Homework 1

Matt Dickenson
CS 527
Fall 2014

Problem 1

$$
\begin{aligned}
X &\sim Unif(0,1) \\
m &= \mathbb{E}(X) = 0.5 \\
\mathbb{E}(|X-m|) &= \mathbb{E}(|X-0.5|) \\
&= -E(X_{x\leq 0.5} - 0.5) + E(X_{x>0.5} - 0.5) \\
&= -(-0.25) + 0.25 \\
&= 0.5
\end{aligned}
$$

Problem 2

By Bayes' Theorem, we can compute the posterior probability that coin 2 was chosen by:

$$
\begin{aligned}
P(c=2|h=1) &= \frac{P(h=1|c=2)P(c=2)}{P(h=1)} \\
&= \frac{P(h=1|c=2)P(c=2)}{P(h=1|c=2)P(c=2) + P(h=1|c=1)P(c=1)} \\
&= \frac{0.8 \times 0.5}{0.8 \times 0.5 + 0.5 \times 0.5} \\
&= \frac{0.4}{0.65} \\
&\approx 0.615
\end{aligned}
$$

Problem 3

Given Equation 2.3 and the fact that x and y are independent, we can show that $Pr(x|y=y^*) = Pr(x)$ by:

$$
\begin{aligned}
Pr(x|y=y^*) &= \frac{Pr(x,y=y^*)}{\int Pr(x,y=y^*dx)} \\
&= \frac{Pr(x,y=y^*)}{Pr(y=y^*)} \\
&= \frac{Pr(x)\cdot Pr(y=y^*)}{Pr(y=y^*)} \\
&= Pr(x)
\end{aligned}
$$

Problem 4

The expected value of one roll of this biased die, x, is:

$$
\begin{aligned}
\mathbb{E}(x) &= 1\cdot\frac{1}{12}+2\cdot\frac{1}{12}+3\cdot\frac{1}{12}+4\cdot\frac{1}{12}+5\cdot\frac{1}{6}+6\cdot\frac{1}{12} \\
&= \frac{10}{12}+\frac{5}{6}+\frac{6}{2} \\
&= \frac{56}{12} \\
&= \frac{14}{3} \\
&\approx 4.67
\end{aligned}
$$

The expected value of the sum of two rolls is:

$$
\begin{aligned}
\mathbb{E}(2x) &= 2\mathbb{E}(x) \\
&= 2(\frac{14}{3}) \\
&= \frac{28}{3} \\
&\approx 9.34
\end{aligned}
$$

Problem 5

Using the relations given in Exercise 2.9, we can show that $\mathbb{E}[(x-\mu)^2] = \mathbb{E}[x^2] - E[x]E[x]$ by:

$$
\begin{aligned}
\mathbb{E}[(x-\mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\
&= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2] \\
&= \mathbb{E}[x^2] - 2\mathbb{E}[x\mu] + \mathbb{E}[x]\mathbb{E}[x] \\
&= \mathbb{E}[x^2] - 2\mathbb{E}[x]\mathbb{E}[x] + \mathbb{E}[x]\mathbb{E}[x] \\
\mathbb{E}[(x-\mu)^2] &= \mathbb{E}[x^2] - E[x]E[x]
\end{aligned}
$$

Problem 6

(a) isProbability.m:

```
% indicate whether a given matrix P is a valid probability distribution
function valid = isProbability(P)
  if ~ismatrix(P)
      error('Input must be a matrix')
  end
  nonnegative = all(all(P >= 0));
  total = sum(sum(P, 1));
  normalized = (abs(total - 1) <= 0.0001);
  valid = nonnegative & normalized;
end
```

(b)

```
% check validity of some matrices
>> isProbability([0 1; 1 0])

ans =

     0

>> isProbability([0 -0.2; 0.7 0.5])

ans =

     0

>> isProbability([1 2 1; 3 0 1] / 8)

ans =

     1
```

(c) marginals.m:

```
% compute marginal distributions from a joint probability distribution
function [Px, Py] = marginals(P)
  if ~isProbability(P)
      error('Input must be a valid probability matrix')
  end
  Px = sum(P, 2);
  Py = sum(P, 1);
end
```

(d)

```
>> P = [0 1 2 1; 0 9 0 3] / 16;
>> [Px, Py] = marginals(P)

Px =

    0.2500
    0.7500


Py =

         0    0.6250    0.1250    0.2500
```

(e) conditionals.m:

```
% compute two conditional probability distributions from a joint
% probability distribution
function [Pxgy, Pygx] = conditionals(P)
  [Px, Py] = marginals(P);
  Pxgy = conditional(P, Py);
  Pygx = transpose(conditional(transpose(P), Px)); % will need to transpose P I think
end

function Pxgy = conditional(Pxy, Py)
  Pxgy = Pxy; % create matrix of same size as P

  % iterate over the rows of Pxy
  [nrows, ncols] = size(Pxy);
  % for each row, the corresponding row of Pxgy = Pxy[i] / Py
  for i = 1:nrows;
    for j = 1:ncols;
```

```
      if Py(j) == 0 % if any columns of y are zero
        Pxgy(i, j) = 1/nrows; % that col of Pxgy should be uniform
      else
        Pxgy(i, j) = Pxy(i, j) / Py(j);
      end
    end
  end
end
```

(f)

```
>> [Pxgy, Pygx] = conditionals(P)

Pxgy =

    0.5000    0.3333    0.7500
    0.5000    0.6667    0.2500


Pygx =

         0    0.4000    0.6000
         0    0.8000    0.2000
```

(g) bayes.m:

todo: check argument validity

```
function Pygx = bayes(Pxgy, Py)
  Pygx = Pxgy; % create Pygx with same dimensions as Pxgy

  [nrow, ncol] = size(Pxgy);
  for i = 1:nrow;
    for j = 1:ncol;
      numer = Pxgy(i, j) * Py(j);
      denom = Pxgy(i, :) * transpose(Py);
      if denom == 0
        Pygx(i, j) = 1 / ncol;
      else
        Pygx(i, j) = numer / denom;
      end
    end
  end
end
```

(h)

```
>> bayes(Pxgy, Py)

ans =

         0    0.4000    0.6000
         0    0.8000    0.2000
```

(i)

```
>> transpose(bayes(transpose(Pygx), transpose(Px)))

ans =

    0.5000    0.3333    0.7500
    0.5000    0.6667    0.2500
```

Problem 7

(a)

$$
\begin{aligned}
p(c) &= 58,299/1024^2 \\
&= 58,299/1,048,576 \\
&\approx 0.0556 \\
p(b) &= 1 - p(c) \\
&\approx 0.9444
\end{aligned}
$$

(b) See Figure 1 below.

(c)

```
pc = 58299 / (1024 ^ 2)
pb = 1 - pc

Pxgy = transpose([transpose(pxgb); transpose(pxgc)])
Py = [pb pc];
Pygx = bayes(Pxgy, Py)
```

See Figure 2 below.

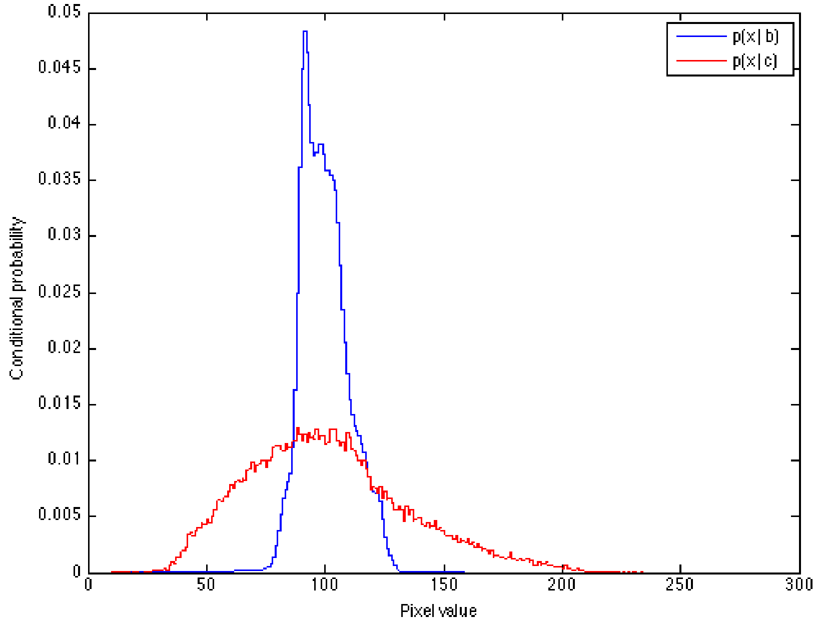

Figure 1: Conditional Probabilities

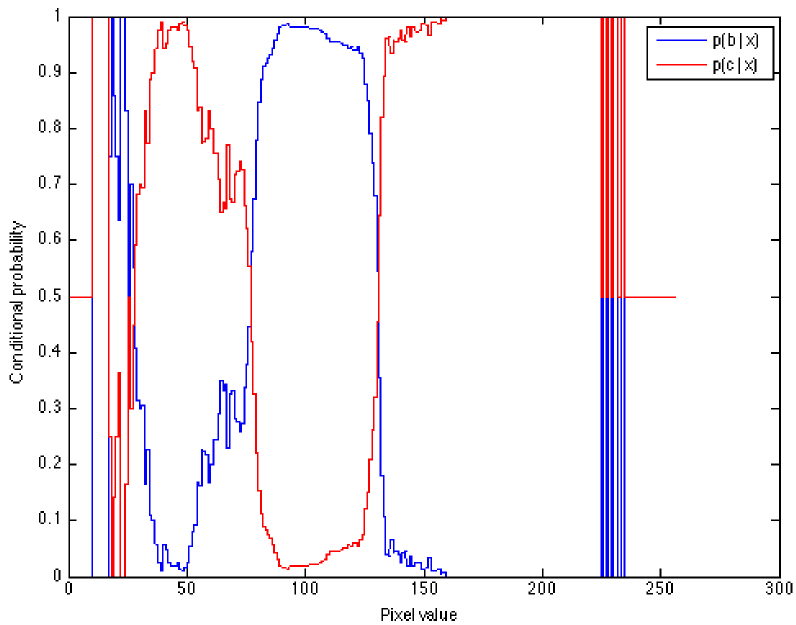

Figure 2: Posterior Probabilities

(d) For any given value of x, the only possibilities are $y = c$ or $y = b$, so $p(c|x) + p(b|x) = 1 \forall x$.

(e) The most natural value of τ in this scenario is $p(c|x) > 0.5$. With $\tau = 0.5$, we are classifying pixels as cells if there is a greater than 0.5 probability that they are cells. Since there are only two possibilities in this case (cell or background), this means pixels that are more likely to be cells than background are classified as cells.

(f) See Figure 3 below.

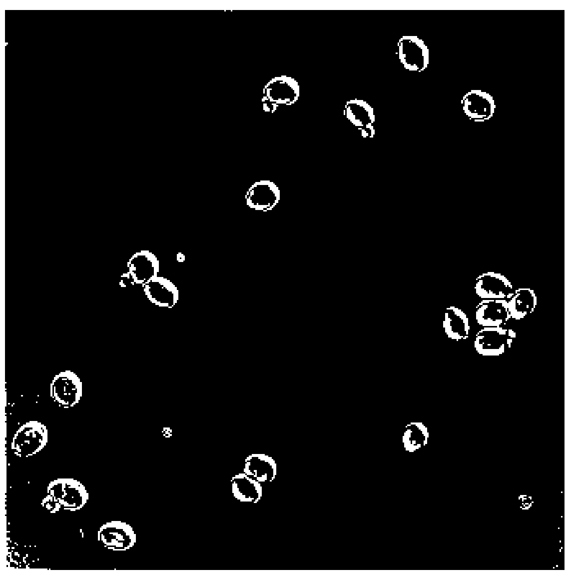

Figure 3: Bayes Classifier Results

(g) Comment on your result, which are inevitably far from perfect. In particular, what does the classifier flag as "cells"? Is this satisfactory? Is it a good start? Are there ways to get much better results based on individual pixel values alone? Why or why not?

(h) In what way is the Bayes classifier more expressive than the classifier based on a single threshold on pixel value x? Another way to ask this question is as follows: Describe the types of subsets of $X = \{0, . . . , 255\}$ that each classifier can yield in principle, as $p(c|x)$ is suitably changed.