Machine Learning and Optimization

Paul D. McNelis

Types of Optimization

- Issue: we need to select how we minimize a function
- This is our first choice or hyperparameter decision: how to we optimize a function
- We have several choices: global vs. local methods, gradient vs stochastic search methods
- We also have to come to terms with non-convex error functions, there may be several local minima or saddlepoints
- The search for better optimization methods for nonlinear functions predates Machine Learning
- But Machine Learning or Computational Learning has contributed to this research agenda.

Local Gradient Search

- The error fuction from estimation is a function of the parameters we are trying to obtain, as well as the data set:
- The estimation problem becomes a function estimation problem
- $\Phi = f(\Omega; x, y)$
- In general this function does not have a closed-form solution
- We have to iterate to find the optimal Ω^* starting with an initial guess, Ω_0
- We use a second-order Taylor expansion:
- $\Phi(\Omega_1) = \Phi(\Omega_0) + f'(\Omega_0)(\Omega_1 - \Omega_0) + \frac{1}{2}(\Omega_1 - \Omega_0)' f''(\Omega_0)(\Omega_1 - \Omega_0)$
- Optimization yields the following: $\Omega_1 = \Omega_0 - \frac{f'(\Omega_0)}{f''(\Omega_0)}$

Problems

- This goes back to Newton. Problem is the second derivative in the denominator
- If Ω is a large vector of coefficients, the second derivative or Hessian may be hard to invert
- So much of the work has been to find ways to approximate the inverse of the Hessian matrix
- Earlier work has been the BFGS method, but the popular one now is called ADAM.
- I will go over these methods in the Jupyther notebook.

Stochastic and Grid Search Methods

- Some methods are quite simple. One is Hill Climbing.
- Take initial guesses of the vector Ω_0 as well as lower and upper bounds.
- Define a new vector $\Omega_1 = \Omega_{LB} + Z_k[\Omega_{UB} - \Omega_{LB}]$, where Z_k is a k by 1 random vector
- Accept the new Ω_1 is better than the initial guess, in terms of lower Error metric.
- Contine for many interations
- Another is the Nelder-Mead Simplex Method. It starts with an initial guess and upper and lower bounds. So there are three vectors. Rank them from worst to best, Ω_i, i=1,2,3
- The method ranks the three vectors and takes an average of two best, Ω_c
- Then we find $\Omega_r = \Omega_c + \alpha[\Omega_c - \Omega_1]$. If Ω_r is best, expand simplex in this direction

SA,GA, and PS

- More elaborate methods are Simulated Annealing (SA), the Genetic Algorithm (GA) and Particle Swarm PS
- The idea is that we start with randomly chose candidate vectors for solution and combine then or mutate them in various ways.
- These are global search methods which can span a large surface of candidate solutions
- One practicle way to do optimization is to start with the global methods, and then its solution be the starting vector for a local search method and then a local gradient method.