CS578/STAT590: Introduction Machine Learning	**Fall 2014**
Problem Set 3	
Handed Out: Nov 3	*Due: Nov 18*

Questions

1. Fitting an SVM classifier by hand (source: Machine Learning, a probabilistic perspective. K Murphy)

   Consider a dataset with 2 points in 1d: ($x_1 = 0$, y_1 = -1) and ($x_2 = \sqrt{2}$, y_2= 1). Consider mapping each point to 3d using the feature vector $\phi(x) = [1, \sqrt{2x}, x^2]$ (This is equivalent to using a second order polynomial kernel). The max margin classifier has the form:

   $$\begin{aligned} &\min ||w||^2 \quad \text{s.t.} && (1)\\ &y_1(w^T\phi(x_1) + w_0) \geq 1 && (2)\\ &y_2(w^T\phi(x_2) + w_0) \geq 1 && (3) \end{aligned}$$

   (a) Write down a vector that is parallel to the optimal vector w. Hint: recall that w is perpendicular to the decision boundary between the two points in the 3d feature space.

   (b) What is the value of the margin that is achieved by this w? Hint: recall that the margin is the distance from each support vector to the decision boundary. Hint 2: think about the geometry of 2 points in space, with a line separating one from the other.

   (c) Solve for w, using the fact the margin is equal to $1/||w||$.

   (d) Solve for w_0 using your value for w and Equations 1 to 3. Hint: the points will be on the decision boundary, so the inequalities will be tight.

   (e) Write down the form of the discriminant function $f(x) = w_0 + w^T\phi(x)$ as an explicit function of x

2. We define a concept space C that consists of the union of k disjoint intervals in a real line. A concept in C is represented therefore using 2k parameters: $a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k$. An example (a real number) is classified as positive by such concept iff it lies in one of the intervals. Give the VC dimension of H (and prove its correctness).

3. The Gradient Descent (GD) algorithm

   (a) Write in one sentence: what are the hyper parameters of the GD algorithm.

   (b) Write in one sentence: What is the difference between $l1$ and $l2$ regularization.

   (c) Write down the gradient descent algorithm applied to hinge loss with $l2$ regularization.

(d) The same as above, but using the $l1$ regularization. **Note:** the $l1$ norm is convex but not differentiable.

Programming Assignment

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

1. **Feature Representation** As in the previous assignment, 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 GD algorithm by completing the function *GD(maxIterations, regularization, stepSize,lambda, featureSet)*. The first argument determines the number of training iterations the algorithm will perform over the dataset. The *regularization* argument determines which regularizer the GD algorithm should use, possible values are $\{l1, l2\}$. The *stepSize* argument (η) determines the step size taken by GD algorithm. The argument *lambda* (λ) determines the impact of the regularizer compared to the training loss. The *featureSet* 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 values of hyper parameters and an explanation of how was the value picked for each feature representation. You should specifically address the impact your choice of regularizer has on each feature set.

[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., $(a = A_1; b = B1)$