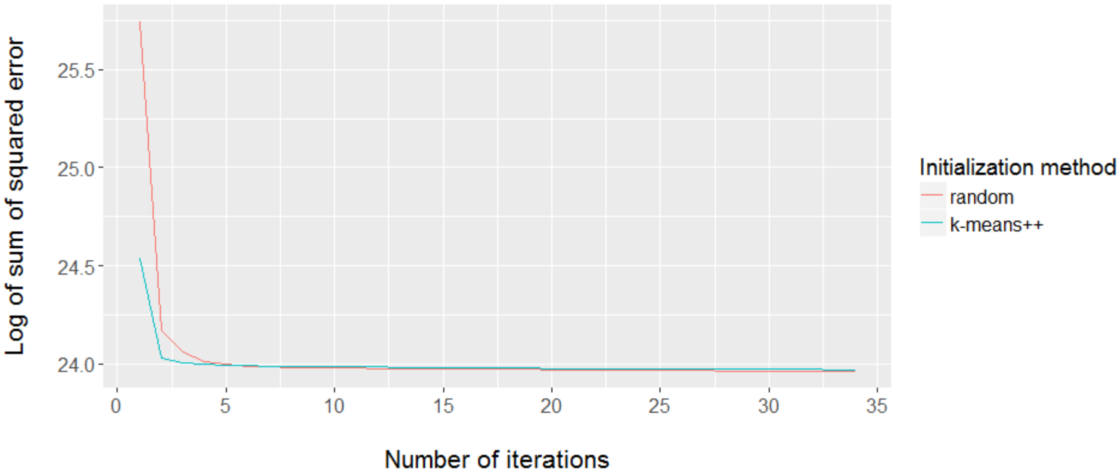
CS 324 Data Mining **Homework #6**
Il Shan Ng **May 10, 2017**

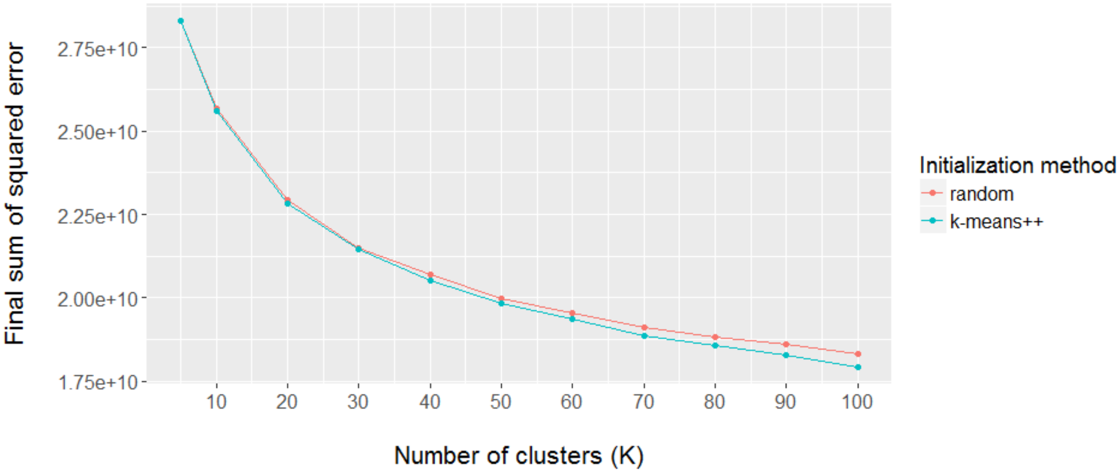
1. Figure 1 gives the requested plot. I use the log of SSE to make the plot look nicer, but because the logarithmic function preserves ordering, the conclusion will not be different. We see that the SSE monotonically decreases for both the `random` and `other` (k-means++) initialization methods.

Figure 1: Convergence of the sum of squared error for $K = 10$



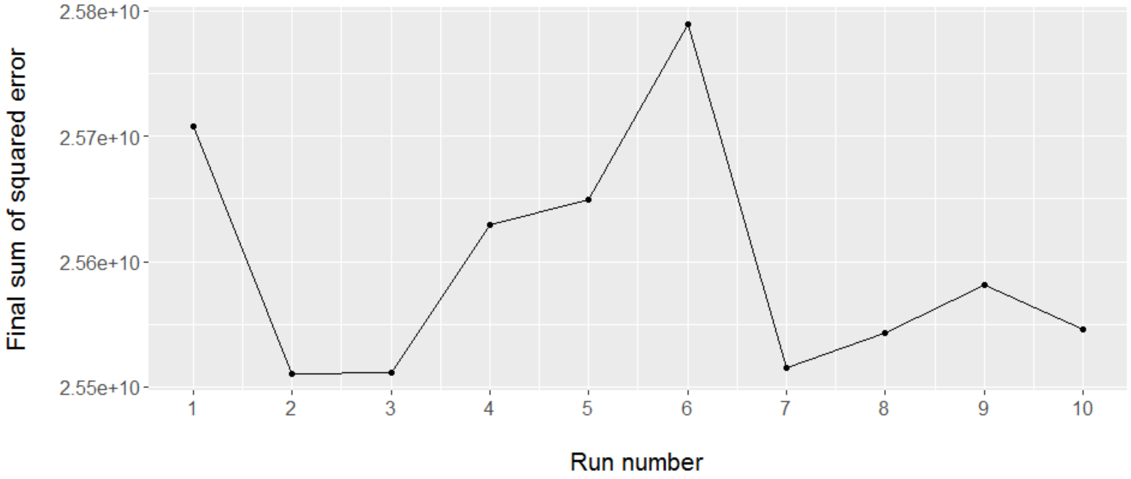
2. Figure 2 gives the requested plot. The value of K that gives the lowest SSE for both the `random` and `other` (k-means++) initialization methods is $K = 100$. The value of K that best fits the dataset is $K = 10$, because we know from the true labels that there are 10 digits (0–9) in the dataset.

Figure 2: Final clustering error for different values of K



3. Figure 3 gives the requested plot. A large change in the final SSE is an indication that the final cluster assignments might have changed between runs. Based on the plot, the final clustering may have changed between runs 1 and 2, between runs 3 and 4, between runs 5 and 6, and between runs 6 and 7.

Figure 3: Final clustering error across ten different runs with `random` and $K = 10$



4. Figure 4 shows a visualization of the final centroids obtained after the algorithm converges using the initialization of `random`. Notice that there are duplicate digits: two centroids show up as zeros, another two centroids show up as ones, and three centroids show up as nines. This happens because the random initialization tends to cause the algorithm to be stuck at local minima, such that multiple centroids end up representing the same digit. Digits that are similar to each other, for example, 3 and 8, also end up contributing to the same centroid, which is why some of the centroids look ambiguous.

Figure 4: Final centroids with initialization of `random` and $K = 10$

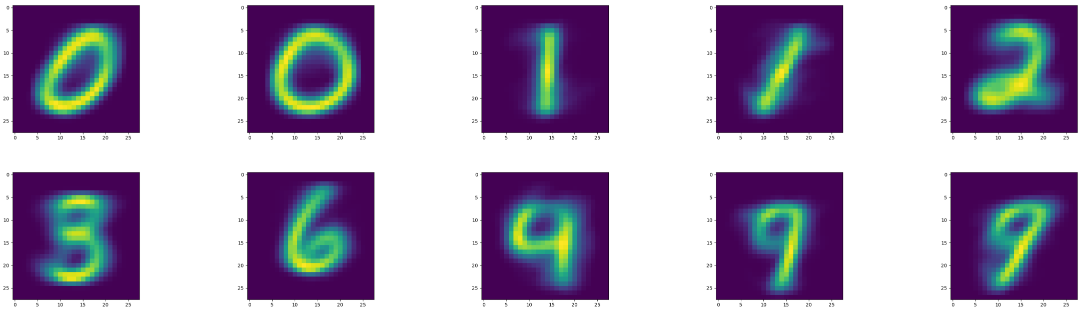
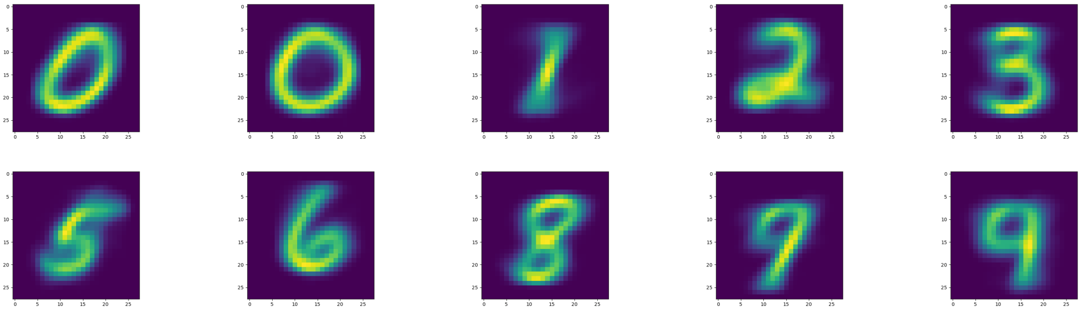


Figure 5 shows a visualization of the final centroids obtained after the algorithm converges using the initialization of `other` (k-means++). Though there are still duplicate digits, there is less duplication compared to Figure 4. This results because the k-means++ algorithm attempts to space out the initial centroids, making it less likely for two centroids to converge on the same cluster.

Figure 5: Final centroids with initialization of `other` (k-means++) and $K = 10$



5. Let SSE_{old} be the SSE in the presence of the empty cluster, and SSE_{new} be the SSE after removing the empty cluster and creating a new centroid using the point x_i that is furthest away from its currently assigned cluster c_i. Then,

$$\text{SSE}_{\text{new}} = \text{SSE}_{\text{old}} - [\text{Euclidean}(x_i, c_i)]^2,$$

because the point x_i will coincide with the new centroid and therefore have zero distance from it. Because x_i is the point that was furthest away from its previously assigned centroid $c_i, \text{Euclidean}(x_i, c_i) > 0$. Hence, $\text{SSE}_{\text{new}} < \text{SSE}_{\text{old}}$ and so the procedure outlined in the assignment reduces clustering error.

6. Figure 6 plots the SSE for the initialization method of `cheating` against the iteration number. The SSE starts off much lower at 2.645×10^{10}, compared to when random initialization was used (1.511×10^{11}) or when k-means++ was used (4.544×10^{10}). Compared to Figure 1, Figure 6 also shows the algorithm converging much faster, requiring only 11 iterations compared to the 34 required for both random initialization and k-means++. This faster convergence results directly from initializing each centroid as the mean of the digits with the label that the centroid is intended to represent.

Figure 6: Convergence of sum of squared error for `cheating`

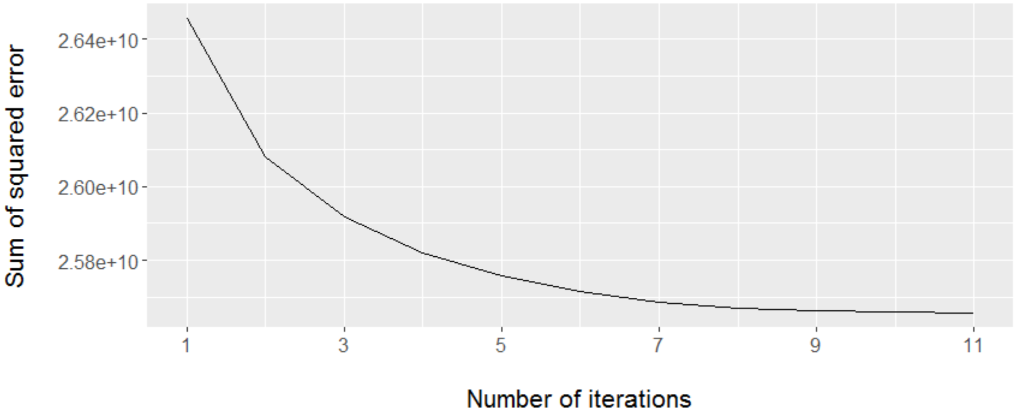


Figure 7 shows a visualization of the final centroids obtained after the algorithm converges using the initialization method of `cheating`. We see that almost every digit is represented, with only 4 missing. This is probably because the handwritten digits in the dataset that were meant to be 4 and those meant to be 9 looked similar. Additionally, there is some ambiguity in the centroids representing 3 and 5, and so even though `cheating` leads to some improvements over random initialization or k-means++, the algorithm is still confusing digits that look like each other. This suggests that the objective function may not be entirely suitable for this dataset, because digits with different labels may sometimes resemble each other (for e.g. 4 and 9, 3 and 8) which causes the final centroids to look like hybrids of digits.

Figure 7: Final centroids with initialization of `cheating` and $K = 10$