1. (Human Learning, 10 points) Can you “learn” a classifier from the data? Please check the iris dataset (the data file and the description) and then devise a classifier with patterns you have “learned” from the data.

i [5 points] Write the classifier with the interface below and test its accuracy with the dataset. Please briefly but precisely report your method and the accuracy of your method on the dataset. The code below only illustrate the structure of the program, and it may not run.

```python
def classify(X, theta):
    """
    Predict labels of instances in X
    Input:
    X : a matrix of N instances and d features
    theta : the parameter to the classifier
    Output:
    a vector of N predicted labels
    """
    # replace the line below with your calculation
    # use 0, 1, or 2 to represent class labels.
    hat_y = predicted_label_vector
    return hat_y

def accuracy(X, y, theta):
    """
    Report the accuracy of the classifier classify()
    Input:
    X : a matrix of N instances and d features
    y : a vector of N true labels
    theta : the parameter to the classifier
    Output:
    a float, the accuracy of the classifier
    """
    y_pred = classify(X, theta)
    correct_pred = (y_pred == y).astype(np.float)
    accuracy = np.mean(correct_pred)
    return accuracy
```

ii [5 points] Is there an approach to code a classifier that achieves accuracy 1 on the iris dataset? If your answer is yes, describe your approach.

2. (Probability, 6 points) Suppose a classifier has been trained. Alice plan to test the accuracy of the classifier with a large test set. However, she accidentally flipped labels of a fraction $\alpha$ of test instances before she touch the classifier. Alice’s test indicates the accuracy of the classifier to be $\mu$.  

---

Due time: 09/24/2018 16:30pm
i [5 points] If we neglect the uncertainty of the test accuracy, what is the true accuracy of the classifier? (hint: what predictions are considered “correct” in Alice’s test? Note that the true accuracy of the classifier should be derived from true labels of the test set.)

ii [1 point] Calculate the true accuracy for \( \alpha = 0.1 \) and \( \mu = 0.8 \).

3. (Probability, 5 points) Show that the covariance of two independent variables is 0.

4. (Linear algebra, 8 points) Suppose an instance matrix is \( X \in \mathbb{R}^{n \times d} \) with each row as an instance and each column as a feature. Express the following values with matrices given using matrix operations, which includes 1) matrix calculations, e.g. matrix addition, subtraction, and multiplication; and 2) element-wise calculations, e.g. element-wise addition, subtraction, multiplication, and division. Please use \( \odot \) to represent element-wise multiplication. You can introduce a few new scalars or vectors if necessary.

i [2 points] The mean vector \( \mu \) of all instances.

ii [3 points] The matrix \( D \) of distances between all pairs of instances, that is, \( D_{ij} \) is the Euclidean distance between \( x_i \) and \( x_j \).

iii [3 points] Suppose we also have a membership matrix \( G \in \mathbb{R}^{n \times K} \) with each \( G_{ik} = 1 \) indicating the instance \( x_i \) belongs to class \( k \) or \( G_{ik} = 0 \) otherwise. Find a matrix \( R \in \mathbb{R}^{K \times d} \) such that the \( k \)-th row of \( R \) is the mean vector of these instances in class \( k \).

Note: you will have chances to use these calculations in this course. In python code, you can use \texttt{numpy.mean} and \texttt{sklearn.metrics.pairwise.pairwise_distances} for the first two calculations then.

5. (Optimal Decision, 10 points) In a binary classification problem, suppose the predictive distribution is \( p(y|x) \). Denote \( \mu_0 = p(y = 0|x) \) and \( \mu_1 = p(y = 1|x) \) for an instance \( x \). Consider the following loss matrix:

<table>
<thead>
<tr>
<th>true label</th>
<th>predicted as</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 10</td>
</tr>
<tr>
<td>1</td>
<td>5 0</td>
</tr>
</tbody>
</table>

i [5 points] What is the expected loss if we predict the class label \( \hat{y} \) to be 1 when \( \mu_1 \geq \mu_0 \) (equivalently \( \mu_1 \geq 0.5 \)) or \( \hat{y} = 0 \) otherwise?

ii [5 points] Can we reduce the expected loss? For example, if we use the following decision rule

\[
\hat{y} = \begin{cases} 
1 & \text{if } \mu_1 \geq \theta \\
0 & \text{if } \mu_1 < \theta 
\end{cases}
\]

What is the optimal \( \theta \)? Please keep in mind: \( \mu_0 \) and \( \mu_1 \) are not two fixed values but vary with different instances. You want to find a \( \theta \) that is optimal for all possible \( \mu_0 \) and \( \mu_1 \) values.