Emtiyaz Khan, Timur Bagautdinov, Carlos Becker, Ilija Bogunovic & Ksenia Konyushkova

7. K-means Clustering

7.1 Goals

The goal of this exercise is to

- Implement and visualize K-means clustering using the "old-faithful" dataset.
- Visualize the behavior with respect to the number of clusters K.
- Implement data compression using K-means.

7.2 Old faithful dataset

We will reproduce Figure 9.1 of Bishop's book. The file `kmeansDemo.m` contains a wrapper code to start with. Please have a look inside.

Exercise 7.1 Let's first implement K-means algorithm using the "old faithful" dataset.

- Fill-in the code to center the data. Visualize the data and make sure that the data is spread around 0. You can also uncomment the code to set the limits of the axes.
- Write the function `kmeansUpdate.m` to update responsibilities $\mathbf{r}$, means $\boldsymbol{\mu}$ and the distance of data points to the means. Your code should work for any number of clusters K (not just $K = 2$).
- Write code to test for convergence.
- Visualize the output. You should get figures similar to Figure 7.1.

■

Exercise 7.2 Now, play with the initial conditions and the number of clusters to understand the behavior of K-means.

- Change the initial conditions and observe the change in convergence. The algorithm must converge for all possible initial conditions, otherwise there is a problem in your implementation.
- Try different values for K. Also try different values of initial condition. Look at the cost function value as K increases.
- BONUS: What is a good value for K? How will you choose it?

■

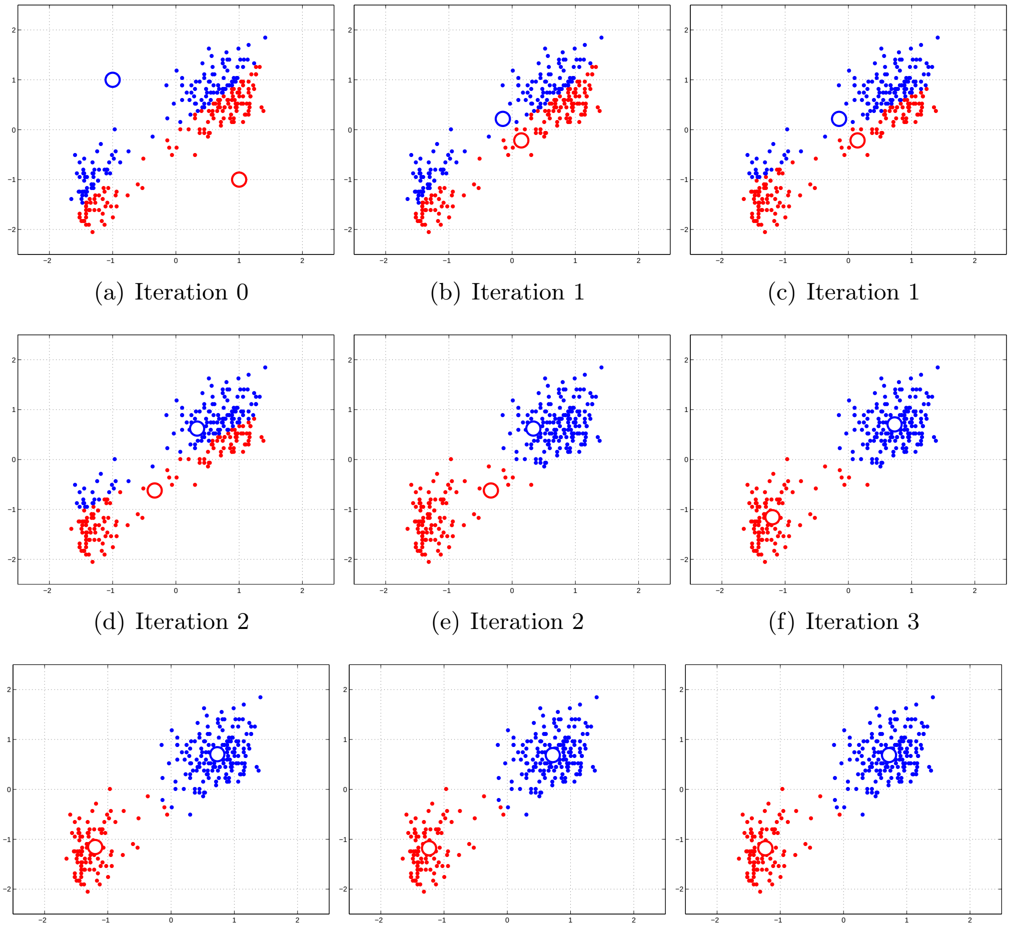
7.3 Data compression using K-means

We will implement data compression using K-means, similar to the examples shown in the class.

Exercise 7.3 Write data compression for `lena.png`.

You output should look like Figure 7.3. You can use the function `imshow()` and `imread()` to show and read a `png` image. You should vectorize the data and convert it to double precision using the function `double()`. Also, normalize the data.

Run K-means with random initializations and observe the convergence. Plot the reconstructed image by setting each pixel's value to the mean value of its cluster. Also,



(a) Iteration 0 (b) Iteration 1 (c) Iteration 1

(d) Iteration 2 (e) Iteration 2 (f) Iteration 3

(g) Iteration 3 (h) Iteration 4 (i) Iteration 4

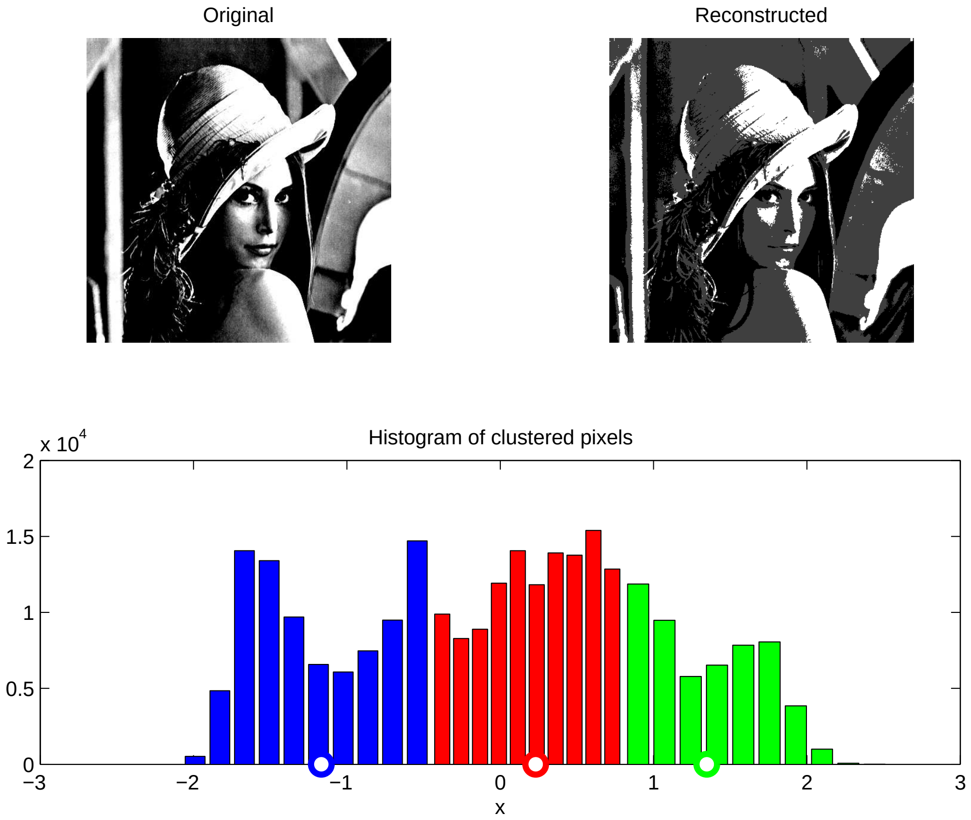
Figure 7.1: K-means for old-faithful data.

plot the histogram for each cluster to make sure that your results make sense. Play with the number of clusters and compare the compression you get in your resulting image (you can write a png file using `imwrite()`). ■

7.4 Bonus: K-means and Project I

You can apply K-means to your regression dataset to cluster your data together. This will cluster the data and then you can learn two regression models on your data. Compare your new test error with what you got before.

Do you think that K-means is ideal for this dataset? What do you think about the points that lie between the two clusters? Should they be assigned to one or the other cluster? The answer lies in something known as soft-clustering using Gaussian Mixture Models.


Original
Reconstructed
Histogram of clustered pixels
x 10^4
2
1.5
1
0.5
0
−3
−2
−1
0
1
2
3
x