ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК XXXXX

Отчет об исследовательском проекте на тему:

Автоматический вывод свойств функции по её выражению

Выполнил:

студент группы БПМИ202
Дмитриев Иван Юрьевич ____________ ____________
(подпись) (дата)

Принял руководитель проекта:

Посыпкин Михаил Анатольевич
Профессор
Факультета компьютерных наук НИУ ВШЭ

____________ ____________
(подпись) (дата)

Москва 2023

Содержание

Аннотация

Автоматическая проверка выпуклости или вогнутости функции по его выражению может быть применена в различных задах оптимизации. В рамках данной работы сравниваются три различных подходы решения этой задачи: дисциплинированное выпуклое программирование, анализ символического выражения гессиана и использование методов интервального анализа, некоторые из которых ранее не имели имплементации. В качестве критериев сравнения выступает множество покрытых функций, скорости вычисления и применимость для глобальной оптимизации методом ветвей и границ.

Ключевые слова

оценка выпуклости, дисциплинированное выпуклое программирование

1 Введение

Необходимость проверки выпуклости или вогнутости функции часто возникает в различных областях оптимизации.

Одной из таких областей является задача выпуклого программирования — одна из классических задач математической оптимизации. Выпуклая оптимизация привлекательна в силу двух причин: некоторый классы задач выпуклого программирования могут быть решены за полиномиальное время, при том что задача математического программирования в общем виде является NP-трудной [9], а так же любой локальный минимум выпуклой целевой функции так же является её глобальным минимумом. В силу этого задача проверки, является ли задача математической оптимизации задачей выпуклой оптимизации является практически применимой.

Другой из таких областей является применение метода ветвей и границ. В рамках этого метода требуется строить миноранты функции на некоторой области, что легко сделать для выпуклых функций, так как любая касательная к выпуклой функции является минорантой, и еще проще сделать для вогнутой функции, так как глобальный минимум вогнутой функции находится на краях области.

Известно, что в общем виде задача проверки выпуклости функции по её выражению является NP-трудной, даже если рассматриваются исключительно многочлены многих переменных степени 4 [12]. Тем не менее, существует множество различных подходов по проверке выпуклости, которые работают для некоторого класса функций. Целью данной работы является сравнение этих методов с точки зрения покрытого множества функций, вычислительной сложности и применимости в задачах оптимизации.

Первым из рассматриваемых подходов по проверке выпуклости функции является дисциплинированное выпуклое программирование. Впервые оно было описано в [8]. Подход получил широкое распространение в связи с его использованием в библиотеке CVX для Matlab [6], [7], которая в дальнейшем была портирована для других языков программирования: Julia [13], Python [3] и R [5]. В рамках проверки строится вычислительный граф функции, который затем рекурсивно анализируется при помощи правил вывода.

Второй из рассматриваемых подходов описан в [10] и основан на анализе гессиана. Для дважды дифференцируемой функции считается её гессиан, он представляется в виде вычислительного графа, который анализируется рекурсивно при помощи правил вывода.

Третий из рассматриваемых подходов описан в [11], где рассматривается анализ выпуклости функции одной переменной при помощи интервальных методов. Авторы используют набор правил вывода, схожий с дисциплинированным выпуклым программированием, совмещают его с использованием интервального анализа для оценки второй производной функции и используют монотонность для проверки выпуклости произведения функций.

В рамках данной работы планируется имплементировать подход по анализу гессиана, описанный исключительно теоретически, оценить применимость и вычислительную эффективность трех подходов для набора функций. Затем планируется использовать совместить полученные подходы с методом ветвей и границ и оценить эффективность полученного метода оптимизации.

2 Обзор литературы

2.1 Дисциплинированное выпуклое программирование

Дисциплинированное выпуклое программирование (ДВП) из двух ключевых компонентов. Первый из низ — набор функций (называемых "атомарными"), про которые известных их свойства: выпуклость, монотонность по аргументам, область значений, является ли функция линейной. Второй — набор правил вывода, определяющий, как атомарные функции могут быть скомбинированы для получения выпуклого результата.

В качестве набора правил обычно выступают следующие четыре правила, описанные в [8], где в качестве f и g_i выступают атомарные функции, в качестве α_i — произвольные константы:

1 $\sum_{i=1}^{n} \alpha_i \cdot g_i$ является выпуклой, если для любого i верно одно из следующего:

- $\alpha_i > 0$ и g_i является выпуклой
- $\alpha_i < 0$ и g_i является вогнутой
- $\alpha_i = 0$

2 $f(g_1, g_2, \ldots, g_n)$ — выпуклая, если f выпуклая и для каждого g_i верно одно из следующего:

- f монотонно возрастает по аргументу i и g_i выпуклая
- f монотонно убывает по аргументу i и g_i вогнутая
- g_i линейная или константа

3 $f(g_1, g_2, \ldots, g_n)$ — вогнутая, если f вогнутая и для каждого g_i верно одно из следующего:

- f монотонно возрастает по аргументу i и g_i вогнутая

- f монотонно убывает по аргументу i и g_i выпуклая
- g_i линейная или константа

4 $f(g_1, g_2, \ldots, g_n)$ — линейная, если f линейная и все функции g_i линейные

В качестве атомарных функций может рассматриваться произвольный набор функций с известными свойствами, например, функции сложения, умножения на константу, нормы вектора. При этом функция произведения двух аргументов не может считаться атомарной, так как произведение не сохраняет свойства выпуклости, и даже произведение линейной функции на выпуклую не обязано быть выпуклой. Это существенно ограничивает возможности ДВП: в рамках данного подхода нельзя показать выпуклость даже сравнительно простых функций, таких как $x^T \cdot \ln(x)$, не прибегая к включению их в множество атомарных.

2.2 Анализ гессиана

В статье [10] описан подход проверки выпуклости функции через анализ её гессиана. Дважды дифференцируемая функция является выпуклой, если её гессиан положительно определен, и вогнутой, если гессиан отрицательно определен. Анализ гессиана описывается следующим образом: рассматривается вычислительный граф для гессиана функции, который обходится рекурсивно. Заранее известно множество атомарных выражений, про которые известна положительная/отрицательная определенность гессиана, комбинации атомарных выражений анализируются при помощи правил вывода. В качестве правил вывода авторы рассматривают следующий набор инструкций, где c — произвольная константа, а M — гессиан:

1 Если $c \geqslant 0$ и $M \geqslant 0$, то $c \cdot M \geqslant 0$

2 Если $c \geqslant 0$ и $M \leqslant 0$, то $c \cdot M \leqslant 0$

3 Если $c \leqslant 0$ и $M \leqslant 0$, то $c \cdot M \geqslant 0$

4 Если $M_1 \geqslant 0$ и $M_2 \geqslant 0$, то $M_1 + M_2 \geqslant 0$

5 Если $M \geqslant 0$ и A — произвольная матрица, то $A \cdot M \cdot A^T \geqslant 0$

Авторы сравнивают свой подход с подходом дисциплинированное выпуклого программирования и доказывают, что анализ гессиана может доказать выпуклость всех функций, выпуклость которых может быть доказана при помощи дисциплинированного выпуклого программирования. Так же приводится ряд функций, выпуклость которых может быть доказана описанным методом, но не методом дисциплинированного выпуклого программирования. Описанный подход остается теоретическим и на данный момент не имеет имплементации.

2.3 Интервальный анализ

Метод, рассматриваемый в рамках статьи [11] работает только для функций одной переменной. Тем не менее, он примечателен тем, что в нем совмещаются два различных подхода: рекурсивный анализ самой функции, схожий с дисциплинированным выпуклым программированием и анализ второй производной функции при помощи интервальных методов, который позволяют оценить множество её значений. Авторы дополняют правила вывода при рекурсивном анализе функции условием, что произведение двух монотонно возрастающих функций всегда является выпуклым.

2.4 Иные подходы

Ряд подходов по выводу выпуклости функции не будут анализироваться, однако заслуживают упоминания.

В рамках [4] рассматривается подход, схожий с дисциплинированным выпуклым программированием. Вместо наличия небольшого количества правил вывода и библиотеки атомарных функций, авторы используют большое число правил вывода, касающихся простых функций и арифметических операций. Работа примечательна наличием результатов большого числа вычислительных экспериментов.

В статье [1] рассматривается метод проверки выпуклости, использующий гессиан. Используя символическое разложение Холецкого, авторы проверяют, что собственные значения матрицы положительно определены. Применимость данного метода сильно ограничена и он работает лишь для небольшого числа функций.

Широко известен подход, в рамках которого для проверки положительности гессиана используется интервальная арифметика, позволяющая оценить собственные значения матрицы через теорему Гершгорина. В силу грубости оценки практическая применимость данного подхода мала.

В статье [2] рассматривается соотношение между количеством шагов градиентного спуска и выпуклостью функции. В результате в рамках данного подхода можно доказать не выпуклость функции, сделав достаточное число шагов. Для доказательства выпуклости функции данный подход неприменим.

Список литературы

[1] J. Camino и др. “Matrix Inequalities: A Symbolic Procedure to Determine Convexity Automatically”. В: *Integral Equations and Operator Theory* 46 (авг. 2003), с. 399—454.

[2] Yair Carmon и др. ““Convex Until Proven Guilty”: Dimension-Free Acceleration of Gradient Descent on Non-Convex Functions”. В: *Proceedings of the 34th International Conference on Machine Learning*. Под ред. Doina Precup и Yee Whye Teh. Т. 70. Proceedings of Machine Learning Research. PMLR, июнь 2017, с. 654—663.

[3] Steven Diamond и Stephen P. Boyd. “CVXPY: A Python-Embedded Modeling Language for Convex Optimization”. В: *Journal of machine learning research : JMLR* 17 (2016).

[4] Robert Fourer и др. “Convexity and Concavity Detection in Computational Graphs: Tree Walks for Convexity Assessment”. В: *INFORMS Journal on Computing* 22 (февр. 2010), с. 26—43.

[5] Anqi Fu, Balasubramanian Narasimhan и Stephen Boyd. “CVXR: An R Package for Disciplined Convex Optimization”. В: *Journal of Statistical Software* 94.14 (2020), с. 1—34.

[6] Michael Grant и Stephen Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx(дата обр. 15.02.2023). Март 2014.

[7] Michael Grant и Stephen Boyd. “Graph implementations for nonsmooth convex programs”. В: *Recent Advances in Learning and Control*. Под ред. V. Blondel, S. Boyd и H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, с. 95—110.

[8] Michael Grant, Stephen Boyd и Yinyu Ye. “Disciplined Convex Programming”. В: янв. 2006, с. 155—210.

[9] Santosh Kabadi Katta Murty. “Some NP-complete problems in quadratic and nonlinear programming”. В: *Mathematical Programming* (1987).

[10] Julien Klaus и др. “Convexity Certificates from Hessians”. В: *Advances in Neural Information Processing Systems*. Под ред. Alice H. Oh и др. 2022.

[11] Posypkin Mikhail и Khamisov Oleg. “Automatic Convexity Deduction for Efficient Function’s Range Bounding”. В: *Mathematics* 9.2 (2021).

[12] “NP-hardness of deciding convexity of quartic polynomials and related problems”. В: *Mathematical Programming* 137.1 (2013), с. 453—476.

[13] Madeleine Udell и др. “Convex Optimization in Julia”. В: (окт. 2014).