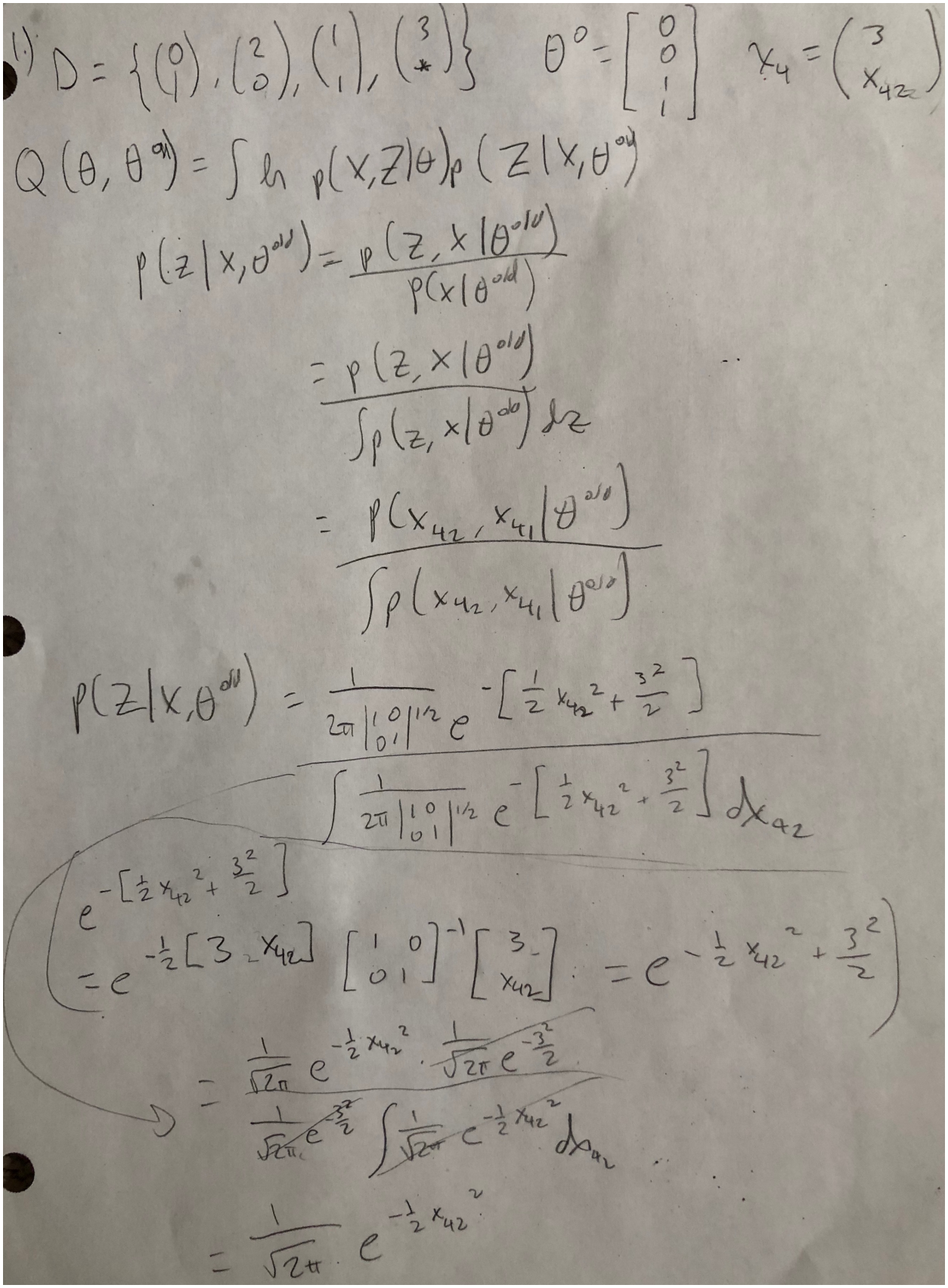
In [82]:
```
import numpy as np
import scipy.io as sio
import matplotlib
import matplotlib.pyplot as plt
from numpy.matlib import repmat
from sklearn.cluster import KMeans

%matplotlib inline
```

1



1.) $D = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ * \end{pmatrix} \right\}$ $\theta^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ $x_4 = \begin{pmatrix} 3 \\ x_{42} \end{pmatrix}$

$$Q(\theta, \theta^{old}) = \int \ln p(X, Z|\theta) p(Z|X, \theta^{old})$$

$$p(z|x, \theta^{old}) = \frac{p(z, x|\theta^{old})}{p(x|\theta^{old})}$$

$$= \frac{p(z, x|\theta^{old})}{\int p(z, x|\theta^{old}) dz}$$

$$= \frac{p(x_{42}, x_{41}|\theta^{old})}{\int p(x_{42}, x_{41}|\theta^{old})}$$

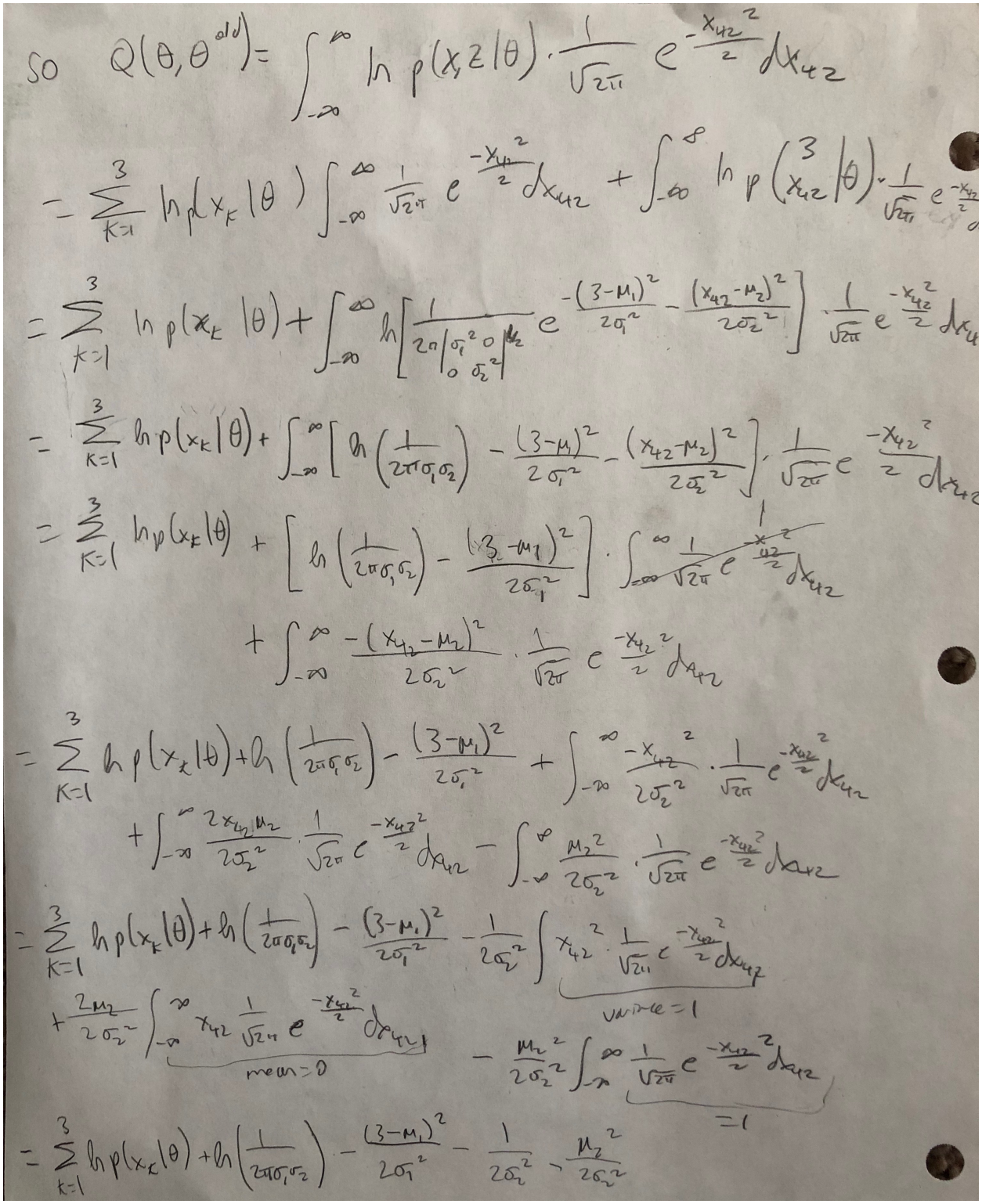
$$p(z|x, \theta^{old}) = \frac{\frac{1}{2\pi \left|\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right|^{1/2}} e^{-\left[\frac{1}{2} x_{42}^2 + \frac{3^2}{2}\right]}}{\int \frac{1}{2\pi \left|\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right|^{1/2}} e^{-\left[\frac{1}{2} x_{42}^2 + \frac{3^2}{2}\right]} dx_{42}}$$

$$e^{-\left[\frac{1}{2} x_{42}^2 + \frac{3^2}{2}\right]}$$

$$= e^{-\frac{1}{2}\begin{bmatrix} 3 & x_{42} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ x_{42} \end{bmatrix}} = e^{-\frac{1}{2} x_{42}^2 + \frac{3^2}{2}}$$

$$= \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_{42}^2} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{3^2}{2}}}{\frac{1}{\sqrt{2\pi}} e^{-\frac{3^2}{2}} \int \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_{42}^2} dx_{42}}$$

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_{42}^2}$$



$$\text{So } Q(\theta,\theta^{old}) = \int_{-\infty}^{\infty} \ln p(x,z|\theta)\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42} + \int_{-\infty}^{\infty} \ln p\left(\begin{matrix}3\\ x_{42}\end{matrix}\middle|\theta\right)\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} d$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \int_{-\infty}^{\infty} \ln\left[\frac{1}{2\pi\begin{vmatrix}\sigma_1^2 & 0\\ 0 & \sigma_2^2\end{vmatrix}^{\frac{1}{2}}} e^{-\frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{(x_{42}-\mu_2)^2}{2\sigma_2^2}}\right]\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \int_{-\infty}^{\infty}\left[\ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{(x_{42}-\mu_2)^2}{2\sigma_2^2}\right]\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \left[\ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{(3-\mu_1)^2}{2\sigma_1^2}\right]\cdot\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$+ \int_{-\infty}^{\infty} -\frac{(x_{42}-\mu_2)^2}{2\sigma_2^2}\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{(3-\mu_1)^2}{2\sigma_1^2} + \int_{-\infty}^{\infty} -\frac{x_{42}^2}{2\sigma_2^2}\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$+ \int_{-\infty}^{\infty} \frac{2x_{42}\mu_2}{2\sigma_2^2}\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42} - \int_{-\infty}^{\infty} \frac{\mu_2^2}{2\sigma_2^2}\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{1}{2\sigma_2^2}\underbrace{\int x_{42}^2\cdot\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}}_{\text{variance}=1}$$

$$+ \frac{2\mu_2}{2\sigma_2^2}\underbrace{\int_{-\infty}^{\infty} x_{42}\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}}_{\text{mean}=0} - \frac{\mu_2^2}{2\sigma_2^2}\underbrace{\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}} e^{-\frac{x_{42}^2}{2}} dx_{42}}_{=1}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{1}{2\sigma_2^2} - \frac{\mu_2^2}{2\sigma_2^2}$$

$$= \sum_{k=1}^{3} \ln \frac{1}{2\pi\sigma_1\sigma_2} - \left[\frac{(x_{k1}-\mu_1)^2}{2\sigma_1^2} + \frac{(x_{k2}-\mu_2)^2}{2\sigma_2^2} \right] - \frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{1}{2\sigma_2^2}$$

$$- \frac{\mu_2^2}{2\sigma_2^2} + \ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right)$$

$$= 4\ln\left(\frac{1}{2\pi\sigma_1\sigma_2}\right) - \frac{\mu_1^2}{2\sigma_1^2} - \frac{(1-\mu_2)^2}{2\sigma_2^2} - \frac{(2-\mu_1)^2}{2\sigma_1^2} - \frac{\mu_2^2}{2\sigma_2^2} - \frac{(1-\mu_1)^2}{2\sigma_1^2}$$

$$- \frac{(1-\mu_2)^2}{2\sigma_2^2} - \frac{(3-\mu_1)^2}{2\sigma_1^2} - \frac{1}{2\sigma_1^2} - \frac{\mu_2^2}{2\sigma_2^2}$$

$$\frac{\partial(Q(\theta,\theta^{old}))}{\partial\mu_2} = \frac{2(1-\mu_2)}{2\sigma_2^2} - \frac{2\mu_2}{2\sigma_2^2} + \frac{2(1-\mu_2)}{2\sigma_2^2} - \frac{2\mu_2}{2\sigma_2^2}$$

$$0 = 2 - 2\mu_2 - 2\mu_2 - 2\mu_2 - 2\mu_2 + 2$$

$$-4 = -8\mu_2 \qquad \mu_2 = \frac{1}{2}$$

2

2.) Bishop (9.10): $p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$

(9.11): $p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}$

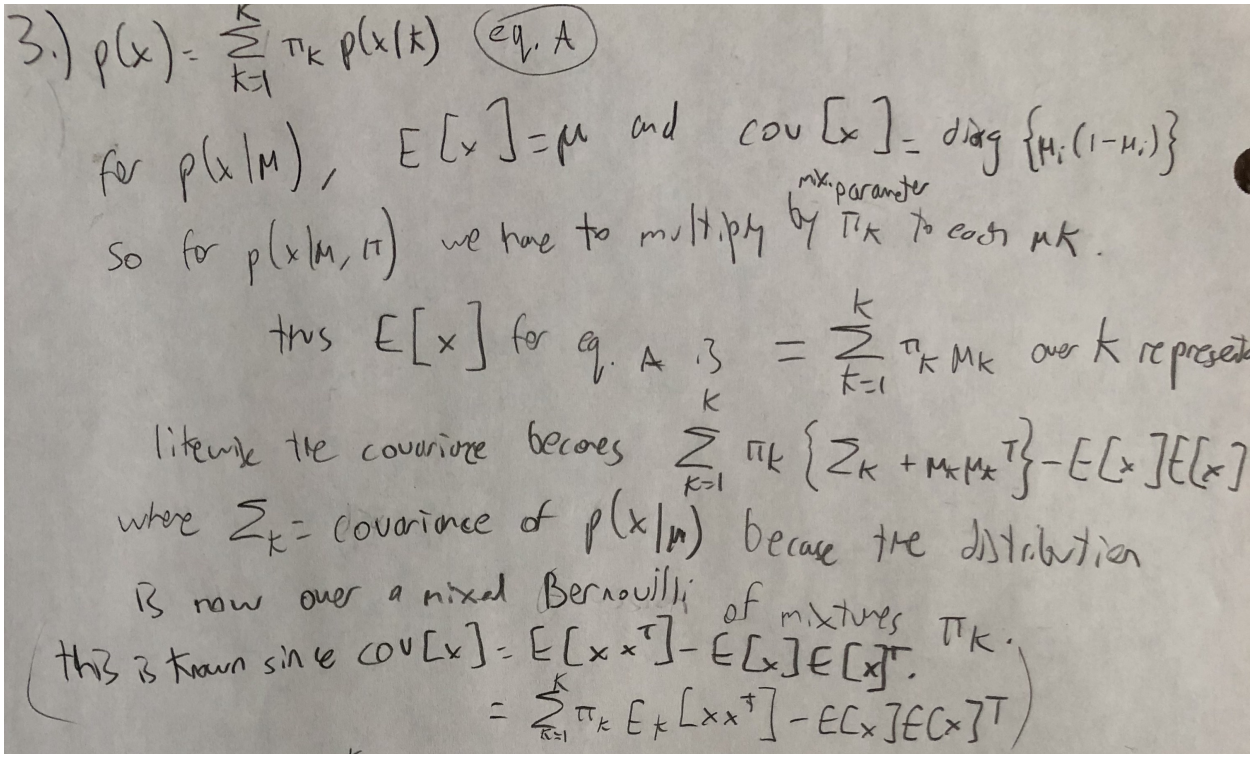
(9.12) $p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$

a mixture of Gaussians takes the form $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$ (9.7)

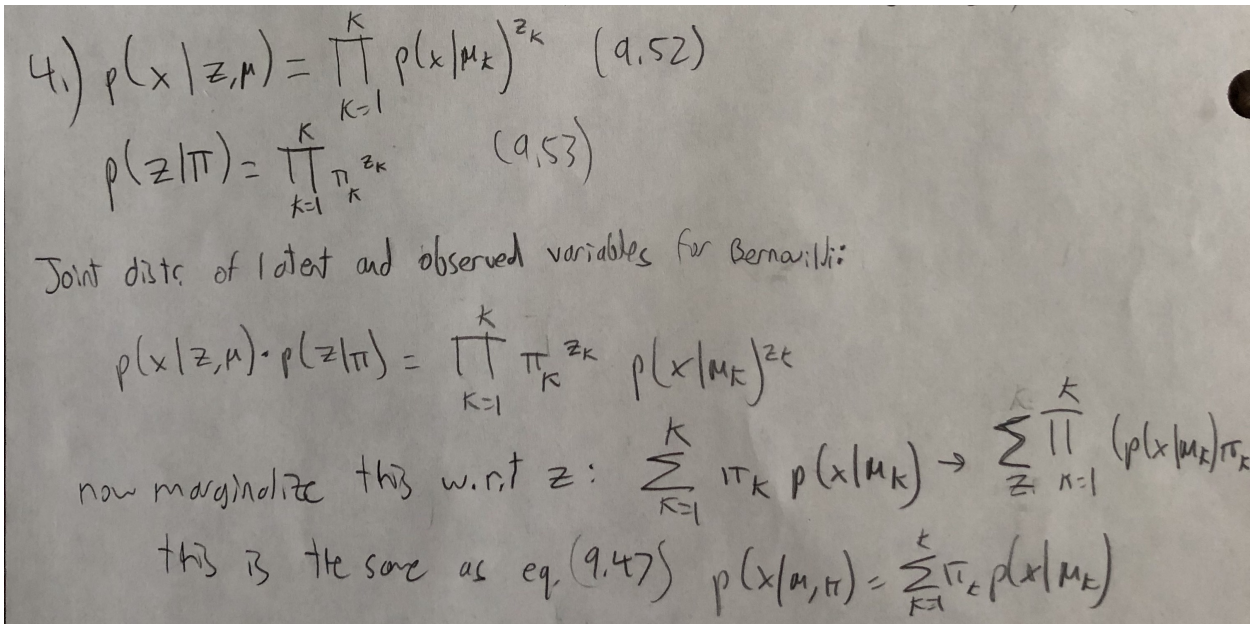
So 9.12, the marginal distribution, is clearly a Gaussian mixture model as we are summing over the K representations and multiplying a mixing coefficient π_k to each normal distribution $\mathcal{N}(x|\mu_k, \Sigma_k)$

$$\sum_z p(z)p(x|z) = \sum_z \prod_{k=1}^{K} (\pi_k \mathcal{N}(x|\mu_k, \Sigma_k))^{z_k} \rightarrow \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

3


3.) $p(x) = \sum_{k=1}^{K} \pi_k p(x|k)$ eq. A
for $p(x|\mu)$, $E[x] = \mu$ and $cov[x] = diag\{\mu_i(1-\mu_i)\}$
mix. parameter
So for $p(x|\mu, \pi)$ we have to multiply by π_k to each μ_k.
thus $E[x]$ for eq. A is $= \sum_{k=1}^{K} \pi_k \mu_k$ over k represents
likewise the covariance becomes $\sum_{k=1}^{K} \pi_k \{\Sigma_k + \mu_k \mu_k^T\} - E[x]E[x]$
where Σ_k = covariance of $p(x|\mu)$ because the distribution
is now over a mixed Bernoulli of mixtures π_k.
this is known since $cov[x] = E[xx^T] - E[x]E[x]^T$.
$= \sum_{k=1}^{K} \pi_k E_k[xx^T] - E[x]E[x]^T$

4


4.) $p(x|z,\mu) = \prod_{k=1}^{K} p(x|\mu_k)^{z_k}$ (9.52)
$p(z|\pi) = \prod_{k=1}^{K} \pi_k^{z_k}$ (9.53)
Joint dist. of latent and observed variables for Bernoulli:
$p(x|z,\mu) \cdot p(z|\pi) = \prod_{k=1}^{K} \pi_k^{z_k} p(x|\mu_k)^{z_k}$
now marginalize this w.r.t z: $\sum_{k=1}^{K} \pi_k p(x|\mu_k) \rightarrow \sum_{z} \prod_{k=1}^{K} (p(x|\mu_k)\pi_k$
this is the same as eq. (9.47) $p(x|\mu,\pi) = \sum_{k=1}^{K} \pi_k p(x|\mu_k)$

5

5.) $E_z[\ln p(X, Z|\mu, \pi)] = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})\left\{\ln \pi_k + \sum_{i=1}^{D}[x_{ni}\ln \mu_{ki} + (1-x_{ni})\ln(1-\mu_{ki})]\right\}$ (9.55)

where $\gamma(z_{nk}) = E[z_{nk}]$ is responsibility.

$$= \frac{\pi_k p(x_n|\mu_k)}{\sum_{j=1}^{K}\pi_j p(x_n|\mu_j)}$$

take $\frac{\partial(E_z[\ln p(X, Z, \mu, \pi)}{\partial \mu_k} = \sum_{n=1}^{N}\gamma(z_{nk})\left(\frac{x_{ni}}{\mu_{ki}} - \frac{1-x_{ni}}{1-\mu_{ki}}\right)$

$$0 = \frac{\sum_n \gamma(z_{nk})\cdot x_{ni} - \sum_n \gamma(z_{nk})\mu_{ki}}{(1-\mu_{ki})\cdot \mu_{ki}}$$

$$\sum_n \gamma(z_{nk}) x_{ni} = \sum_n \gamma(z_{nk})\mu_{ki}$$

$$\mu_{ki} = \frac{\sum_n \gamma(z_{nk}) x_{ni}}{\sum_n \gamma(z_{nk})} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk}) x_n$$

but $\bar{x}_k$ is $\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$

So $\mu_k = \bar{x}_k$ in vector form.

6

6.) $A \cdot A^T \cdot v = \lambda \cdot v$

multiply both sides by A^T

$A^T A \cdot A^T \cdot v = A^T \cdot \lambda \cdot v$

$A^T A \cdot (A^T \cdot v) = \lambda \cdot (A^T \cdot v)$

Thus $A^T \cdot v = A'v$ is an eigen vector of $A'A$

eigsort.m

In [83]:
```
# [Vsort,Dsort] = eigsort(V, eigvals)
#
# Sorts a matrix eigenvectors and a array of eigenvalues in order
# of eigenvalue size, largest eigenvalue first and smallest eigenvalue
# last.
#
# Example usage:
# di, V = np.linarg.eig(L)
# Vnew, Dnew = eigsort(V, di)
#
# Tim Marks 2002
```

In [84]:
```
def eigsort(V, eigvals):

    # Sort the eigenvalues from largest to smallest. Store the sorted
    # eigenvalues in the column vector lambd.
    lohival = np.sort(eigvals)
    lohiindex = np.argsort(eigvals)
    lambd = np.flip(lohival)
    index = np.flip(lohiindex)
    Dsort = np.diag(lambd)

    # Sort eigenvectors to correspond to the ordered eigenvalues. Store sor
    # eigenvectors as columns of the matrix vsort.
    M = np.size(lambd)
    Vsort = np.zeros((M, M))
    for i in range(M):
        Vsort[:,i] = V[:,index[i]]
    return Vsort, Dsort
```

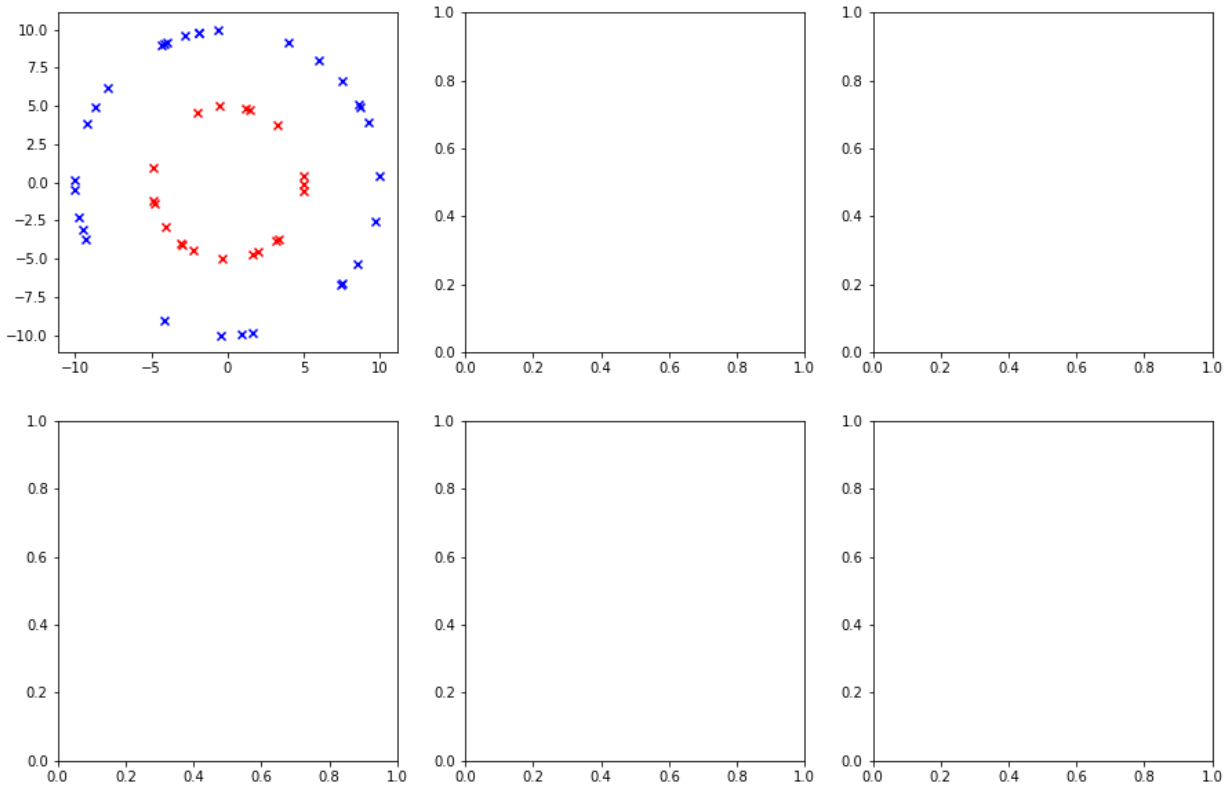
spectclust.m

In [85]:
```
efs
1]   Shi, J., and J. Malik (1997) "Normalized Cuts and Image Segmentation",
 in Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition,
 Puerto Rico.

2]   Kannan, R., S. Vempala, and A. Vetta (2000) "On Clusterings - Good, Bac

his code is from ref [3]
3]   Ng, A. Y., M. I. Jordan, and Y. Weiss (2001) "On Spectral Clustering:
 Analysis and an algorithm", in Advances in Neural Information Processing%%


4]   Weiss, Y. (1999) "Segmentation using eigenvectors: a unifying view",
 Tech. Rep., CS. Dept., UC Berkeley.
```

In [86]:
```
#For question 9 uncomment this code to make your own random data of the sam
r1 = 5
r2 = 10
set1 = np.random.uniform(low=0,high=2*np.pi,size=(1,20))
set2 = np.random.uniform(low=0,high=2*np.pi,size=(1,30))
d1 = np.vstack((r1*np.cos(set1),r1*np.sin(set1)))
d2 = np.vstack((r2*np.cos(set2),r2*np.sin(set2)))
```

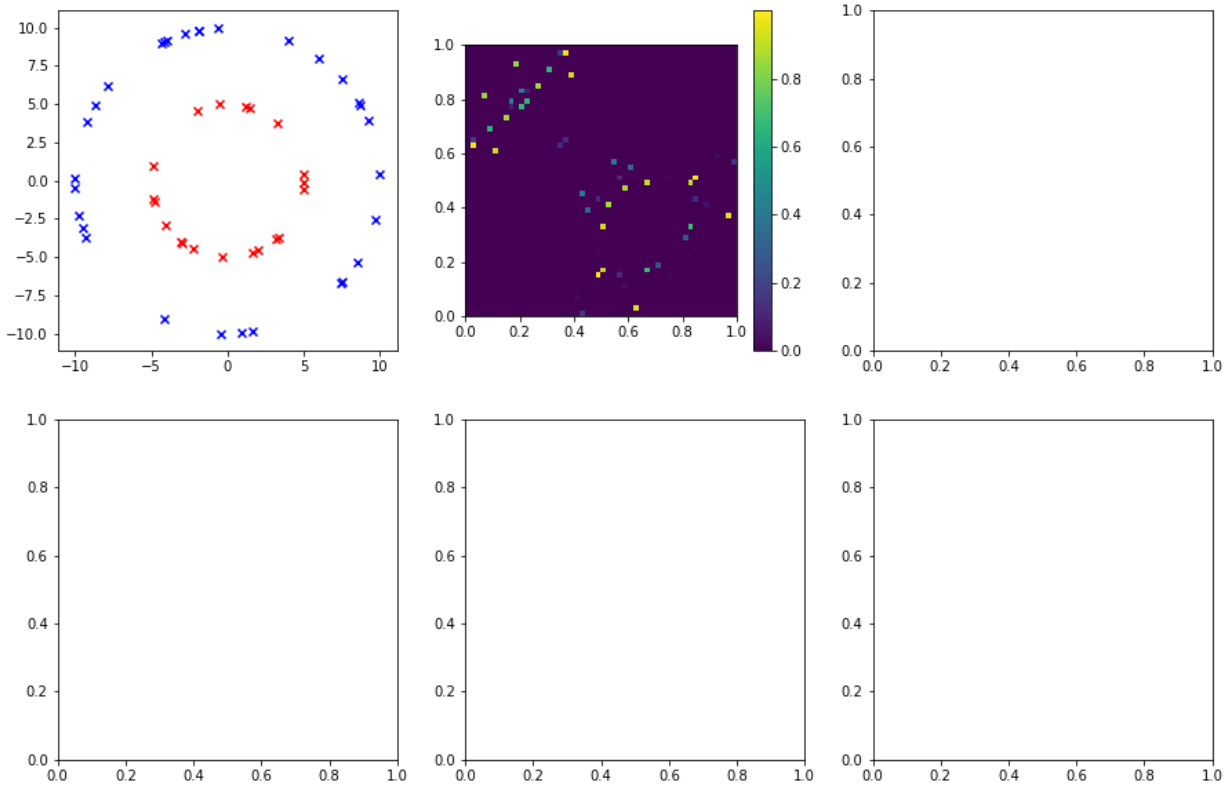
In [87]:
```
# # For question 9 comment out the next 3 lines
# allpts = sio.loadmat('HW3.mat')['allpts']
# d1 = allpts[0:20].T
# d2 = allpts[20:50].T
```

In [109]:
```
fig, axes = plt.subplots(2, 3)
fig.set_figheight(10)
fig.set_figwidth(15)

ax = axes[0,0]
ax.scatter(d1[0,:], d1[1,:], c='r', marker='x')
ax.scatter(d2[0,:], d2[1,:], c='b', marker='x')

cluster1 = d1.T
cluster2 = d2.T

allpts = np.vstack((cluster1, cluster2))
goto = np.shape(allpts)[0]
```



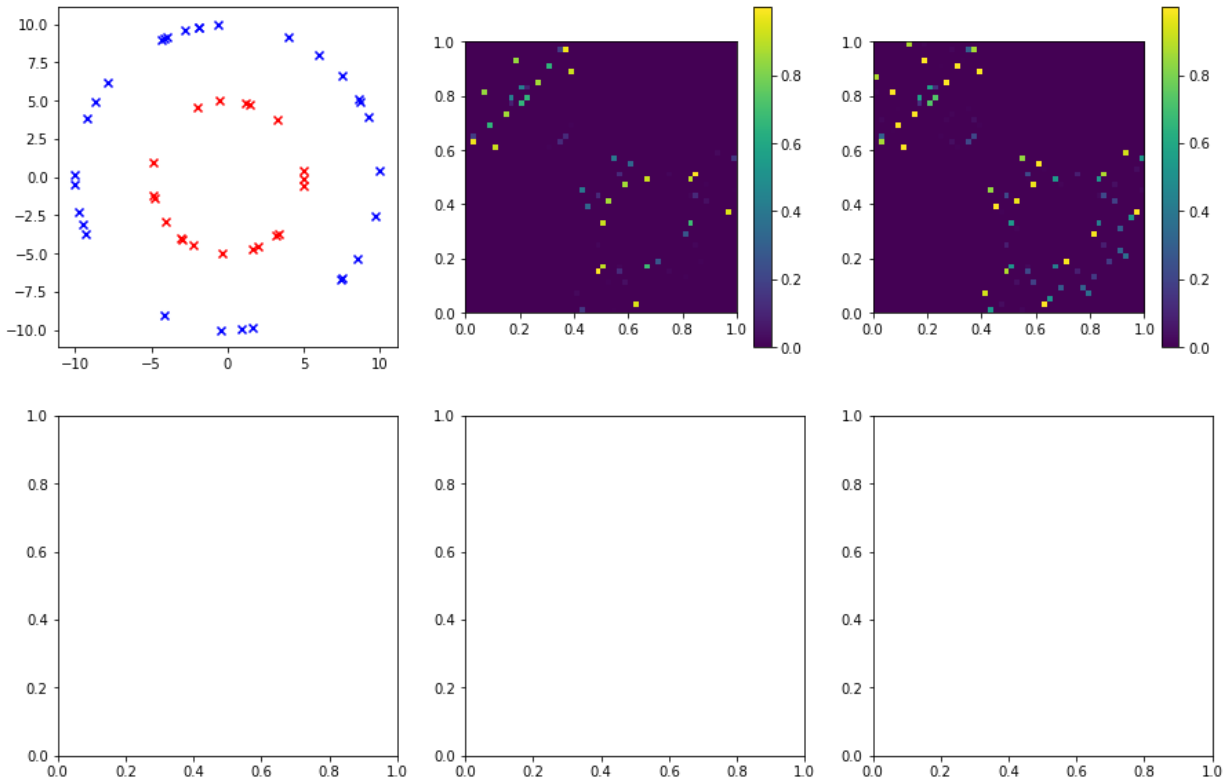
In [110]:
```
# compute A (step 1)
# experiment with sigsq in question 8
sigsq = .2
Aisq = np.power(allpts[:,0], 2) + np.power(allpts[:,1], 2)
Dotprod = allpts.dot(allpts.T)

distmat = - repmat(Aisq, goto, 1) - repmat(Aisq.reshape(-1,1), 1, goto)  +
Afast = np.exp(distmat / (2*sigsq))
A = Afast - np.diag(np.diag(Afast))

ax = axes[0,1]
im = ax.imshow(A, extent=[0, 1, 0, 1])
fig.colorbar(im, ax=ax)
fig
```

Out[110]:



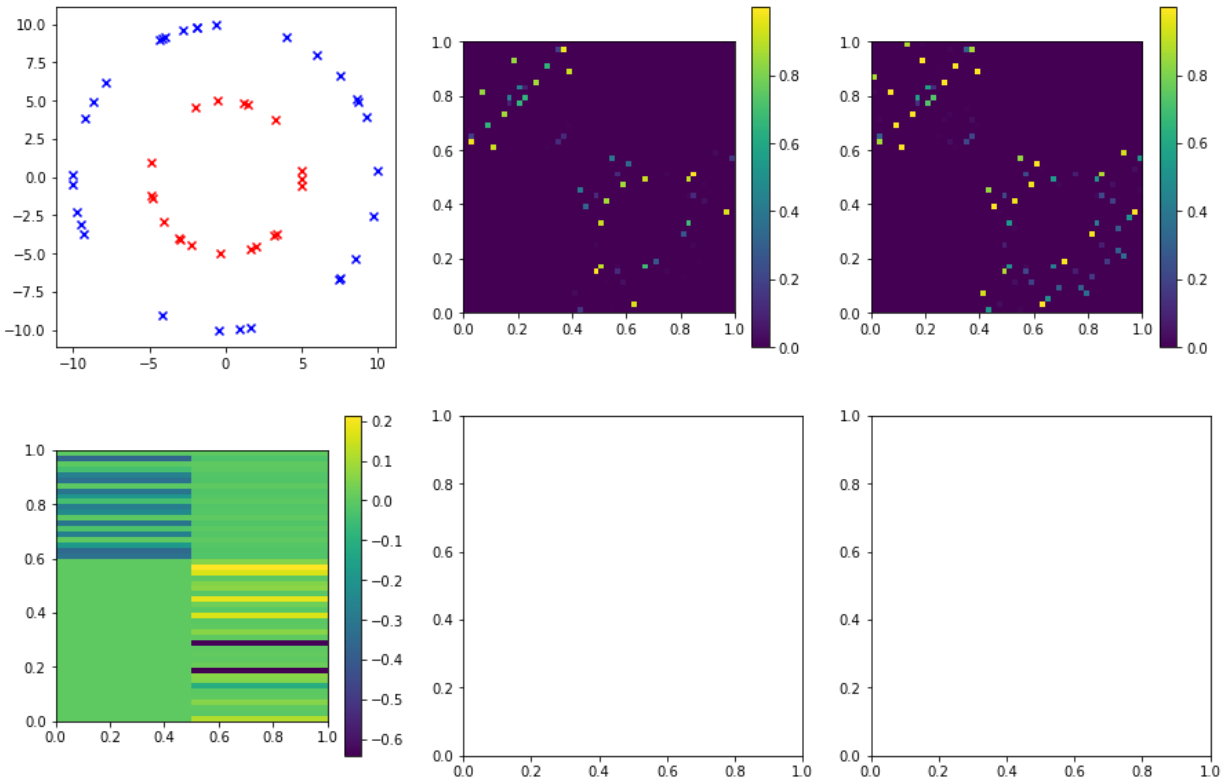
In [111]:
```
# step 2
D = np.diag(np.sum(A.T, axis=0))
L = np.linalg.inv(np.sqrt(D)).dot(A).dot(np.linalg.inv(np.sqrt(D)))
ax = axes[0,2]
im = ax.imshow(L, extent=[0, 1, 0, 1])
fig.colorbar(im, ax=ax)
fig
```

Out[111]:



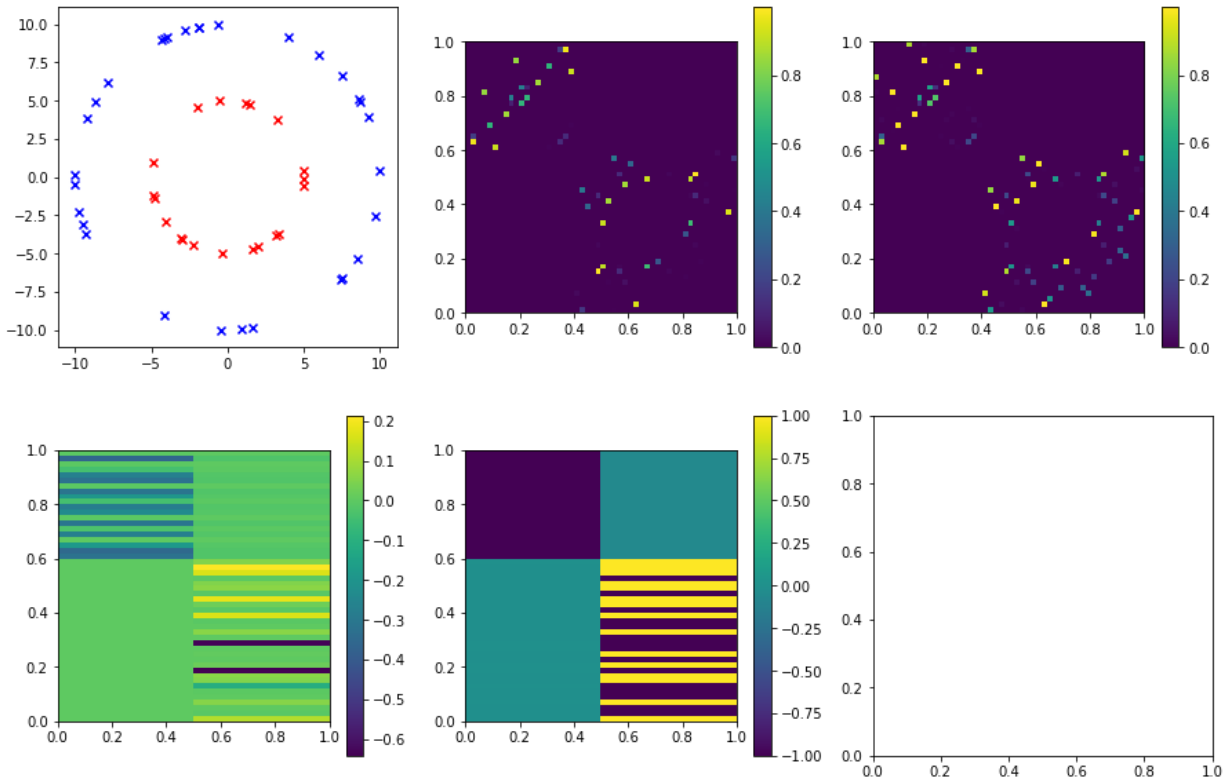
In [112]:
```
# step 3
di, X = np.linalg.eig(L)
Xsort, Dsort = eigsort(X,di)
Xuse = Xsort[:, :2]
ax = axes[1,0]
im = ax.imshow(Xuse, extent=[0, 1, 0, 1])
fig.colorbar(im, ax=ax)
fig
```

Out[112]:



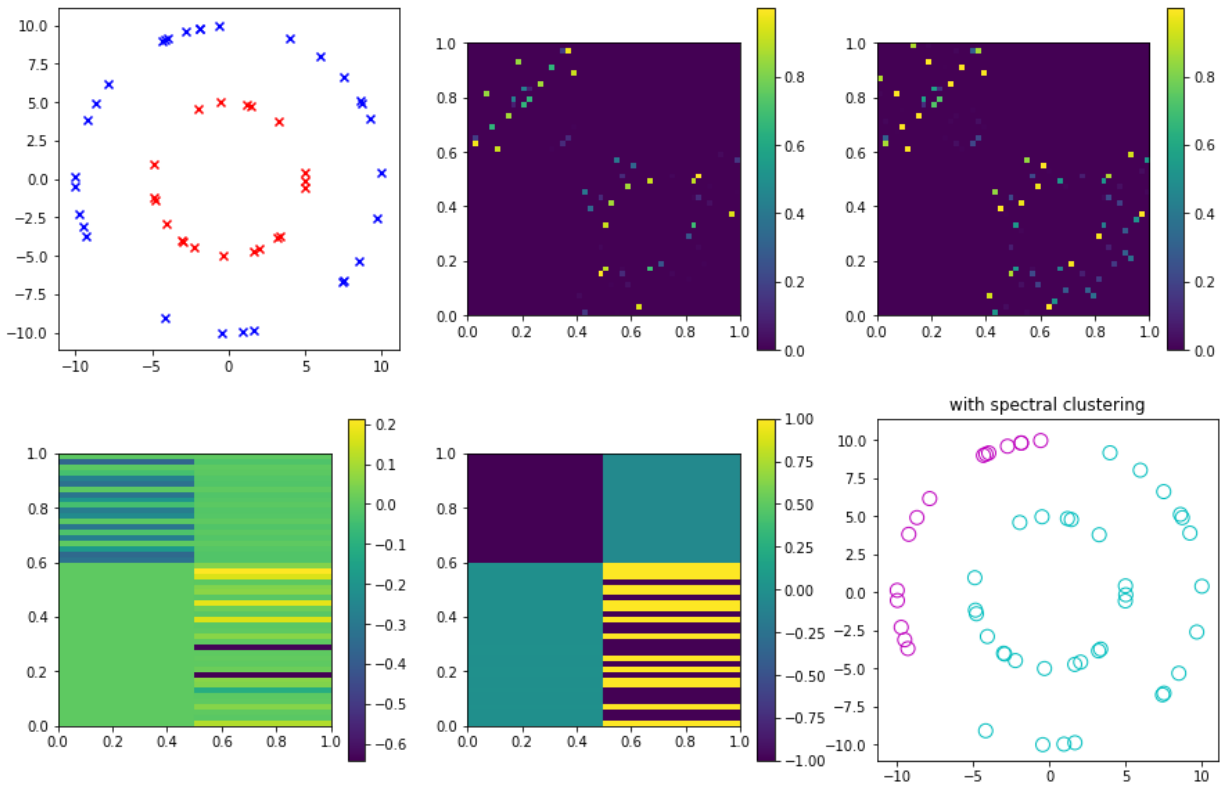
In [113]:
```
# normalize X to get Y (step 4)
Xsq = np.multiply(Xuse, Xuse)
divmat = repmat(np.sqrt(np.sum(Xsq.T, axis=0)).reshape(-1,1), 1, 2)
Y = np.divide(Xuse, divmat)
ax = axes[1,1]
im = ax.imshow(Y, extent=[0, 1, 0, 1])
fig.colorbar(im, ax=ax)
fig
```

Out[113]:



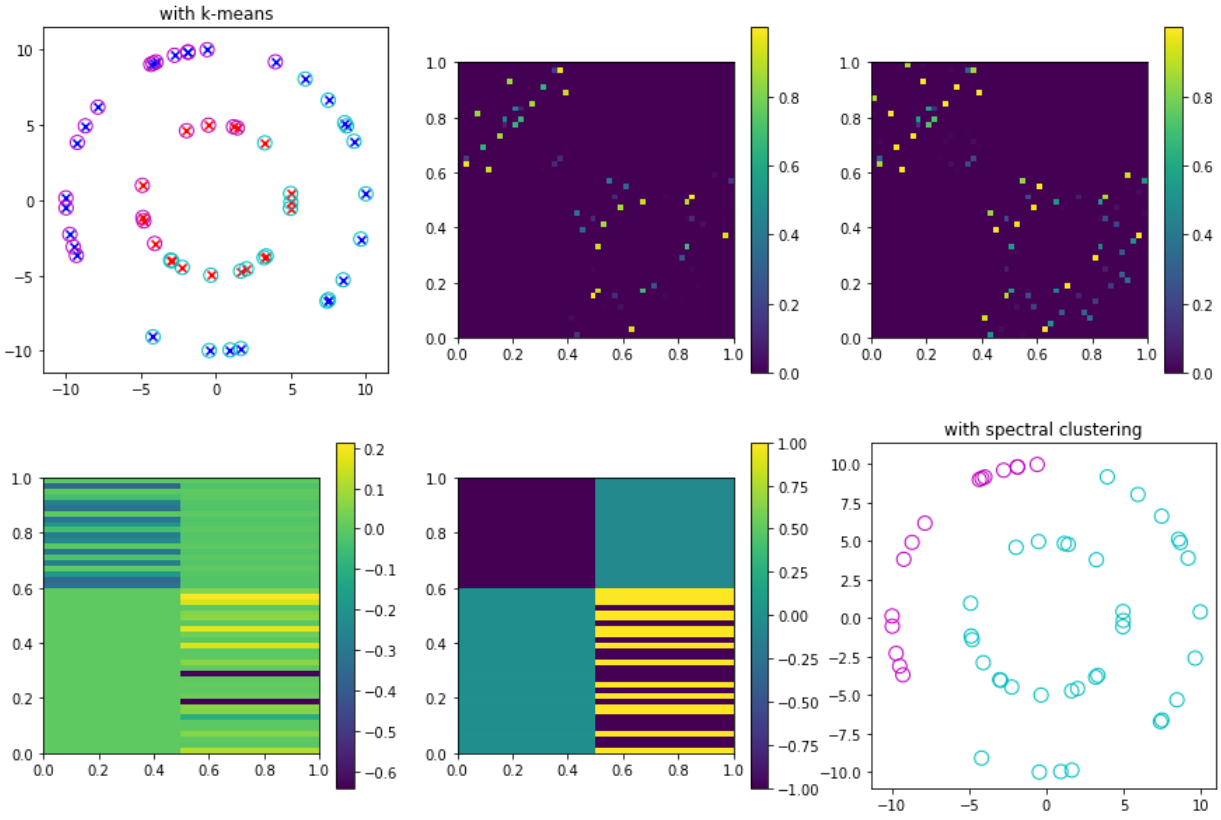
In [114]:
```
# step 5/6
kmeans = KMeans(n_clusters=2).fit(Y)
kk = kmeans.labels_
c1 = np.argwhere(kk == 0)
c2 = np.argwhere(kk == 1)
ax = axes[1,2]
ax.scatter(allpts[c1][:,0][:,0], allpts[c1][:,0][:,1], edgecolor='c', marke
ax.scatter(allpts[c2][:,0][:,0], allpts[c2][:,0][:,1], edgecolor='m', marke
ax.set_title('with spectral clustering')
fig
```

Out[114]:



In [115]:
```
# For comparison run kmeans on original data
kmeans = KMeans(n_clusters=2).fit(allpts)
kk = kmeans.labels_
c1 = np.argwhere(kk == 0)
c2 = np.argwhere(kk == 1)
ax = axes[0, 0]
ax.scatter(allpts[c1][:,0][:,0], allpts[c1][:,0][:,1], edgecolor='c', marke
ax.scatter(allpts[c2][:,0][:,0], allpts[c2][:,0][:,1], edgecolor='m', marke
ax.set_title('with k-means')
fig
```

Out[115]:



7

The first subfigure shows the two data sets plotted, with red points in the inner ring and blue points in the outer.

The second one shows the similarity/affinity matrix. Notice how this fills two quadrants.

The third one shows the normalized graph laplacian matrix. The weights of the colors are now more evenly distributed.

The fourth one shows the top k eigenvalues.

The fifth one shows the normalized version of the above.

The last subfigure shows the result of spectral clustering. The data seems to be clustered as desired.

K means works not too bad.

8

Sigsq represents the variance. Too low sigsq results in inaccurate clustering, with emphasis on one rather than another. Too high, over 1, and the same thing happens but reverse emphasis.

9

Sigsq sometimes does not work well, as it will miss for example 4 points in the clustering algorithm, in which case I have to make it smaller. Some data sets probably don't work well because the randomization of the data sometimes clumps points in certain areas.