

```

function [class_distribution_matrix, delta_vector] = MASS_Unsup_Modal_BlackTree(dataset, class_vector, N)

num_rows = size(dataset,1);
num_classes = size(class_vector,1);

%identifies testing rows
testing_rows = find(dataset(:,N+2) == -1);
num_testing_rows = size(testing_rows,1);

%calculates fixed maximum cluster radius s
s = std(dataset(:,1:N));
s = norm(s)/e;

%return value, distribution of classes in each cluster
class_distribution_matrix = zeros(num_testing_rows, num_classes);

%initializes delta
delta_vector = zeros(num_testing_rows,1);

%finds a cluster for each testing row
for i = 1 : num_testing_rows

    testing_vector = dataset(testing_rows(i),:);
    break_loop = 0;

    %Lefthand-----

    %if true, then the row is the first entry
    if(testing_rows(i) == 1)

        break_loop = 1;

    endif

    j = 0; %iterator

    %iterates until distance exceeds s, or we hit the first entry
    while(break_loop == 0 && (i - j) > 1)

        j = j + 1;
        current_vector = dataset(testing_rows(i) - j,:);
        diff = norm(current_vector(1:N) .- testing_vector(1:N));

        %if true, then we exceeded the fixed radius, break loop
        if(diff > s)

            break_loop = 1;

        %if true, then the current vector is a training row, include in cluster
        elseif(current_vector(N+2) != -1)

            %finds the index of the class
            current_class = current_vector(N+1);
            class_index = find(class_vector == current_class);
            class_distribution_matrix(i,class_index) = class_distribution_matrix(i,class_index) + 1;

            %if true, then we store diff as delta
            if(diff > delta_vector(i))

                delta_vector(i) = diff;
            end
        end
    end
end

```

```

        endif

    endif

endwhile

%Righthand-----

%if true, then the row is the last entry
if(testing_rows(i) == num_rows)

    break_loop = 1;

endif

j = 0; %iterator

%iterates until distance exceeds s, or we hit the first entry
while(break_loop == 0 && (i + j) < num_rows)

    j = j + 1;
    current_vector = dataset(testing_rows(i) + j,:);
    diff = norm(current_vector(1:N) .- testing_vector(1:N));

    %if true, then we exceeded the fixed radius, break loop
    if(diff > s)

        break_loop = 1;

    %if true, then the current vector is a training row, include in cluster
    elseif(current_vector(N+2) != -1)

        %finds the index of the class, increments it
        current_class = current_vector(N+1);
        class_index = find(class_vector == current_class);
        class_distribution_matrix(i,class_index) = class_distribution_matrix(i,class_index) + 1;

        %if true, then we store diff as delta
        if(diff > delta_vector(i))

            delta_vector(i) = diff;

        endif

    endif

endwhile

endfor

endfunction

```