CPSC-340 Machine Learning and Data Mining 2004

Homework # 6

Due Friday, Dec 3 at 4pm.

NAME:____________________________________

Signature:__________________________________

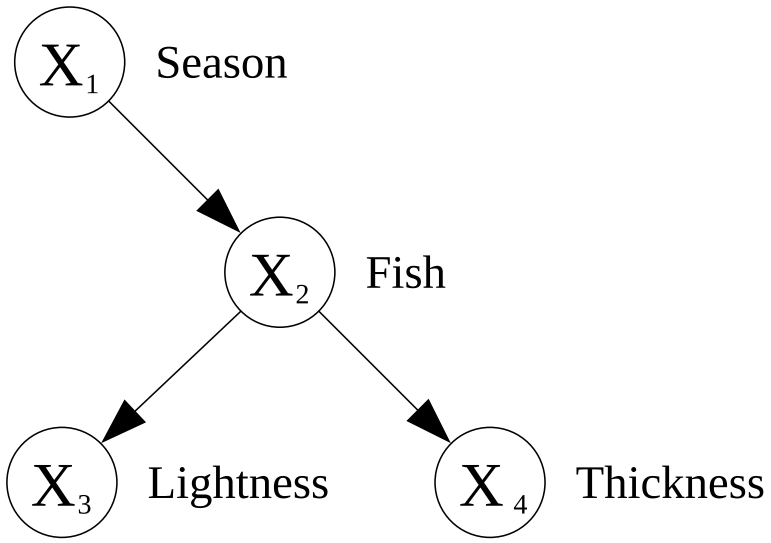
STD. NUM: ________________________________

Email: ____________________________________

General guidelines for homeworks:

You are encouraged to discuss the problems with others in the class, but all write-ups are to be done on your own.
Homework grades will be based not only on getting the "correct answer," but also on good writing style and clear presentation of your solution. It is your responsibility to make sure that the graders can easily follow your line of reasoning.
Try every problem. Even if you can't solve the problem, you will receive partial credit for explaining why you got stuck on a promising line of attack. More importantly, you will get valuable feedback that will help you learn the material.
Please acknowledge the people with whom you discussed the problems and what sources you used to help you solve the problem (e.g. websites, books from the library). This won't affect your grade but is important as academic honesty.
When dealing with Matlab exercises, please attach a printout with all your code and show your results clearly.

1. **Inference in Graphical models**:



Consider the graphical model shown above. Here, the nodes represent the following variables

$$\begin{aligned} X_1 &\in \{\text{winter, spring, summer, autumn}\} \\ X_2 &\in \{\text{salmon, sea bass}\} \\ X_3 &\in \{\text{light, medium, dark}\} \\ X_4 &\in \{\text{wide, thin}\} \end{aligned}$$

The corresponding conditional probability tables are

$$p(x_1) = \begin{pmatrix} .25 & .25 & .25 & .25 \end{pmatrix}$$

$$p(x_2|x_1) = \begin{pmatrix} .9 & .1 \\ .3 & .7 \\ .4 & .6 \\ .8 & .2 \end{pmatrix}$$

$$p(x_3|x_2) = \begin{pmatrix} .33 & .33 & .34 \\ .8 & .1 & .1 \end{pmatrix}$$

$$p(x_4|x_2) = \begin{pmatrix} .4 & .6 \\ .95 & .05 \end{pmatrix}$$

Let's answer the following two queries

- Suppose the fish was caught on December 20 — the end of autumn and the beginning of winter — and thus let $p(x_1) = (.5\ 0\ 0\ .5)$. Suppose the lightness has not been measured but it is known that the fish is thin. Classify the fish as salmon or sea bass.
- Suppose all we know is that the fish is thin and medium lightness. What season is it now, most likely?

To answer this question using the Junction Tree Algorithm, you should

- Download the Bayes Net toolbox of Kevin Murphy from
  `http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html`
- Familiarise yourself with the software by reading the usage instructions file and the tutorial at:
  `http://www.cs.ubc.ca/~murphyk/Software/BNT/usage.html`

2. **Boosting**:

Your task is to apply the boosting algorithm with classification stumps to the tremor dataset. Since you already applied kernel classifiers to this dataset in the previous homework, you should hand in plots comparing the decision boundaries obtained with kernel machines and boosting for both training and test data sets. You should also compare the number of errors on both datasets. To get you started, here is the matlab code for the demo shown in class:

```
clear; echo off;

% PARAMETERS:
% ==========

N = 20;     % Number of data.
d = 2;      % Input data dimension.
c = 2;      % Number of classes (Fixed).
M = 10;     % Number of iterations.

% GENERATE THE DATA:
% ==================

x1 = zeros(N/2,d);
x2 = zeros(N/2,d);
for i=1:N/2,
  x1(i,:) = .1*ones(1,d) + 0.7*randn(1,d);
  x2(i,:) = 0*ones(1,d) + 0.7*randn(1,d);
end;
x=[x1; x2];                     % Input data.
y=[ones(N/2,1); -ones(N/2,1)];  % Class labels.

% TRAIN BOOSTED CLASSIFIER:
% ========================

w = ones(N,M)/N;       % Initial uniform weights.
ylower = zeros(M,1);   % Sign to left of stump.
cutPoint = zeros(M,1); % Location of stump.
dimCut = zeros(M,1);   % Cutting dimension.
error = zeros(M,1);    % error incurred by stump.
```

```
alpha = zeros(M,1);    % Coefficients.
ystump = zeros(N,1);

for m = 1:M,
  % Fit stump:
  [ylower(m),cutPoint(m),dimCut(m),error(m)] = classStump(x,y,w(:,m));
  % Compute stump predictions:
  for i=1:N,
    if x(i,dimCut(m))<=cutPoint(m),
      ystump(i) = ylower(m);
    else
      ystump(i) = -1*ylower(m);
    end;
  end;
  alpha(m) = log((1-error(m))/error(m));
  % Update weights:
  w(:,m+1) = w(:,m) .* exp(alpha(m)*[ystump~=y]);
end;


% GENERATE PREDICTIONS:
% ====================

yp = zeros(N,M);    % predictions for each stump.
ypred = zeros(N,1); % Final predictions.
for i=1:N,
  for m=1:M,
    if x(i,dimCut(m))<=cutPoint(m),
      yp(i,m) = ylower(m);
    else
      yp(i,m) = -1*ylower(m);
    end;
  end;
  ypred(i) = sign(yp(i,:)*alpha);
end;


errorStump1 = sum([yp(:,1) ~= y])
errorStump2 = sum([yp(:,2) ~= y])
errorBoost = sum([ypred ~= y])


% PLOT RESULTS:
% ============

if d==2 % only plot decision boundary for 2d.
  figure(1)
  clf;
  plot(x(y>0,1),x(y>0,2),'r+',x(y<0,1),x(y<0,2),'bx','linewidth',1);
  hold on;
  plot(x(ypred>0,1),x(ypred>0,2),'ro',x(ypred<0,1),x(ypred<0,2),'bo','linewidth',2);
  legend('y=1','y=-1');
  xlabel('x_1');
  ylabel('x_2');
  for m=1:M,
```

```
    if dimCut(m)==1,
      plot(cutPoint(m),[-0.5+min(x(:,2)):.05:0.5+max(x(:,2))],'k');
    else
      plot([-0.5+min(x(:,1)):.05:0.5+max(x(:,1))],cutPoint(m),'k')
    end;
    hold on;
  end;

  % Plot the decision boundary

  rangex1 = [-0.5+min(x(:,1)):.05:0.5+max(x(:,1))];
  rangex2 = [-0.5+min(x(:,2)):.05:0.5+max(x(:,2))];
  [x1,x2] = meshgrid(rangex1,rangex2);
  [mx,nx] = size(x1);
  N = nx*mx;
  x = [reshape(x1,N,1) reshape(x2,N,1)];
  yp = zeros(N,M); % predictions for each stump.
  ypred = zeros(N,1); % Final predictions.
  for i=1:N,
    for m=1:M,
      if x(i,dimCut(m))<=cutPoint(m),
        yp(i,m) = ylower(m);
      else
        yp(i,m) = -1*ylower(m);
      end;
    end;
    ypred(i) = yp(i,:)*alpha;
  end;
  ypred = reshape(ypred,mx,nx);
  contour(x1,x2,ypred,[0 0],'b');
  hold off;
end;
```

You'll also need the code for the weak (stump) classifier:

```
function [ylower,cutPoint,dimCut,minError] = classStump(x,y,w)
% PURPOSE : classification stump. ylower is the sign of the prediction for
% points lower than or equal to the threshold along the dimension dimCut. The
% input to the classifier is the data (x,y) and the weights w. minError is
% the weighted error incurred by the stump.

% CLASSIFICATION STUMP:
% ====================

doplot = 0; % Set to 1 if you want iteration plots for d=2.
[N,d] = size(x);
c = 2; % Fixed: binary classification.
sumw = sum(w);
```

```
lprop=zeros(N-1,c);
ly = zeros(N-1,1);  % Prediction to the left of the stump.
ry = zeros(N-1,1);  % Prediction to the right of the stump.
error=zeros(N-1,d);
for p=1:d,
  [z,idz] = sort(x,1);
  xsorted = x(idz(:,p),:);
  ysorted = y(idz(:,p));
  wsorted = w(idz(:,p));
  for j=1:N,
    lprop(j,1) = (1/j)*sum(ysorted(1:j)==1);  %proportion of class 1 on left.
    lprop(j,2) = (1/j)*sum(ysorted(1:j)==-1); %proportion of class 2 on left.
    ly(j,p) = 1*(lprop(j,1)>lprop(j,2)) + (-1)*(lprop(j,1)<=lprop(j,2));
    ry(j,p) = -1*ly(j,p);

    errorLeft = sum(wsorted(1:j).*[ysorted(1:j)~=ly(j,p)]);
    errorRight = sum(wsorted(j+1:N).*[ysorted(j+1:N)~=ry(j,p)]);
    error(j,p) =  (errorLeft + errorRight)/sumw;

    if ((doplot==1)&(p==1)),
      figure(1)
      clf;
      plot(x(y>0,1),x(y>0,2),'r+',x(y<0,1),x(y<0,2),'bx','linewidth',1);
      hold on;
      plot(xsorted(ly(1:j,p)>0,1),xsorted(ly(1:j,p)>0,2),'ro',xsorted(ly(1:j,p)<0,1),xsorted(ly(1:j,p)<0,2),'
      plot(xsorted(j),[min(x(:,2)):.01:max(x(:,2))],'k');
      legend('y=1','y=-1');
      xlabel('x_1');
      ylabel('x_2');
      hold off;
    end;
  end;
end;

% Look at error to find best cut (mins) and best dimension (minp)
k=min(error);
[j,minp]=min(k);
[i,mins] = min(error(:,minp));
ylower = ly(mins,minp);        % RETURN: sign to the left of stump.
dimCut = minp;                 % RETURN: dimension of cut.
minError = min(min(error));    % RETURN: weighted error.
[z,idz] = sort(x,1);
xsorted = x(idz(:,minp),:);
criticalx = xsorted(mins,:);
cutPoint = criticalx(1,minp);  % RETURN: location of stump.
```