Homework 3, Due April 26

MPCS 53111 Machine Learning, University of Chicago

Practice problems, do not submit

1. Please solve the following questions by hand, showing your calculations clearly.

x_1	x_2	Class Label
1	1	0
2	2	0
3	4	0
2	4.5	1
3	6	1

   (a) Roughly plot the position of the examples along x_1 and x_2 and choose a simple decision boundary such that all the five training examples are correctly classified.

   (b) Compute by hand the parameters of a logistic regression model that correspond to the above decision boundary.

   (c) Determine the probability of the 0 class for the first and the last training examples according to the your logistic regression model.

Graded problems, submit

3. Run the perceptron learning rule, by hand, on the dataset in Problem 1 (go through the examples in the order shown), initializing $\mathbf{w}$ to $(0, 0, 1)$, and using a learning rate α of 0.5. Stop when the corresponding perceptron classifies all examples correctly. State clearly your final weights.

4. Please solve the following questions by hand, showing your calculations clearly.

x_1	x_2	Class Label
0.5	0.5	+
−0.5	0.5	+
−0.5	−0.5	+
2	2	−
2	−2	−
−2	2	−
−2	−2	−

(a) Roughly plot the position of the examples along x_1 and x_2 and choose a simple decision boundary such that all the seven training examples are correctly classified.

(b) Compute by hand the parameters of a logistic regression model that corresponds to the above decision boundary. (Hint: For features use x_1^2, x_2^2.)

(c) Determine the probability of the + class for the first and the last training examples according to the your logistic regression model.

5. Implement a python function `gradient_descent(X, y, alpha, T)` that takes an input `numpy` matrix `X` of shape (m,n), a output vector `y` of shape (m,1), a scalar learning rate `alpha`, the number of iterations `T`, and returns a vector `theta` of shape (n+1,1). `theta` should be the logistic regression parameter vector θ found by executing a gradient descent algorithm for `T` iterations on the given inputs. The function should also plot the value of the cost (loss) function $J(\theta)$ vs the iternation number. Use the cost function used in Andrew Ng's Notes.

   [You may find it useful to watch the two videos on "Gradient Descent in Practice" on Coursera. Enroll in the course for free to access the videos. Also while developing, it is best to test your function on a toy dataset with few examples and just a couple of features where you have worked out what should happen at each step and so can debug your code efficiently.]

6. Use the above implemenation for fitting a logistic regression model on the accompanying breast cancer data, with 30 input variables, and two classes. State the θ values you obtain, the value of the learning rate you used, and submit a plot of the cost function vs the iteration number for that learning rate.

7. Fit a logistic regression model on the above breast cancer data using `sklearn.linear_model.LogisticRegression` and state the θ values it returns. (If the θ here is very different from that in Problem 6, you may want to understand why, including debugging your gradient descent code.)

8. Fit a logistic regression model on the just the features `mradius` and `mtexture` using `sklearn.linear_model.LogisticRegresion`, but include terms up to degree 3. Draw a scatter plot in which the malignant cases are in red, the benign cases are in green, and the decision boundary corresponding to your model is in blue.

As usual please submit your homework as `.py` and `.txt`/`.pdf` files on SVN.