Homework Four, for Thu 2/11 CSE 250B

Your homework must be typeset, and the PDF should be uploaded to Gradescope by midnight on the due date.

1. Suppose we have data $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^p \times \mathbb{R}$ and we wish to find a vector $w \in \mathbb{R}^p$ such that $y \approx w \cdot x$. This is a *regression problem.*

   A common loss function to use in this context is:

$$L(w) = \sum_{i=1}^{n} (y^{(i)} - w \cdot x^{(i)})^2.$$

   (a) Establish that $L(w)$ is a convex function of w.

   (b) Derive the gradient descent update rule.

   (c) Derive the Newton-Raphson update rule.

2. Show that the following functions $f : \mathbb{R}^p \to \mathbb{R}$ are convex.

   (a) $f(x) = x^T M x$, where $M \in \mathbb{R}^{p \times p}$ is positive semidefinite.

   (b) $f(x) = e^{u \cdot x}$, for some $u \in \mathbb{R}^p$.

   (c) $f(x) = \max(g(x), h(x))$, where g and h are convex.

3. *Logistic regression using gradient descent.* In this problem, you'll implement a logistic regression solver using the gradient descent equations we went over in class.

   (a) Write a function that takes as input:

   - a data matrix $x \in \mathbb{R}^{n \times p}$ (one point per row),
   - a vector of binary labels $y \in \{-1, 1\}^n$, and
   - an integer m,

   and returns a classification vector $w \in \mathbb{R}^p$ obtained by gradient descent on the logistic regression loss function

$$L(w) = \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})}),$$

   starting at $w = 0$ and running for at most m iterations. You will have to decide how to choose the step size on each iteration.

   The resulting classifier is *homogeneous*: that is, it goes through the origin. Recall that we allow for this simplification by assuming that the data has been augmented with an extra, constant-valued, feature.

   (b) Now run your algorithm on a toy data set consisting of the following six points in $\mathbb{R}^2$:

   - $(2, 1), (1, 20), (1, 5)$ with label -1
   - $(4, 1), (1, 40), (3, 30)$ with label 1

   (Remember that this will become three-dimensional once you add a constant feature.) How many iterations does it take to get a good decision boundary? Plot this boundary, and also show the locations of the six points. Separately, plot two or three of the other boundaries you get along the way. (Plotting the boundary corresponding to a vector $w \in \mathbb{R}^3$ is easy: just solve for the x_1- and x_2-intercepts and draw a line between these.)

   (c) This data set is extremely simple, but does have different scaling along the two features. Suppose you scale down the x_2-axis by a factor of 10, to make the two axes more similar. Now try logistic regression: is there a change in the number of iterations needed for convergence?

(d) Now create a 2-d data set of your own, with 100 points (50 per label) and overlapping classes. (Examples: you could generate these using bivariate Gaussians, or take a data set like 'wine' and pick out two features and two labels). Run your logistic regression solver and plot the points together with the final boundary.

A couple of things to be aware of:

- Getting a good solution might require more iterations than you would have expected!
- It is easy to run into overflow problems because of $\exp(z)$ terms. To handle this, just check whether z is large, and if so, do something different.

What to turn in:

- The plots for part (b), along with information about how many iterations each corresponds to.
- A few sentences for part (c).
- The plot for part (d), along with information about number of iterations.
- Anything else that you feel might be of interest.