

```

% Read the dataset
data = csvread('MNIST_HW3.csv', 1, 0);

% Calculate distortions for different numbers of clusters
max_clusters = 20;
distortions = zeros(max_clusters, 1);
for num_clusters = 1:max_clusters
    [~, labels] = fit(data, num_clusters, 100);
    centers = update_centers(data, labels, num_clusters);
    distortion = calculate_distortion(data, labels, centers);
    distortions(num_clusters) = distortion;
end

% Print the distortions
for num_clusters = 1:max_clusters
    fprintf('Number of clusters: %d, Distortion: %.4f\n', num_clusters,
    distortions(num_clusters));
end

% Plot the distortions
plot(1:max_clusters, distortions, 'x-');
xlabel('Number of Clusters');
ylabel('Distortion');
title('Elbow Method');

% Find the best k using the elbow method
distortion_diff = diff(distortions);
[~, best_k_index] = max(distortion_diff);
best_k = best_k_index + 2;
fprintf('The optimal number of clusters (k): %d\n', best_k);

% Function to initialize cluster centers
function centers = initialize_centers(data, num_clusters)
    % Randomly initialize cluster centers
    indices = randperm(size(data, 1), num_clusters);
    centers = data(indices, :);
end

% Function to assign data points to cluster centers
function labels = assign_clusters(data, centers)
    % Assign each data point to the nearest cluster center
    distances = pdist2(data, centers);
    [~, labels] = min(distances, [], 2);
end

% Function to update cluster centers
function centers = update_centers(data, labels, num_clusters)
    % Update cluster centers as the mean of the assigned data points
    centers = zeros(num_clusters, size(data, 2));
    for i = 1:num_clusters
        cluster_points = data(labels == i, :);
        centers(i, :) = mean(cluster_points, 1);
    end
end
end

```

```

% Function to calculate distortion
function distortion = calculate_distortion(data, labels, centers)
    distortion = 0;
    for i = 1:size(data, 1)
        cluster_center = centers(labels(i), :);
        distortion = distortion + norm(data(i, :) - cluster_center)^2;
    end
end

% Fit the data to find cluster centers
function [centers, labels] = fit(data, num_clusters, max_iterations)
    centers = initialize_centers(data, num_clusters);

    for iteration = 1:max_iterations
        old_centers = centers;

        labels = assign_clusters(data, centers);
        centers = update_centers(data, labels, num_clusters);

        if isequal(old_centers, centers)
            break;
        end
    end
end

```