Homework 2

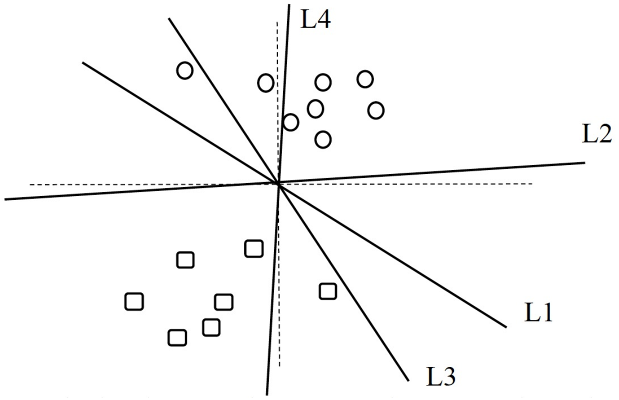
Due: See Canvas

There are two parts: individual problems and group problems. Each student should upload one submission for individual problem. Each group should upload one submission for group problems.

Individual Problem (15pt)

Problem 1. **(6pt)** Recall that a linear threshold function or a linear classifier is given by: If $(w_0 + \sum w_i x_i) > 0$ then class is positive, otherwise it is negative. Assume that 1 is true and 0 is false. Consider a function over n Binary features, defined as follows. If at least k variables are false, where $k \leq n$ is a constant, then the class is positive, otherwise the class is negative. Can you represent this function using a linear threshold function. If your answer is YES, then give a precise numerical setting of the weights. Otherwise, clearly explain, why this function cannot be represented using a linear threshold function.

Problem 2. **(9pt)** In this problem, we will refer to the binary classification task depicted in the figure given below.



Consider the following logistic regression (LR) model: $P(y = 1|x_1, x_2, w_1, w_2) = \frac{1}{1+\exp(w_1 x_1 + w_2 x_2)}$. Notice that the model is assuming that the bias term w_0 equals 0, namely the induced classifiers will pass through the origin. Let L1 be the solution (line) output by a gradient ascent algorithm using the maximum likelihood estimation (MLE) criteria. Consider a regularization approach where we try to maximize: $\sum_i log\{P(y_i|x_{i,1}, x_{i,2}, w_1, w_2)\} - \frac{C}{2}(w_1)^2$ for large C. Note that only w_1 is penalized. We'd like to know which of the lines in the figure above could arise as a result of such regularization. For each potential line L2, L3 or L4 determine whether it can result from regularizing w_1. If not, explain very briefly why not.

- L2 (answer Yes/NO and briefly explain why):
- L3 (answer Yes/NO and briefly explain why):
- L4 (answer Yes/NO and briefly explain why):

What to Turn in

- a pdf file with your answer

Group Problem (35pt)

In this homework you will implement and evaluate **Naive Bayes, Perceptron and Logistic Regression** for text classification. You can use either **Java or Python** to implement your algorithms.

- Download the spam/ham (ham is not spam) datasets available on Canvas. The datasets were used in the Metsis et al. paper [1]. There are three datasets. You have to perform the experiments described below on all three datasets. Each data set is divided into two (sub)sets: training set and test set. Each of them has two directories: spam and ham. All files in the spam folders are spam messages and all files in the ham folder are legitimate (non spam) messages.

- (10 pt) Implement the multinomial Naive Bayes algorithm for text classification described here: `https://nlp.stanford.edu/IR-book/pdf/13bayes.pdf` (see Figure 13.2) (have uploaded to Canvas). Note that the algorithm uses add-one Laplace smoothing. Make sure that you do all the calculations in log-scale to avoid under flow. Use your algorithm to learn from the training set and report accuracy on the test set.

- (10 pt) Implement the MCAP Logistic Regression algorithm with L2 regularization (see Sec 3.3 Mitchell's new book chapter)(have uploaded to Canvas). Try different values of λ. Divide the given training set into two sets using a 70/30 split (namely the first split has 70% of the examples and the second split has the remaining 30%). Learn parameters using the 70% split, treat the 30% data as validation data and use it to select a value for λ. Then, use the chosen value of λ to learn parameters from the full training set and report accuracy on the test set. Use gradient ascent for learning the weights (you have to set the learning rate appropriately. Otherwise, your algorithm may diverge or take a long time to converge). Do not run gradient ascent until convergence; you should put a suitable hard limit on the number of iterations.

- (10 pt) Implement the perceptron algorithm (use the perceptron training rule and not the gradient descent rule). Notice that unlike logistic regression which is a batch algorithm, the perceptron algorithm is an incremental or stochastic algorithm. Treat number of iterations in the perceptron algorithm as a hyper-parameter and use the 70-30 split method described earlier to choose a suitable value for this hyper-parameter. Then, use the chosen value of hyper-parameter, train on the full training dataset and report accuracy on the test set.

- (5 pt) Write a detailed write up that reports the accuracy obtained on the test set, and, parameters used (e.g., values of λ, hard limit on the number of iterations, etc.). We should be able to replicate your results based on your writeup.

What to Turn in

- Your **code and a Readme file** for compiling the code.

- A report.

Reference

[1] V. Metsis, I. Androutsopoulos and G. Paliouras, Spam Filtering with Naive Bayes - Which Naive Bayes?". Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006.