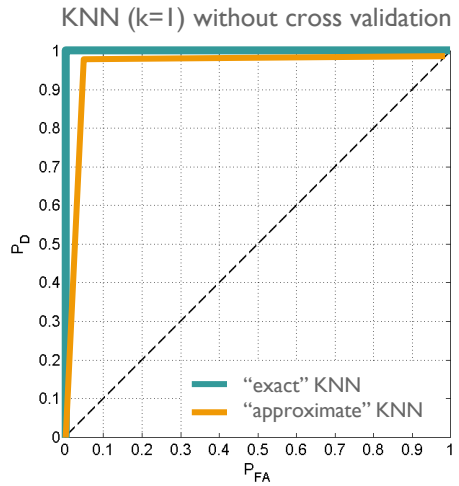
Computational Considerations

kNN training: *No learning of model parameters*
"Training" consists of storing the training observations
Hyperparameter k selected through cross-validation

kNN testing: *Calculate distance between x_{test} and all training observations to find k nearest neighbors*
→ Computational load is in testing, not training

("exact" KNN)



How to ease computation ("approximate" KNN)?

Generally, 2 strategies:
1) *Store only training observations that are near/define decision boundary*
   → requires choosing the decision rule ahead of time
2) *Strategically search over training observations, perhaps accepting the "almost nearest" neighbors*

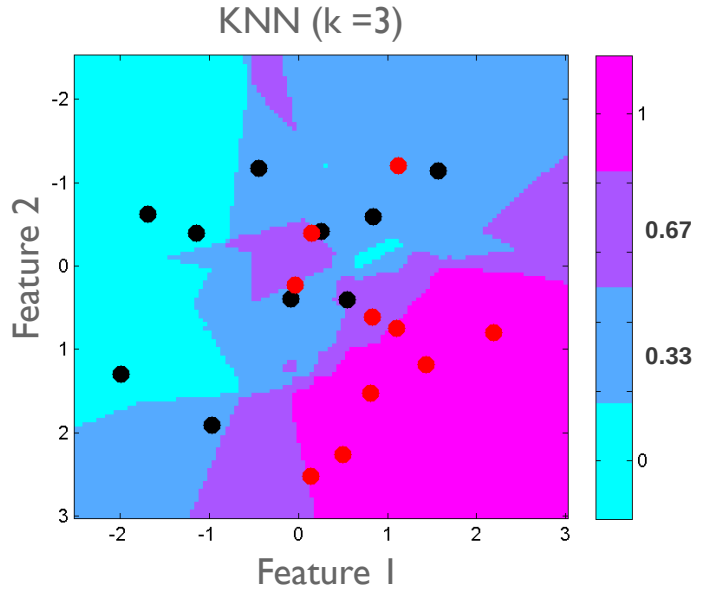
Classifiers: Visualizing Decision Statistic Surfaces

Hypothesize a grid of test data

Calculate the decision statistic at each grid point

- by running the classifier with the list of hypothesized test data as the test data for the classifier

Image (`imagesc`) the grid of decision statistics (the decision statistic surface)



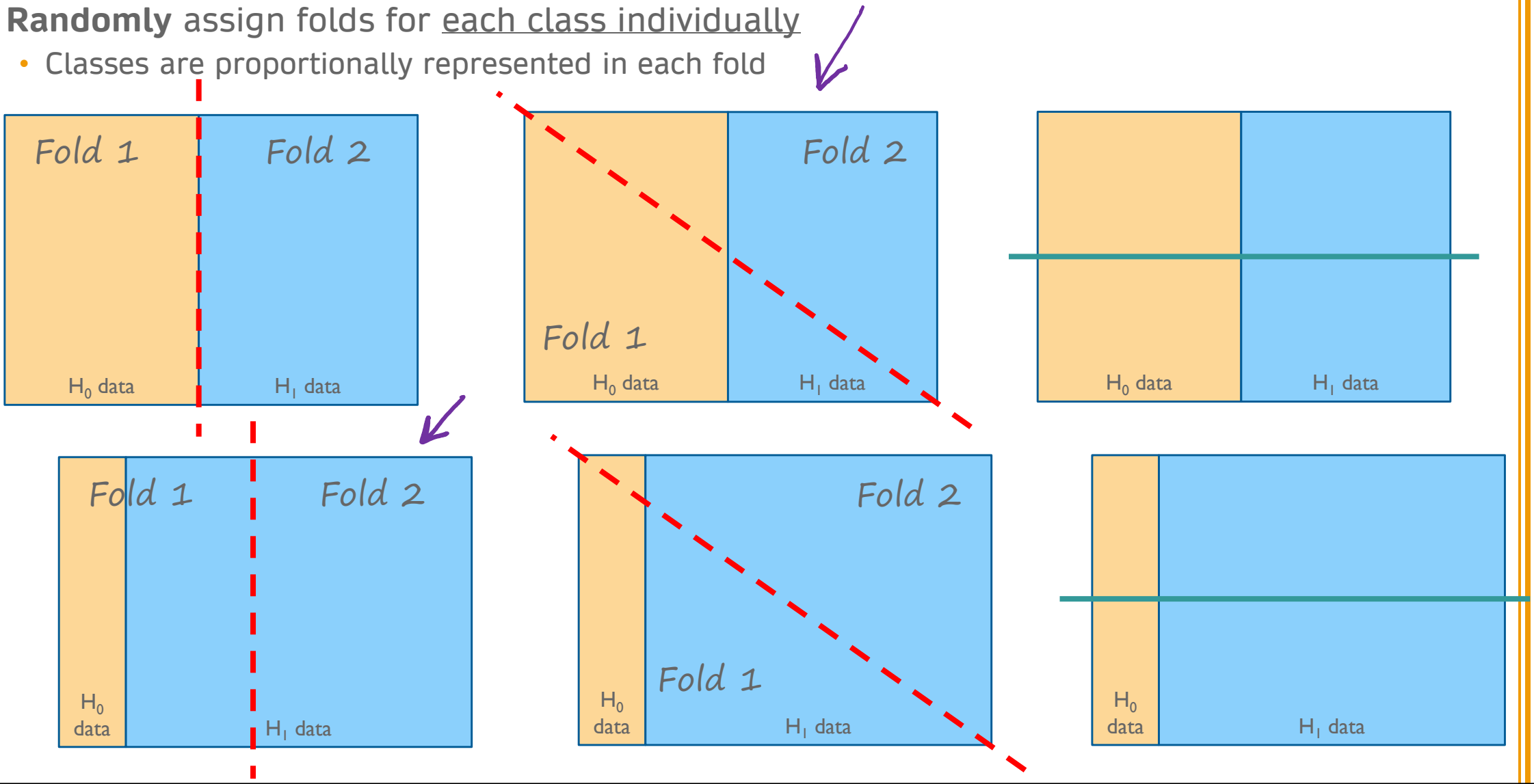
Using only information stored in the classifier structure:

```
x1Range = max(xTrain(:,1))- min(xTrain(:,1));
x2Range = max(xTrain(:,2))- min(xTrain(:,2));
x1 = linspace(min(xTrain(:,1))-0.2*x1Range,
        max(xTrain(:,1))+0.2*x1Range,251);
x2 = linspace(min(xTrain(:,2))-0.2*x2Range,
        max(xTrain(:,2))+0.2*x2Range,251);

% Create the grid of test data points
[xTest1,xTest2] = meshgrid(x1,x2);

% Each column is a feature, each row an observation
xTest = [xTest1(:) xTest2(:)];

% Run the classifier with these test data
dsTest = runClassifier(classifierStructure,xTest);

% dsTest is a vector, reshape it to a matrix
dsTest = reshape(dsTest,length(x2),length(x1));

% Image the decision statistic surface
imagesc(x1([1 end],x2([1 end]),dsTest)

% Add the training data points to the surface
hold on
% H0
plot(xTrain(truth==0,1),xTrain(truth==0,2),'ko')
% H1
plot(xTrain(truth==1,1),xTrain(truth==1,2),'ro')
```

Assigning Folds

Randomly assign folds for <u>each class individually</u>

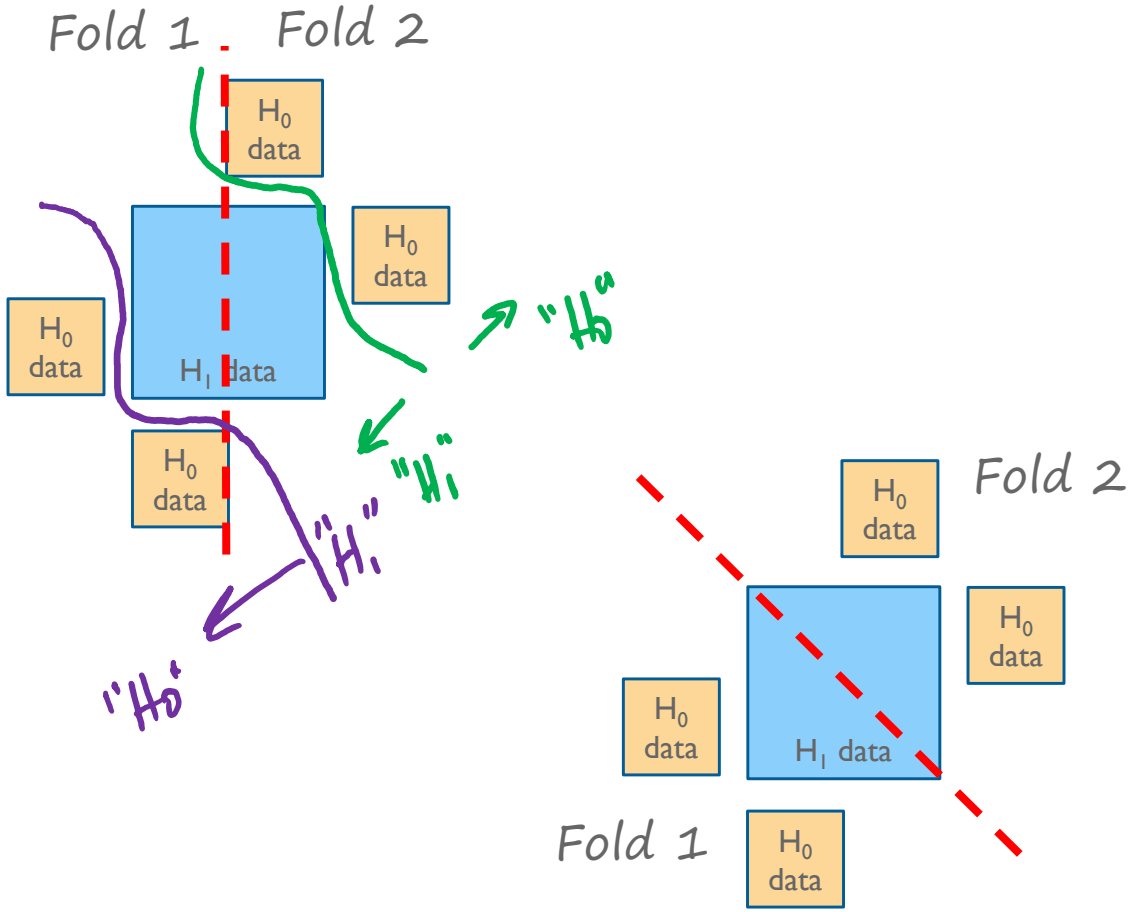
- Classes are proportionally represented in each fold

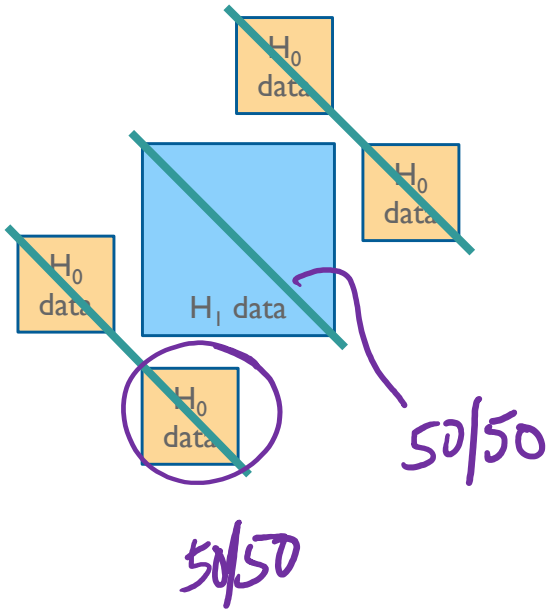


Stratified Cross-Validation

Randomly assign folds for <u>each sub-class individually</u>

- "Clusters" in the data are distributed among folds





Cross-Validation Framework