

© 2024 г. П.А. АРХИПОВ (pavel.arkhipov@ist.ac.at)
(Институт науки и технологий (ISTA), Австрия)

АЛГОРИТМ НАХОЖДЕНИЯ ОБОБЩЕННОГО ЧЕБЫШЕВСКОГО ЦЕНТРА МНОЖЕСТВ, ЗАДАННЫХ ОПОРНЫМИ ФУНКЦИЯМИ

Статья посвящена задаче оптимизации. Пусть $A, B \subset \mathbb{R}^n$ – выпуклые компакты. Рассмотрим минимальное число $t^0 > 0$ такое, что $t^0 B$ покрывает A после сдвига на вектор $x^0 \in \mathbb{R}^n$. Цель – найти t^0 и x^0 . В частном случае, когда B является единичным шаром с центром в нуле, x^0 и t^0 известны как чебышевский центр и чебышевский радиус A . В данной статье рассматривается случай, когда A и B определяются с помощью своих опорных функций. Предложен алгоритм в духе градиентного спуска Б.Т. Поляка для эффективного решения таких задач. Алгоритм имеет сверхлинейную скорость сходимости и может решать стомерные тестовые задачи за разумное время, однако для гарантии наличия сходимости необходимы некоторые дополнительные условия на A и B . Дополнительно исследовано поведение алгоритма для простого частного случая, что приводит к ряду теоретических результатов. Изучаются также возмущения этого частного случая.

Ключевые слова: оптимизация, чебышевский центр, градиентный спуск.

DOI: 10.31857/S0005231024060059, **EDN:** XXDBSM

1. Введение

Пусть A и B – выпуклые компакты в \mathbb{R}^n . Пусть $t^0 \in \mathbb{R}$ – наименьшее неотрицательное число, такое что некоторая сдвинутая копия $t^0 B$ покрывает A ,

$$(1) \quad t^0 = \min\{t \geq 0 \mid \exists x^0 \in \mathbb{R}^n : x^0 + t^0 B \supset A\}.$$

На рис. 1 показан пример оптимальной конфигурации A и B .

Можно заметить, что та же задача имеет смысл, если A невыпукло. Действительно, если в такой задаче A заменить на ее выпуклую оболочку $\text{conv}(A)$, то оптимальные x^0 и t^0 останутся прежними. Итак, без ограничения общности можно считать, что A выпукло.

Задача нахождения чебышевского центра множества является частным случаем описанной выше задачи: если $B = B_1(0) = \{y \in \mathbb{R}^n \mid \|y\| \leq 1\}$, то x^0 – чебышевский центр A . Известно, что ее можно быстро решить, если A конечно [4]. Но в целом подобные задачи оказываются вычислительно сложными. В последние годы им уделялось некоторое внимание. Например, задача нахождения чебышевского центра пересечения шаров NP-сложная [5]. Даже

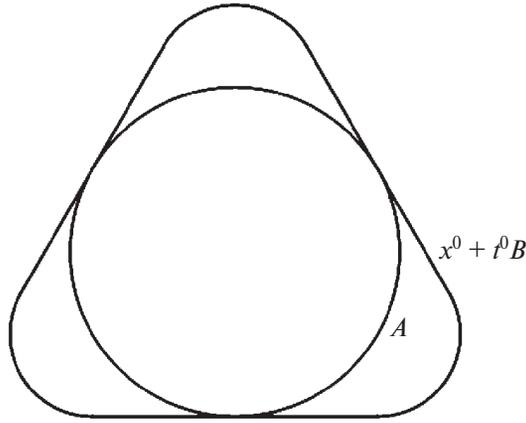


Рис. 1. Оптимальная конфигурация A и B .

вычисление чебышевского центра пересечения двух эллипсоидов является существенной проблемой [16]. У чебышевского центра есть практические применения. Он может быть полезен при решении задач долгосрочного гидротермального планирования [17], задач регрессии с шумом [18], идентификации линейных динамических систем с шумом [19], кибербезопасности [20, 22] и робототехники [21].

Рассмотрим такие задачи, в которых оптимальный x^0 единственный.

Цель состоит в том, чтобы разработать алгоритм, который эффективно решает задачу, если множества A и B определены со своими опорными функциями. Напомним читателю, что для данного множества $M \subset \mathbb{R}^n$ значение опорной функции M в точке $p \in \mathbb{R}^n$ определяется следующим образом:

$$h_M(p) = \sup\{(p, y) \mid y \in M\}.$$

В этих обозначениях

$$(2) \quad A = \{y \in \mathbb{R}^n \mid (y, p) \leq h_A(p) \forall p \in S^{n-1}\},$$

$$(3) \quad B = \{y \in \mathbb{R}^n \mid (y, p) \leq h_B(p) \forall p \in S^{n-1}\},$$

где $S^{n-1} = \{y \in \mathbb{R}^n \mid \|y\| = 1\}$ – стандартная единичная сфера. Затем проблеме (1) можно переформулировать:

$$(4) \quad t^0 = \min \{t \mid \exists x^0 \in \mathbb{R}^n : (x, p) + t \cdot h_B(p) \geq h_A(p) \forall p \in S^{n-1}\}.$$

В некотором смысле это задача линейного программирования с бесконечным числом ограничений. Действительно, минимизируется линейный функционал $(w, c) = t$, $w = (t, x_1, \dots, x_n)^T \in \mathbb{R}^{n+1}$, $c = (1, 0, \dots, 0)^T$, с учетом ограничений $(w, b_p) \leq h_A(p)$, где $b_p = (h_B(p), p_1, \dots, p_n)^T$, что должно выполняться для любого p единичной нормы.

Один известный подход к проблеме описан в [1]. Автор выбирает конечное число линейных ограничений, соответствующих сетке на единичной сфере, и решает задачу линейного программирования с учетом этих ограничений. Недостатком этого подхода является то, что достаточно мелкая сетка на единичной сфере большой размерности содержит слишком много элементов. На практике вычисления становятся слишком трудными при $n > 4$. Таким образом, разработка алгоритма, способного решить поставленную задачу в больших размерностях, является актуальной исследовательской задачей.

Прежде чем описывать алгоритм, имеет смысл напомнить читателю, что для строго выпуклого компакта $M \subset \mathbb{R}^n$ его опорная функция имеет следующий градиент:

$$(5) \quad \nabla h_M(p) = \arg \max_{y \in M} (p, y).$$

Очевидно, $(p, y) = h_M(p)$. Если M выпукло, но не строго выпукло, можно рассмотреть субдифференциал $\partial h_M(p)$:

$$(6) \quad \partial h_M(p) = H(p) \cap M,$$

где $H(p) = \{y \in \mathbb{R}^n | (p, y) = h_M(p)\}$. Субдифференциалом выпуклой функции $f: \mathbb{R}^n \rightarrow \mathbb{R}$ в точке x_0 называется множество всех векторов $v \in \mathbb{R}^n$ таких, что $f(x) - f(x_0) \geq (v, x - x_0)$ для всех $x \in \mathbb{R}^n$. Такие векторы v называются субградиентами f в точке x_0 . Легко видеть, что если M строго выпуклая, то субдифференциал ее опорной функции является одноэлементным множеством в каждой точке:

$$(7) \quad \begin{aligned} \partial h_M(p) &= H(p) \cap M = \{y \in M | (p, y) = h_M(p)\} = \\ &= \left\{ \arg \max_{y \in M} (p, y) \right\} = \{\nabla h_M(p)\}. \end{aligned}$$

Обозначим выпуклую оболочку множества M как $\text{conv}(M)$, а r -сильно выпуклую оболочку M как $\text{strconv}_r(M)$. Множество называется r -сильно выпуклым, если его можно представить в виде пересечения шаров радиуса r . Предположим, $M \subset \mathbb{R}^n$ можно покрыть шаром радиуса r . Тогда $\text{strconv}_r(M)$ – это пересечение всех шаров радиуса r , содержащих M (также как выпуклая оболочка M – это пересечение всех полупространств, содержащих M).

2. Алгоритм и теоретические результаты

Чтобы описанный ниже алгоритм наверняка сошелся, A и B должны удовлетворять условию: функция $f: S^{n-1} \rightarrow \mathbb{R}$, $f(p) = (x^0, p) + t^0 h_B(p) - h_A(p)$ должна иметь ровно $n + 1$ локальных минимумов p_1^0, \dots, p_{n+1}^0 , и выпуклая оболочка множества этих минимумов должна содержать ноль: $0 \in \text{int}(\text{conv}(p_1^0, \dots, p_{n+1}^0))$. Неформально, эти условия означают, что задача линейного программирования (4) имеет ровно $n + 1$ активных ограничений.

Пусть p_1, \dots, p_{n+1} – векторы единичной длины. Будем использовать следующие обозначения:

$$(8) \quad a_i = h_A(p_i), \quad a = (a_1, \dots, a_{n+1})^T,$$

$$(9) \quad b_i = h_B(p_i), \quad b = (b_1, \dots, b_{n+1})^T,$$

$$(10) \quad M_a = \begin{pmatrix} a_1 & (p_1)_1 & \dots & (p_1)_n \\ a_2 & (p_2)_1 & \dots & (p_2)_n \\ \vdots & \vdots & \ddots & \vdots \\ a_{n+1} & (p_{n+1})_1 & \dots & (p_{n+1})_n \end{pmatrix},$$

$$(11) \quad M_b = \begin{pmatrix} b_1 & (p_1)_1 & \dots & (p_1)_n \\ b_2 & (p_2)_1 & \dots & (p_2)_n \\ \vdots & \vdots & \ddots & \vdots \\ b_{n+1} & (p_{n+1})_1 & \dots & (p_{n+1})_n \end{pmatrix}.$$

Лемма 1. Если $0 \in \text{conv}(\{p_1, \dots, p_{n+1}\})$, то для t, x таких, что $(p_j, x) + th_B(p_j) \geq h_A(p_j)$ и t минимально, справедливо равенство

$$(12) \quad M_b \begin{pmatrix} t \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = a.$$

Следствие 1. Используя правило Крамера, имеем

$$(13) \quad t = \frac{\det M_a}{\det M_b}.$$

Алгоритм 1.

Начнем с точек $p_1, \dots, p_{n+1} \in S^{n-1}$, равных вершинам случайно повернутого правильного симплекса. На каждой итерации будем делать следующее:

1. Найдем решение системы линейных уравнений (12), t, x .
2. Заменим каждый p_i на p'_i , который является локальным минимумом функции $f_{t,x}(p) = (x, p) + th_B(p) - h_A(p)$, найденным с помощью градиентного спуска из начальной точки p_i . Градиент $f_{t,x}$ можно вычислить следующим образом:

$$(14) \quad \nabla f_{t,x}(p) = x + t \arg \max_{y \in B} (p, y) - \arg \max_{y \in A} (p, y).$$

Если существует несколько глобальных максимумов (p, y) , выберем произвольный.

3. Если среди p'_i нет дубликатов, положим p_i равными p'_i и завершим эту итерацию.

В противном случае удалим дубликаты из $\{p'_i\}$. Пусть L – множество оставшихся минимумов p'_i .

Выполним градиентный спуск до минимума $f_{t,x}$ начиная с (равномерно) случайного элемента единичной сферы. Если этот минимум не совпадает ни с одним элементом L , добавим его в L . Будем делать так до тех пор, пока в L не будет $n + 1$ элементов. Затем положим p_i равными p'_i и завершим эту итерацию. Если количество выполненных попыток больше K , но еще не найдены $n + 1$ различных минимумов, перезапустим весь алгоритм.

Остановимся, когда t и x изменяются достаточно незначительно за одну итерацию или когда количество итераций превышает некоторый предел.

Градиентный спуск описан, например, в классической книге [2].

Далее обсудим типы задач, которые можно решить с помощью алгоритма 1.

Лемма 2. Если B строго выпуклая, то функция $f_{t,x} = (x, p) + th_B(p) - h_A(p)$, $t \in \mathbb{R}^+$, $x \in \mathbb{R}^n$ дифференцируема в любом своем минимуме.

В частности, $f_{t,x}$ дифференцируема в своих минимумах для задач нахождения чебышевского центра. Дифференцируемость $f_{t,x}$ в минимумах также может быть полезна для правильной сходимости градиентного спуска. Но существуют модификации подхода градиентного спуска, которые могут сходиться и к негладким минимумам.

Теорема 1. Пусть p_1^0, \dots, p_{n+1}^0 - минимумы $f(p) = (x^0, p) + t^0 h_B(p) - h_A(p)$, $0 \in \text{int}(\text{conv}(p_1^0, \dots, p_{n+1}^0))$. Пусть B строго выпукла. Предположим, что для любого $j \in \{1 \dots n + 1\}$ и для любой точки p из некоторой окрестности p_j^0

$$(15) \quad M \|p - p_j^0\|^\alpha \geq f(p) \geq \mu \|p - p_j^0\|^2,$$

где M, μ, α - константы, $1 < \alpha \leq 2$. Тогда в некоторой окрестности решения алгоритм (1) сходится со степенью сходимости α . То есть для t, x из некоторой окрестности t^0, x^0

$$(16) \quad \|(t' - t^0, x'_1 - x_1^0, \dots, x'_n - x_n^0)^T\| \leq C \|(t - t^0, x_1 - x_1^0, \dots, x_n - x_n^0)^T\|^\alpha,$$

где t', x' - состояние алгоритма на следующей итерации, если текущее состояние равно t, x .

Теорема 2. Пусть $B = B_1(0)$, $A = \text{conv}((v_1 + C_1) \cup \dots \cup (v_{n+1} + C_{n+1}))$, где $v_j \in \mathbb{R}^n$, $C_j \subset \mathbb{R}^n$ - сильно выпуклые множества с радиусом сильной выпуклости $r < t^0$. Также пусть f имеет хотя бы $n + 1$ нулей. Тогда алгоритм 1 имеет квадратичную сходимость.

3. Численные эксперименты

Код, который использовался для численных экспериментов, а также рисунки находятся в публичном доступе на GitHub, <https://github.com/Paul566/GraduateThesis>. Класс солвера находится в файле GradientDescentSolver.py.

Алгоритм 1 был реализован и протестирован. Во время всех тестов 15 итераций хватало, чтобы алгоритм сошелся с машинной точностью. Во всех случаях погрешность итогового результата составляла около 10^{-14} – 10^{-16} . В большинстве случаев 5–7 итераций было достаточно.

Во всех тестах алгоритм имел следующие параметры: максимальное количество итераций: 20, параметр скорости обучения в градиентном спуске: 1, количество попыток найти $n + 1$ минимумов: $10n$, максимальное количество шагов при градиентном спуске: 1000, градиентный спуск прекращается, если f изменяется менее чем на 10^{-10} , два минимума считаются дубликатами, если норма их разницы меньше 10^{-6} , максимальное количество перезапусков 1000. Использовался обычный градиентный спуск без модификаций. Для некоторых серий тестов будет представлен график зависимости ошибки выходных данных от времени выполнения. Вычисления проводились на CPU на обычном ноутбуке.

1. Начнем с очень простой серии тестов. Пусть $B = B_1(0)$, A – случайный симплекс, содержащий центр описанной сферы. Результаты численных экспериментов для размерности 100 представлены на рис. 2. Конечно, решение подобных задач с помощью алгоритма 1 не имеет практического смысла, данная серия симуляций проводилась в целях тестирования. Как видно, даже стомерные задачи решаются за несколько секунд. Найти опорные функции в этом случае несложно:

$$(17) \quad h_B(p) = \|p\|,$$

$$(18) \quad \nabla h_B(p) = \frac{p}{\|p\|},$$

$$(19) \quad h_A(p) = \max_i (v_i, p),$$

$$(20) \quad \nabla h_A(p) = \arg \max_{v_i} (v_i, p).$$

2. Теперь перейдем к более сложной серии тестов. Пусть множество A – выпуклая оболочка $n + 1$ эллипсоидов, а B – шар. Результаты численных экспериментов для размерности 100 представлены на рис. 3.

В этом случае на решение 100-мерной задачи алгоритму требуются десятки секунд, что примерно на порядок дольше, чем в предыдущей серии. Однако в большинстве случаев для сходимости все равно требовалось 5–7 итераций. Увеличение времени выполнения связано с усложнением вычисления опорной функции A . Действительно, для эллипсоида $E = v + MB_1(0)$, где M – матрица, $v \in \mathbb{R}^n$,

$$(21) \quad h_E(p) = (p, v) + \max_{y \in MB_1(0)} (p, y) = (p, v) + \max_{u \in B_1(0)} (p, Mu) =$$

$$(22) \quad = (p, v) + \max_{u \in B_1(0)} (M^T p, u) = (p, v) + \|M^T p\|,$$

$$(23) \quad \nabla h_E(p) = v + M \frac{M^T p}{\|M^T p\|} = v + \frac{MM^T p}{\|M^T p\|}.$$

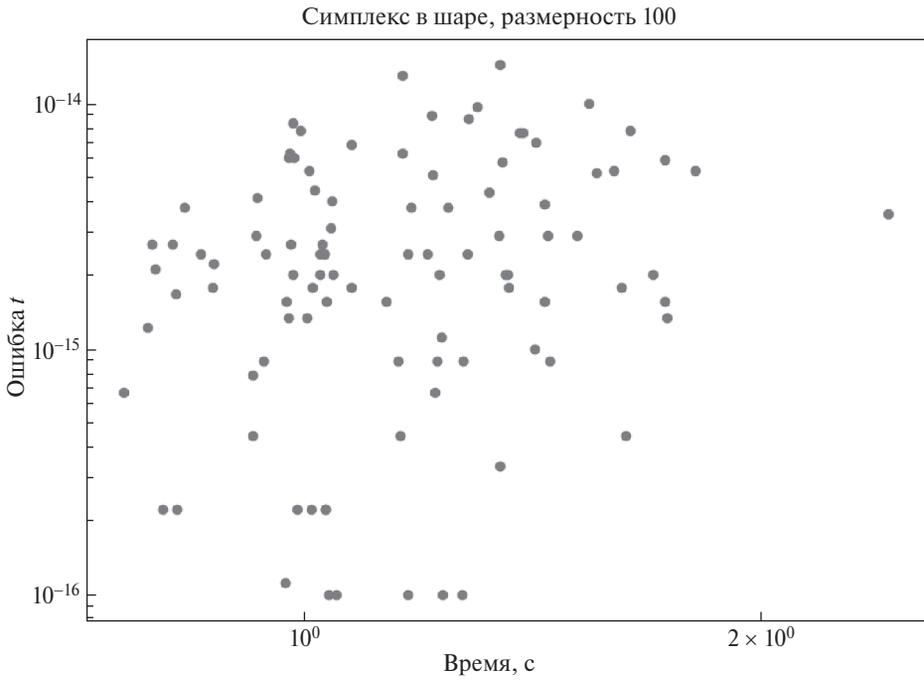


Рис. 2. Погрешность t в зависимости от времени выполнения, $n = 100$, результаты 100 тестов.

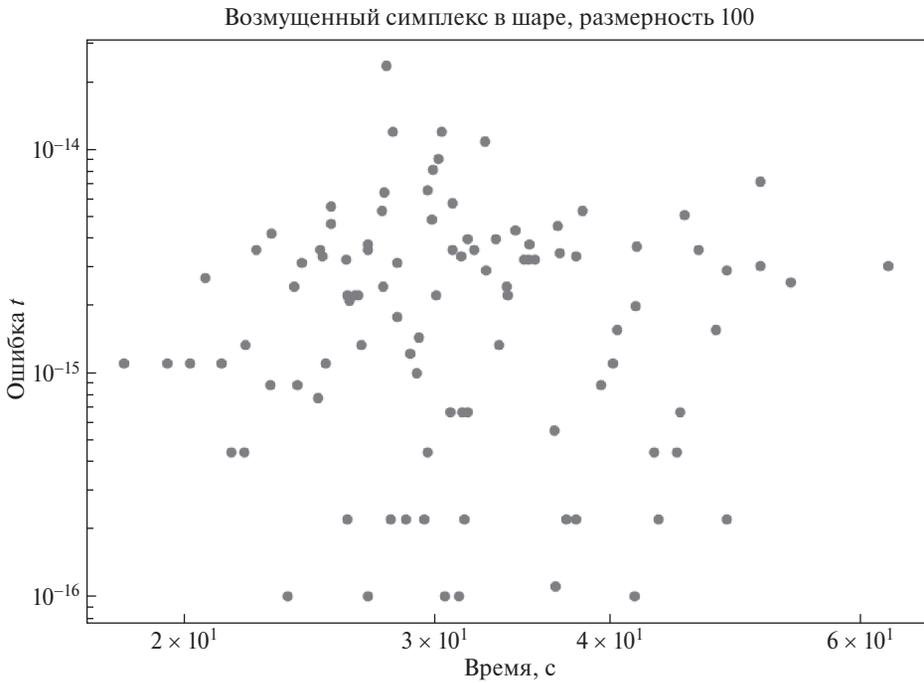


Рис. 3. Погрешность t в зависимости от времени выполнения, $n = 100$.

Если $A = \text{conv}(E_1 \cup \dots \cup E_{n+1})$, $E_j = v_j + M_j B_1(0)$, опорную функцию A можно вычислить следующим образом:

$$(24) \quad h_A(p) = \max_i h_{E_i}(p).$$

Затем градиент h_A вычисляется как градиент h_{E_k} , где E_k – эллипсоид с наибольшим $h_{E_k}(p)$.

Таким образом, вычисление опорной функции выпуклой оболочки $n + 1$ эллипсоидов вычислительно гораздо затратнее.

4. Частный случай симплекса в шаре

Пусть в этом разделе B – шар, а A – симплекс, содержащий центр описанной сферы. В этом случае $x^0 + t^0 \partial B$ совпадает с описанной сферой A . Достаточно рассмотреть случай $x^0 = 0$, $t^0 = 1$, $B = B_1(0)$, поэтому эти равенства будут предполагаться на протяжении всей главы.

Лемма 3. Минимумы $f_{t,x}$ зависят только от x и не зависят от t .

Определение 1. Множество $F \subset \mathbb{R}^n$ является замыканием множества таких $x \in \mathbb{R}^n$, что $f_{t,x} = (x, p) + h_B(p) - h_A(p)$ имеет $n + 1$ минимумов на единичной сфере.

Поскольку для $x \in F$ алгоритм 1 имеет шанс сойтись (иначе алгоритм просто не сможет найти $n + 1$ минимумов $f_{t,x}$), изучим некоторые свойства F , особенно ограничения на его объем.

Лемма 4. Если $n = 2$, то $F = \text{conv}(A \cup -A)$, $\text{Vol}(F) = 2\text{Vol}(A)$.

Пусть v_1, \dots, v_{n+1} – вершины A . Следующая лемма описывает F в терминах v_1, \dots, v_{n+1} .

Лемма 5.

$$(25) \quad F = \{x \in \mathbb{R}^n | (v_j - x, v_j) \geq (v_j - x, v_i) \forall i, j\},$$

$$(26) \quad F = \{x \in \mathbb{R}^n | (v_i - x, v_i - x) \geq (v_i - x, v_j - x) \forall i, j\}.$$

F – многогранник, ограниченный гиперплоскостями, содержащими v_i , и ортогональными $v_i - v_j$, для всех $i \neq j$.

Теорема 3. Пусть $A \subset \mathbb{R}^n$ – n -мерный симплекс с вершинами v_1, \dots, v_{n+1} , $F = \{x \in \mathbb{R}^n | (v_i - x, v_i - x) \geq (v_i - x, v_j - x) \forall i, j\}$. Затем

$$(27) \quad \text{Vol}(F) \geq n! \text{Vol}(A),$$

и если выполняется равенство, то F замощает \mathbb{R}^n .

Теорема 2 утверждает, что существует некоторая окрестность решения, такая что алгоритм 1 сходится с некоторой скоростью. Однако она ничего не говорит о размерах этой окрестности. Следующие две теоремы дают некоторую информацию об этой окрестности.

Теорема 4. Пусть $D = \max_{i,j} \|v_i - v_j\|$. Если на текущей итерации Алгоритм 1 находится в состоянии t , x , где $x \in F$ и

$$(28) \quad \|x\| \leq \frac{7}{16\sqrt{4 - D^2}},$$

$t \leq 1$, то $\|x'\| \leq \|x\|$, где t' , x' – состояние алгоритма на следующей итерации.

Теорема 5. Пусть $d = \min_{i,j} \|v_i - v_j\|$ и $d > 1$. Тогда для состояния t , x , где $x \in \text{int}(F)$, $t \leq 1$, на следующей итерации $\|x'\| < \|x\|$. Следовательно, $x' \in F$.

Поскольку при больших n все ребра случайного симплекса, вписанные в единичную сферу, почти наверняка имеют длину больше 1, приведенная выше теорема почти наверняка применима. В этих случаях достаточно $\|x\| < 0,5$ на начальной итерации для сходимости алгоритма 1.

5. Связь чебышевского центра симплекса и возмущенного симплекса

В предыдущем разделе описано поведение алгоритма в случае, когда B – шар, а A – симплекс, содержащий центр своей описанной сферы. Решение такой задачи очевидно: нужно просто вписать симплекс в сферу. Поэтому пока утверждения, приведенные в предыдущем разделе, на практике бесполезны. Содержательным случаем являлся бы анализ возмущенной задачи из предыдущего раздела.

Теорема 2 описывает класс множеств A , центр Чебышева которых можно найти с помощью алгоритма 1. Эти множества описываются как «возмущенные симплексы»: $A = \text{conv}((v_1 + C_1) \cup \dots \cup (v_{n+1} + C_{n+1}))$, где v_j – точки, C_j являются r -сильно выпуклыми множествами, а f гарантированно имеет $n + 1$ нулей. Интуитивно можно надеяться, что поведение алгоритма 1 для «возмущенного» симплекса не должно сильно отличаться от его поведения для симплекса, изученного в предыдущем разделе. Обсудим это в данном разделе.

Пусть B – шар на протяжении этого раздела. В теореме 2 можно сделать $C_j \subset B_r(0)$ с помощью сдвига, поскольку C_j r -сильно выпукло, а значит, содержится в шаре радиуса r . Сформулируем теорему о множестве F , определенном в предыдущем разделе.

Теорема 6. Пусть $B = B_1(0)$, $A = \text{conv}((v_1 + C_1) \cup \dots \cup (v_{n+1} + C_{n+1}))$, C_j – r -сильно выпуклые множества, как в теореме 2. Пусть $C_j \subset B_r(0)$. F – это множество $x \in \mathbb{R}^n$ таких, что $f_{t,x}(p) = (x, p) - h_A(p)$ имеет $n + 1$ минимумов на единичной сфере. Пусть \tilde{A} – симплекс с вершинами v_j , $\tilde{A} = \text{conv}(v_1, \dots, v_{n+1})$. Пусть \tilde{F} – множество $x \in \mathbb{R}^n$, таких что $f_{t,x}(p) =$

$= (x, p) - h_{\tilde{A}}(p)$ имеет $n + 1$ минимумов на единичной сфере. Тогда

$$(29) \quad F + B_{r/d}(0) \supset \tilde{F},$$

где $d = \min_{i \neq j} \|v_i - v_j\|$.

Замечание. Справедливо несколько более сильное утверждение:

$$(30) \quad F + B_\varepsilon(0) \supset \tilde{F},$$

где

$$(31) \quad \varepsilon = \frac{\max_{i,j} \max_{p \in S^{n-1}} (h_{C_i}(p) - h_{C_j}(p))}{d} \leq \frac{r}{d}.$$

Например, если все множества C_j одинаковы, то $F \supset \tilde{F}$.

6. Заключение

Рассматривается задача оптимизации (1). Предложен, реализован и обсужден алгоритм 1, решающий задачу для конкретного класса множеств A, B . Алгоритм выполняет несколько итераций. На каждой итерации он ищет минимумы «зазора» между A и $x + tB$, используя градиентный спуск. На практике алгоритм сходится очень быстро даже для 100-мерных задач, демонстрируя квадратичную скорость сходимости. Однако, чтобы гарантировать наличие сходимости, необходимо наложить дополнительные ограничения на A и B . Было теоретически исследовано поведение алгоритма для задач вида «симплекс в шаре». Исследованы свойства множества F таких начальных условий, что существует $n + 1$ минимумов $f_{t,x}$, и, таким образом, алгоритм 1 может дойти до следующей итерации. Далее было показано, что если немного возмутить симплекс, то F тоже изменится несильно (теорема 6). Основные теоретические результаты: теорема 1 о скорости сходимости алгоритма; теорема 2, описывающая класс задач, к которым применим алгоритм; теорема 3, ограничивающая объем F и дающая симпатичный факт о замощении \mathbb{R}^n ; теоремы 4, 5, описывающие такие начальные условия, при которых сходимость алгоритма 1 гарантирована; теорема 6, которая устанавливает связь между множеством F , когда A – симплекс и когда A – возмущенный симплекс.

ПРИЛОЖЕНИЕ

Доказательство леммы 2. Функции (x, p) и $th_B(p)$ дифференцируемы. Достаточно доказать, что h_A дифференцируема в минимумах $f_{t,x}$.

Пусть p_0 будет минимумом $f_{t,x}$; h_A – выпуклая функция, имеющая субдифференциал ∂h_A . Построим доказательство от противного. Пусть $\partial h_A(p_0)$ имеет более одного элемента.

Для любого $\varepsilon > 0$ существует δ такой, что для любого $p \in B_\delta(p_0)$ выполняется следующее неравенство:

$$(II.1) \quad |\varphi(p) - \varphi(p_0) - (g, p - p_0)| < \varepsilon \|p - p_0\|.$$

Для всех $y \in \partial h_A(p_0)$

$$(II.2) \quad h_A(p) \geq h_A(p_0) + (y, p - p_0),$$

$$(II.3) \quad -h_A(p) + h_A(p_0) \geq (-y, p - p_0).$$

Уравнения (II.1) и (II.3) дают

$$(II.4) \quad \varphi(p) - \varphi(p_0) - h_A(p) + h_A(p_0) \leq \varepsilon \|p - p_0\| + (g, p - p_0) - (y, p - p_0),$$

$$(II.5) \quad f(p) - f(p_0) \leq \varepsilon \|p - p_0\| + (g - y, p - p_0).$$

Поскольку в $\partial h_A(p_0)$ больше одного элемента, существует такой $y \in \partial h_A(p_0)$, что $y \neq g$. Тогда при $p - p_0 = \delta \frac{y-g}{\|y-g\|}$

$$(II.6) \quad f(p) - f(p_0) \leq \varepsilon \delta - \delta \|g - y\|,$$

что меньше нуля для достаточно малых ε . Это противоречит тому, что f имеет минимум в p_0 .

Доказательство теоремы 1. Достаточно рассмотреть случай $x^0 = 0$, поскольку другие случаи можно свести к этому подходящим сдвигом. Будем считать, что $x^0 = 0$. Для p из некоторой окрестности $U(p_j^0)$ справедливы следующие неравенства:

$$(II.7) \quad \begin{cases} f_{t,x}(p) = (x, p) + th_B(p) - h_A(p) \geq (x, p) + (t - t^0)h_B(p) + \mu \|p - p^0\|^2, \\ f_{t,x}(p) = (x, p) + th_B(p) - h_A(p) \leq (x, p) + (t - t^0)h_B(p) + M \|p - p^0\|^2. \end{cases}$$

Пусть p' – минимум $f_{t,x}$ в $U(p_j^0)$. Сначала оценим $\|p' - p_j^0\|$.

$$(II.8) \quad \begin{cases} f_{t,x}(p') = (x, p') + th_B(p') - h_A(p') \geq \\ \geq (x, p') + (t - t^0)h_B(p') + \mu \|p' - p^0\|^2, \\ f_{t,x}(p') \leq \min_{p \in U(p_j^0)} \{(x, p) + (t - t^0)h_B(p) + M \|p - p^0\|^2\} \leq \\ \leq (x, p_j^0) + (t - t^0)h_B(p_j^0). \end{cases}$$

Тогда

$$(II.9) \quad (x, p') + (t - t^0)h_B(p') + \mu \|p' - p^0\|^2 \leq f_{t,x}(p') \leq (x, p_j^0) + (t - t^0)h_B(p_j^0),$$

$$(II.10) \quad (x, p') + (t - t^0)h_B(p') + \mu \|p' - p^0\|^2 \leq (x, p_j^0) + (t - t^0)h_B(p_j^0).$$

Откуда получаем

$$(П.11) \quad \mu \|p' - p^0\|^2 \leq (x, p_j^0 - p') + (t - t^0)(h_B(p_j^0) - h_B(p')).$$

Используя неравенство Коши–Буняковского и определение градиента опорной функции, получим

$$(П.12) \quad \mu \|p' - p^0\|^2 \leq (x, p_j^0 - p') + (t - t^0)(h_B(p_j^0) - h_B(p')) \leq$$

$$(П.13) \quad \leq \|x\| \|p' - p_j^0\| + |t - t^0| \|\nabla h_B(p_j^0)\| \|p' - p_j^0\| (1 + o(1)).$$

Тогда

$$(П.14) \quad \|p' - p_j^0\| \leq \text{Const}(\|x\| + |t - t^0|)$$

для некоторой константы Const , не зависящей от x и t в некоторой окрестности (t^0, x^0) . Теперь есть оценка погрешности $\|p' - p_j^0\|$.

Для состояния алгоритма на следующей итерации t', x' ,

$$(П.15) \quad (x', p'_j) + t' h_B(p'_j) - h_A(p'_j) = 0 \quad \forall j \in \{1, \dots, n+1\},$$

$$(П.16) \quad (x', p'_j) + (t' - t^0) h_B(p'_j) = h_A(p'_j) - t^0 h_B(p'_j).$$

Это система линейных уравнений относительно x' и $t' - t^0$. Поскольку при любом фиксированном j правая часть уравнения равна $h_A(p'_j) - t^0 h_B(p'_j) = -f(p'_j)$, то норма правой стороны во всей системе не превосходит $\sqrt{n+1} M \|p'_j - p_j^0\|^\alpha \leq \text{Const}(\|x\| + |t - t^0|)^\alpha$. Матрица этой системы не вырождена, если $0 \in \text{conv}(\{p'_1, \dots, p'_{n+1}\})$, что справедливо, если t и x достаточно близки до t^0, x^0 . Обратная матрица имеет ограниченную норму для (p'_1, \dots, p'_{n+1}) из некоторой окрестности $(p_1^0, \dots, p_{n+1}^0)$. Таким образом, справедлива оценка

$$(П.17) \quad \|(t' - t^0, x'_1 - x_1^0, \dots, x'_n - x_n^0)^T\| \leq \text{Const}(\|x\| + |t - t^0|)^\alpha \leq$$

$$(П.18) \quad \leq \text{Const} \|(t - t^0, x_1 - x_1^0, \dots, x_n - x_n^0)^T\|^\alpha,$$

что завершает доказательство.

Доказательство теоремы 2. Достаточно доказать утверждение для $x^0 = 0, t^0 = 1$.

1. Заметим, что $(A \cap \partial B) \subset \bigcup_j (v_j + C_j)$.

Действительно, по теореме Каратеодори A является объединением симплексов с вершинами из множеств $v_j + C_j$. Симплекс, являющийся подмножеством шара, может пересекать границу шара только по своим вершинам. Итак, каждая точка из $A \cap \partial B$ принадлежит $v_j + C_j$ для некоторого j .

2. Для каждого j множество $(v_j + C_j) \cap \partial B$ содержит не более одного элемента.

Докажем от противного, т.е. если для некоторого j существуют различные $a, b \in (v_j + C_j) \cap \partial B$. Пусть M – r -сильно выпуклая оболочка $\{a, b\}$: $M = \text{strconv}_r(\{a, b\})$. Поскольку множества $v_j + C_j$ сильно выпуклые,

$$(P.19) \quad M \subset v_j + C_j.$$

Поскольку $a, b \in \partial B = \partial B_1(0)$ и $r < 1$,

$$(P.20) \quad M \cap (\mathbb{R}^n \setminus B) \neq \emptyset.$$

Тогда

$$(P.21) \quad (v_j + C_j) \cap (\mathbb{R}^n \setminus B) \neq \emptyset,$$

что противоречит тому, что $(v_j + C_j) \subset B$.

3. Для каждого j множество $(v_j + C_j) \cap \partial B$ содержит ровно один элемент. По предыдущему пункту $v_j + C_j$ пересекает ∂B не более чем в одной точке. По условию теоремы f имеет не менее $n + 1$ нулей, тогда $A \cap \partial B$ имеет не менее $n + 1$ элементов. Итак, $(v_1 + C_1) \cup \dots \cup (v_{n+1} + C_{n+1})$ пересекает ∂B как минимум по $n + 1$ точкам. Таким образом, каждое множество $v_j + C_j$ пересекает ∂B .

4. В некоторой окрестности каждого своего нуля f имеет квадратичную верхнюю оценку.

Пусть p_1^0, \dots, p_{n+1}^0 – нули f . Для симплекса $S = \text{conv}(\{p_1^0, \dots, p_{n+1}^0\})$ и точки p из некоторой окрестности p_j^0 справедливо равенство: $h_S(p) = (p, p_j^0) = 1 - \frac{1}{2}\|p - p_j^0\|^2$. Поскольку $S \subset A$, $h_A(p) \geq h_S(p)$, это дает искомую оценку $1 - h_A(p) \leq 1 - h_S(p) = \frac{1}{2}\|p - p_j^0\|^2$.

5. В некоторой окрестности каждого своего нуля f имеет квадратичную оценку снизу.

$v_j + C_j \subset B_r((1 - r)p_j^0)$, следовательно, $A \supset \text{conv}(B_r((1 - r)p_1^0) \cup \dots \cup B_r((1 - r)p_{n+1}^0)) = B_r(0) + (1 - r)\text{conv}(\{p_1^0, \dots, p_{n+1}^0\})$. Если A – это сумма симплекса и шара, то искомая оценка справедлива. Тогда и для данного A эта оценка справедлива.

Таким образом, показано, что все условия теоремы 1 выполнены, причем $\alpha = 2$. Тогда в этой задаче присутствует квадратичная сходимость алгоритма 1.

Доказательство теоремы 3. Определим решетку Λ , порожденную $v_i - v_{n+1}$, $1 \leq i \leq n$:

$$(P.22) \quad \Lambda = \{c_1(v_1 - v_{n+1}) + \dots + c_n(v_n - v_{n+1}) \mid c_i \in \mathbb{Z}\}.$$

Объем элементарной ячейки этой решетки равен $n! \text{Vol}(A)$.

Рассмотрим диаграмму Вороного для Λ . Пусть C – ячейка диаграммы Вороного, содержащая нуль.

$$(П.23) \quad C = \left\{ x \in \mathbb{R}^n \mid (x, y) \leq \frac{1}{2}(y, y) \forall y \in \Lambda \setminus \{0\} \right\}.$$

Можно заметить, что

$$(П.24) \quad F = \left\{ x \in \mathbb{R}^n \mid (x, v_i - v_j) \leq \frac{1}{2}(v_i - v_j, v_i - v_j) \forall i, j \right\}.$$

Поскольку $v_i - v_j \in \Lambda$, набор линейных ограничений, определяющих F , является подмножеством множества линейных ограничений, определяющих C . Тогда $C \subset F$, и

$$(П.25) \quad \text{Vol}(F) \geq n! \text{Vol}(A)$$

следует немедленно.

Если $\text{Vol}(F) = \text{Vol}(C) = n! \text{Vol}(A)$, то $F = C$ и F замощает \mathbb{R}^n , поскольку C замощает \mathbb{R}^n .

Доказательство теоремы 6.

1. Если для каждого j существует такой $p_j \in S^{n-1}$, что для любого $i \neq j$

$$(П.26) \quad h_{C_j}(p_j) + (v_j, p_j) \geq h_{C_i}(p_j) + (v_i, p_j),$$

то $x \in F$. Действительно, каждому минимуму $f_{t,x}(p)$ соответствует свой C_j (см. доказательство теоремы 2, раздел 3), и для того, чтобы C_j имел связанный с ним минимум, достаточно выполнения $(x, p_j) - h_{v_j+C_j}(p_j) \leq (x, p_j) - h_{v_i+C_i}(p_j)$ для некоторого p_j , что эквивалентно (П.26).

2. Рассмотрим предыдущий пункт для $p_j = \frac{v_j - x}{\|v_j - x\|}$. Тогда если для каждого j и любого $i \neq j$

$$(П.27) \quad h_{C_j}(v_j - x) + (v_j, v_j - x) \geq h_{C_i}(v_j - x) + (v_i, v_j - x),$$

тогда $x \in F$.

3.

$$(П.28) \quad 0 \leq h_{C_j}(p) \leq r,$$

поскольку $C_j \subset B_r(0)$. Это дает следующее.

4. Если для каждых j и $i \neq j$

$$(П.29) \quad (v_j, v_j - x) \geq (v_i, v_j - x) + r,$$

то $x \in F$.

5. Сравним:

$$(П.30) \quad \{x \in \mathbb{R}^n \mid (v_i - v_j, x) - (v_i - v_j, v_j) \geq r\} \subset F,$$

$$(П.31) \quad \{x \in \mathbb{R}^n \mid (v_i - v_j, x) - (v_i - v_j, v_j) \geq 0\} = \tilde{F}.$$

Если $x \in \tilde{F}$, то $x + r \frac{v_i - v_j}{\|v_i - v_j\|^2} \in F$. Тогда

$$(П.32) \quad F + B_{r/d}(0) \supset \tilde{F},$$

что и требовалось.

СПИСОК ЛИТЕРАТУРЫ

1. *Балашов М.В.* Покрытие множества выпуклым компактом: оценки погрешности и вычисление // Матем. заметки. 2022. Т. 112. № 13. С. 337–349.
2. *Поляк Б.Т.* Введение в оптимизацию. М.: Наука, 1983.
3. *R. Tyrrell Rockafellar.* Convex Analysis. Princeton University Press, 1997.
4. *Botkin N., Turova V.* An algorithm for finding the Chebyshev center of a convex polyhedron // Appl. Mat. Optim. 1995. V. 29. P. 211–222.
5. *Xia Y., Yang M.* Chebyshev center of the intersection of balls: complexity, relaxation and approximation // Mathematical Programming. 2021. V. 187. P. 287–315.
6. *Frankowska H., Olech C.* R-convexity of the integral of set-valued functions // Contribut. Anal. Geometry. 1980. V. 117–129.
7. *Vial J.-Ph.* Strong and Weak Convexity of Sets and Functions // Mathematics of Operations Research. 1983. V. 8. No. 2. P. 231–259.
8. *Boyd S., Vandenberghe L.* Convex optimization. Cambridge University Press, 2004.
9. *Li X., Zhu Z., Man-Cho S., Lee J.D.* Incremental Methods for Weakly Convex Optimization // arXiv, 2019. V. 1907.11687v1.
10. *Davis D., Drusvyatskiy D., MacPhee K.J., Paquette C.* Subgradient Methods for Sharp Weakly Convex Functions // arXiv, 2018. V. 1803.02461v1.
11. *Schneider R., Uschmajew A.* Convergence results for projected line search methods on varieties of low-rank matrices via Lojasiewicz inequality // SIAM J. Optim. 2015. V. 25. No. 1. P. 622–646.
12. *Balashov M.V.* About the Gradient Projection Algorithm for a Strongly Convex Function and a Proximally Smooth Set // J. Convex Anal. 2017. V. 24. No. 2. P. 493–500.
13. *Bello-Cruz Y., Li G., Nghia T.T.A.* On the Linear Convergence of Forward-Backward Splitting Method: Part I – Convergence Analysis // J. Optim. Theory Appl. 2021. V. 188. P. 378–401.
14. *Ioffe A.D.* Metric regularity – a survey Part I // J. Austral. Math. Soc. 2016. V. 101. P. 1–56.
15. *Absil P.-A., Mahony R., Sepulchre R.* Optimization Algorithms on Matrix Manifolds. Princeton University Press, 2008.

16. *Cen X., Xia Y., Gao R., Yang T.* On Chebyshev Center of the Intersection of Two Ellipsoids // Optimization of Complex Systems: Theory, Models, Algorithms and Applications. Springer International Publishing, 2020. P. 135–144.
17. *Beltran F., Finardi E.C., Fredo G.M., Oliveira W.* Improving the performance of the stochastic dual dynamic programming algorithm using Chebyshev centers // Optim. Engineer. 2022. V. 23. P. 147–168.
18. *Beck A., Eldar Y.C.* Regularization in Regression with Bounded Noise: A Chebyshev Center Approach // SIAM J. Matrix Anal. Appl. 2007. V. 29. No. 2. P. 606–625.
19. *Cerone V., Piga D., Regruto D.* Set-Membership Error-in-Variables Identification Through Convex Relaxation Techniques // IEEE Transact. Autom. Control. 2012. V. 57. No. 2. P. 517–522.
20. *Hou J., Teng F., Yin W., Song Y., Hou Y.* A Cost-Effective Cyber-Defense Strategy: Attack-Induced Region Minimization and Cybersecurity Margin Maximization // arXiv. 2023. V. 2302.07597.
21. *Samadi S., Roux J., Tanguy A., Caron S., Kheddar A.* Some journal publication in English // IEEE Robot. Autom. Lett. 2021. V. 6. No. 2. P. 4032–4039.
22. *Ren X., Mo Y., Chen J., Johansson K.H.* Secure state estimation with byzantine sensors: A probabilistic approach // IEEE Transact. Autom. Control. 2020. V. 65. No. 9. P. 3742–3757.

Статья представлена к публикации членом редколлегии П.С. Щербаковым.

Поступила в редакцию 25.01.2024

После доработки 12.03.2024

Принята к публикации 20.03.2024