Mircoarray Analysis - Machine Learning

Table of Contents

Template by [Thinh Nguyen Fernando Ramirez] Code by [Thinh Nguyen Fernando Ramirez] Adapted from [Dr. Ahmet Sacan - svmdemo]

Background

In this study you will analyze a Breast Cancer dataset, GSE7390, and identify a gene signature for prediction of Breast Cancer replase. Use SVM (support-vecot machine) to predict relapse. Use a forward-selection strategy and 10-fold crossvalidation to determine the best gene signature

Download and parse the dataset. You may use bmes_downloadandparsegse('GSE7390')

```
%(which downloads the series file and parses it using
 geoseriesread()).

gse = bmes_downloadandparsegse('GSE7390')
data = gse.Data';
genenames = data.colnames;
%genes in columns, samples in rows
%gene ids have the 1007_s_at
%compare and have the 76-genes, update to variable X, no need to keep
 the
%remaining genes
```

```
Downloading https://ftp.ncbi.nlm.nih.gov/geo/series/GSE7nnn/GSE7390/
matrix/GSE7390_series_matrix.txt.gz ...
Reading C:\Users\Fernando A. Ramirez\AppData\Local\Temp
\GSE7390.txt ...
```

```
gse =

  struct with fields:

    Header: [1×1 struct]
      Data: [22283×198 bioma.data.DataMatrix]
```

you do not need to to tranlsate the Probe names to gene IDS; hence you

do not need to download the GPL platform file for this dataset.

```
%Finding the e.rfs with 1 indicating the releaps and 0 indicative of
 no
%relapse
```

in the Header of the Series, you can find the patient characteristics.

Cancer relapse status is given as "e.rfs" with 1 indicating relapse, and 0 indicating no relapse.

```
%where gene located in the datamatrix
%if possible use Map to the genes

e_rfs = strcmp(gse.Header.Samples.characteristics_ch1(:,1),'e.rfs: 0')
 |...
    strcmp(gse.Header.Samples.characteristics_ch1(:,1),'e.rfs: 1');
```

Instead of using all of the genes, use only the 76-genes listed in Table 3

of the `Gene-expression profile to predict distant mestatsis of lymph-node- negative primary breast cancer`. Filter out the genes that are not part of the 76-genes, you won't need them for the rest of the assigment

```
load_sevensix = fileread('76_genes.txt');
%entries  = strsplit(load_sevensix, newline);
sevensix_hold = regexp([newline load_sevensix], '\n([\d_a-zA-Z]+)
 ','tokens');
sevensix = [sevensix_hold{:}]';

%logical vector to iterate over
Isevensix= logical(zeros(size(genenames)));

for i=1:length(sevensix)
    Isevensix = Isevensix | strcmp(sevensix(i),genenames);
end

%sum(Isevensix(:) == 1) should produce samples that hold true, which
 are
```

```
%found in the list of the sevensix genes and genenames.
```

Randomly pick out of 90% of the samples to serve as training data, and

the remaining 10 to serve as test data. report on the

```
%extract the e_rfs for training dataset
T = gse.Header.Samples.characteristics_ch1(e_rfs,:);
rawdata = data(:,sevensix);

x = double(rawdata);

cv = cvpartition(T, 'k', 10);

%using more robust method for the cvpartition

Itrain = cv.training(1);
Itest = cv.test(1);
```

Get SVM predctions using predict() on the test data. Calculate and

report the accuracy rate (for a single partition/fold) -- after training models are set

```
%setting-up training models
Xtrain = x(Itrain,:);
Xtest = x(Itest,:);
Ttrain = T(:,Itrain)';
Ttest = T(:,Itest)';
```

Train a SVM model using fitcsvm() on the training data. Note that SVM

considers each row as a sample to be predicted and consideres each column as features (genes).

```
mdl = fitcsvm(Xtrain,Ttrain,'KernelFunction','rbf');
Ytest = mdl.predict(Xtest);
%training modle by Gaussian or Radial Basis Function (RBF) kernel,
 default
%for one-class learning.

numcorrect = sum(strcmp(Ytest,Ttest));
numerror = sum(~strcmp(Ytest, Ttest));

accuracy = numerror / numel(Ttest);
fprintf('Accuracy of model is %.2f%%\n',(accuracy*100))

%errorate and the accuracy should equal 1.
```

```
Accuracy of model is 47.37%
```

Write an evaluation function numerror=hwmaml_breastcancer_trainandtest(Xtrain,Ttrain,Xtest,Ttest)

that trains an SVM using Xtrain & Ttrain, where Xtrain contains gene expression data for a subset of samples, and Ttrain is binary vector of class labels (indicating cancer relaspse status) and calculates the number of errors on the test data Xtest & Ttest.

```
% Back in hwmaml_breastcancer.m, calculate and report the accuracy
% (for a single partition/fold), this time by calling the
% hwmaml_breastcancer_trainandtest() function you wrote.


[numerrortrain accuracytrain] =
 hwmaml_breastcancer_trainandtest(Xtrain,Ttrain,Xtest,Ttest);

fprintf('Trained numerror of model is %.1f\n',(numerrortrain));
fprintf('Trained Accuracy of model is %.2f%%\n',(accuracytrain*100))
```

```
Trained numerror of model is 9.0
Trained Accuracy of model is 52.63%
```

Perform forward selection of features (genes) that give the best prediction

results (as measured by accuracy). Use sequentialfs() > Create a 10-fold cross-validation of all data samples using cvpartition(). You will pass this to sequentialfs(). Report the names of the genes that were selected by sequentialfs to have the best accuracy.

```
Iselection =
 sequentialfs(@hwmaml_breastcancer_trainandtest,x,T','cv',cv,'options',...
    statset('display','iter'),'direction','forward');
fprintf('Names of the genes selected by sequentialfs with best
 accuracy are: %s \n',...
    strjoin(rawdata.colnames(Iselection),' , '))
errors = crossval(@hwmaml_breastcancer_trainandtest,
 x(:,Iselection),T','partition',cv);
```

```
Start forward sequential feature selection:
Initial columns included:  none
Columns that can not be included:  none
Step 1, added column 15, criterion value 0.363636
Step 2, added column 61, criterion value 0.308081
Step 3, added column 70, criterion value 0.29798
Step 4, added column 51, criterion value 0.292929
Step 5, added column 35, criterion value 0.257576
Step 6, added column 9, criterion value 0.247475
Final columns included:  9 15 35 51 61 70
```

```
Names of the genes selected by sequentialfs with best accuracy are:
 210314_x_at , 211382_s_at , 217815_at , 215633_x_at , 218430_s_at ,
 202239_at
```

Using the list of genes selected, report the 10-fold cross-validation

accurary of the SVM model. results are the same as the sequentialfs..

```
tenfold_cross_val_acc = 1 - sum(errors)/numel(T);
fprintf('The Accuracy after cross validation is %.2f%%\n',
(tenfold_cross_val_acc)*100)
```

```
The Accuracy after cross validation is 75.25%
```

Published with MATLAB® R2020b