CS578/STAT590: Introduction Machine Learning	**Fall 2014**
Problem Set 2	
Handed Out: Sep 30	*Due: Oct 14*

Questions

1. Given n boolean variables $(x_1, .., x_n)$, we define our target classification function $f(\cdot)$ as $f(\cdot) = 1$ *if at least 3 variables are active*. For n=5 show how this function can be represented as (1) Boolean function (2) Linear function.

2. Let CON_B be the set of all different monotone conjunctions defined over n boolean variables. What is the size of CON_B?

3. Let CON_L be the set of all linear functions defined over n boolean variables that are consistent with the functions in CON_B . We say that two functions $f_b \in CON_B$ and $f_l \in CON_L$ are consistent if $\forall x, f_b(x) = f_l(x)$. What is the size of CON_L?

4. Define in one sentence: Mistake bound (your answer should include all the components described in class).

5. Suggest a mistake bound algorithm for learning Boolean conjunctions (*hint: recall the elimination algorithm for monotone conjunctions*). Show that your algorithm is a mistake bound algorithm for Boolean conjunctions.

6. Given a linearly separable dataset consisting of 1000 positive examples and 1000 negative examples, we train two linear classifier using the perceptron algorithm. We provide the first classifier with a sorted dataset in which all the positive examples appear first, and then the negative examples appear. The second classifier is trained by randomly selecting examples at each training iteration. (1) Will both classifiers converge? (2) what will be the training error of each one of the classifiers?

7. We showed in class that using the Boolean kernel function K(x,y)=$2^{same(x,y)}$ we can express all conjunctions over the attributes of x and y (with both the positive and negative literals). This representation can be too expressive. Suggest a kernel function K(x,y) that can express all conjunctions of size at most k (where k is a parameter, k $<$ n).

Programming Assignment

In this part of the assignment you are required to implement the perceptron algorithm and observe its performance in practice by running the algorithm over the credit card approval dataset.

1. **Feature Representation** Your implementation should consider three variations of the feature set as described below [1]
   (1) original attributes e.g., $\{(a = A_1), (b = B_1), (c = C_1)\}$
   (2) feature pairs e.g., $\{(a = A_1; b = B_1), (a = A_1; c = C_1), (b = B_1, c = C_1)\}$
   (3) use all the features in 1-2.

2. **Implementation** Implement the Perceptron algorithm by completing the function *Perceptron(maxIterations, featureSet)*. The first argument determines the number of training iterations the algorithm will perform over the dataset. The second argument determines which feature representation will be used. The argument takes integer values ({1,2,3}) corresponding to three feature representations described above.

3. **Experiments** In addition to the code implementation you should also submit a short report describing the results of your algorithm. For each feature representation, report the accuracy, precision, recall and F1 as described in class, over the training, validation and testing sets used in the previous assignment. Your report should include the value of *maxIterations* and an explanation of how was the value picked for each feature representation.

[1] Examples: given three attributes a,b,c each taking different values, we denote the feature corresponding to each value assignment using parentheses e.g., $(a = A_1)$, when the feature is defined over multiple attributes we use ";" to denote the pair of attributes e.g., $(a = A_1; b = B1)$