

ROCHESTER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING
CMPE 677 Machine Intelligence
HW #6

Name: _____

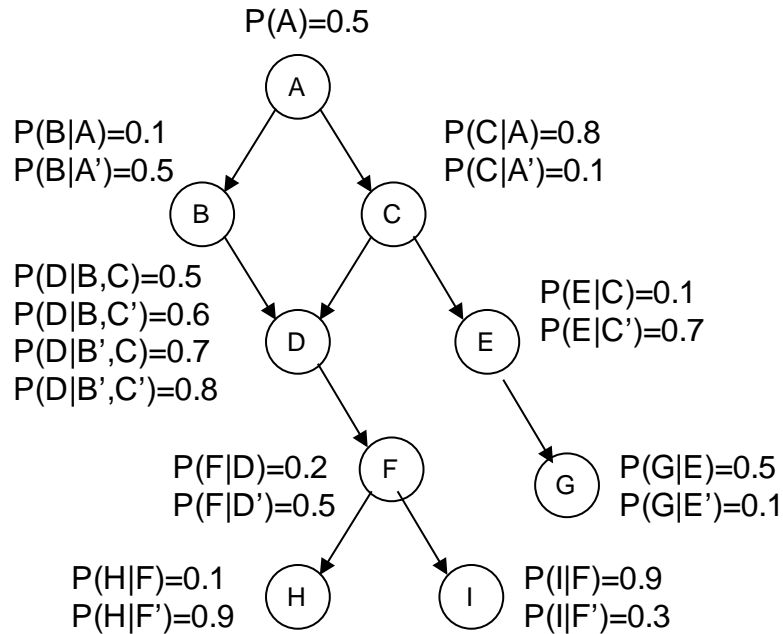
Due 11:55pm, Fri, 10/27/2017, submit via Dropbox. Work alone. All questions pertain to Matlab/Octav.

1. (5 pts) The probability of your robot sensors seeing a traffic light is 0.25. The probability of your robot sensors seeing a red light camera is 0.2. The probability of your sensors seeing a traffic light and red light camera at the same time is 0.05. What is the probability of your robot sensors seeing a red light camera given that they are seeing a traffic light?

Ans:

2. (10 pts)) Given the Bayesian Net, solve for:

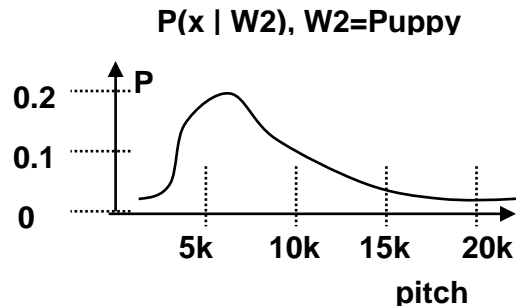
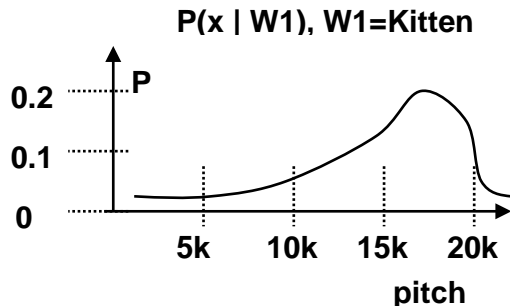
- a) $P(A,B,C,D,E,G)$, show work using repeated use of the chain rule
- b) $P(BC)$



Ans:

3. (5 pts) Using the Bayes net in the previous problem, what is the formula for $P(A, B', C, D, E', F, I)$ via visual inspection:

4. (10 pts) You are tasked to program a robot to find three kittens in a crowded room of 27 puppies. After much trial and error, your best predictor is a sound sensor. You analyze sound clips from several kittens and sound clips from several puppies and get the following data:

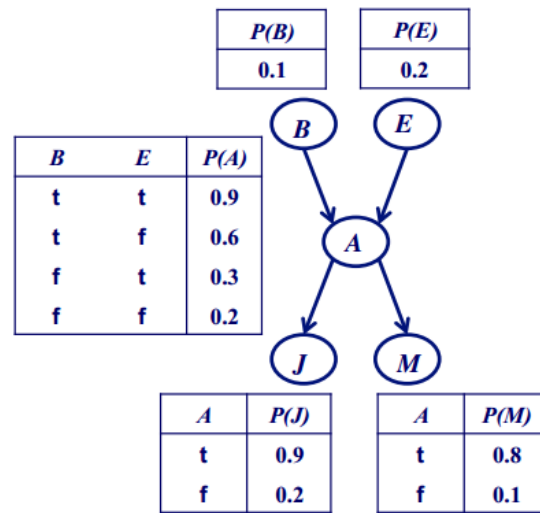


If your robot hears a sound pitch of 14k, use Maximum Likelihood and Bayesian classifiers to estimate if you are hearing a kitten or a puppy. Please explain and give numerical estimates for each.

Ans:

(-5 if explanation does not include description of how kitten is higher for max likelihood. -5 if no formulas used for bayes rule classification.)

5. (15 pts) You are given the following Bayes net:



Ten sample measurements were made, but the A data was lost. Using the EM algorithm, starting with the initial parameters below, one can fill out the right-most column using the result after the first expectation step for the A column. Show how the 'A from E step' entry is calculated for sample 1 and 2.

| sample | B | E | A | J | M | A from E step |
|--------|---|---|---|---|---|--------------------|
| 1 | 0 | 0 | ? | 0 | 0 | 0.006896552 |
| 2 | 0 | 0 | ? | 1 | 0 | 0.2 |
| 3 | 1 | 0 | ? | 1 | 0 | 0.9818 |
| 4 | 0 | 0 | ? | 0 | 1 | 0.2 |
| 5 | 0 | 1 | ? | 1 | 0 | 0.3 |
| 6 | 0 | 0 | ? | 0 | 1 | 0.2 |
| 7 | 1 | 1 | ? | 1 | 1 | 0.9969 |
| 8 | 0 | 0 | ? | 0 | 0 | 0.006896552 |
| 9 | 0 | 0 | ? | 1 | 0 | 0.2 |
| 10 | 0 | 0 | ? | 0 | 1 | 0.2 |

Ans (show calculations):

6. (9 pts) After solving for the expected values for A in the previous problem (in the ‘A from E step’ column), the EM algorithm would next update the probability statistics relating to A by computing the maximization step. Solve for $P(A|BE)$, $P(A|B'E)$, $P(A|BE')$, $P(A|B'E')$, $P(J|A)$, $P(J|A')$, $P(M|A)$, $P(M|A')$ after the first maximization step for the for the Bayes net in the previous problem.

To simplify the math, use the following estimated values of “A from E step”:

| sample | B | E | A | J | M | A from E step |
|--------|---|---|---|---|---|---------------|
| 1 | 0 | 0 | ? | 0 | 0 | 0.01 |
| 2 | 0 | 0 | ? | 1 | 0 | 0.1 |
| 3 | 1 | 0 | ? | 1 | 0 | 0.8 |
| 4 | 0 | 0 | ? | 0 | 1 | 0.1 |
| 5 | 0 | 1 | ? | 1 | 0 | 0.2 |
| 6 | 0 | 0 | ? | 0 | 1 | 0.1 |
| 7 | 1 | 1 | ? | 1 | 1 | 0.9 |
| 8 | 0 | 0 | ? | 0 | 0 | 0.01 |
| 9 | 0 | 0 | ? | 1 | 0 | 0.1 |
| 10 | 0 | 0 | ? | 0 | 1 | 0.1 |

We have two scenarios here, when A is the dependent variable and when A is the independent variable.

- 1) When A is the dependent variable:

Because the values of A are floating point, you can calculate these using:

$$P(A|B, E) = \frac{\sum A_{for\ which\ B\ and\ E\ are\ true}}{number\ of\ entries\ in\ which\ B\ and\ E\ are\ true}$$

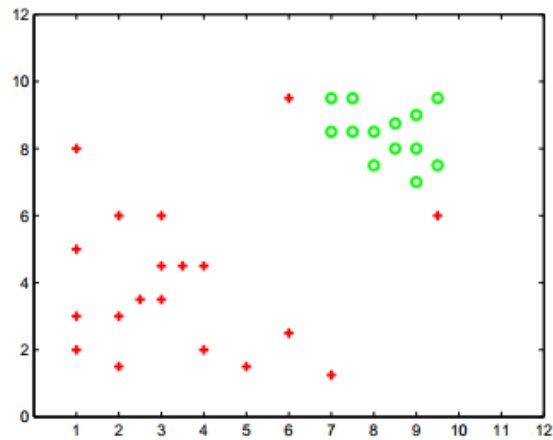
(To solve for $P(A|B, E)$, we find all row's for which B and E are true, sum up the corresponding A values, and divide by the number of such rows)

- 2) When A is the independent variable, such as $P(J / A)$: First threshold A to $[0,1]$, where all A values ≥ 0.5 are set to 1, all A values < 0.5 are set to 0, then compute statistics as in the class slides.

Ans:

7. (8 pts) Consider a point that is correctly classified and distant from an SVM decision boundary. Why would SVM's decision boundary be unaffected by this point, but the decision boundary learned by logistic regression be affected?

8. (12 pts) You are given the following dataset and use SVM with a quadratic kernel- a polynomial kernel with degree 2. The slack penalty C determines the cost with misclassification, where the higher C , the higher the cost.



- Draw a decision boundary for a very high value of C in solid line.
- Draw a decision boundary for a very low value of C in dashed line.
- Circle a point that will not change either decision boundary significantly.
- Insert a new point which will significantly change the decision boundary if a large value of C is used.

9. (21 pts) Download `ex6data1.mat` from MyCourses Modify the following code to produce the plot as shown. (Note, this sample code works for Windows 64bit. If you have Windows 32 bit, Linux, MacOS, or any other OS, see Appendix on how to compile libsvm.) This problem requires you to:
- Update the paths in lines 2 and 3.
 - Compute `w`, note- when doing this, convert `model.SVs` from sparse to full
 - Compute `predictionsTrain` and `predictionsTrainError`
 - Compute `predictionsTest` and `predictionsTestError`

```
close all ; clear all;
cd C:\Users\rwpeec\Desktop\rwpeec\rit\CMPE-677_MachineIntelligence\hwk\hwk6
addpath C:\Users\rwpeec\Desktop\rwpeec\rit\CMPE-677_MachineIntelligence\hwk\hwk6\libsvm-3.18\windows
```

```
load('ex6data1.mat'); %load Andrew Ng data
% Find Indices of Positive and Negative Examples
pos = find(y == 1); neg = find(y == 0);
```

```
% Plot Examples
hold off
plot(X(pos, 1), X(pos, 2), 'g+', 'LineWidth', 3, 'MarkerSize', 12)
hold on;
plot(X(neg, 1), X(neg, 2), 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 7)
```

```
% Cost value
C = 1;
```

```
% Call svmtrain from LIBSVM
% -t 0 says to do a linear kernel
% -c <value> sets cost to <value>, higher c means SVM will weigh errors more
eval(['model = svmtrain(y,X,"-t 0 -c ' num2str(C) '");']);
```

```
% SVM solves  $wx+b$ 
% alpha values are stored in model.sv_coef
% Note: alpha values are really  $y(i) * \alpha(i)$ - so no need to include  $y$  when solving for  $w$ 
% support vectors are stored in model.SVs
% solve for  $w$  here:
%  $w = \text{<insert code here>}$ 
 $w = (\text{model.sv\_coef} * \text{full}(\text{model.SVs}));$  %full converts from sparse to full matrix representation
%  $b$  is stored in -model.rho
 $b = -\text{model.rho};$ 
```

```
%plot boundary ontop of data
xp = linspace(min(X(:,1)), max(X(:,1)), 100);
yp = - (w(1)*xp + b)/w(2);
hold on;
plot(xp, yp, 'b-', 'linewidth', 3);
```

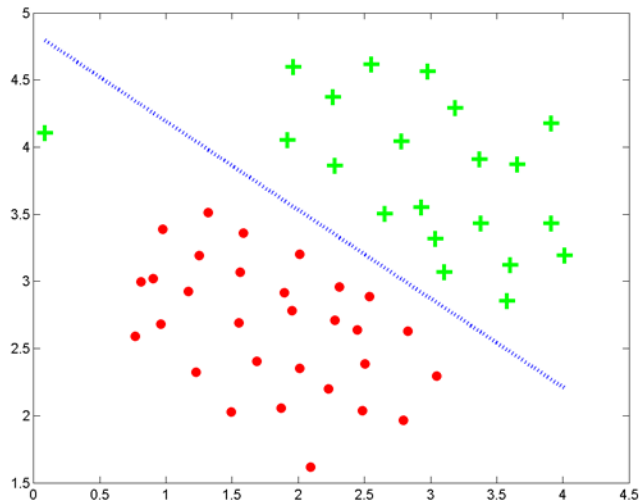
```
%Predictions are  $\text{sign}(\langle x, w \rangle + b)$ , note: can do all predictions in one line
%predictionsTrain = <insert code here>
predictionsTrain = sign(X * w' + b);
predictionsTrain(predictionsTrain== -1) = 0; %change -1 to 0 to match GT
```

```

%compute training error
%predictionsTrainError = <insert code here>
predictionsTrainError = sum(predictionsTrain~= y)/length(y);
fprintf('Error on train set = %0.2f%%\n',predictionsTrainError*100);

%Now we will see how this does on a test set
Xtest = [ 1 3; 2 3; 3 3; 4 3; 1 4; 2 4; 3 4; 4 4];
ytest = [0 0 0 1 0 1 1 1]';
%predictionsTest = <insert code here>
predictionsTest = sign(Xtest * w' + b);
predictionsTest(predictionsTest==-1) = 0; %change -1 to 0 to match GT
%predictionsTest = <insert code here>
predictionsTestError = sum(predictionsTest~= ytest)/length(ytest);
fprintf('Error on test set = %0.2f%%\n',predictionsTestError*100);
%print -dpng hwk6_q5.png

```



What is D ?
 What is n ?
 How many classes are there?
 How many support vectors are there?
 What is the Error on the train set?
 What is the Error on the test set?
 What is the smallest integer value of the Cost value C , such that the train error is 0%?

Ans (only answer questions, do not show code or plots)

10. (10 pts) Download `ex6data2.mat` from MyCourses. Update the code from the previous problem to show the linear SVM hyperplane with $C=1$. Show your plot at 50% size:

11. (15 points) The linear fit is obviously not appropriate for such a non-linear dataset. LIBSVM supports many different types of kernels. The most popular type of kernel is the radial basis function which implements:

$$K(x^{(i)}, x^{(j)}) = \exp\left(-\gamma(x^{(i)} - x^{(j)})^2\right)$$

The value of γ needs to be specified. You can use the radial basis function via the syntax:

```
model = svmtrain( ytrain, Xtrain, '-t 2 -c 1 -g 10' );
```

```
% -t 2 says use radial basis function
```

```
% -c <value> sets cost to <value>, higher c means SVM will weigh errors more
```

```
% -g <value> sets gamma to <value>, lower g means SVM will be smoother
```

```
predict = svmpredict( ytest, Xtest, model, '-q' );
```

```
% -q suppresses interactive output
```

Download `visualizeBoundary2D.m` from MyCourses. Using a Cost $c=1$, solve for the smallest gamma (use integer values, starting with 0 and incrementing by 10) value that correctly classifies all training points. Show your plot at 50% size with the g value in title, show code.

12. (13 points) The result in the previous problem no doubt overfit the data. To get around this, we can use cross validation to choose the optimal value of c and g . Download `classify677_hwk6.m` from MyCourses. This file has been modified to allow SVM classification. Test with the following code:

```
close all ; clear all;
cd C:\Users\rwpeec\Desktop\rwpeec\rit\CMPE-677_MachineIntelligence\hwk\hwk6
addpath C:\Users\rwpeec\Desktop\rwpeec\rit\CMPE-677_MachineIntelligence\hwk\hwk6\libsvm-3.18\windows

load('ex6data2.mat'); %load Andrew Ng data
%convert from y=[0,1] to y={1,2} for confusion matrix indexing
y(y==1)=2;
y(y==0)=1;

options.method = 'SVM';
options.numberOfFolds = 5;
options.svm_t=2;
options.svm_c=1;
options.svm_g=200;
[confusionMatrix,accuracy] = classify677_hwk6(X,y,options);
```

The resulting confusionMatrix and accuracy should be about:

SVM: Accuracy = 93.40%

Confusion Matrix:

| | |
|-----|-----|
| 327 | 56 |
| 1 | 479 |

Solve for the optimum c and g , allowing c to go from 1:2:100 and g to go from 0:10:300. Show corresponding accuracy, confusion matrix, and plot of data (50% size) with title using command:

```
str = sprintf('The best accuracy is: %0.2f%%:c=%d, g=%d\n',maxAccuracy*100,best_c,best_g);
title(str,'fontsize',14);
```

Appendix- compiling libsvm for an OS other than 64 bit windows

- 1) Download a compiler.
 - a. For pc:
 - i. Download c++ compiler:
<http://www.microsoft.com/en-us/download/details.aspx?id=5555>
(Microsoft visual C++ 2010 redistributable)
OR
Install any visual studio
 - b. For mac:
 - i. Install Xcode,
 - ii. Open Xcode, go to Preferences -> Downloads -> Component, click
Install for Command Line Tools
- 2) Get compiler working within Matlab:
 - a. >> mex -setup

For PC:

(Note commands are from within matlab)

- 1) >> cd('C:<insertPathHere>\libsvm-3.18\matlab')
- 2) >> make
- 3) >> addpath('C:<insertPathHere>\libsvm-3.18\windows')

For mac:

- 1) From command window of matlab,(edit file mexopts.sh)
cd (matlabroot)
cd bin
edit mexopts.sh
edit following 4 variables:
CC='xcrun -sdk macosx10.X clang'
MW_SDK_TEMP="find `xcode-select -print-path` -name MacOSX10.X.sdk"
MACOSX_DEPLOYMENT_TARGET='10.X '
CXX='xcrun -sdk macosx10.X clang++'
where X is the version of OSX: 8,9,10,11
- 2) cd to the matlab folder of libsvm-18, add the following line before line #include "mex.h" of all *.c files:
typedef uint16_t char16_t;
- 3) Using command "make" of matlab to compile the library from matlab folder of libsvm-18