# Problem Set 1

CS 6375

Due: 9/9/2015 by 11:59pm

Note: all answers should be accompanied by explanations for full credit. Late homeworks will not be accepted.

## Warm-Up: Subgradients & More (4 pts)

1. Compute a subgradient at the specified points for each of the following convex functions.

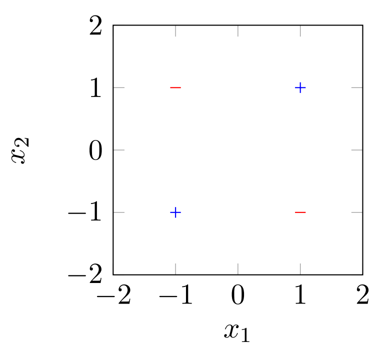
   (a) $f(x) = \max\{x^2, (x+1)^2\}$ at $x = 1$ and $x = 0.5$.

   (b) $g(x) = \max\{x^2, \exp(x), 10x\}$ at $x = -1$ and $x = 0.01$.

## Problem 1: Separability & Feature Vectors (25 pts)

1. Suppose we perform subgradient descent using the perceptron loss as described in class with $\gamma_t = 1$ for all $t$. Explain why the algorithm may not converge if the training data is not linearly separable.

2. Consider the following data set.




Under which of the following feature vectors is the data linearly separable? For full credit, you must justify your answer by either providing a linear separator or explaining why such a separator does not exist.

(a) $\phi(x_1, x_2) = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$ (c) $\phi(x_1, x_2) = \begin{bmatrix} x_1 x_2 \\ 1 \end{bmatrix}$

(b) $\phi(x_1, x_2) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ 1 \end{bmatrix}$ (d) $\phi(x_1, x_2) = \begin{bmatrix} x_1 \cdot \sin(x_2) \\ 1 \end{bmatrix}$

## Problem 2: Perceptron Learning (20 pts)

Consider the data set (perceptron.data) attached to this homework. This data file consists of $n$ data elements of the form $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}, y_i)$ where $x_1^{(i)}, \ldots, x_4^{(i)} \in \mathbb{R}$ define a data point in $\mathbb{R}^4$ and $y_i \in \{-1, 1\}$ is the corresponding label. In class, we saw how to use the perceptron algorithm to minimize the following objective.

$$\sum_{i=1}^{n} \max\{0, -y_i \cdot (w^T x^{(i)} + b)\}$$

In this problem, you are to implement the perceptron algorithm using two different gradient descent strategies (you can use any programming language that you would like - you don't need to turn in any code). For each strategy below, report the number of iterations that it takes to find a perfect classifier for the data, the values of $w$ and $b$ for the first three iterations, and the final weights and bias. Each descent procedure should start from the initial point

$$w^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad b^0 = 0$$

and use the step size $\gamma_t = 1$ for each iteration.

1. Standard gradient descent.
2. Stochastic gradient descent where exactly one component of the sum is chosen to approximate the gradient at each iteration. Instead of picking a random component at each iteration, you should iterate through the data set starting with the first element, then the second, and so on until the $n^{th}$ element, at which point you should start back at the beginning again.

## Problem 3: Support Vector Machines (25 pts)

In class, we saw how to formulate SVMs as a quadratic optimization problem that can be solved with standard algorithmic techniques (e.g., `quadprog()` in MATLAB). However, when all of the data points are support vectors, the optimization problem can often be solved by hand. For this problem, you should use the SVM formulation without slack variables.

| $x_1$ | $x_2$ | $x_3$ | y |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| -1 | -1 | -1 | 1 |
| 1 | 1 | -2 | -1 |

1. Find the linear SVM classifier, by hand, for the above data points under the assumption that each point is a support vector. Give a precise setting of the weights and bias. You must show your work to receive credit for this problem. Simply providing the weights and bias will not earn any points.

2. What is the size of the margin?

## Problem 4: Lagrange Multipliers & Duality (25 pts)

The method of Lagrange multipliers is quite useful in a number of applications. For some integer $n > 0$, consider the following constrained convex optimization problem.

$$\min_{p_1,\ldots,p_n} \sum_{i=1}^{n} p_i \log p_i$$

such that $p_1, \ldots, p_n \geq 0$ and $\sum_{i=1}^{n} p_i = 1$.

1. Using the method of Lagrange multipliers, construct a dual optimization problem for the above primal problem. Hint: you will not need to add Lagrange multipliers for the non-negativity constraints.

2. Solve the dual optimization problem.

3. In this case, strong duality holds. Use the solution to the dual optimization problem to find a solution to the primal problem.