Classification in the Presence of Missing Data

[View all machine learning examples](https://www.mathworks.com/solutions/machine-learning/examples.html)

Missing data is quite common when dealing with real world datasets. There are several ways to improve prediction accuracy when missing data in some predictors without completely discarding the entire observation. This example shows how decision trees with surrogate splits can be used to improve prediction accuracy in the presence of missing data.

Load Data for Classification

rng(5); % For reproducibility

load ionosphere;

labels = unique(Y);

Partition 70% of the Data into a Training Set and 30% into a Test Set

cv = cvpartition(Y,'holdout',0.3);

Xtrain = X(training(cv),:);

Ytrain = Y(training(cv));

Xtest = X(test(cv),:);

Ytest = Y(test(cv));

Use Bagged Decision Trees to Classify the Ionosphere Data

Bagging (bootstrap aggregating), is an ensemble approach which involves training several weak learners to create a strong classifier.

% Classification Tree is chosen as the learner

mdl1 = ClassificationTree.template('NVarToSample','all');

RF1 = fitensemble(Xtrain,Ytrain,'Bag',150,mdl1,'type','classification');

% Classification Tree with surrogate splits is chosen as the learner

mdl2 = ClassificationTree.template('NVarToSample','all','surrogate','on');

RF2 = fitensemble(Xtrain,Ytrain,'Bag',150,mdl2,'type','classification');

Suppose half of the values in the test set are missing:

Xtest(rand(size(Xtest))>0.5) = NaN;

Predict Responses Using Both Approaches

y\_pred1 = predict(RF1,Xtest);

confmat1 = confusionmat(Ytest,y\_pred1);

y\_pred2 = predict(RF2,Xtest);

confmat2 = confusionmat(Ytest,y\_pred2);

disp('Confusion Matrix - without surrogates')

disp(confmat1)

disp('Confusion Matrix - with surrogates')

disp(confmat2)

Confusion Matrix - without surrogates

67 1

24 13

Confusion Matrix - with surrogates

65 3

4 33

Visualize Misclassification Error

Decreasing value with number of trees indicates good performance.

figure

subplot(2,2,1:2)

plot(loss(RF1,Xtest,Ytest,'mode','cumulative'),'LineWidth',3);

hold on;

plot(loss(RF2,Xtest,Ytest,'mode','cumulative'),'r','LineWidth',3);

legend('Regular trees','Trees with surrogate splits');

xlabel('Number of trees');

ylabel('Test classification error','FontSize',12);

subplot(2,2,3)

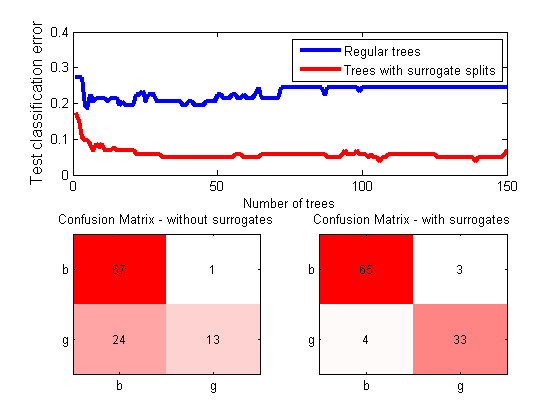
[hImage, hText, hXText] = heatmap(confmat1, labels, labels, 1,'Colormap','red','ShowAllTicks',1);

title('Confusion Matrix - without surrogates')

subplot(2,2,4)

heatmap(confmat2, labels, labels, 1,'Colormap','red','ShowAllTicks',1);

title('Confusion Matrix - with surrogates')



The misclassification error is much lower when surrogate splits are used with decision trees.

The heatmap visualization in the subplot was generated using [Customizable Heat Maps](https://www.mathworks.com/matlabcentral/fileexchange/24253-customizable-heat-maps).