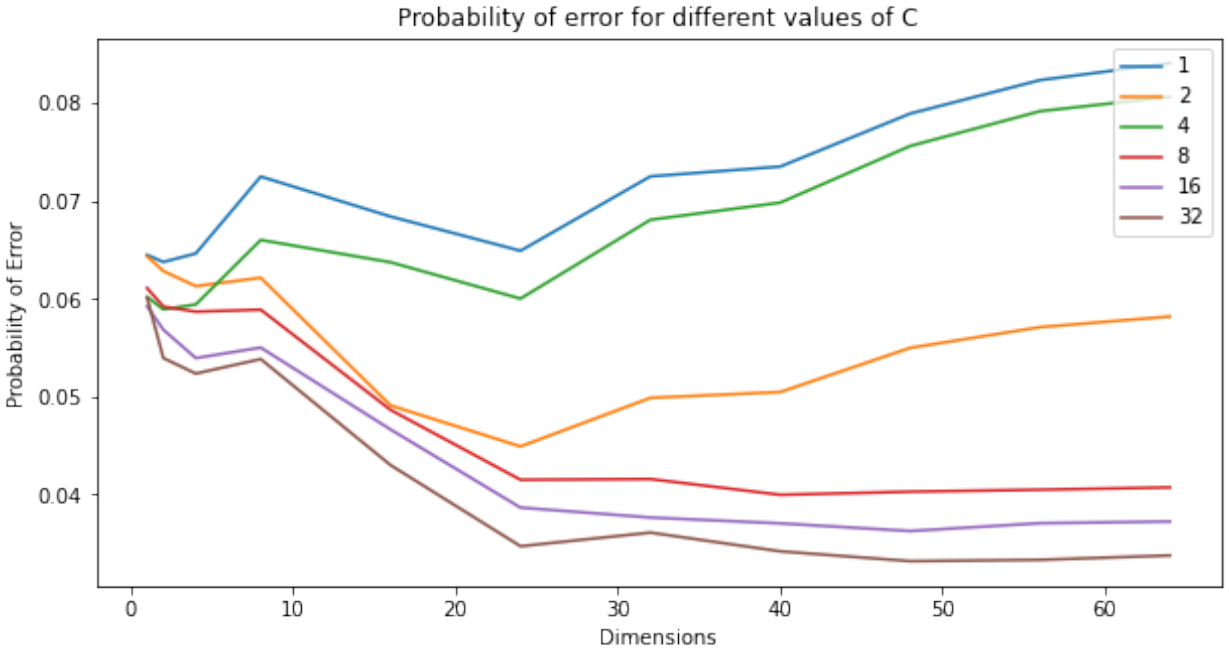
b) For each class, learn mixtures with C 2 f1; 2; 4; 8; 16; 32g. Plot the probability of error vs. dimension for each number of mixture components. What is the effect of the number of mixture components on the probability of error?

Solution:

Below is the plot for Probabilities of Error versus different mixture components C = {1, 2, 4, 8, 16, 32}

It can be observed from the plot that using just 1 components to fit the data will result in high probability of error which is expected as not all data points can be fit perfectly. For the given randomly initialized parameters, 32 components yielded the best result. As the number of components increases, PoE decreases and a similar trend is followed for larger components. When the dimension is low, regardless of the number of components used, the value of PoE is high.

For a given dimension, PoE is dependent on the value of the number of componets and the random initialization.



In [1]:
```
import numpy as np
from scipy.io import loadmat
import matplotlib.pyplot as plt
import scipy
import scipy.fftpack
import cv2
from scipy import stats
%matplotlib inline
```

In [2]:
```
# Helper functions
def dct2d(feature):
    return scipy.fftpack.dct(scipy.fftpack.dct(feature, axis=0, norm='

#ZigZag transform
zigzag = np.array([[0,1,5,6,14,15,27,28],
                   [2,4,7,13,16,26,29,42],
                   [3,8,12,17,25,30,41,43],
                   [9,11,18,24,31,40,44,53],
                   [10,19,23,32,39,45,52,54],
                   [20,22,33,38,46,51,55,60],
                   [21,34,37,47,50,56,59,61],
                   [35,36,48,49,57,58,62,63]])
zigzagFlat = zigzag.flatten()
def zig_zag_transform(a):
    result = np.zeros(64)
    for i in range(64):
        result[zigzagFlat[i]] = a[i]
    return result

# Helper function that takes in the input of components and outputs ra
def rand_init(components):
    pi = np.ones(components) * (1 / components) # Normalized values
    mu = np.random.randn(components, 64)
    cov = []
    for i in range(components):
        cov_temp = np.random.normal(5, 0.3, size=64)
        cov.append(np.diag(cov_temp))
    cov = np.array(cov)
    return pi,mu,cov

#Helper function to perform BDR
def gaussian_mixture_decision(pi_FG, mu_FG, cov_FG, pi_BG, mu_BG, cov_
    c = mu_FG.shape[0]
    im_blocks = calculate_dct(image)[:,:dim]

    #Vectorizing BDR
    p_y_x_cheetah, p_y_x_grass, A = np.zeros(247*262),np.zeros(247*262
    #Calculate foreground probability
    for k in range(c):
        p_y_x_cheetah += stats.multivariate_normal.pdf(im_blocks,mean

    #Calculate background probability
    for k in range(c):
        p_y_x_grass += stats.multivariate_normal.pdf(im_blocks,mean =
    A = p_y_x_cheetah - p_y_x_grass
    #print(A)
    A = np.where(A > 0,1,0)
    decision = np.reshape(A,(247,262))
    decision = np.lib.pad(decision,(4,4),'constant',constant_values =
```

```
    return A, decision

#Helper function to calcualte dct of input image
def calculate_dct(image):
    result = []
    for i in range(image.shape[0]-8):
        for j in range(image.shape[1]-8):
            row_start,row_end = i,i+8
            col_start,col_end = j,j+8
            block = image[row_start:row_end,col_start:col_end]
            block_dct = dct2d(block).flatten()
            block_dct= zig_zag_transform(block_dct)
            result.append(block_dct)
    result = np.array(result)
    return result

def expectation_maximization(c,sample,max_iter):
    pi,mu,cov = rand_init(c)
    for i in range(max_iter):
        # E-step
        H = []
        for j in range(c):
            H_temp = stats.multivariate_normal.pdf(sample,mean=mu[j,:]
            H.append(H_temp)
        H = np.array(H).T
        H = H / np.sum(H,axis = 1)[:,np.newaxis]
        H_sum = np.sum(H,axis = 0)
        # M-step
        pi = 1 / sample.shape[0] * H_sum
        mu_update = []
        for j in range(c):
            mu_temp = np.sum(H[:,j][:,np.newaxis] * sample,axis = 0) /
            mu_update.append(mu_temp)
        # update covariance
        cov_update = []
        for j in range(c):
            x_temp = sample - mu[j,:]
            cov_temp = np.sum((x_temp ** 2) * H[:,j][:,np.newaxis],axi
            cov_temp[cov_temp < 1e-6] = 1e-6 # Ensuring the covariance
            cov_temp = np.diag(cov_temp)
            cov_update.append(cov_temp)
        cov = np.array(cov_update)
        mu = np.array(mu_update)
    return pi,mu,cov

def probability_of_error(predicted, image_mask, rows, cols):
    return np.sum(np.absolute(image_mask - predicted)) / (rows*cols)
```

In [3]:
```
TrainingSet = loadmat('TrainingSamplesDCT_8_new.mat')
foreground,background = TrainingSet['TrainsampleDCT_FG'],TrainingSet['
```

In [4]:
```
[fg_rows, fg_cols] = foreground.shape
[bg_rows, bg_cols] = background.shape

total_samples = fg_rows + bg_rows;

#Prior calculation, used later
prior_foreground = round(fg_rows / total_samples, 4);
prior_background = round(bg_rows / total_samples, 4);
print("Foreground prior value is {0} and Background prior value is {1}
```

```
Foreground prior value is 0.1919 and Background prior value is 0.8081
```

In [5]:
```
import imageio
mask_image = cv2.imread('../cheetah_mask.bmp', 0)
mask_image = np.array(mask_image)
mask_image = mask_image / 255

input_image = cv2.imread('../cheetah.bmp', 0)
input_image = input_image / 255 - 0.23529412
image_size_row = input_image.shape[0]
image_size_col = input_image.shape[1]
```

In [6]:
```
num_mixtures = 5
num_components = 8
```

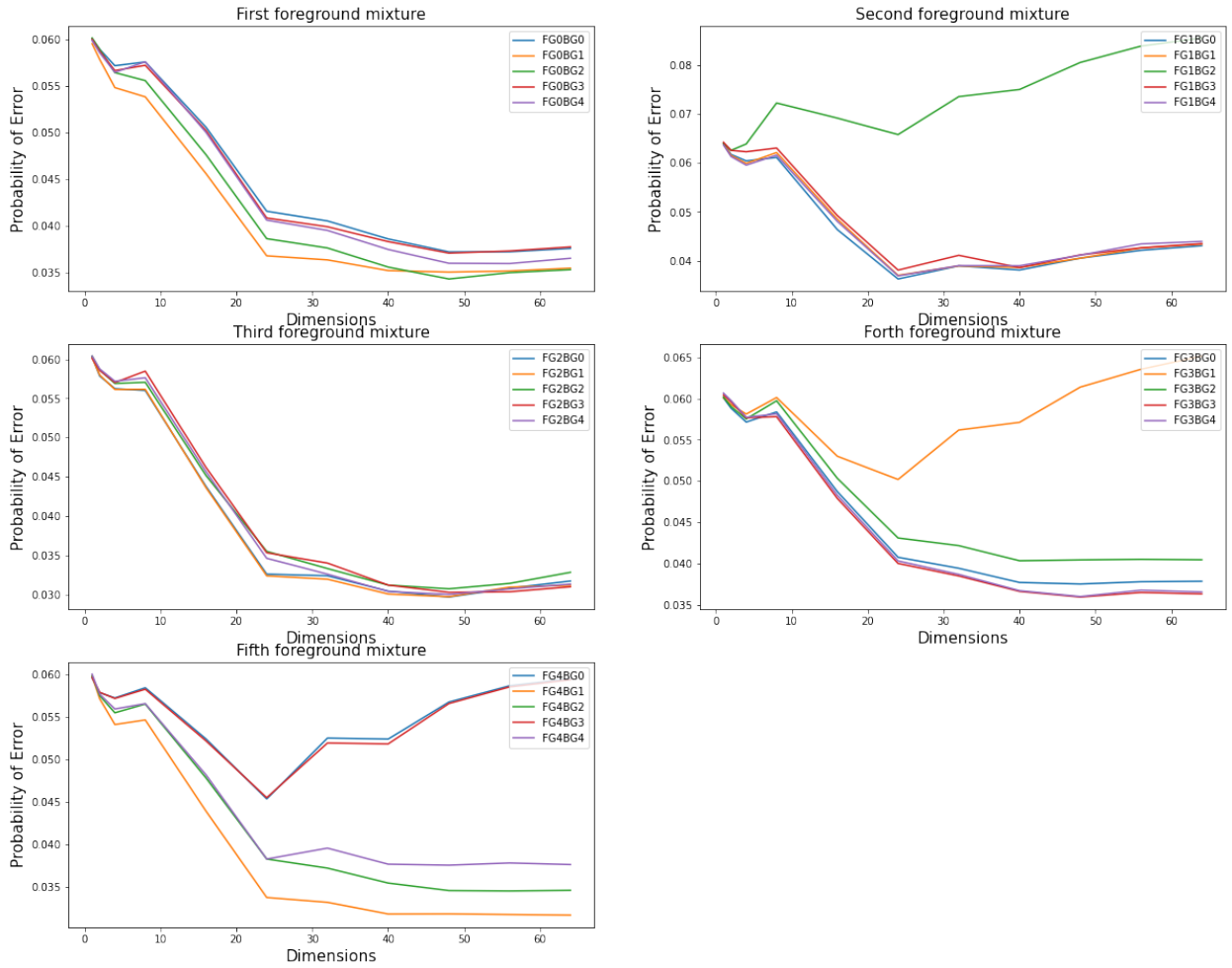
In [7]:
```
error_dic = {}
dimensions = np.array([1,2,4,8,16,24,32,40,48,56,64])
for i in range(num_mixtures):
    pi_FG,mu_FG,cov_FG = expectation_maximization(8,foreground,200)
    print("EM for 5 different BG values for FG" + str(i) + " started."
    for j in range(num_mixtures):
        pi_BG,mu_BG,cov_BG = expectation_maximization(8,background,200
        error_list = []
        for dim in dimensions:
            mu_FG_cur,cov_FG_cur = mu_FG[:,:dim],cov_FG[:,:dim,:dim]
            mu_BG_cur,cov_BG_cur = mu_BG[:,:dim],cov_BG[:,:dim,:dim]

            A, bdr_out = gaussian_mixture_decision(pi_FG,mu_FG_cur,cov
                                                   cov_BG_cur, input_image, di

            #plt.imshow(bdr_out, cmap='gray')
            #plt.show()
            error = probability_of_error(bdr_out.flatten(),mask_image.
            error_list.append(error)
            print("Processing dimension {0} completed and the error fo
        error_list = np.array(error_list)
        label = 'FG' + str(i) + 'BG' + str(j)
        error_dic[label] = error_list
```

```
Processing dimension 2 completed and the error for current dim is 0.0
61554103122730575
Processing dimension 4 completed and the error for current dim is 0.0
5991285403050109
Processing dimension 8 completed and the error for current dim is 0.0
6213507625272331
Processing dimension 16 completed and the error for current dim is 0.
04851125635439361
Processing dimension 24 completed and the error for current dim is 0.
03702251270878722
Processing dimension 32 completed and the error for current dim is 0.
03895424836601307
Processing dimension 40 completed and the error for current dim is 0.
038634713144517066
Processing dimension 48 completed and the error for current dim is 0.
040522875816993466
Processing dimension 56 completed and the error for current dim is 0.
04261437908496732
Processing dimension 64 completed and the error for current dim is 0.
04339869281045751
```

In [11]:
```
fig=plt.figure(figsize=(20,16))
voc = ['First','Second','Third','Forth','Fifth']
for i in range(num_mixtures):
    fig.add_subplot(3,2,i+1)
    plt.plot(dimensions,error_dic['FG'+str(i)+'BG0'],label = 'FG'+str(
    plt.plot(dimensions,error_dic['FG'+str(i)+'BG1'],label = 'FG'+str(
    plt.plot(dimensions,error_dic['FG'+str(i)+'BG2'],label = 'FG'+str(
    plt.plot(dimensions,error_dic['FG'+str(i)+'BG3'],label = 'FG'+str(
    plt.plot(dimensions,error_dic['FG'+str(i)+'BG4'],label = 'FG'+str(
    plt.title(voc[i] + ' foreground mixture', fontdict={ 'size'   : 15
    plt.legend(loc='upper right')
    plt.xlabel('Dimensions', fontdict={ 'size'   : 15})
    plt.ylabel('Probability of Error', fontdict={'size': 15})
```



b)

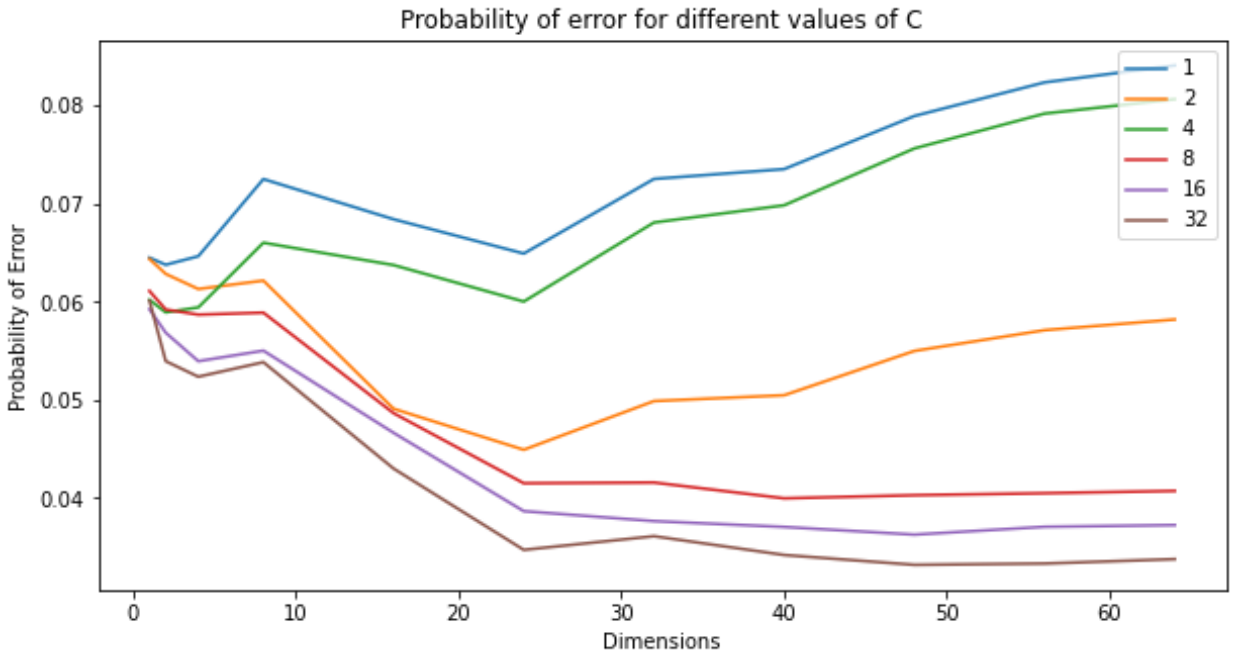
In [12]:
```
error_mixtures = {}
dimensions = np.array([1,2,4,8,16,24,32,40,48,56,64])
mixtures = np.array([1,2,4,8,16,32])
for c in mixtures:
```

```
for c in mixtures:
    error_list = []
    pi_FG, mu_FG, cov_FG = expectation_maximization(c,foreground,200)
    pi_BG, mu_BG, cov_BG = expectation_maximization(c,background,200)
    for dim in dimensions:
        mu_FG_cur,cov_FG_cur = mu_FG[:,:dim],cov_FG[:,:dim,:dim]
        mu_BG_cur,cov_BG_cur = mu_BG[:,:dim],cov_BG[:,:dim,:dim]
        A, bdr_out = gaussian_mixture_decision(pi_FG,mu_FG_cur,cov_FG_
        error = probability_of_error(bdr_out.flatten(),mask_image.flat
        error_list.append(error)
        print("Probability of error for C = {0} and dim = {1} is {2}".
    error_list = np.array(error_list)
    label = str(c)
    error_mixtures[label] = error_list
```

```
Probability of error for C = 1 and dim = 1 is 0.06450254175744372
Probability of error for C = 1 and dim = 2 is 0.0637763253449528
Probability of error for C = 1 and dim = 4 is 0.0646477850399419
Probability of error for C = 1 and dim = 8 is 0.07251997095134351
Probability of error for C = 1 and dim = 16 is 0.0684241103848947
Probability of error for C = 1 and dim = 24 is 0.06492374727668845
Probability of error for C = 1 and dim = 32 is 0.07253449527959333
Probability of error for C = 1 and dim = 40 is 0.07353667392883079
Probability of error for C = 1 and dim = 48 is 0.07895424836601307
Probability of error for C = 1 and dim = 56 is 0.08238198983297022
Probability of error for C = 1 and dim = 64 is 0.08411038489469862
Probability of error for C = 2 and dim = 1 is 0.06437182280319535
Probability of error for C = 2 and dim = 2 is 0.06284676833696441
Probability of error for C = 2 and dim = 4 is 0.06130718954248366
Probability of error for C = 2 and dim = 8 is 0.06216412490922295
Probability of error for C = 2 and dim = 16 is 0.049092229484386345
Probability of error for C = 2 and dim = 24 is 0.04490922294843863
Probability of error for C = 2 and dim = 32 is 0.04987654320987654
Probability of error for C = 2 and dim = 40 is 0.0504720406681191
Probability of error for C = 2 and dim = 48 is 0.05498910675381263
Probability of error for C = 2 and dim = 56 is 0.05710965867828613
Probability of error for C = 2 and dim = 64 is 0.058198983297022513
Probability of error for C = 4 and dim = 1 is 0.06020334059549746
Probability of error for C = 4 and dim = 2 is 0.05892519970951343
Probability of error for C = 4 and dim = 4 is 0.05943355119825708
Probability of error for C = 4 and dim = 8 is 0.06602759622367466
Probability of error for C = 4 and dim = 16 is 0.06374727668845316
Probability of error for C = 4 and dim = 24 is 0.060029048656499634
Probability of error for C = 4 and dim = 32 is 0.06809005083514888
Probability of error for C = 4 and dim = 40 is 0.06984749455337691
Probability of error for C = 4 and dim = 48 is 0.07564270152505446
Probability of error for C = 4 and dim = 56 is 0.07920116194625998
Probability of error for C = 4 and dim = 64 is 0.08066811909949165
Probability of error for C = 8 and dim = 1 is 0.0611038489469862
Probability of error for C = 8 and dim = 2 is 0.05921568627450981
Probability of error for C = 8 and dim = 4 is 0.05869281045751634
```

```
Probability of error for C = 8 and dim = 8 is 0.058896151053013795
Probability of error for C = 8 and dim = 16 is 0.04864197530864198
Probability of error for C = 8 and dim = 24 is 0.0414960058097313
Probability of error for C = 8 and dim = 32 is 0.04156862745098039
Probability of error for C = 8 and dim = 40 is 0.03995642701525055
Probability of error for C = 8 and dim = 48 is 0.04027596223674655
Probability of error for C = 8 and dim = 56 is 0.04047930283224401
Probability of error for C = 8 and dim = 64 is 0.0407116920842411
Probability of error for C = 16 and dim = 1 is 0.05924473493100944
Probability of error for C = 16 and dim = 2 is 0.0568482207697894
Probability of error for C = 16 and dim = 4 is 0.05394335511982571
Probability of error for C = 16 and dim = 8 is 0.055018155410312276
Probability of error for C = 16 and dim = 16 is 0.04663761801016703
Probability of error for C = 16 and dim = 24 is 0.038649237472766884
Probability of error for C = 16 and dim = 32 is 0.03763253449527959
Probability of error for C = 16 and dim = 40 is 0.037037037037037035
Probability of error for C = 16 and dim = 48 is 0.03625272331154684
Probability of error for C = 16 and dim = 56 is 0.03705156136528685
Probability of error for C = 16 and dim = 64 is 0.03722585330428468
Probability of error for C = 32 and dim = 1 is 0.06011619462599855
Probability of error for C = 32 and dim = 2 is 0.05394335511982571
Probability of error for C = 32 and dim = 4 is 0.0523602033405955
Probability of error for C = 32 and dim = 8 is 0.05384168482207698
Probability of error for C = 32 and dim = 16 is 0.0429920116194626
Probability of error for C = 32 and dim = 24 is 0.03469862018881627
Probability of error for C = 32 and dim = 32 is 0.03609295570079884
Probability of error for C = 32 and dim = 40 is 0.0341757443718228
Probability of error for C = 32 and dim = 48 is 0.03317356572258533
Probability of error for C = 32 and dim = 56 is 0.0333042846768337
Probability of error for C = 32 and dim = 64 is 0.03375453885257807
```

In [14]:
```
plt.figure(figsize=(10, 5))
for i in range(mixtures.shape[0]):
    plt.plot(dimensions, error_mixtures[str(2**(i))],label = str(2**(i
plt.title('Probability of error for different values of C')
plt.legend(loc='upper right')
plt.xlabel('Dimensions', )
plt.ylabel('Probability of Error')
plt.show()
```



Type *Markdown* and LaTeX: α^2