Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 4707-4717
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.482

Design And Optimization Of Optimization Algorithms For Convex Programming Problems

Akshay Kumar Asst. Professor, Department of Mathematics, Graphic Era Hill University, Dehradun Uttarakhand India.

Abstract

Convex programming issues are fundamental to many fields of science and engineering, from signal processing and control systems to machine learning and data analysis. Due to its extensive applicability and influence, the effective resolution of these problems is of paramount importance. In this article, we give a thorough investigation into the creation and improvement of optimisation algorithms specifically suited for convex programming issues. We first explore the theoretical underpinnings of convex optimisation and talk about the characteristics that make these issues accessible to effective methods for solution. We emphasise the significance of convexity, duality, and optimality conditions that direct the creation of successful optimisation techniques. The design concepts and optimisation strategies for first-order methods, which depend on gradient information, are next examined. We discuss more complex techniques including accelerated gradient methods and proximal algorithms, as well as traditional algorithms like gradient descent and its variations. We go over their computational challenges, convergence characteristics, and accuracy vs. speed trade-offs. On benchmark convex programming problems, we give numerical tests and comparative analyses to assess how well the suggested optimisation strategies perform. We go over their advantages and disadvantages as well as the consequences of our research for practical use. This paper offers a thorough examination of the design and optimisation of algorithms specifically suited for resolving these issues, which advances the subject of convex programming as a whole. For practitioners and scholars working on convex optimisation, the offered methodologies provide insights and guidelines, supporting the effective resolution of challenging issues across a variety of disciplines.

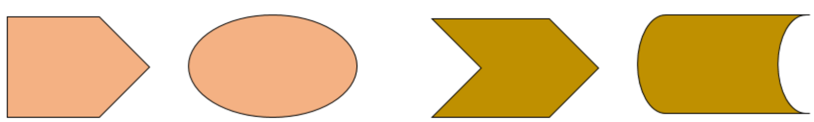
Keywords: Convex Programming, Proximal Algorithm, Gradient descend method, complex technique.

I. Introduction

Due to its vast applicability and manageable solution qualities, convex programming problems have attracted significant attention in the domains of mathematics, computer science, and engineering. These [3] issues can be solved by effective algorithms by optimising a convex objective function across a convex collection of viable solutions. In order to create effective and scalable approaches to handle these challenges in practise, convex programming problems have been the subject of substantial study on algorithm design and optimisation. Convex programming problems need the design of efficient optimisation algorithms, which is essential for effectively completing real-world optimisation tasks. These algorithms seek to use the structure and characteristics of the problem to efficiently and accurately discover the best solution. The choice of an optimisation technique is influenced by a number of variables, including as the size of the issue, its sparsity, how smooth the objective function is, and the computer resources that are available.

Convex functions serve as both the objective function and the constraints in convex optimisation issues. When minimising or maximising in these issues, the objective function is convex, and the constraints are also convex functions. Since linear functions are convex, linear programming issues are a particular type of convex optimisation problems.

Convex [2] optimisation problems also include conic optimisation problems, which are an extension of linear programming. Conic sets like second-order cones or semidefinite cones, which define convex constraints, are involved in these issues. Convex optimisation problems are distinguished by the feasible region, which is a convex region formed by the intersection of convex constraint functions. As seen in the image below, this indicates that any two places within the feasible zone can be connected by a straight line that completely encircles the area shown in figure 1 (a).



(a) (b)

Figure 1: (a) Convex region (b) Non-Convex region

Situations where either the objective function or any of the constraints are non-convex fall under the category of non-convex optimisation problems. Non-convex optimisation issues are not guaranteed to have a single globally optimal solution, in contrast to convex optimisation problems. The non-convexity can make it more difficult and challenging to come up with satisfactory solutions. The objective function or constraint functions in a non-convex optimisation problem may have characteristics like local optima, saddle points, or discontinuities. As a result, as seen in the figure below, the viable zone created by the intersection of the constraint functions might have irregular forms and several disconnected sections.

II. Review of Literature

There is a wealth of literature on the design and optimisation of algorithms for convex programming issues. A [14] number of academics have significantly advanced this field by creating algorithms that have higher convergence rates and are more computationally effective. We give a quick summary of a few significant papers that have helped progress optimisation techniques for convex programming issues in this section.

A thorough textbook on convex optimisation [4] covers many facets of convex programming and optimisation methods. Convex sets, convex functions, and convex optimisation issues are introduced throughout the book as a foundation for understanding the design and analysis of optimisation methods.

An [5] accelerated gradient technique that yields the best convergence rates for smooth convex functions is called Nesterov's accelerated gradient (NAG). The faster optimization algorithms that have since been developed as a result of the NAG method's widespread adoption in the optimisation field.

The [6] idea of "online convex optimisation," where judgements must be made without knowledge of future data and the objective function is revealed sequentially. The author addresses the regret bounds of online gradient descent algorithms, which measure how well they perform in comparison to an offline optimal solution. Convex programming issues in dynamic or streaming contexts may be affected by this research [1].

The proximity operator and gradient-based updates are combined in the proximal gradient method, sometimes referred to as the proximal algorithm, to handle non-smooth convex functions. In many fields, like machine learning and signal processing, where sparsity-inducing regularisation is common, the proximal gradient method has become increasingly prominent.

Scalable and distributed methods have also been the focus of recent breakthroughs in optimisation algorithms, [6] for example, describe a distributed subgradient method for resolving convex optimisation issues in networked systems, where several agents cooperate to resolve a global optimisation problem by exchanging knowledge with little communication.

Even though [7] the aforementioned studies have significantly improved the design and development of optimisation algorithms for convex programming issues, much more work has to be done in this area. Using this corpus of previous work as a foundation, we offer new methods in this study that tackle the problems brought on by massive data sets, intricate constraints, and various objective functions.

III. Conditions for Convex Programming Problems

In order to address the complexity that non-convexity introduces, it is frequently necessary to use specialised algorithms and heuristics while solving non-convex optimisation issues. Local search, evolutionary algorithms, simulated annealing, and other methods of global optimisation may be used in these procedures. In order to find interesting regions that might hold optimal or nearly optimal solutions, these algorithms seek to fully explore the solution space.

It is [16] significant to remember that non-convex optimisation tasks typically require more processing power than their convex equivalents. In the case of non-convexity, the search for global optimality or high-quality solutions may necessitate a deeper study of the problem space, perhaps requiring more computational time and resources.

A. Solving Convex Optimization Problems

There are [5] numerous ways to solve convex optimisation issues, however Interior Point or Barrier approaches are particularly effective in this context. Even for linear programming (LP) problems, these techniques handle linear, quadratic, conic, and smooth nonlinear functions uniformly by using a smooth convex nonlinear barrier function to express the constraints.

Even for large-scale situations, interior point approaches have made addressing convex issues feasible. Particularly for second-order issues involving quadratic and second-order cone programming (SOCP) functions, when the Hessians of the problem functions are constant, they demonstrate outstanding performance. Theoretical study and actual experience show that Interior Point techniques converge to an ideal solution in a very short period of time (usually less than 50 iterations). Convex problems can be successfully solved using Frontline Systems' Solver technology, which can handle a variety of functions, including linear, quadratic, conic, and nonlinear functions. The Frontline Systems Solver products offer a wide variety of techniques, allowing users to select the best strategy based on the features and needs of the problem [2].

B. Types of Problem

Depending on [2] the convexity of the objective function and the constraints, optimisation problems can range in complexity. The problem becomes easier to solve and offers various benefits in terms of feasibility assessment, global optimality, and scalability when all associated functions are convex. However, the problem becomes substantially more difficult and there is less confidence in its feasibility, optimality, and scalability when non-convex functions are included.

The sorts of optimisation problems are listed below, with the degree of difficulty of the solutions increasing:

- Problems of Convex Optimisation
  - Constraints and the objective function are all convex.
  - Feasibility: It is trustworthy to determine feasibility.
  - Globally Optimal Solution: It is confidently possible to locate the globally optimal solution.
  - Scalability: Even for very high sizes, these challenges can be effectively solved.

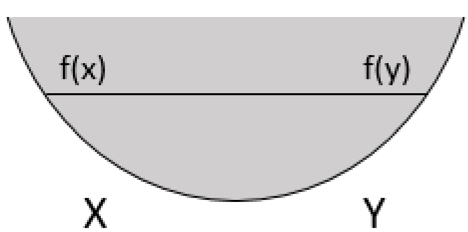
In terms of geometry, a function is convex if the chord from x to y, which is a line segment connecting any two points (x, f(x)) and (y, f(y)), falls on or above the graph of f.

According to algebra [9], a function f is said to be convex if it satisfies the inequality: f(tx + (1-t)y) t f(x) + (1-t) f(y) for any given points x and y and any value t between 0 and 1. According to this inequality, the value of f at the convex combination of x and y is lower or equal to the value of f at the convex combination of x and y.

A function [10] is also said to be concave if its negation, -f, is convex. In other words, the inequality -f(tx + (1-t)y) t(-f(x)) + (1-t)(-f(y)) holds for any x, y, and t between 0 and 1. Geometrically speaking, this indicates that the chord spanning x and y is located on or below the concave function f's graph.

- Problems with Non-Convex Optimisation
  - Non-convex constraints or the objective function both exist.
  - Feasibility: Deciding whether something is feasible gets more difficult and unpredictable.
  - Globally Optimal Solution: It is not certain that the globally optimal solution will exist or be unique. The global optimum becomes challenging to find.
  - Scalability: As a problem's size grows, it may become computationally challenging to solve non-convex problems.

Geometrically, a function is said to be convex if the chord from x to y, a line segment connecting any two points (x, f(x)) and (y, f(y)), wholly resides on or above the graph of the function, as seen in the image below.



By proving that a problem is convex, we arrive to an important conclusion: Any solution that satisfies the necessary conditions will automatically satisfy the sufficient conditions for optimality[18]. A global minimum will also be provided by such a solution. We use the technique taking into account various instances depending on the switching criteria stated in order to determine the best design. When a solution satisfies the prerequisites, we can call an end to the search because this solution reflects the overall idealised design[12].

Conditions that are Necessary for Convex Programming Issues:

The first-order Karush-Kuhn-Tucker (KKT) criteria for a convex cost function f(x) defined on a convex feasible set are both sufficient and necessary to find a global minimum.

For minimizing

$f(x) = (2x1 - 1.5)2 + (2x2 - 1.5)2$

Subject to

$$f(x) = x1 + x2 - 2 \leq 0$$

Solution to above condition as given below:

The cost function's Hessian matrix can be written as follows:

$$H = [\frac{\partial x12}{\partial 2f} \quad \frac{\partial x2 \partial x1}{\partial 2f \partial x1 \partial x2} \; \frac{\partial 2f}{\partial x22} \partial 2f]$$

The entire Hessian matrix H is positive definite according condition states that the cost function f(x) is strictly convex as a result. As a result, the issue can be categorized as convex. The solutions 1 = 1 x 1 =1 and 2 = 1 x 2 =1, which meet the sufficiency condition stated in Theorem 4.11, serve as a stringent global minimum point for the issue.

IV. Optimization Problem

The constraints in the problem under consideration are linear equality constraints of the type Ax = b. In this case, the positive orthant in Rm and matrix A both belong to the set of positive semi definite matrices in ℝ^(m×n). Additionally, the variables are subject to nonnegativity requirements, xi > 0. Discuss changes to our strategy that make it possible to handle limitations imposed by linear inequality. However, in this particular scenario, we assume that the matrix A comprises rows that are linearly independent and that the number of constraints m is fewer than or equal to the number of variables n.

The i^{th} column of the matrix A is shown by the notation a_i = [A1i, ..., Ami]^T for each i = 1,..., n. The vectors b and a_i are given to node i in this distributed configuration. It cannot, however, access the remaining columns of matrix A.

The following convex optimisation issue is what we refer to as the "primal problem":

Minimize f(x)

Subject to Ax =b,

X_i => 0, i= 1, 2, ...

The Lagrangian function L(x,, v) is related to the primary issue (P) and is defined as:

$$L(x,\lambda,\nu) = f(x) + \lambda T(Ax - b) - \nu Tx$$

where x stands for the fundamental variables, Rm stands for the Lagrange multipliers related to the linear equality constraints, and Rn stands for the Lagrange multipliers related to the nonnegativity constraints.

The infimum of the Lagrangian function over the primary variables is known as the Lagrange dual function, indicated as g(λ,ν).

$$g(\lambda,\nu) = infxL(x,\lambda,\nu)$$

It is significant to remember that the Lagrange dual function g(λ,ν) offers a lower bound on the primal problem (P)'s ideal value.

The Lagrange dual problem to (P) is

Minimize $g(\lambda,\nu)$

Subject to Ax =b,

V_i => 0, i= 1, 2, ...

We present a logarithmic barrier strategy to manage the nonnegativity restrictions without directly enforcing them. We formulate the fundamental barrier problem as follows for a given parameter > 0:

Minimize f(x)+θ∑i=1n−log(xi)

subject to Ax=b

Here, x stands for the fundamental variables, while f(x) is the cost function. The expression $\theta\sum i = 1n - log(xi)$ functions as a logarithmic barrier that penalises transgressions of the nonnegativity restrictions 0 x i 0. The barrier's strength is determined by the parameter.

The following is a representation of the Lagrange dual function related to the primal barrier problem (P):

$$g(\lambda, \nu) = infxL(x, \lambda, \nu) = infx[f(x) + \lambda T(Ax - b) - \nu Tx - \theta \sum i = 1nlog(xi)]$$

Where, $\in \lambda \in Rm \; and \; \in \nu \in Rn$ the linear equality constraints and the nonnegativity constraints, respectively, are related to the Lagrange multipliers.

For selecting parameter

We see that the variation in curvature depends on two elements when we look at the Hessian of the Lagrange dual function, abbreviated H in equation. The variance in (h(-1))T, which essentially reflects the variation in the curvature of each individual function fi, is the first factor that affects this. Second, it depends on how the singular values of the AT matrix change over time.

More specifically, it should be emphasised that variations in the singular values of the transpose of matrix A as well as changes in the individual function curvatures both affect the curvature variation. Together, these variables affect the curvature features of the Hessian matrix, which in turn affects the convergence characteristics of the optimisation procedure. It makes sense that fewer measures must be taken to ensure convergence when dealing with areas of extreme curvature. On the other hand, minor steps result in slow progress towards an ideal solution in areas with little curvature.

For analysing the behaviour of the Hessian matrix and creating successful optimisation strategies, it is essential to comprehend and take into account the variance in the curvature of the individual functions and the singular values of AT.

Iteration k

1. For $j = 1, \ldots, m$, the nodes compute an estimate $\hat{s}_j^{k-1}$ of $s_j^{k-1} = \sum_{i=1}^{n} A_{ji} x_i(\lambda^{k-1})$.
2. The nodes check the following two stopping conditions:

$$(7) \qquad (1-\epsilon_1)\left(1 - \frac{2}{3}\epsilon\right)\|b\| \leq \left\|\hat{s}^{k-1}\right\| \leq (1+\epsilon_1)\left(1 + \frac{2}{3}\epsilon\right)\|b\|$$

   and

$$(8) \qquad \left\|\hat{s}^{k-1} - b\right\| \leq \left(\frac{2}{3}\epsilon + \epsilon_1\left(\frac{1+\epsilon_1}{1-\epsilon_1}\right)\left(1 + \frac{2}{3}\epsilon\right)\right)\|b\|.$$

   If both conditions (7) and (8) are satisfied, the inner run terminates, producing as output the vector $x\left(\lambda^{k-1}\right)$.
3. The nodes update the dual vector by setting $\Delta\lambda^{k-1} = \hat{s}^{k-1} - b$ and $\lambda^k = \lambda^{k-1} + t\Delta\lambda^{k-1}$.

Figure 2: The i^{th} iteration when run in inner loop

Nodes employ the accuracy parameter 1 = α/3 for the summation subroutine, where is the distributed algorithm's specified error tolerance. Nodes compute and use the following step size for gradient ascent.

The equation can be written as

$$t = 21 - \alpha + 63(2 - 3) + 2121 + 23\alpha + 2321 + 23\alpha \cdot QR$$

The factorsand 1 QR1 ensure that the condition > 0 t>0 is true. It's crucial to remember that t can be approximated by Theta left(frac q Q 2). Up until both halting criteria are satisfied, iterations are carried out in an inner run with rising values of k. The algorithm's outer loop ends the inner run.

V. Conclusion

Numerous issues with convex objectives and constraints have been successfully solved by designing and refining optimisation methods for convex programming problems. Convex optimisation issues have advantageous characteristics, such as a globally optimal solution and effective solvability even for big issues. Particularly Interior Point approaches, which treat smooth nonlinear, linear, quadratic, and conic functions in a single framework, have become effective methods for solving convex issues. Convex optimisation techniques have given rise to necessary and sufficient conditions for global optimality through the use of convexity properties and the application of first-order KKT conditions. Barrier functions and Lagrange dual techniques have made it possible to handle nonnegativity requirements and have improved the algorithms' convergence characteristics.

VI. Future Work

Even though there have been tremendous advancements in the design and optimisation of algorithms for convex programming issues, there are still a number of areas that present opportunities for additional study and advancement. Some areas that could be investigated further include:

Creation of specialised algorithms for particular convex problem types: Numerous types of problem structures, such as linear programming, conic optimisation, quadratic programming, and others are covered by convex optimisation. Further increases in efficiency and scalability can be achieved by creating customised algorithms that take use of the unique characteristics of these issue classes.

Combining convex optimisation methods with machine learning algorithms can help solve challenging optimisation issues that arise in fields like data analysis, pattern recognition, and high-dimensional space optimisation.

Exploration of distributed and parallel optimisation: As distributed computing resources and parallel processing architectures become more widely available, there is a rising demand for optimisation techniques that can take use of these features. Investigating distributed and parallel optimisation methods for convex programming issues might greatly improve the algorithms' scalability and effectiveness.

Robust optimisation and uncertainty: In real-world optimisation issues, there is frequently uncertainty in the data or the problem parameters. Convex optimisation techniques can be extended to manage uncertainty and robustness, which can result in more dependable and robust solutions under uncertain circumstances.

References:

[1] Y. Bartal, J. W. Byers, and D. Raz, Fast, distributed approximation algorithms for positive linear programming with applications to flow control, SIAM J. Comput., 33 (2004), pp. 1261–1279.

[2] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 1989.

[3] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, UK, 2004.

[4] R. G. Gallager, A minimum delay routing algorithm using distributed computation, IEEE Trans. Comm., 25 (1977), pp. 73–85.

[5] N. Garg and N. E. Young, On-line end-to-end congestion control, in Proceedings of the 43rd Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, IEEE, Washington, DC, 2002, pp. 303–312.

[6] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, J. Oper. Res. Soc., 49 (1998), pp. 237– 252.

[7] M. Luby and N. Nisan, A parallel approximation algorithm for positive linear programming, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 448–457.

[8] D. Mosk-Aoyama and D. Shah, Fast distributed algorithms for computing separable functions, IEEE Trans. Inform. Theory, 54 (2008), pp. 2997–3007.

[9] C. H. Papadimitriou and M. Yannakakis, Linear programming without the matrix, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 121–129.

[10] R. T. Rockafellar, Network Flows and Monotropic Optimization, John Wiley & Sons, New York, 1984.

[11] R. Srikant, The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications), Birkh¨auser Boston, Boston, 2004.

[12] S.-P. Wu and S. Boyd. SDPSOL: A parser/solver for semidefinite programs with matrix structure. In L. El Ghaoui and S.-I. Niculescu, editors, Recent Advances in LMI Methods for Control, chapter 4, pages 79-91. SIAM, 2000.

[13] F. Wang and V. Balakrishnan. Robust Kalman filters for linear time-varying systems with stochastic parametric uncertainties. IEEE Transactions on Signal Processing, 50(4):803-813, April 2002.

[14] J. Weickert and Christoph Schnorr. A theoretical framework for convex regularizers in pde-based computation of image motion. International Journal of Computer Vision, Band 45, 3:245-264, 2001.

[15] S. Wang, L. Xie, and C. Zhang. Hz optimal inverse of periodic FIR digital filters. IEEE Transactions on Signal Processing, 48(9):2696-2700, September 2000

[16] H. Zhou, L. Xie, and C. Zhang. A direct approach to Hz optimal deconvolution of periodic digital channels. IEEE Transactions on Signal Processing, 5O(7): 1685-1698, July 2002.

[17] R. Tapia, Y. Zhang, and L. Velazquez. On convergence of minimization methods: Attraction, repulsion and selection. Journal of Optimization Theory and Applications, 107:529-546, 2000.

[18] Z. Tan, Y. Soh, and L. Xie. Envelope-constrained H, filter design: an LMI optimization approach. IEEE Transactions on Signal Processing, 48(10):2960- 2963, October 2000