Newton Direction in Descent Methods for Optimization
Project Proposal

Caleb Maddry

1 Project Overview

In this project, I will introduce root finding and the motivation for descent methods. I will discuss gradient descent and its limitations on difficult problems and introduce the Newton direction as tool to avoid these limitations. The introductory material will introduce and derive the Newton direction descent method. I will then carry out numerical experiments and implement the Newton descent variations as a tool to improve the cost of the method. The numerical experiments will look at convergence rates, cost, and error to compare the Newton direction and its variations to gradient descent. The extension of this project will investigate portfolio optimization problems using the Newton descent direction and its variations.

1.1 Introduction

Here, I will introduce root finding its applications to different problems. A brief look at gradient descent will be used to further establish the importance of descent methods and to motivate the Newton descent method.

1.2 Mathematical background and derivations

This will provide context for the descent direction, then derive the Newton direction. I will include graphs/images to complement the derivation. I will give a detailed description of the requirements for the method to converge. This will establish assumptions, conditions, and convergence rates.

1.3 Implementation and testing

I will discuss my implementation of the Newton direction and test it. After creating the descent algorithm, I will compare the algorithm's order of convergence with the expected convergence order from an analytical calculation. Testing will also include comparison cost and convergence rate to the gradient descent method.

1.4 Variations of Newton descent

Similarly to the Jacobian in Broyden's method, the Hessian matrix in the Newton direction can be expensive to compute. To bypass this, I will be implementing a lazy version that only computes the Hessian at the first step. I will compare the lazy method with the full implementation to see how it performs. Another way to bypass the expensive computation of the Hessian matrix is to implement the Broyden-Fletcher-Goldfarb-Shannon update. When compared to the original Newton descent step update, this should have better performance, and I will compare them to verify this.

1.5 Conclusions for introductory material

2 Independent Extension

2.1 Introduction

For my independent extension, I will be solving a portfolio optimization problem. This will use the Cvxportfolio python package: a portfolio specific extension of cvxpy.

2.2 Derivations and pseudocode for proposed scalar hybrid methods

Here, I will further explain Cvxportfolio and its implementation in my extension. This will also involve a description about how my implementation of the Newton direction and the package will work together (inputs/outputs/etc.).

2.3 Numerical experiments

After a specific problem has been selected, I will compare the different implementations of Newton descent and gradient descent. The numerical experiments will include analysis of convergence reliability, convergence rates, error, etc. These metrics will be compared

2.4 Potential extensions

A potential extension to this problem would involve a real-time optimization solver. As this is real-time (and would thus need to run fast), using the Newton descent variations, that enable faster computation of the Newton, would most be used here. A comparison to see if the Hessian can be computed fast enough for real time solving of the optimization problem would also be considered here.

2.5 Conclusions

3 Project Timeline

Date range	Proposed work
Before Nov 1	Create GitHub and Overleaf repos. Complete project proposal.
Nov 1	***Project Proposal due***
Nov 4 – 8	Discuss proposal with professor and make any necessary changes.
Nov 9 – 12	Begin writing the introduction and mathematical formulation.
Nov 12 – 14	Build implementation of Newton method.
Nov 14 – 16	Test Implementation and work on variations.
Nov 16 – 22	Finish writing the conclusion to introductory material and begin extension.
Nov 22	***Rough Draft due***
Nov 23 – 24	Discuss rough draft with professor and make any necessary changes.
Nov 24 – 30	Implement extension.
Nov 30 – Dec 1	Meet with professor to discuss extension.
Dec 1 – Dec 7	Finish implementation of extension and run numerical experiments.
Dec 7 – Dec 10	Create final presentation.
Dec 10 – Dec 17	Make and final edits before the final paper is due.
Dec 17	***Final Paper due***

References

[1] Stephen Boyd et al. "Multi–Period Trading via Convex Optimization". In: *Foundations and Trends in Optimization* 3.1 (Aug. 2017), pp. 1–76. URL: %5Curl%7Bhttps://stanford.edu/~boyd/papers/pdf/cvx_portfolio.pdf%7D.

[2] Enzo Busseti, Steven Diamond, and Stephen Boyd. *Cvxportfolio*. https://github.com/cvxgrp/cvxportfolio. Portfolio Optimization and Back–Testing. Jan. 2017.