Homework 4, CS 4220, Spring 2021
Instructor: Austin R. Benson
Due Friday, April 16, 2021 at 3:44pm ET on CMS (before lecture)

## Policies

**Submission.** Submit your write-up as a single PDF on CMS: https://cmsx.cs.cornell.edu.
**Coding questions.** You can use any programming language for the coding parts of the assignment. Include your code in your write-up.
**Typesetting.** Your write-up must be typeset with LaTeX.
**Collaboration.** Please discuss and collaborate on the homework, but you have to write your own solutions and code.
**Resources and attribution.** Feel free to use any resources that might be helpful, and provide attribution for any key ideas. We only ask that you work on the problems in earnest. Please do not hunt for solutions with a search engine.

## Problems

1. *Power laws.*

   Suppose that we have a bunch of data points $x_1, \ldots, x_n$, where each $x_i$ is a positive integer. We will represent this data as a vector $x \in \mathbb{R}^n$. We suspect that the data points were sampled i.i.d. from the *Zipfian distribution*:

   $$\Pr(x_i = k) = \frac{k^{-\alpha}}{Z}, \; k = x_{\min}, \ldots, x_{\max},$$

   where $Z = \sum_{k=x_{\min}}^{x_{\max}} k^{-\alpha}$ is a normalization constant, $\alpha > 0$ is an unknown parameter, and $x_{\min} < x_{\max}$ are the minimum and maximum values that the $x_i$ can take. In this problem, we will work out a way estimate a good value for $\alpha$ and apply this method on empirical data.

   (a) For now, assume that $x_{\min}$ and $x_{\max}$ are given constants. Suppose that the data were generated with some known $\alpha$, and consider the function

   $$L(\alpha; x) = \text{probability of observing the data } x \text{ under parameter } \alpha.$$

   Find an expression for $L(\alpha; x)$.

   (b) A reasonable way to estimate $\alpha$ is to find $\hat{\alpha} = \arg\max_\alpha L(\alpha; x)$. This is called maximum likelihood estimation and is a 1-dimensional optimization problem in our setting here. Describe some numerical issues we might run into if we tried to compute $\hat{\alpha}$ via a root-finding method with the function $L'(\alpha; x)$ (the derivative of $L$).

   (c) Let $N(\alpha; x) = -\log(L(\alpha; x))$ ($N$ is called the negative log likelihood function). Explain why a minimizer for $N$ is a maximizer for $L$.

   (d) By part (c), we could compute $\hat{\alpha} = \arg\min_\alpha N(\alpha; x)$. First, find expressions for $N(\alpha; x)$ and $N'(\alpha; x)$. Second, explain whether or not this formulation alleviates any of the numerical issues in part (b).

   (e) **Optional, ungraded question that can count towards class participation credit.** Using your expression for $N'(\alpha; x)$, find some sufficient conditions for when there is an interval $[a, b]$ with $0 < a < b < \infty$ that contains a local minimum of $N$, assuming that $N'$ is continuous.

(f) Download the data at https://github.com/arbenson/cs4220_2021sp/blob/main/tag-counts.txt. This is a dataset of the number of times a *tag*[1] has been used in a question on https://stackoverflow.com. Each line of the file has the question count and the name of the tag. Discard any tag that has appeared fewer than 10 times, and construct a dataset where $x_i$ is the question count of tag $i$.

We can set $x_{\min} = 10$. Describe how might you choose $x_{\max}$ for this dataset. Then estimate $\hat{\alpha}$ using an existing library such as `Optim.jl` (you can also feel free to implement your own algorithm instead). Report $\hat{\alpha}$ and plot your estimated probability distribution compared to the empirical probability distribution $p_e(k) = |\{j \mid x_j = k\}|/n$.

2. *BB steps.*

Recall the gradient descent update

$$x_{k+1} = x_k - \alpha_k g_k,$$

where $g_k = \nabla f(x_k)$ and $f \colon \mathbb{R}^n \to \mathbb{R}$. One way to choose $\alpha_k$ is the following:

$$\alpha_k = (\arg\min_{\alpha} \|(\alpha I)(x_k - x_{k-1}) - (g_k - g_{k-1})\|_2^2)^{-1}.$$

This is called the *Barzilai-Borwein method*, which is a type of quasi-Newton method that uses $\alpha_k I$ as a Hessian approximation.

(a) Derive a closed-form expression for $\alpha_k$.

(b) Show that if $n = 1$, this method is equivalent to applying the secant method to find a root of $f'$.

(c) Consider the special case where

$$f(x) = \frac{1}{2} x^T A x + x^T b + c, \tag{1}$$

where $A$ is symmetric positive definite. Show that in this case

$$\alpha_k = \frac{g_{k-1}^T g_{k-1}}{g_{k-1}^T A g_{k-1}}. \tag{2}$$

This differs from the optimal step size for gradient descent,

$$\lambda_k = \arg\min_{\lambda} f(x - \lambda_k g_k) = \frac{g_k^T g_k}{g_k^T A g_k}. \tag{3}$$

But $\alpha_k = \lambda_{k-1}$, which is interesting!

(d) Implement the gradient-based algorithms for the $f$ given in eq. (1) with step sizes from eqs. (2) and (3). Show the convergence of these two methods on a test problem, and briefly explain what you find.

[1] https://stackoverflow.com/help/tagging