Python与MATLAB小练习：计算准确度Accuracy

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

分别使用Python与MATLAB编程，计算聚类准确度。思路为：首先利用匈牙利算法将训练后的标签进行调整，然后再计算准确度。

1．Python程序

```python
# Python demo
#  -*- coding: utf-8 -*-
# Author：凯鲁嘎吉 Coral Gajic
# https://www.cnblogs.com/kailugaji/
# Python小练习：计算准确度Accuracy
# 先用匈牙利算法调整标签，然后再计算准确度
import numpy as np
# 已经调整过标签了
def cluster_acc(y_true, y_pred):
    y_true = y_true.astype(np.int64)
    assert y_pred.size == y_true.size
    D = max(y_pred.max(), y_true.max()) + 1
    w = np.zeros((D, D), dtype=np.int64)
    for i in range(y_pred.size):
        w[y_pred[i], y_true[i]] += 1
    from sklearn.utils.linear_assignment_ import linear_assignment
    # 匈牙利算法调整标签
    ind = linear_assignment(w.max() - w)
    return sum([w[i, j] for i, j in ind]) * 1.0 / y_pred.size

y_true = np.array([2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1])
y_pred_1 = np.array([1, 1, 2, 1, 1, 2, 2, 2, 3, 2, 2, 3, 1, 3, 3, 2, 3]) # 未调整的标签
y_pred_2 = np.array([2, 2, 3, 2, 2, 3, 3, 3, 1, 3, 3, 1, 2, 1, 1, 3, 1]) # 调整后的标签
result_1 = cluster_acc(y_true, y_pred_1)
result_2 = cluster_acc(y_true, y_pred_2)
print('1: ', result_1)
print('2: ', result_2)
```

结果：

```
1：  0.6470588235294118
2：  0.6470588235294118
```

2. MATLAB程序

```
%% MATLAB demo
clear
clc
y_true = [2 2 2 2 2 2 3 3 3 3 3 3 1 1 1 1 1 ];
y_pred_1 = [1 1 2 1 1 2 2 2 3 2 2 3 1 3 3 2 3];
results = Evaluate(y_true,y_pred_1);
fprintf('未调整标签的准确度：%f\n', results(1));
% ---------------------------------------------------
% 实际采用下面这个：先用匈牙利算法对标签进行调整，然后再计算准确度Accuracy
y_pred_2 = label_map(y_pred_1, y_true);
results = Evaluate(y_true,y_pred_2);
fprintf('调整标签后的准确度：%f\n', results(1));

%% MATLAB实例：Munkres指派算法 - 凯鲁嘎吉 - 博客园
% 来自：https://www.cnblogs.com/kailugaji/p/11765596.html
function [assignment,cost] = munkres(costMat)
% MUNKRES   Munkres Assign Algorithm
%
% [ASSIGN,COST] = munkres(COSTMAT) returns the optimal assignment in ASSIGN
% with the minimum COST based on the assignment problem represented by the
% COSTMAT, where the (i,j)th element represents the cost to assign the jth
% job to the ith worker.
%

% This is vectorized implementation of the algorithm. It is the fastest
% among all Matlab implementations of the algorithm.

% Examples
% Example 1: a 5 x 5 example
%{
[assignment,cost] = munkres(magic(5));
[assignedrows,dum]=find(assignment);
disp(assignedrows'); % 3 2 1 5 4
disp(cost); %15
%}
% Example 2: 400 x 400 random data
%{
n=5;
A=rand(n);
tic
[a,b]=munkres(A);
toc
%}
```

```
% Reference:
% "Munkres' Assignment Algorithm, Modified for Rectangular Matrices",
% http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html

% version 1.0 by Yi Cao at Cranfield University on 17th June 2008

assignment = false(size(costMat));
cost = 0;

costMat(costMat~=costMat)=Inf;
validMat = costMat<Inf;
validCol = any(validMat);
validRow = any(validMat,2);

nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
    return
end

dMat = zeros(n);
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);

%*************************************************
% Munkres' Assignment Algorithm starts here
%*************************************************

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   STEP 1: Subtract the row minimum from each row.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 dMat = bsxfun(@minus, dMat, min(dMat,[],2));

%*************************************************************************
%   STEP 2: Find a zero of dMat. If there are no starred zeros in its
%           column or row start the zero. Repeat for each zero
%*************************************************************************
zP = ~dMat;
starZ = false(n);
while any(zP(:))
    [r,c]=find(zP,1);
    starZ(r,c)=true;
    zP(r,:)=false;
    zP(:,c)=false;
end

while 1
```

```
%**************************************************************************
%   STEP 3: Cover each column with a starred zero. If all the columns are
%           covered then the matching is maximum
%**************************************************************************
    primeZ = false(n);
    coverColumn = any(starZ);
    if ~any(~coverColumn)
        break
    end
    coverRow = false(n,1);
    while 1
        %************************************************************************
        %   STEP 4: Find a noncovered zero and prime it.  If there is no starred
        %           zero in the row containing this primed zero, Go to Step 5.
        %           Otherwise, cover this row and uncover the column containing
        %           the starred zero. Continue in this manner until there are no
        %           uncovered zeros left. Save the smallest uncovered value and
        %           Go to Step 6.
        %************************************************************************
        zP(:) = false;
        zP(~coverRow,~coverColumn) = ~dMat(~coverRow,~coverColumn);
        Step = 6;
        while any(any(zP(~coverRow,~coverColumn)))
            [uZr,uZc] = find(zP,1);
            primeZ(uZr,uZc) = true;
            stz = starZ(uZr,:);
            if ~any(stz)
                Step = 5;
                break;
            end
            coverRow(uZr) = true;
            coverColumn(stz) = false;
            zP(uZr,:) = false;
            zP(~coverRow,stz) = ~dMat(~coverRow,stz);
        end
        if Step == 6
            % ************************************************************************
            % STEP 6: Add the minimum uncovered value to every element of each covered
            %         row, and subtract it from every element of each uncovered column.
            %         Return to Step 4 without altering any stars, primes, or covered lines.
            %************************************************************************
            M=dMat(~coverRow,~coverColumn);
            minval=min(min(M));
            if minval==inf
                return
            end
            dMat(coverRow,coverColumn)=dMat(coverRow,coverColumn)+minval;
```

```
            dMat(~coverRow,~coverColumn)=M-minval;
        else
            break
        end
    end
    %*************************************************************************
    % STEP 5:
    %  Construct a series of alternating primed and starred zeros as
    %  follows:
    %  Let Z0 represent the uncovered primed zero found in Step 4.
    %  Let Z1 denote the starred zero in the column of Z0 (if any).
    %  Let Z2 denote the primed zero in the row of Z1 (there will always
    %  be one).  Continue until the series terminates at a primed zero
    %  that has no starred zero in its column.  Unstar each starred
    %  zero of the series, star each primed zero of the series, erase
    %  all primes and uncover every line in the matrix.  Return to Step 3.
    %*************************************************************************
    rowZ1 = starZ(:,uZc);
    starZ(uZr,uZc)=true;
    while any(rowZ1)
        starZ(rowZ1,uZc)=false;
        uZc = primeZ(rowZ1,:);
        uZr = rowZ1;
        rowZ1 = starZ(:,uZc);
        starZ(uZr,uZc)=true;
    end
end

% Cost of assignment
assignment(validRow,validCol) = starZ(1:nRows,1:nCols);
cost = sum(costMat(assignment));
end

%% MATLAB实例：为匹配真实标签，对训练得到的标签进行调整 - 凯鲁嘎吉 - 博客园
% 来自：https://www.cnblogs.com/kailugaji/p/11771226.html
function [ new_label ] = label_map( label, gnd )
%为匹配真实标签，对标签重新调整
K = length(unique(gnd));
cost_mat = zeros(K,K);
for i=1:K
    idx = find(label==i);
    for j=1:K
        cost_mat(i,j) = length(find(gnd(idx)~=j));
    end
end
[assignment, ~] = munkres(cost_mat);
[assignedrows, ~]=find(assignment');
```

```
new_label = label;
for i=1:K
    idx = find(label==i);
    new_label(idx) = assignedrows(i);
end
end

%% MATLAB聚类有效性评价指标（外部 成对度量） - 凯鲁嘎吉 - 博客园
% 来自：https://www.cnblogs.com/kailugaji/p/11926253.html
function result = Evaluate(real_label,pre_label)
% This fucntion evaluates the performance of a classification model by
% calculating the common performance measures: Accuracy, Sensitivity,
% Specificity, Precision, Recall, F-Measure, G-mean.
% Input: ACTUAL = Column matrix with actual class labels of the training
%                 examples
%        PREDICTED = Column matrix with predicted class labels by the
%                    classification model
% Output: EVAL = Row matrix with all the performance measures
% https://www.mathworks.com/matlabcentral/fileexchange/37758-performance-measures-for-classification

idx = (real_label()==1);

p = length(real_label(idx));
n = length(real_label(~idx));
N = p+n;

tp = sum(real_label(idx)==pre_label(idx));
tn = sum(real_label(~idx)==pre_label(~idx));
fp = n-tn;
fn = p-tp;

tp_rate = tp/p;
tn_rate = tn/n;

accuracy = (tp+tn)/N; %准确度
sensitivity = tp_rate;  %敏感性：真阳性率
specificity = tn_rate; %特异性：真阴性率
precision = tp/(tp+fp);  %精度
recall = sensitivity;  %召回率
f_measure = 2*((precision*recall)/(precision + recall));  %F-measure
gmean = sqrt(tp_rate*tn_rate);
Jaccard=tp/(tp+fn+fp); %Jaccard系数

result = [accuracy sensitivity specificity precision recall f_measure gmean Jaccard];
end
```

结果:

未调整标签的准确度：0.294118
调整标签后的准确度：0.647059

完成。