Machine Learning *CS 726, Fall 2011*

HW05: Gradient descent and friends

Hand in at: `http://www.cs.utah.edu/~hal/handin.pl?course=cs726`. Remember that only PDF submissions are accepted. We encourage using LaTeX to produce your writeups. See `hw00.tex` for an example of how to do so. You can make a `.pdf` out of the `.tex` by running "`pdflatex hw00.tex`".

1. Show that logistic loss (Equations 6.5 on p87 of the book) is convex for a fixed value of $y \in \pm 1$ and *as a function of* $\hat{y}$. It's easiest (shortest, least cumbersome) to do in terms of derivatives, but you could also do it directly from the definition of convexity in terms of chords if you prefer.

2. Show that if $f(z)$ is convex in z, then $f(\boldsymbol{w} \cdot \boldsymbol{x})$ is convex in $\boldsymbol{w}$ for a fixed $\boldsymbol{x}$. Note: show it *directly*: simply saying that linear functions are convex and composition of convex functions is convex is *not* an acceptable answer. In particular, show it in terms of the chord definition of convexity. I've started the solution below to set up some notation that you're free to use or erase.

   Let $\boldsymbol{u}$ and $\boldsymbol{w}$ be given, and let $\beta \in [0, 1]$. Let $\boldsymbol{v} = \beta \boldsymbol{u} + (1 - \beta)\boldsymbol{w}$ (so that v is between u and w). Define $f_u = f(\boldsymbol{u} \cdot \boldsymbol{x})$ and similarly for f_v and f_w. We wish to show that $f_v \leq \beta f_u + (1 - \beta) f_v$.
   TODO: your part here

3. You might notice that Algorithm 23 (for subgradient descent on regularized hinge loss) looks a *lot* like Algorithm 5 (the original perceptron algorithm). In fact, the most substantial differences are in line 5, where HingeRegularizedGD compares $y(\boldsymbol{w} \cdot \boldsymbol{x} + b) \leq 1$ whereas Perceptron comparse $\cdots \leq 0$; and line 10 in which the weights are regularized. How would you have to change the hinge loss and change the regularizer to make these two differences go away? (Note even with these changes that Perceptron would make updates after *each* example, while HingeRegularizedGD doesn't update until after processing *all* examples, so they're not completely identical.)