Machine Learning Assignment5

Xiaoxu Gao | 4504348 | highsmallxu@gmail.com

The Navie MIL Classifier

Read image

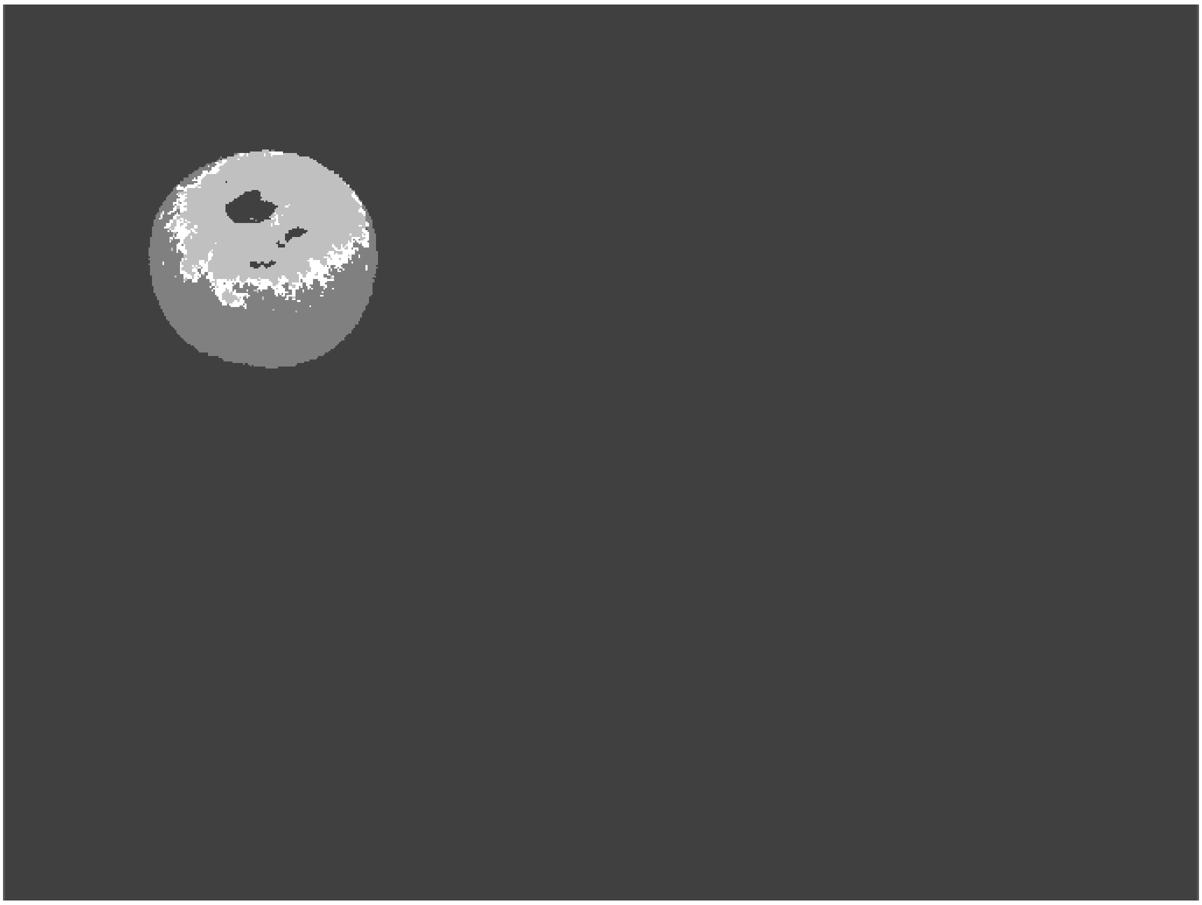
```
img_apple=imread(fullfile(path,[apple.name]));
img_banana=imread(fullfile(path,[banana.name]));
```

Extract instances

```
function segement = extractinstances(img)
        width = 30;
    img_seg = im_meanshift(img,width);
    seg_num = length(unique(img_seg));
    segement = zeros(seg_num,3);

    red = img(:,:,1);
    blue = img(:,:,2);
    green = img(:,:,3);

    for i = 1:seg_num
        target = (img_seg==i);
        red_pix = red(target==1);
        red_av = sum(red_pix)/length(red_pix);
        blue_pix = blue(target==1);
        blue_av = sum(blue_pix)/length(blue_pix);
        green_pix = green(target==1);
        green_av = sum(green_pix)/length(green_pix);

        segement(i,1) = red_av;
        segement(i,2) = blue_av;
        segement(i,3) = green_av;
    end
end
```

* results:*

```
value of width: 30
```

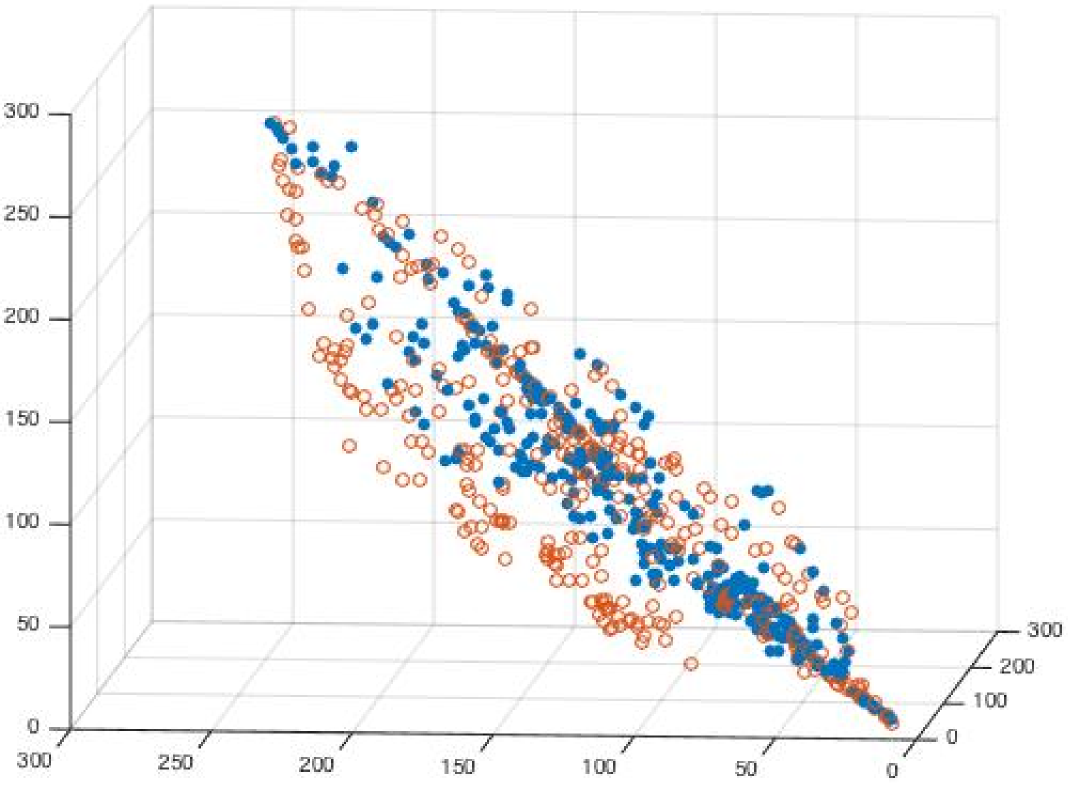


make dataset

```
bags = [apple_cell;banana_cell];
baglab = [2*ones(length(apple_cell),1);ones(length(banana_cell),1)];
dataset = bags2dataset(bags,baglab);
```

results:

```
number of bags: 120 {60 apples, 60 bananas}
number of features: 3 {average red, average blue, average green}
number of instances: 4~8 {each bag may have different number of instances}
```



combine instance labels

```
function label_out = combineinstlabels(labels)
        label_set = unique(labels);
        label_num = length(unique(labels));
        label_len_ini = 0;

    for i = 1:label_num
        label_len = sum(labels==label_set(i));
        if label_len > label_len_ini
                label_out = label_set(i);
                label_len_ini = label_len;
        end
        end
end
```

train classifier

```
ins = dataset.data;
labels = dataset.nlab;
fishclf = fitcdiscr(ins,labels);
ins_pre = predict(fishclf,ins);

s = 1;
result = [];
for i = 1:length(bags)
        ins_len = size(bags{i},1);
    ins_set = ins_pre(s:s+ins_len-1);
    s = s+ins_len;
    label_out = combineinstlabels(ins_set)
    result = [result;label_out];
end
```

results:

```
misclassified apples: 10 bags(17%)
misclassified bananas: 24 bags(40%)

Explanation: This estimation is not trustful because if a bag has more negative
instances which are not either banana or apple related, like background, in this
case it will detect it as a banana. However, it is an apple bag. So, sometimes,
the background instances can disrupt the result. We can also say that the result
can be easily confused by the false positive instances in positivce bags.
```

optimization

```
method 1: create some new features like HSV(hue, saturation and value).
method 2: use K-mean clustering to do segmentation.
```

MILES

bagembed

```
function m = bagembed(bags,instance)
        sigma = 25;
        distinct_f = 0;
    m_i = []; m = [];
    for i = 1:length(bags)
        bag_ins = bags{i};
        for j = 1:size(instance,1)
                for z = 1:size(bag_ins,1)
                distance_z = exp(-sum((instance(j,:)-bag_ins(z,:)).^2)/sigma);
                if(distance_z > distance_f)
                        distance_f = distance_z;
                end
            end
            s_j = distance_f;
            m_i = [m_i s_j];
            distance_f = 0;
        end
        m = [m;m_i];
        m_i = [];
    end
end
```

result

```
size: 120*662
experimentation 1: test set = train set
                                 accuracy = 100%
experimentation 2: train set 70% = 84(42 for each class), test set 30% = 36(18
for each class)
                             apple accuracy = 94.4%
                  banana accuracy = 72.2%
*explanation*: the result got from MILES Classifier is better than the one we got
from Navie Classifier. MILES maps each bag into a feature space defined by the
instances via an instance similarity measure. However, this mapping may provide
some unrelated features. So, 1-norm SVM is used to select important features and
construct classifier simultaneously, which can improve the accuracy.
*optimization*: In (7), the size of M(Bi) is defined by the size of bags and the
size of instances. Sometimes, instances would be much more than bags which may
cause storage problem. Considering this situation, instance similarity could be
measured to generate sparase features and then reduce the storage space.
```

Another MIL Classifier - Citation kNN

```
It's a kind of method which is performed in the space of bags. We can define a
distance function D(X,Y) that compares any two bags X and Y, and plug this
distance function into a standard distance-based classifier such as K-NN. In this
case, I used Hausdorff distance. This is the distance between the closet points
of X and Y. This method considers not only the bags as the nearest neighbours of
bag B, but also the bags that count B as their neighbours based on the minimum
Hausdorff distance.

for i=1:length(bags)
    bag_i = bags{i};
    for j=1:length(bags)
        bag1 = bag_i;
        bag2 = bags{j};
        d_j = bag_distance(bag1,bag2); % calculate the distance between two bags
        d = [d;d_j];
    end
    d_full = [d_full d];
    [B,I] = sort(d);
    knn = [knn I(1:k,:)];
    d = [];
end

result = [];
for i=1:length(bags)
    % references
    [bag1_rn,bag2_rn]= knn_ref(i,knn);
    % citers
    [bag1_cn,bag2_cn]= knn_cit(i,knn);
    if(bag1_rn+bag1_cn)>=(bag2_rn+bag2_cn)
        label = 1;
    else
        label = 2;
    end
    result = [result;label];
end
acc = [ones(length(apple_cell),1);(2)*ones(length(banana_cell),1)];
sum(acc==result)/120;
```

result

```
k = 4;
AUC = 61%;
```