Name: Joyce Le
ID#: 82549113
UCI NetID: lejy

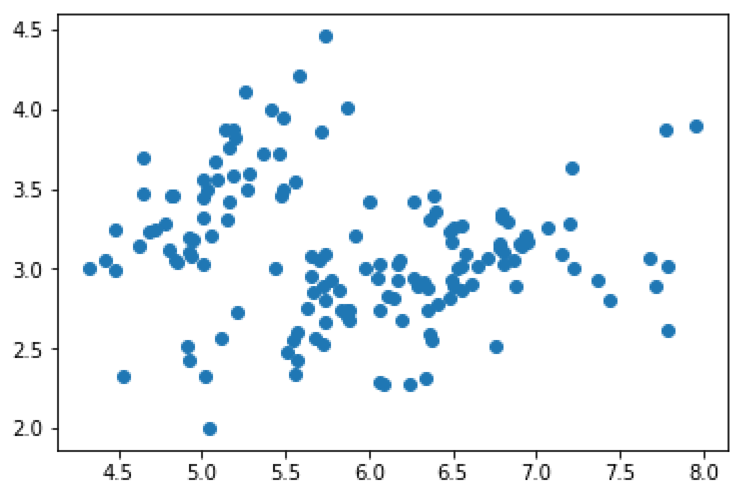
CS 178 Homework 5

Problem 1: Clustering

In [1]:
```
import numpy as np
import mltools as ml
import matplotlib.pyplot as plt
```

In [2]:
```
print('1)')

# load data
iris = np.genfromtxt("data/iris.txt",delimiter=None)
X = iris[:,0:2]
Y = iris[:,-1]

# select first two features and plot them
plt.figure()
plt.scatter(X[:, 0], X[:, 1])
plt.show()
```

```
1)
```

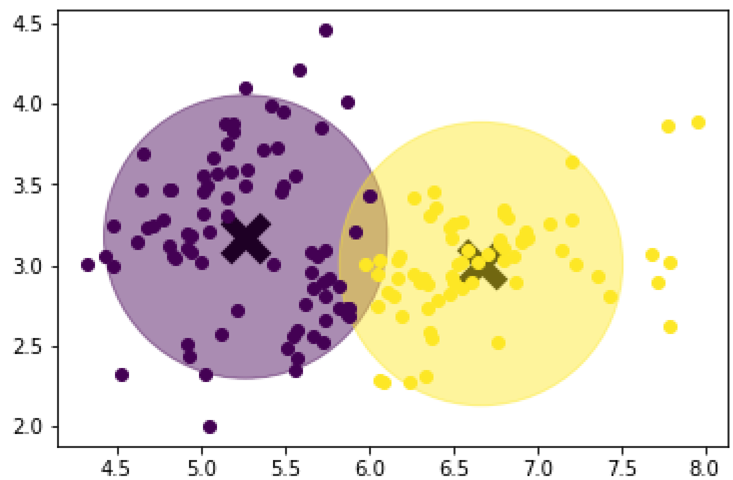


The data does look to be clustered. I think two clusters exist because the data for these two clusters seem to be nicely divided. There could also possibly be another third cluster at the bottom beccause the bottom cluster is quite large.

In [3]:
```
print('2)\n')
print('k=2:\n')

ssd_best = np.inf
for i in range(5):
    Z_i, mu_i, ssd_i = ml.cluster.kmeans(X, K=2, init='k++', max_iter=5)
    print(ssd_i)
    # different initializations do NOT all find the same solution
    # pick best one
    if ssd_i < ssd_best:
        Z, mu, ssd_best = Z_i, mu_i, ssd_i
ml.plotClassify2D(None,X,Z)
plt.scatter(mu[:, 0], mu[:, 1], s=400, marker='x', facecolor='black', lw=8)
plt.scatter(mu[:, 0], mu[:, 1], s=20000, alpha=.45, c=np.unique(Z))
```

```
2)

k=2:

57.87966196118197
57.87966196118197
57.87966196118197
58.01355452847487
57.87966196118197
```

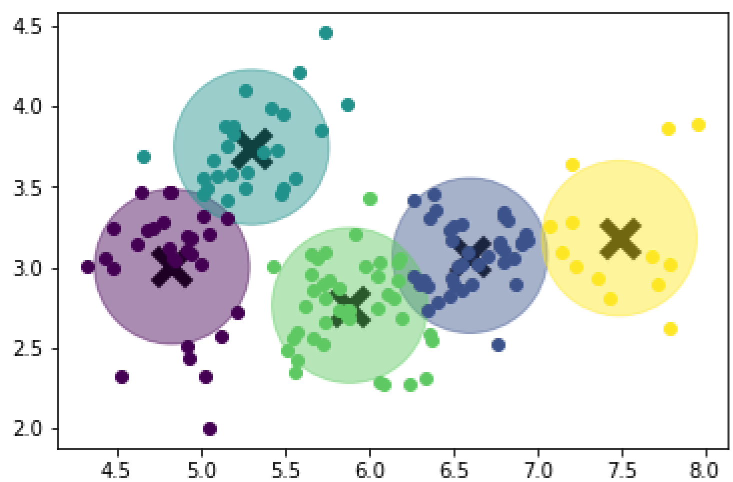
Out[3]:
```
<matplotlib.collections.PathCollection at 0x21d89d58860>
```



In [4]:
```
print('k=5:\n')

ssd_best = np.inf
for i in range(5):
    Z_i, mu_i, ssd_i = ml.cluster.kmeans(X, K=5, init='k++', max_iter=5)
    print(ssd_i)
    # different initializations do NOT all find the same solution
    # pick best one
    if ssd_i < ssd_best:
        Z, mu, ssd_best = Z_i, mu_i, ssd_i
ml.plotClassify2D(None,X,Z)
plt.scatter(mu[:, 0], mu[:, 1], s=300, marker='x', facecolor='black', lw=6)
plt.scatter(mu[:, 0], mu[:, 1], s=6000, alpha=.45, c=np.unique(Z))
```

```
k=5:

21.325838512733114
20.90559762715885
28.963012289610635
25.109620287979148
21.567280736715215
```

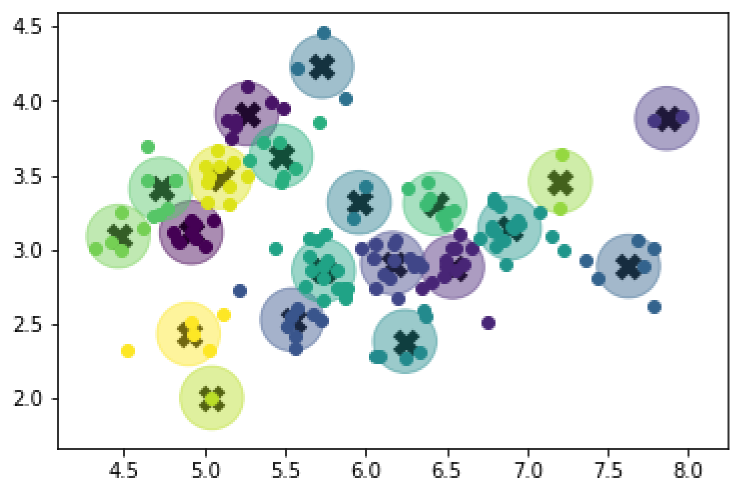
Out[4]:
```
<matplotlib.collections.PathCollection at 0x21d89c86208>
```



In [5]:
```
print('k=20:\n')

ssd_best = np.inf
for i in range(5):
    Z_i, mu_i, ssd_i = ml.cluster.kmeans(X, K=20, init='k++', max_iter=5)
    print(ssd_i)
    # different initializations do NOT all find the same solution
    # pick best one
    if ssd_i < ssd_best:
        Z, mu, ssd_best = Z_i, mu_i, ssd_i
ml.plotClassify2D(None,X,Z)
plt.scatter(mu[:, 0], mu[:, 1], s=100, marker='x', facecolor='black', lw=5)
plt.scatter(mu[:, 0], mu[:, 1], s=1000, alpha=.45, c=np.unique(Z))
```

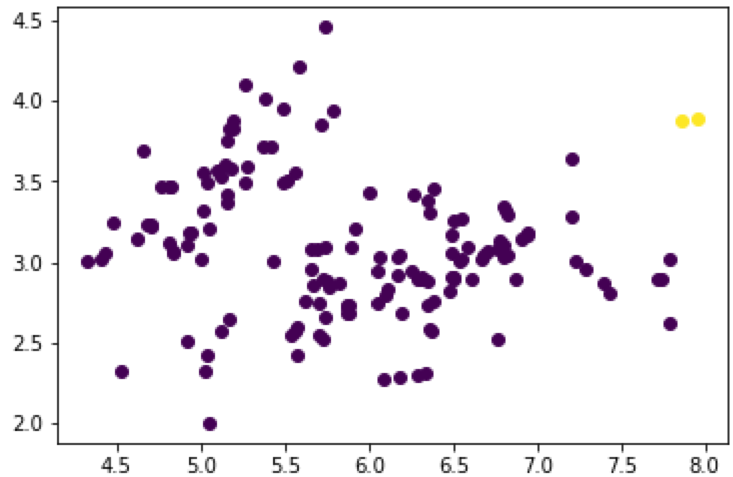
```
k=20:

4.4800293480559255
4.513257894284805
4.258703733850255
5.282172698257471
4.1895286826698515
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x21d89e05fd0>
```

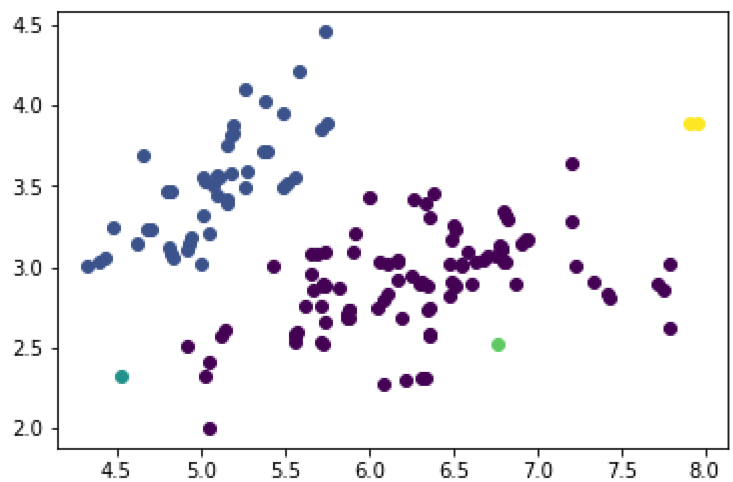


In [6]:
```
print('3)')

# single linkage
for k in [2,5,20]:
    print('Single linkage, k =', k)
    Z, dendogram = ml.cluster.agglomerative(X, K=k, method='min')
    plt.figure()
    ml.plotClassify2D(None,X,Z)
    plt.show()

# complete linkage
for k in [2,5,20]:
    print('Complete linkage, k =', k)
    Z, dendogram = ml.cluster.agglomerative(X, K=k, method='max')
    plt.figure()
    ml.plotClassify2D(None,X,Z)
    plt.show()
```

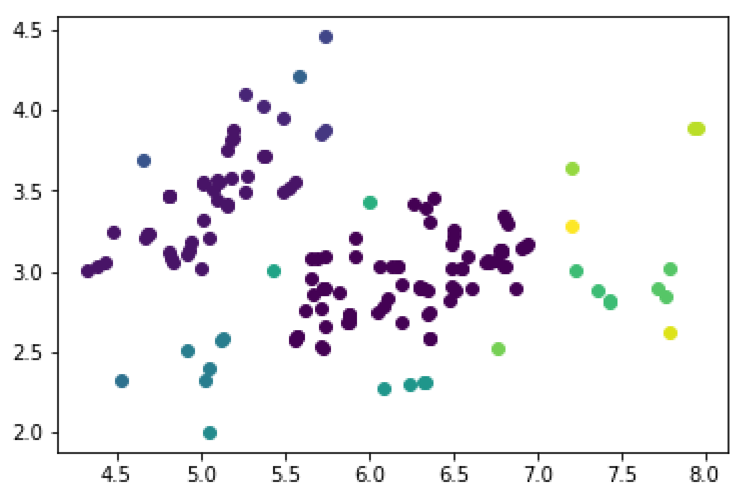
3)
Single linkage, k = 2



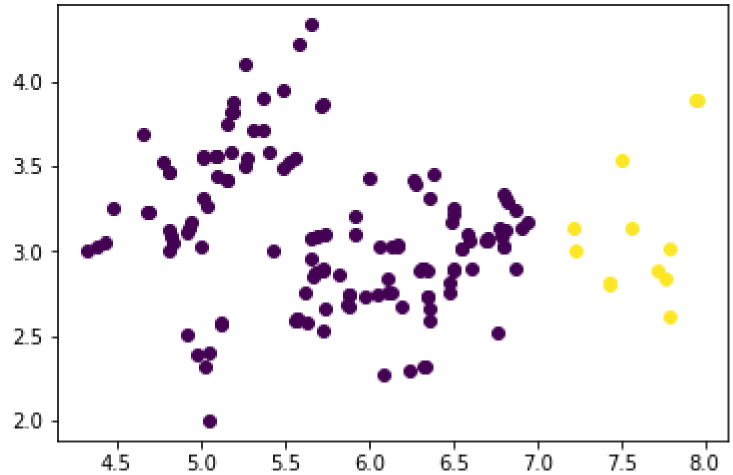
Single linkage, k = 5



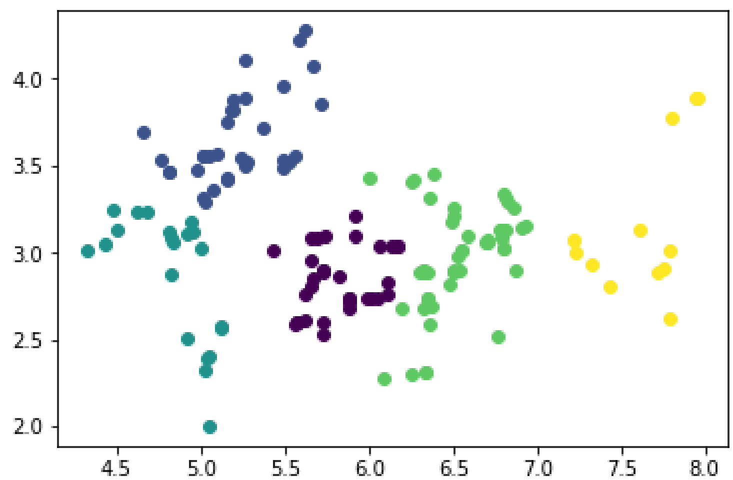
Single linkage, k = 20



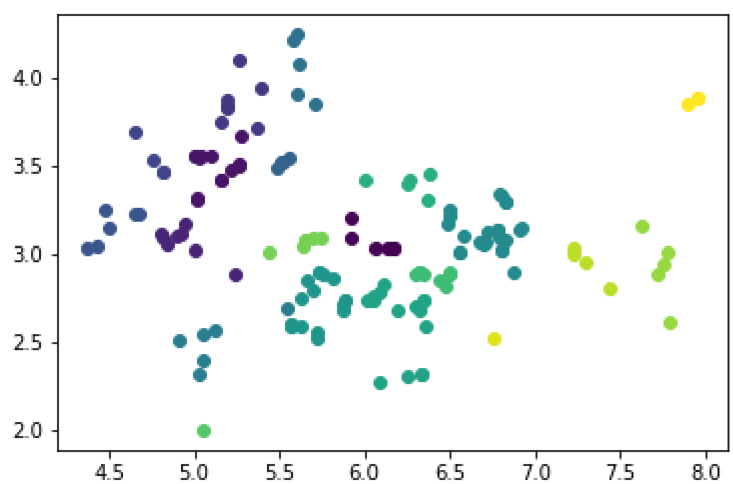
Complete linkage, k = 2



Complete linkage, k = 5



Complete linkage, k = 20

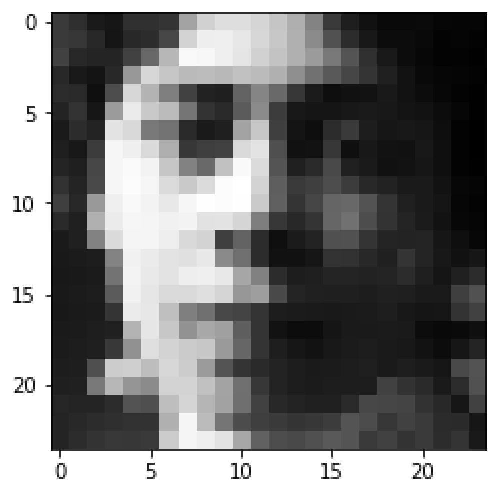


4) K-means is different from agglomerative because k-means produces clusters that are more circular and similar to each other. Agglomerative clustering forms clusters that are differently shaped and sized according to each data cluster. This is more true for single-linkage agglomerative clustering than it is complete linkage. The similarities are that k-means clustering is similar to complete-linkage agglomerative clustering.

Problem 2: EigenFaces

In [7]:
```
X = np.genfromtxt("data/faces.txt", delimiter=None) # load face dataset
plt.figure()
# pick a data point i for display
img = np.reshape(X[i,:],(24,24)) # convert vectorized data to 24x24 image p
atches
plt.imshow( img.T , cmap="gray") # display image patch; you may have to squ
int
```

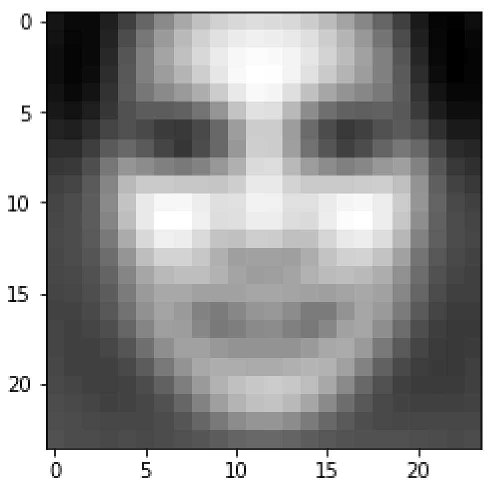
Out[7]:
```
<matplotlib.image.AxesImage at 0x21d8a1628d0>
```



In [8]:
```
print('1)')

mu = X.mean(axis=0)
X_0 = X - mu
plt.figure()
img = np.reshape(mu, (24,24))
plt.imshow(img.T, cmap='gray')
```

```
1)
```

Out[8]: <matplotlib.image.AxesImage at 0x21d9a384e48>



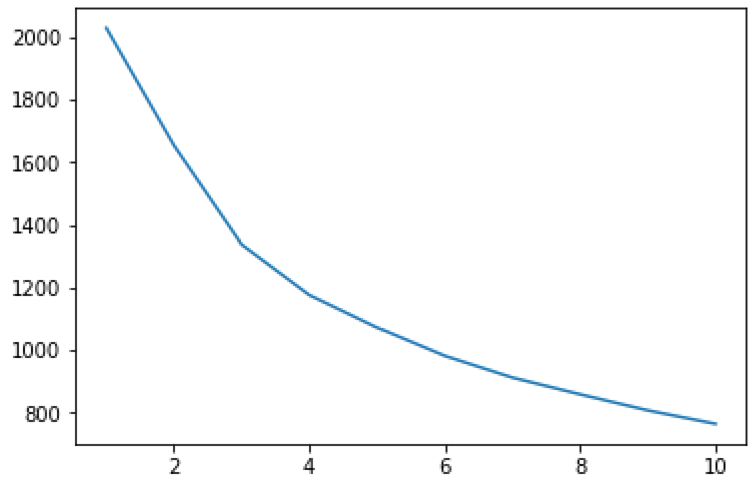
In [9]:
```
print('2)')
import scipy.linalg

u,s,v = scipy.linalg.svd(X_0, full_matrices=False)
w = u.dot(np.diag(s))
print('W shape:', w.shape)
print('V shape:', v.shape)
```

```
2)
W shape: (4916, 576)
V shape: (576, 576)
```

In [10]:
```
print('3)')

mse = np.zeros(10)
for k in range(1,11):
    xhat_0 = w[:,:k].dot(v[:k,:])
    mse[k-1] = ((X_0-xhat_0)**2).mean()

# plot mse as a function of k
plt.plot(range(1,11), mse)
```

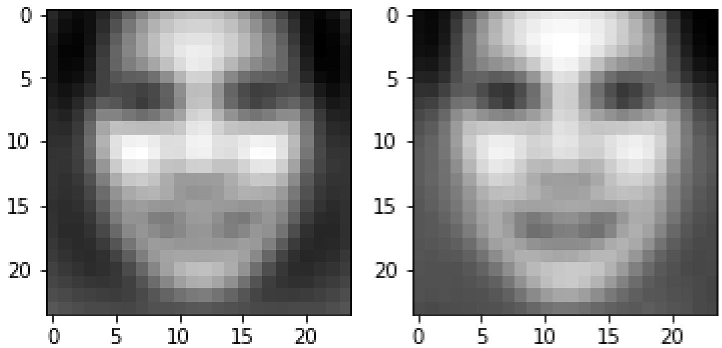
```
3)
```

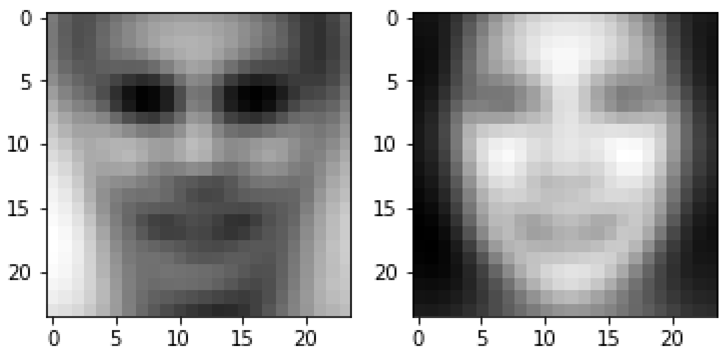
Out[10]:
```
[<matplotlib.lines.Line2D at 0x21d8a132128>]
```

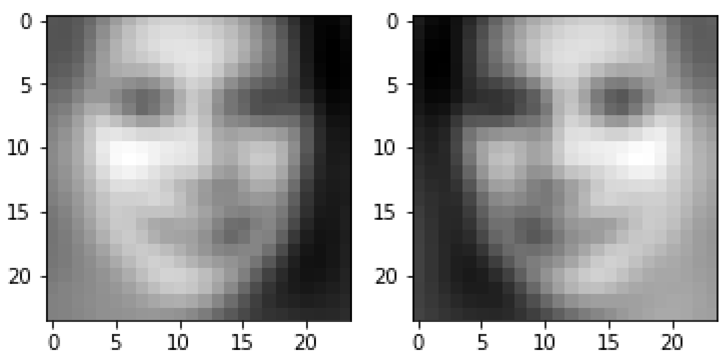


In [11]:
```
print('4)')

for j in range(3):
    a = 2*np.median(np.abs(w[:,j]))
    plus = np.reshape(mu + a*v[j,:], (24,24))
    minus = np.reshape(mu - a*v[j,:], (24,24))
    f,ax = plt.subplots(1,2)
    ax[0].imshow(plus.T, cmap='gray')
    ax[1].imshow(minus.T, cmap='gray')
```

4)

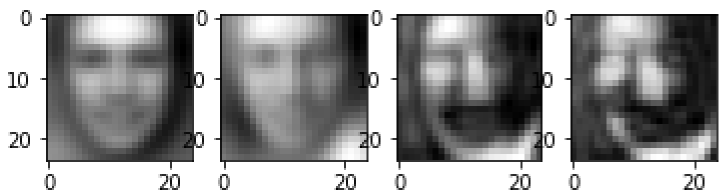


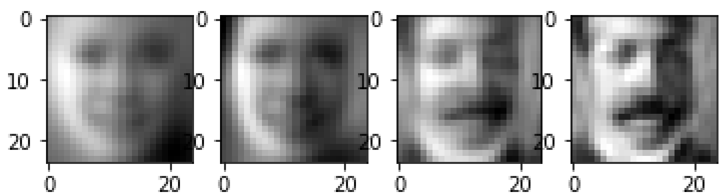




In [12]:
```
print('5)')

for i in range(2):
    f,ax = plt.subplots(1,4)
    for j,k in enumerate([5,10,50,100]):
        image = np.reshape(mu + w[i,0:k].dot(v[0:k,:]), (24,24))
        ax[j].imshow(image.T, cmap='gray')
```

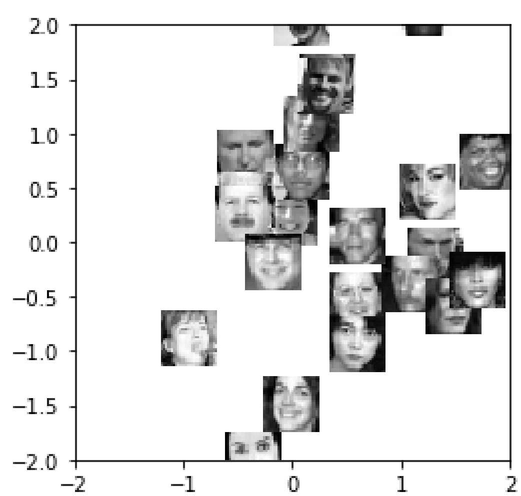
5)





In [14]:
```
print('6)')

idx = np.random.randint(X.shape[0]-1, size=25)

import mltools.transforms
coord,params = ml.transforms.rescale( w[:,0:2] ) # normalize scale of "W" l
ocations
plt.figure();
for i in idx:
    # compute where to place image (scaled W values) & size
    loc = (coord[i,0],coord[i,0]+0.5, coord[i,1],coord[i,1]+0.5)
    img = np.reshape( X[i,:], (24,24) ) # reshape to square
    plt.imshow( img.T , cmap="gray", extent=loc ) # draw each image
    plt.axis( (-2,2,-2,2) ) # set axis to a reasonable scale
```

6)



Statement of Collaboration

I did not collaborate with anyone on this assignment. I only looked on CampusWire for help on how to do some parts of Problem 2.