1. Fit logistic regression: define s. This procedure is really threshold independent!
2. Get training set and calculate .
3. For and define threshold such that for will be as close as possible to . As close as possible means have maximal BA.

X and Y are for two classes

function [bestT, bestErr] = oneDClass(x, y, name)

% oneDClass applied classification with one input attribute by searching

% the best threshold.

%Inputs

% x contains values for Class 1

% y contains values for Class 2

% name is array of strings:

% name(1) is name of attribute

% name(2) is name of the first class

% name(2) is name of the second class

%

%Outputs

% bestT is optimal threshold

% bestErr is minimal error which corresponds to threshold bestT. Error is

% one minus half of sencsitivity + specificity or

% 1 - 0.5\*(TP/Pos+TN/Neg), where

% TP is true positive or the number of correctly recognised casses of

% the first class,

% Pos is the number of casses of the first class,

% TN is true negative or the number of correctly recognised casses of

% the secong class,

% Neg is the number of casses of the second class.

%

%Define numbers of cases

Pos = length(x);

Neg = length(y);

tot = Pos + Neg;

%Define set of unique values

thr = unique([x; y])'; % Get all possible points

%Add two boders

thr = (thr(2:end) + thr(1:end - 1)) / 2;

errs = zeros(1, length(thr));

%Define meaning of "class 1"

xLt = mean(x) > mean(y);

%Define variabled to search

bestErr = tot;

bestT = -Inf;

%Check each threshold

for k = 1:length(thr)

t = thr(k);

nX = sum(x < t);

nY = sum(y >= t);

if xLt

nX = Pos - nX;

nY = Neg - nY;

end

err = 1 - (nX / Pos + nY / Neg) / 2;

if err < bestErr

bestErr = err;

bestT = t;

end

errs(k) = err;

end

end

Meaningless area

Meaningless area

(thr(2:end) + thr(1:end - 1)) / 2;