```
function [cluster_matrix final_delta] = fully_vectorized_delta_clustering(dataset, N)
  dataset = dataset(:,1:N);
  num_rows = size(dataset,1);
  num_cols = size(dataset,2);
  s = std(dataset(:,1:N)); %calculates the standard deviation of the dataset in each
dimension
  s = mean(s); %takes the average standard deviation
  alpha = 1.5; %this is a constant used to adjust the standard deviation
  s = s*alpha;
  num_iterations = 25; %this is a hypothetical that is vectorized
  temp_matrix = repmat(dataset', [1 1 num_rows]);
  ref_dataset = shiftdim(temp_matrix,2);
  diff_matrix = (dataset(:,1:N) .- ref_dataset).^2;
  diff_vector = sum(diff_matrix,2);
  delta_vector = [1/num_iterations : s/num_iterations : s];
  delta_vector = delta_vector.^2;
  num_delta_tests = size(delta_vector,2);
  delta_matrix = repmat(delta_vector, [num_rows 1]); %housekeeping repitition of the
entries in delta_vector
  final_matrix = diff_vector < delta_matrix;
  LH = final_matrix(:,1:num_delta_tests - 1, :);
  RH = final_matrix(:,2:num_delta_tests,:);
  change_count_matrix = (LH .- RH).^2; %counts the number of times a test condition
changes
  change_count_vector = sum(change_count_matrix,1); %takes the sum over the
changes by column
  change_count_vector = sum(change_count_vector,3); %takes the sum over the
changes by page
  [a b] = max(change_count_vector);
  final_delta = delta_matrix(1,b);
  final_delta = final_delta(1,1);
  cluster_matrix = diff_vector < final_delta;
endfunction
```