Homework 3

Matt Dickenson
CS 527
Fall 2014

Problem 1

(a) kmeans.m

```
function [M, R] = kmeans(Z, M)
  % dim(Z) = d x I
  % dim(M) = d x K
  % dim(R) = K x I

  % initialize f(S) values for iteration
  f_p = intmax
  f_c = f_p - 1

  while f_p > f_c + sqrt(eps)
    R = closest_cluster(Z, M)
    M = (Z * transpose(R)) ./ (transpose(sum(R, 2) * ones(1, 2)))
    f_p = f_c
    f_c = sum(sum(dists(Z, M) .* R, 1), 2)
  end
end

function [assignments] = closest_cluster(Z, mu)
  % partition the points in Z by their closes mean
  K = size(mu, 2)
  all_dists = dists(Z, mu)
  min_dists = ones(K, 1) * min(all_dists, [], 1)
  assignments = all_dists == min_dists
end

function [all_dists] = dists(Z, mu)
  % compute distance of each point in Z from each centroid
  K = size(mu, 2)
  I = size(Z, 2)
  all_dists = zeros(K, I)
  for k=1:K
    means = mu(:, k) * ones(1, I)
    dists = sqrt(sum((Z - means).^2, 1))
    all_dists( k , :) = dists
  end
```

```
end
```

Output:

```
M =

    0.6816    0.8271   -1.0890
   -0.6389    1.1812   -0.6784
```

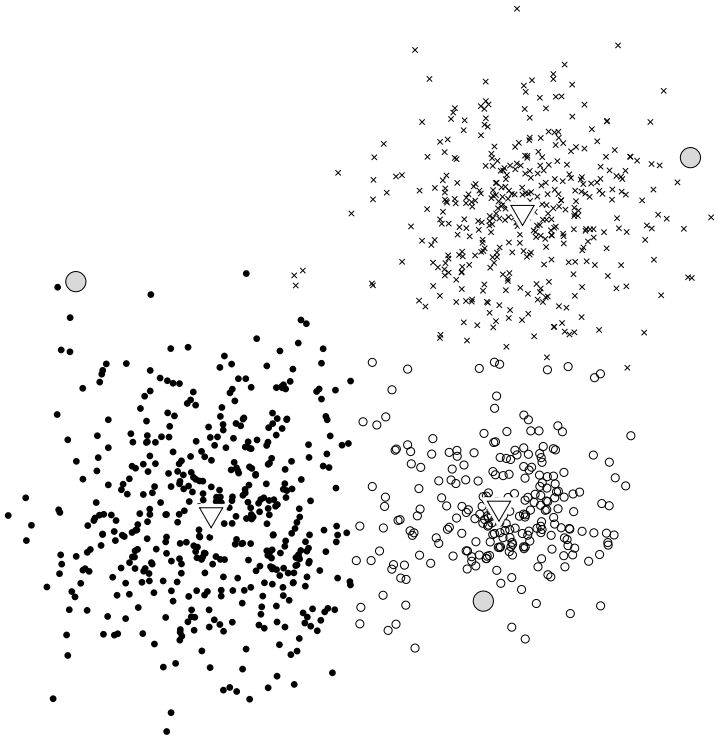

Figure 1: Result of K-Means with blobs data for 1(a)

(b) Output:

```
M =

   -1.0371    1.0207
   -0.0176    0.0173
```

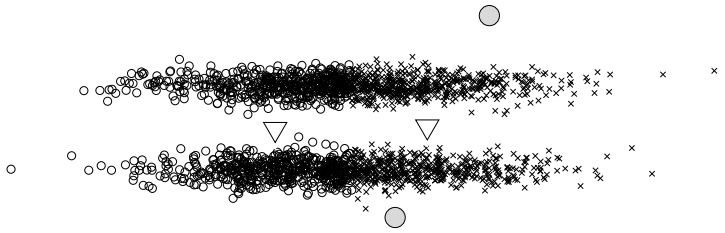

Figure 2: Result of K-Means with cigars data for 1(b)

(c) Output:

```
M =

   -1.0402    1.0175
   -0.0199    0.0195
```

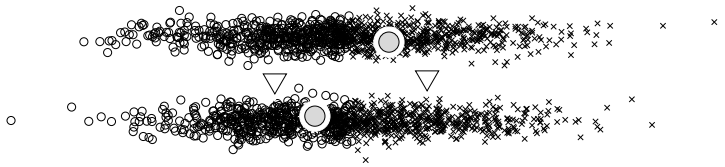

Figure 3: Result of K-Means with cigars data for 1(c)

(d) The shortcomings are not that the K-means algorithm computes a local minimum. The problem is that both dimensions are treated equally for the purposes of computing the error rate despite the fact that they may have very different variances.

The problem is that the data points vary much more along the x-dimension than the y dimension. Although they appear to be well-described by two clusters with the same x range, one above the other, the high variance in the x dimension would force two such centroids to still have a high error measure $f(S)$.

(e) I constructed four clusters whose points form approximately the outline of a unit square. The four initial centroids m all lie at the origin. The points and centroids are displayed in Figure 4 below. Note that the final centroids and initial centroids are exactly equal.

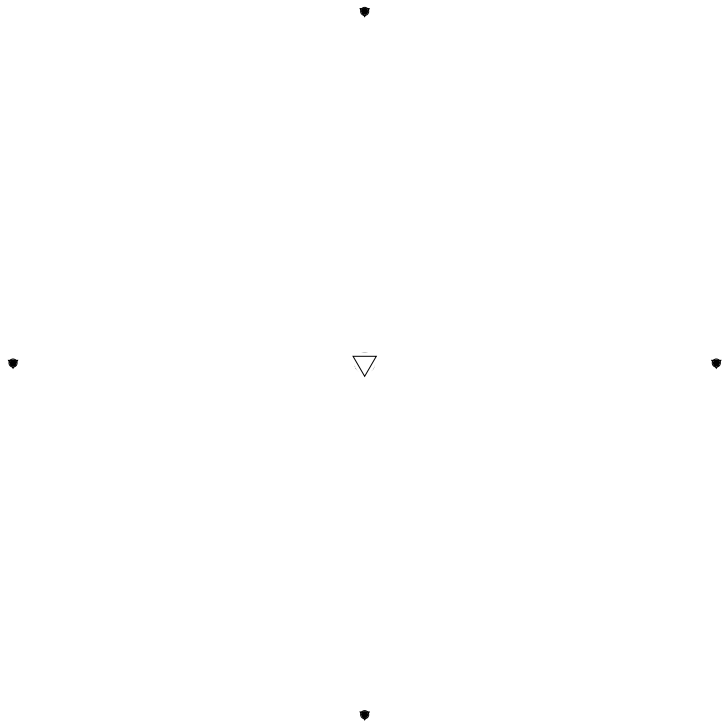

Figure 4: Result of K-Means with pathological input

The true centroids are $\mu_1^* = (0, 1), \mu_2^* = (1, 0), \mu_3^* = (0, -1), mu_4^* = (-1, 0)$. The initial (and final) centroids are $m_1 = m_2 = m_3 = m_4 = (0, 0)$.

The points in each cluster are

$$
\begin{aligned}
S_1 &= \{(\epsilon, 1), (1+\epsilon, 0), (-\epsilon, -1), (-1-\epsilon, 0)\} \\
S_2 &= \{(-\epsilon, 1), (1-\epsilon, 0), (\epsilon, -1), (-1+\epsilon)\} \\
S_3 &= \{(0, 1+\epsilon), (1, \epsilon), (0, -1-\epsilon), (1, -\epsilon)\} \\
S_4 &= \{(0, 1-\epsilon), (1, -\epsilon), (0, -1+\epsilon), (-1, \epsilon)\},
\end{aligned}
$$

although any random, symmetric perturbation about the true centroids would work in expectation.

The algorithm makes no progress because the initial centroids m are each symmetrically located between all four clusters and all clusters are equally sized. Moving any centroid nearer to one cluster would make it closer to a few points but farther away from all other points, cancelling out the advantage of relocating the centroid.

(f) The previous answer shows that the success K-means algorithm depends upon the initial centroids. K-means also depends on knowing the true number of clusers. It can also fail when clusters are symmetric and of equal size.