1. Bayesian Statistics [13 points]  
(from Murphy, chapter 5)  
Let \( x \in \{0,1\} \) denote the results of a coin toss (\( x = 0 \) for tails, \( x=1 \) for heads). The coin is potentially biased, so that heads occur with probability \( \theta_1 \). Suppose that someone else observes the coin flip and reports to you the outcome, \( y \). But this person is unreliable and only reports the result correctly with probability \( \theta_2 \); i.e., \( p(y|x,\theta_2) \) is given by

\[
\begin{array}{c|cc}
  & y = 0 & y = 1 \\
  x = 0 & \theta_2 & 1 - \theta_2 \\
  x = 1 & 1 - \theta_2 & \theta_2 \\
\end{array}
\]

Assume that \( \theta_2 \) is independent of \( x \) and \( \theta_1 \).

(a) Write down the joint probability distribution \( p(y|x,\theta) \) as a 2 x 2 table, in terms of \( \theta=(\theta_1,\theta_2) \). [5 points]

(b) Suppose we have the following dataset: \( x = (1, 1, 0, 1, 1, 0, 0) \), \( y = (1, 0, 0, 0, 1, 0, 1) \). What are the MLEs for \( \theta_1 \) and \( \theta_2 \)? Justify your answer. Hint: note that the likelihood functions factorizes, \( p(x,y|\theta) = p(y|x,\theta_2) p(x|\theta_1) \). What is \( p(D|\hat{\theta}, M_2) \) where \( M_2 \) denotes this 2-parameter model? (You may leave your answer in fractional form). [8 points]

2. Manual calculation of Multinomial Naive Bayes [12 points]  
Consider the following training dataset for predicting the class of a document.

<table>
<thead>
<tr>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Russia Moscow</td>
<td>r</td>
</tr>
<tr>
<td>2</td>
<td>Washington U.S. Moscow</td>
<td>u</td>
</tr>
<tr>
<td>3</td>
<td>Russia St Petersburg</td>
<td>r</td>
</tr>
<tr>
<td>4</td>
<td>St Paul U.S.</td>
<td>u</td>
</tr>
<tr>
<td>5</td>
<td>U.S. Russia Syria</td>
<td>u</td>
</tr>
</tbody>
</table>
Using a multinomial Naive Bayes classifier, with Laplace (add-1) smoothing, what would be the predicted classes for the following entries? Show your work.

<table>
<thead>
<tr>
<th>Doc</th>
<th>Words</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moscow Moscow St Russia</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>U.S. St Petersburg</td>
<td>?</td>
</tr>
</tbody>
</table>

3. Perceptron and Kernels [15 points]
   (a) Implement the perceptron learning algorithm, as described in class, and run it on the dataset provided (percep1). Keep track of how many iterations you perform until convergence, as well as how many total updates (corresponding to mistakes) that occur through each iteration. After convergence, your code should output the raw weights, as well as the normalized weights corresponding to the linear classifier
   \[ w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 = 1 \]
   (You will create the normalized weights by dividing your perceptron weights \( w_1, w_2, w_3, \) and \( w_4 \) by \(-w_0, \) the weight corresponding to the special "offset" feature)
   Report the results of your perceptron learning algorithm as described above. [7 points]

   (b) In class, you have seen the primal formulation for perceptron, but there is also the dual form ([http://www.ccs.neu.edu/home/vip/teach/MLcourse/6_SVM_kernels/lecture_notes/kernels/KernelTrick.pdf](http://www.ccs.neu.edu/home/vip/teach/MLcourse/6_SVM_kernels/lecture_notes/kernels/KernelTrick.pdf)). Implement the dual version of perceptron, keeping in mind that you have to make it work with kernels. Apply it to the same dataset as in item a), using a linear kernel, in order to make sure your dual is equivalent to the primal (same solution). Then apply it to a linearly non-separable dataset (percep2) and check the results. Choose an appropriate kernel and report the results. [8 points]

4. SVM [30 points]
   In the next two problems, you will perform a digit recognition task using different algorithms. Given an image of a handwritten digit, you want to predict what number it represents. The problem has been simplified to a binary classification between two numbers that sometimes are hard to distinguish (3 vs 5).
   The digit images have been taken from a Kaggle competition ([http://www.kaggle.com/c/digit-recognizer/data](http://www.kaggle.com/c/digit-recognizer/data)) and are originally from the MNIST database of handwritten digits. They are in a format that is easy to read: each row represents an image; the first column is the label and the remaining columns give the grayscale values for each pixel. The version that you will be working on has been filtered to just those datapoints whose labels are 3 or 5 and it has been divided in training and test sets.

   Implement SVM with the SMO algorithm and train it on the modified digit dataset. For your implementation, you only have to use the linear kernel. Also run SVM from a package, try different kernels and compare the results. You can implement the simplified SMO, as described in [http://cs229.stanford.edu/materials/smo.pdf](http://cs229.stanford.edu/materials/smo.pdf)
5. Multi-Layer Perceptron [30 points]
Implement a multilayer perceptron, with one hidden layer, using the backpropagation algorithm and the sigmoid activation function. Run it on the modified digit dataset. Try different number of nodes in the hidden layer, like N=10, 20, 30, 50, 100, and see if there is any difference in the results.