Continuous Optimization 2025
Project Description

Projects can be done in teams of two people, or individually. A report written in Latex (see details below) has to be submitted through ADAM. The deadline for the report is **21 of July, 2025, 12:00 Basel time**.

List of Projects

You have to fill-in the following spreadsheet where you should **enter a ranked list of three projects** (ranked choice where 1 is your favorite project and 3 is your least favorite) from the list below:

https://docs.google.com/spreadsheets/d/1n9T1VA7C3au9ESEyssLTbpSEOlSkGwiwKddaR0awsBc/edit?usp=sharing

(make sure to fill-in both tabs, one for the team ID, and one for the ranked list).
We will then assign projects by trying to satisfy your ranked choice (our goal is to assign one project to a single team only). If not possible, we might resort to random assignments to break ties.

Stochastic Second-Order Methods

Newton's method is a popular optimization algorithm commonly used to solve optimization problems. It is a second-order optimization algorithm since it uses second-order information of the objective function. However, computing full Hessian might be infeasible in many applications. Therefore, there is an active field of analyzing stochastic second-order methods, i.e. when the full Hessian is not computed. This project aims to survey and analyze stochastic Newton-type methods in convex and non-convex regimes.

Objectives

- Comprehensively survey the existing literature on the convergence of Newton's method for convex and non-convex functions.
- Identify the possible sources of stochasticity, i.e., how to compute Hessian approximation cheaper than full Hessian.
- Identify the conditions under which Newton's method converges globally to an optimum for convex and non-convex functions.
- Identify the main differences in the results for convex and non-convex regimes: convergence rates, assumptions, global or local rates, and convergence measures.
- Implement 2-3 selected stochastic Second-order methods and apply them to two datasets.

References (non-exhaustive)

Kovalev, Dmitry, Konstantin Mishchenko, and Peter Richtárik. "Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates." arXiv preprint arXiv:1912.01597 (2019).

Agafonov, A., Kamzolov, D., Gasnikov, A., Antonakopoulos, K., Cevher, V., & Takáč, M. (2023). Advancing the lower bounds: An accelerated, stochastic, second-order method with optimal adaptation to inexactness. arXiv preprint arXiv:2309.01570.

Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In Doina Precup and Yee Whye Teh (eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70, pp. 1895–1904. PMLR, 5 2017.

Antonakopoulos, Kimon, Ali Kavis, and Volkan Cevher. "Extra-newton: A first approach to noise-adaptive accelerated second-order methods." Advances in Neural Information Processing Systems 35 (2022): 29859-29872.

Global convergence of Newton’s method

Newton's method is a popular optimization algorithm that is commonly used to solve optimization problems. It is a second-order optimization algorithm since it uses second-order information of the objective function. Newton's method is known to have fast local convergence guarantees for convex functions. However, the global convergence properties of Newton's method are still an active area of research. The purpose of this project is to survey and analyze various strategies to achieve global convergence.

Objectives

- To conduct a comprehensive survey of the existing literature on the global convergence properties of Newton's method for convex functions.
- To identify the conditions under which Newton's method converges globally to an optimum for convex functions.
- To evaluate various modifications and variations of Newton's method (including line search strategies, regularization techniques, and trust region method) on its global convergence properties.
- To implement selected strategies of Newton’s methods and apply them to two datasets.

References (non-exhaustive)

Hanzely, Slavomir, et al. "A Damped Newton Method Achieves Global O (1/k2) and Local Quadratic Convergence Rate." (2022).
Mishchenko, Konstantin. "Regularized Newton Method with Global $ O (1/k^ 2) $ Convergence." arXiv preprint arXiv:2112.02089 (2021).

Quasi-Newton methods

Quasi-Newton methods are a family of optimization algorithms that approximate the Hessian matrix of a function, without computing it directly. These methods are widely used in many fields, such as engineering, economics, and machine learning, to solve optimization problems efficiently. The purpose of this project is to survey and analyze various quasi-Newton methods.

Objectives

- To study the theory and background of quasi-Newton methods, including their historical development, mathematical foundations, and properties.
- To investigate the different types of quasi-Newton methods, such as BFGS, LBFGS, DFP, SR1, and other variants, and compare their advantages and disadvantages.
- To evaluate the performance of quasi-Newton methods by comparing them with other optimization algorithms, such as gradient descent, and Newton's method, (and potentially conjugate gradient) in terms of convergence rate, accuracy, and robustness.
- To implement selected quasi-Newton methods and apply them to two datasets.

References (non-exhaustive)

Xu, Chengxian, and Jianzhong Zhang. "A survey of quasi-Newton equations and quasi-Newton methods for optimization." Annals of Operations research 103 (2001): 213-234.
Byrd, Richard H., et al. "A stochastic quasi-Newton method for large-scale optimization." SIAM Journal on Optimization 26.2 (2016): 1008-1031.

Multi-objective optimization

Multi-objective optimization is a subfield of optimization that deals with the optimization of multiple (potentially conflicting) objectives simultaneously. The purpose of this project is to survey and analyze various multi-objective optimization methods and their applications.

Objectives

- To study the theory and background of multi-objective optimization, including its historical development, mathematical foundations, and properties.
- To investigate the different types of multi-objective optimization methods, such as evolutionary algorithms, scalarization methods, and constraint handling methods, and compare their advantages and disadvantages.
- To evaluate the performance of multi-objective optimization methods by comparing them with other optimization algorithms, such as single-objective optimization, and assessing their convergence and diversity, in terms of various metrics (including Pareto optimality).
- To implement selected multi-objective optimization methods and apply them to two datasets.

References (non-exhaustive)

Fliege, Jörg, A. Ismael F. Vaz, and Luís Nunes Vicente. "Complexity of gradient descent for multiobjective optimization." Optimization Methods and Software 34.5 (2019): 949-959.
Emmerich, Michael TM, and André H. Deutz. "A tutorial on multiobjective optimization: fundamentals and evolutionary methods." Natural computing 17 (2018): 585-609.

Constrained optimization

Constrained optimization is a subfield of optimization that deals with optimizing an objective function subject to constraints. The purpose of this project is to survey and analyze various constrained optimization methods and their applications.

Objectives:

- To study the theory and background of constrained optimization, including its historical development, mathematical foundations, and properties.
- To investigate the different types of constrained optimization methods, such as penalty function methods, Lagrangian methods, and Frank-Wolfe, and compare their advantages and disadvantages.
- To evaluate the performance of constrained optimization methods by comparing them with other optimization algorithms, such as unconstrained optimization, and assessing their convergence and feasibility.
- To implement selected constrained optimization methods and apply them to two datasets.

References (non-exhaustive)

Wright, Stephen, and Jorge Nocedal. "Numerical optimization." Springer Science 35.67-68 (1999). Chapter 17.
Jaggi, Martin. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization." International conference on machine learning. PMLR, 2013.
Freund, Robert M., and Paul Grigas. "New analysis and results for the Frank–Wolfe method." Mathematical Programming 155.1-2 (2016): 199-230.

Sketching methods

Sketching optimization is a subfield of optimization that deals with solving large-scale optimization problems efficiently by approximating the objective function and constraints using sketching techniques. The purpose of this project is to survey and analyze various sketching optimization methods and their applications.

Objectives

- To study the theory and background of sketching optimization, including its historical development, mathematical foundations, and properties.
- To investigate the different types of sketching optimization methods, such as sketching-based Newton methods, subspace Newton methods, and compare their advantages and disadvantages.
- To evaluate the performance of sketching optimization methods by comparing them with other optimization algorithms, such as gradient descent and Newton's method, and assessing their convergence and accuracy.
- To implement selected sketching optimization methods and apply them to two datasets.

References (non-exhaustive)

Pilanci, Mert, and Martin J. Wainwright. "Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares." The Journal of Machine Learning Research 17.1 (2016): 1842-1879.
Pilanci, Mert, and Martin J. Wainwright. "Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence." SIAM Journal on Optimization 27.1 (2017): 205-245.
Gower, Robert, et al. "Rsn: Randomized subspace newton." Advances in Neural Information Processing Systems 32 (2019).

Variance-reduced methods

Variance-reduced methods are a class of optimization algorithms that aim to reduce the variance of the stochastic gradient estimate in order to improve the convergence rate and efficiency of stochastic optimization. The purpose of this project is to survey and analyze various variance-reduced methods and their applications.

Objectives

- To study the theory and background of variance-reduced methods, including their historical development, mathematical foundations, and properties.
- To investigate the different types of variance-reduced methods, such as SVRG (Stochastic Variance Reduced Gradient), SAG (Stochastic Average Gradient), SAGA, and compare their advantages and disadvantages.
- To evaluate the performance of variance-reduced methods by comparing them with other optimization algorithms, such as standard stochastic gradient descent and batch gradient descent, and assessing their convergence and accuracy.
- To implement selected variance-reduced methods and apply them to two datasets.

References (non-exhaustive)

Johnson, Rie, and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction." Advances in neural information processing systems 26 (2013).
Schmidt, Mark, Nicolas Le Roux, and Francis Bach. "Minimizing finite sums with the stochastic average gradient." Mathematical Programming 162 (2017): 83-112.
Defazio, Aaron, Francis Bach, and Simon Lacoste-Julien. "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives." Advances in neural information processing systems 27 (2014).

Stochastic Optimization

Stochastic optimization methods are increasingly popular due to their ability to efficiently optimize models with large datasets. Stochastic gradient descent (SGD) is a widely used stochastic optimization algorithm. However, the convergence of SGD is often slow, especially for high-dimensional problems with complex objective functions. To address this challenge, researchers have developed various stochastic optimization methods, including first and second-order methods, which can significantly improve the convergence rate.

Objectives

- To survey stochastic first and second-order optimization methods and analyze their effectiveness.
- To investigate the theoretical foundations of both first and second-order stochastic optimization methods, and compare their advantages and disadvantages.
- To evaluate the performance of these methods by assessing their convergence and accuracy, also depending on the choice of hyper-parameters (e.g. batch-size).
- To implement selected stochastic optimization methods and apply them to two datasets.

References (non-exhaustive)

Byrd, Richard H., et al. "A stochastic quasi-Newton method for large-scale optimization." SIAM Journal on Optimization 26.2 (2016): 1008-1031.

Cubic regularization

Cubic regularization is a class of optimization algorithms that aim to improve the convergence rate and efficiency of non-linear optimization problems. The purpose of this project is to survey and analyze various cubic regularization methods and their applications.

Objectives

- To study the theory and background of cubic regularization methods, including their historical development, mathematical foundations, and properties.
- To investigate the different types of cubic regularization methods, such as Cubic Regularization Newton Method, Adaptive Cubic Regularization, and compare their advantages and disadvantages.
- To evaluate the performance of cubic regularization methods by comparing them with other optimization algorithms, such as gradient descent, conjugate gradient method, and quasi-Newton methods, and assessing their convergence and accuracy.
- To implement selected cubic regularization methods and apply them to two datasets.

References (non-exhaustive)

Nesterov, Yurii, and Boris T. Polyak. "Cubic regularization of Newton method and its global performance." Mathematical Programming 108.1 (2006): 177-205.
Cartis, Coralia, Nicholas IM Gould, and Philippe L. Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results." Mathematical Programming 127.2 (2011): 245-295.
Cartis, Coralia, Nicholas IM Gould, and Philippe L. Toint. "Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity." Mathematical programming 130.2 (2011): 295-319.

Adaptive Algorithms

Adaptive algorithms are a class of optimization methods that aim to improve the efficiency of gradient-type methods by changing the step size adaptively. The adaptive stepsize is typically set

based on the observed trajectory (e.g., gradient norm or distance). The purpose of this project is to survey and analyze various adaptive methods, their differences, and applications.

Objectives

- To study the theory and background of cubic regularization methods, including their historical development, mathematical foundations, and properties.
- To investigate the different types of adaptive methods, such as Adam, Adagrad, Adagrad-norm, RMSprop, and compare their advantages and disadvantages.
- To evaluate the performance of adaptive methods by comparing them with other optimization algorithms, such as (stochastic) gradient descent and quasi-Newton methods, and assessing their convergence and accuracy.
- To implement selected adaptive methods and apply them to two datasets.

References (non-exhaustive)

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
Ward, Rachel, Xiaoxia Wu, and Leon Bottou. "Adagrad stepsizes: Sharp convergence over nonconvex landscapes." Journal of Machine Learning Research 21.219 (2020): 1-30.
Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747 (2016).

Coordinate Descent

Coordinate Descent (CD) is a class of optimization methods in which only a subset of parameters is updated at each iteration. The purpose of this project is to survey and analyze various coordinate sampling schemes and their theoretical and practical difference.

Objectives

- To study the theory and background of coordinate-descent-type methods, including their historical development, mathematical foundations, and properties.
- To investigate the different types of coordinate sampling, how to evaluate coordinates' importance, and compare their advantages and disadvantages against full sampling (i.e., gradient descent).
- To evaluate the performance of coordinate-descent-type methods by comparing them with other optimization algorithms, such as gradient descent and accelerated gradient descent, and assessing their convergence and accuracy.
- To implement selected CD methods and apply them to two datasets.

References (non-exhaustive)

Qu, Zheng, and Peter Richtárik. "Coordinate descent with arbitrary sampling I: Algorithms and complexity." Optimization Methods and Software 31.5 (2016): 829-857.
Richtárik, Peter, and Martin Takáč. "On optimal probabilities in stochastic coordinate descent methods." arXiv preprint arXiv:1310.3438 (2013).
Yu. Nesterov. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems." SIAM Journal on Optimization.

Experiments

Datasets
You should run experiments on at least two of the following four datasets:

- Covtype: This is a dataset containing information about forest cover types. It is often used for classification problems.
- A9a: This is a dataset that is commonly used binary classification and that contains information about census records of people from the United States.
- IJCNN1: This is a benchmark dataset used in the field of machine learning and neural networks.
- MNIST: This is a dataset of handwritten digits that is often used for classification problems. Note this is a multi-class problem.

These datasets can be downloaded at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Loss function
You should run your experiments using the following two losses:

- Logistic regression loss (cross-entropy loss) is a popular classification algorithm used to model the probability of an outcome. The logistic regression loss measures the difference between the predicted probability and the actual label. For binary classification problems, the formula for logistic regression loss is:
  $L(y, f(x)) = -[y * \log(f(x)) + (1 - y) * \log(1 - f(x))]$
  where y is the true label (0 or 1), f(x) is the predicted probability of the positive class, and log is the natural logarithm. The model f(x) should be a linear model, i.e. $f(x) = \langle w, x \rangle$ where w are the weights of the model to be optimized.
- Logistic regression with the non-convex regularizer r(x) defined in https://arxiv.org/pdf/1603.06159.pdf, page 9.

Methodology

The methodology of this project includes the following steps:

Literature review: Collecting and reviewing the relevant literature including research papers, books, and online resources.

Theoretical analysis: Studying the mathematical foundations and properties of the relevant optimization methods.

Comparative analysis: Comparing different types of optimization methods in terms of their computational complexity, memory requirements, and convergence properties.

Experimental evaluation: Implementing selected optimization methods using Python and applying them to two datasets (among the ones listed above). The performance of the methods will be evaluated by comparing them with other optimization algorithms and assessing their convergence. Among other things, you should produce figures that show the rate of convergence as a function of the number of iterations. Since the behavior of most optimization algorithms depends on random factors (such as initialization), you should average your results over several runs (3 to 5 runs).

Expected outcome

Report
Each group writes a report using the following latex template:
https://neurips.cc/Conferences/2022/PaperInformation/StyleFiles

The report must not exceed 8 pages excluding references. An additional appendix containing proofs or experiments can be included as well.

The report should contain an introduction section that explains the aim of the report and what is covered in the report. It should then contain various sections that cover the material described in the methodology section.

Reproducibility and source code
Reproducibility is important in science as it allows for the independent verification and validation of research findings. We will therefore ask that you make sure that your results can be reproduced. To do so, we ask that you deliver a python notebook that can easily be run and produce the figures presented in your report. These figures should be numbered and documented appropriately.