University of Central Florida
CAP5415 - Computer Vision
Problem Set 2
Fall 2008
Assigned: Thursday, September 18, 2008
Due: Tuesday, September 30, 2008

Introduction

Each of the sections below constitute one problem. If the problem asks for images, you should turn in a print-out with the requested images. Ideally, your assignment should be composed in a word-processor, such as LaTeXor Microsoft Word. You are welcome to write out derivations by hand.

In your writeup, described the steps you completed for each problem and show the results. Readability will be part of your grade. For the questions that require you to code, please turn in the code.

If you are using Python instead of MATLAB, please contact me for Python versions of all of the files mentioned below.

Problem 1

Given a set of N training examples $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and labels $l_1, \ldots, l_N$ the log-likelihood of the data is

$$L = \sum_{i=1}^{N} \log\Big(1 + \exp\left(-l_i\left(\mathbf{x_i} \cdot \theta\right)\right)\Big)$$

Where θ is a vector of line parameters. Note that $\mathbf{x_i} \cdot \theta$ is the vector inner product (or dot-product) between two vectors.

In the files associated with this problem set, we have included data and labels. Complete the function `calculate_log_loss(pts, labels)`, using the function header that we have provided as a starting point.

To check your solution, we have gotten the following values in our implementation:

θ	L
$[0; 0; 1]$	81.32
$[1; 0; 1]$	48.15
$[1; 1; 1]$	28.60
$[1; 2; 2]$	36.46

Problem 2

If $f(x, y) = (x - 2y)^2 + (y - 60)^2$, compute the gradient vector of f.

Problem 3

Complete the framework code provided in `grad_desc_mod.m` to implement steepest descent optimization of the function from Problem 2. You should be able to achieve a value very close to zero. We have also included visualization code.

Problem 4

Using the vectors from Problem 1, we'll assume that the components of $\mathbf{x}$ are denoted using additional subscripts, or $\mathbf{x}_i = [x_{i,1} x_{i,2} x_{i,3}]$. Derive the gradient vector ∇L with respect to each entry in the θ vector, or in other words, derive the equation for:

$$\frac{\partial L}{\partial \theta} = \begin{bmatrix} \frac{\partial L}{\partial \theta_1} \\ \frac{\partial L}{\partial \theta_2} \\ \vdots \end{bmatrix}$$

Problem 5

Complete the code framework provided in `logistic_regression.m` to implement steepest descent optimization of the classifier parameters. Again, we have included a visualization. For reference, you should be able to achieve values of L less than $L = 1.6e1$.

To implement the gradient calculations, you should complete the function `calculate_gradient_log_loss.m`, which is included with the files for this problem set. We have also included the function `check_grad.m` which you can use to numerically check your gradient function.