

## Homework 9

CSC-432

Due: 4/22/13

### Clustering

#### 1. Generate some Data (2 points)

There are a number of ways you could go about generating clusters of data. One way might be to first generate data from a multivariate Gaussian, then to shift the locations of subsets of the data. The multivariate Gaussian is much like the normal distribution you have already seen except that is multivariate rather than univariate.

1. Set `'numpy.random.seed(12345)'` before generating the data
  2. Generate 1000 observations of x,y points using one call to `'numpy.random.multivariate_normal'`. Assign these to a variable `'X'`.
  3. Use a mean of `'[0,0]'` and a covariance of `'[[.1,0],[0,.1]]'`.
  4. Shift the *\*location\** of each fifth observation by an array of shape `'(2,)'` drawn from a normal distribution with location 0 and standard deviation 2. **\*\*Hint:\*\*** Recall that you can use a step in slice notation, so that every fifth observation of X starting from zero, might be shifted by calling  
`X[::5] += np.random.normal(0, 2, size=2)`
  5. Use `'matplotlib.pyplot.scatter'` to plot your clusters.
  6. What does the covariance used above indicate about the relationship between the 'x' and 'y' coordinates?
- 
2. Using `sklearn.cluster.KMeans` with `K=5` find the cluster centroids of this data. Print the centers. (1 points)
  3. Plot the cluster centers along with your data using `matplotlib.pyplot.scatter`. Indicate the cluster centers with a red point and the data with blue points. Make sure that the cluster points stand out. Use an alpha of .25 for the data points and point size for 64 for the cluster centers. (1 points)
  4. There is an image in the notebooks directory called `blue_flower_600x600.png`. Load this image using `scipy.ndimage.imread` in grayscale. Scale the image by 1/2 using `scipy.misc.imresize`. Plot it using `matplotlib.pyplot.imshow`. (1 points)
  5. Using `sklearn.cluster.Ward` determine a good number of clusters to select the flower (or enough of it, so that you know where it

is). Use the labels to crop the background of the *original* image (ie., not the grayscale image) so that it contains the flower and the background that is left. Do this using numpy and display the result using `imshow`. (5 points)