MATH 462 ASSIGNMENT 3
VERSION October 19, 2023

ADAM M. OBERMAN

Instructions. You are also free to consult the textbook references or the internet, but be careful of conflicting notation and definitions. Submit your work on the MyCourses course page.

Math Solutions are preferred handwritten, but typed (good quality latex) is permitted. You can get help from other students, but you should do the write up yourself. Code solutions can be a PDF of a workbook, you can use any language.

3.1. **Losses.**

Exercise 3.1. Consider the normalized log loss,

$$\ell_{\log,2}(s,y) = \frac{1}{\log 2}\log(1+\exp(-ys))$$

(1) Plot it, (do one plot for each value of $y \in \pm 1$), and show in the plot that it is an upper bound for the zero-one loss.
(2) Show that it is balanced, normalized, monotone, and convex.

Exercise 3.2. Consider the exponential loss $\ell(s,y) = \exp(-ys)$.

(1) Plot it, (do one plot for each value of $y \in \pm 1$), and show in the plot that it is an upper bound for the zero-one loss.
(2) Show that it is balanced, normalized, monotone, and convex.

Exercise 3.3 (Multi-class losses).

(1) Let $s = (5.2, 3, .5)$, calculate the $K = 3$ class $\log$ loss when $y = 1$ and when $y = 2$.
(2) Let $s = (5.2, 3, .5)$, calculate the $K = 3$ class margin loss when $y = 1$ and when $y = 2$.
(3) Let ℓ be (a) the multiclass log loss and (b) the multiclass margin loss. In each case, plot the function $\ell(s(t), y = 3)$ for $s(t) = (1, 2, t)$ with $t \in [-4, 4]$.

3.2. **Gradient Descent and SGD.**

Exercise 3.4 (Convergence rates and log plots). We say an algorithm converges exponentially with rate $c < 1$ if the error $e(t)$ satisfies $\log e(t)/e(0) \le ct$. Consider the sequence $a(t) = 25(2/3)^t$.

(1) Show that it convergences exponentially and find the rate.
(2) Plot $a(t)$ a log-plot, so that the slope shows the rate of convergence. The x-axis should be the iteration count, and the y-axis should be the $\log$ of the error.
(3) Combine the previous plot with a log-plot for the sequences $(.99)^t$, $100(.99)^t$ and $.04^t$.

Exercise 3.5 (Gradient Descent and SGD Implementation). You may use the code provided as a starting point, or write your own.

`https://colab.research.google.com/drive/1-YoLDf3OyH3SxLJtC5W4qG3L1zYxkyMf?usp=sharing`

Consider the model problem, for $w \in \mathbb{R}$,

$$L(w) = \frac{1}{m}\sum_{i=1}^{m}\frac{(w-y_i)^2}{2}$$

for $w \in \mathbb{R}$, where $m = 500$ and y_i are uniformly generated over $[-1, 1]$.

(a) Run Gradient Descent on the model problem with learning rates $\alpha = .95, .75, .5, .25, .1$. In this case, you know the exact w^*. Plot the error, $e(w^t) = \|w^t - w^*\|^2$, on a log-plot, so that the slope shows the rate of convergence. The x-axis should be the iteration count, and the y-axis should be the $\log$ of the error, see sample below.
(b) Run the SGD algorithm, corresponding to example 2.3 in the notes. (Drawing balls with replacement). Consider a data set with $R = 10$ red ball and $B = 15$ blue balls, and let p_t be the estimate of the fraction of red balls. Do the update,

$$p_{t+1} = p_t - \frac{1}{t+1}(p_t - y_t)$$

where y_t is 1 if the ball is red, and zero otherwise. Plot the error, $e(t) = (p_t - p^*)^2$ as a function of t.

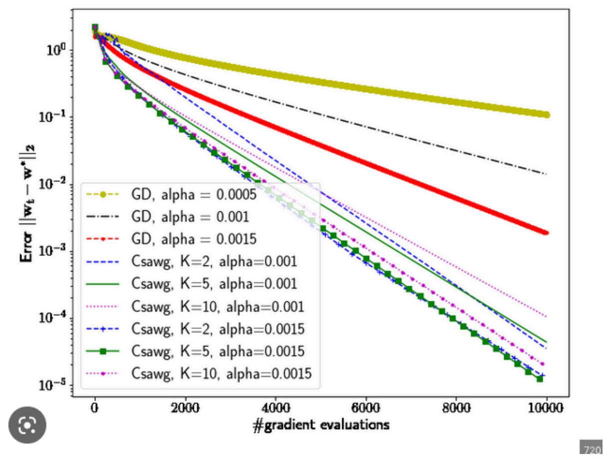

FIGURE 1. Sample convergence rate plot for gradient descen