

Методы отсекающей гиперплоскости

Максимова Анна Михайловна

9 января 2025 г.

Аннотация

Методы отсекающей гиперплоскости (Cutting-Plane Methods) — класс методов решения общих задач выпуклой и квазивыпуклой оптимизации, основанных на использовании отсекающих плоскостей, которые представляют собой гиперплоскости, отделяющие текущую точку от оптимальных точек.

В ходе данной работы будут рассмотрены общая постановка задачи, основная идея решения и два алгоритма: метод центров тяжести и метод эллипсоидов. Так же будет приведена реализация этих алгоритмов на одном из языков программирования.

1 План

1. Введение
2. Необходимые определения
3. Постановка задачи
4. Основная идея решения
5. Методы выбора точки для построения отсекающей гиперплоскости
 - (a) Метод центров тяжести
 - (b) Метод эллипсоидов
6. Реализация
7. Литература

2 Введение

В этом проекте рассматривается решение задачи выпуклого программирования, которая имеет следующий вид:

$$\min_x \{f_0(x) : f_i(x) \leq 0, i = 1, \dots, m, x \in G \subset \mathbb{R}^n\} \quad (1)$$

, где

- G - область определения задачи - является замкнутым выпуклым множеством в \mathbb{R}^n ,
- рассматриваемая функция f_0 и ограничения $f_i, i = 1, \dots, m$, являются выпуклыми функциями на \mathbb{R}^n (для простоты, в дальнейшем предполагается, что области этих функций - весь \mathbb{R}^n)

Речь пойдет о методах отсекающих гиперплоскостей. Эти методы существенно отличаются от методов внутренних точек. Они обычно менее эффективны для задач, к которым применимы методы внутренних точек, но имеют ряд преимуществ, которые могут сделать их выгодными в определенных ситуациях.

- Методы отсекающих гиперплоскостей не требуют дифференцируемости минимизируемых функций и ограничений и могут напрямую решать как квазивыпуклые, так и выпуклые задачи. Каждая итерация требует вычисления субградиента минимизируемой функции или функции ограничения.
- Методы отсекающих гиперплоскостей могут использовать определенные типы структуры в больших и выпуклых задачах. Метод отсекающих гиперплоскостей, использующий структуру, может быть быстрее, чем общий метод внутренних точек для той же задачи.
- Методы отсекающих гиперплоскостей не требуют оценки минимизируемой функции и всех функций ограничений на каждой итерации. (В отличие от этого, методы внутренних точек требуют оценки всех функций, а также их первые и вторые производные). Это делает данный тип решения полезным для задач с очень большим числом ограничений.
- Методы отсекающих гиперплоскостей могут использоваться для декомпозиции задач на более мелкие, которые можно решать последовательно или параллельно.

Чтобы применять эти методы к недифференцируемым задачам, необходимо знать о субградиентах используемых функций.

3 Необходимые определения

Выпуклая функция

Определение 1. Выпуклая функция - функция, надграфик или подграфик которой является выпуклым множеством. Формально, для числовой функции на некотором интервале (в общем случае — на выпуклом подмножестве некоторого векторного пространства) выпуклость (вниз) можно определить как выполнение неравенства Йенсена — если для любых двух значений аргумента x, y и для любого числа $t \in [0, 1]$ имеет место:

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) \quad (2)$$

Субградиент функции

Определение 2. В задачах выпуклой оптимизации нет необходимости в дифференцируемости функции, вместо этого роль градиентов выполняют субградиенты функции. Если g - выпуклая функция на \mathbb{R}^n , то в каждой точке $x \in \mathbb{R}^n$ существует по крайней мере один субградиент $g'(x)$ функции g - такой вектор, что

$$g(y) \geq g(x) + (y - x)^T g'(x) \quad (3)$$

Геометрически: график линейной функции

$$y \vdash g(x) + (y - x)^T g'(x)$$

находится везде под графиком g самой функции, а в точке x оба графика касаются друг друга. Вообще говоря, субградиент g в точке x не является единственным, он единственный тогда и только тогда, когда g дифференцируема в точке x , и в этом случае субградиент g в точке x - это как раз градиент $\nabla g(x)$ g в точке x .

Говоря о методах решения (1), предполагается, что у нас всегда есть оракул первого порядка, т.е. процедура, которая, получая на вход $x \in \mathbb{R}^n$, возвращает на выходе значения $f_i(x)$ и некоторые субградиенты $f'_i(x), i = 1, \dots, m$ минимизируемой функции и ограничений в точке x

Отсекающая гиперплоскость (cutting/separating plane)

Задача методов отсекающей гиперплоскости состоит в том, чтобы найти точку на выпуклом множестве $X \subseteq \mathbb{R}^n$, которое называется целевое множество (target set), или, в некоторых случаях, определить, что X пусто. В задаче оптимизации целевое множество X можно рассматривать как множество оптимальных (или ϵ -субоптимальных) точек для данной задачи, и главная цель - найти оптимальную (или ϵ -субоптимальную) точку для оптимизационной задачи.

Чаще всего не имеется прямого доступа к какому-либо описанию целевого множества X , кроме как через оракул. Запрашивая оракул в точке $x \in \mathbb{R}^n$,

оракул возвращает следующую информацию: либо сообщает что $x \in X$, либо возвращает разделяющую гиперплоскость между x и X , то есть $a \neq 0$ и b такие что

$$a^T z \leq b \quad \forall z \in X, \quad a^T x \geq b$$

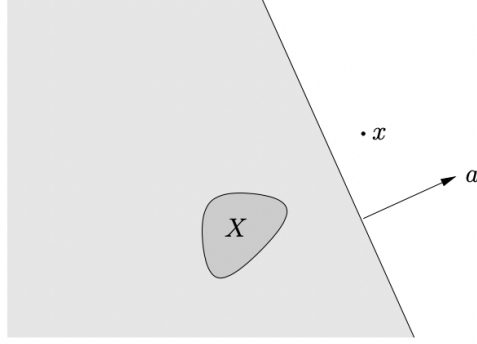


Рис. 1: Разделяющая гиперплоскость, когда $x \notin X$

Определение 3. Гиперплоскость, описанная выше, называется отсекающей или разделяющей гиперплоскостью, поскольку она «разрезает» или устаняет полупространство $\{z | a^T z > b\}$ из поиска. Ни одна такая точка не может находиться в целевом множестве X . Иллюстрацию данного определения можно увидеть на рисунке 1. Предполагается, что $\|a\|_2 = 1$, потому что деление a и b на $\|a\|_2$ определяет ту же гиперплоскость

Когда разделяющая гиперплоскость $a^T z = b$ содержит рассматриваемую точку x , она называется нейтральной. Когда $a^T z > b$, то это означает, что x лежит во внутренней части полупространства, которое выкидывается из рассмотрения, и эта ситуация называется глубоким разрезом. Интуиция подсказывает, что глубокий разрез лучше, т.е. более информативнее, чем нейтральный разрез, поскольку исключает из рассмотрения больший набор точек.

4 Постановка задачи

В этом разделе рассматривается поиск отсекающих гиперплоскостей для нескольких стандартных задач выпуклой оптимизации. Целевое множество X является оптимальным множеством для данной задачи, поэтому оракул должен либо объявить точку x оптимальной, либо вывести гиперплоскость, отделяющую x от оптимального множества.

Минимизация без ограничений

Сначала рассмотрим задачу без ограничений:

$$\text{minimize } f_0(x), \quad (4)$$

где f_0 - выпуклая функция. Для того, чтобы найти разделяющую гиперплоскость для данной проблемы, сначала необходимо найти субградиент $g \in \partial f(x)$. (если f дифференцируема в точке x , то выбором будет $g = \nabla f(x)$) Если $g = 0$, то x - точка минимума, то есть лежит в целевом множестве X . Предположим, что $g \neq 0$. Вспомним, что для всех z мы имеем:

$$f_0(z) \geq f_0(x) + (z - x)^T g$$

по определению субградиента. Поэтому, если z удовлетворяет $(z - x)^T g > 0$, тогда выполняется и $f_0(z) > f_0(x)$, и тем самым не может лежать в X .

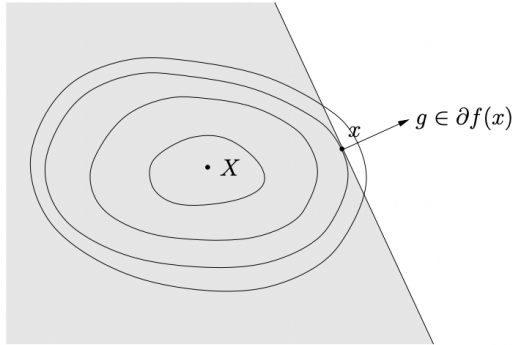


Рис. 2: Линии уровня функции f_0 . Здесь оптимальное множество X - это просто точка, минимум f_0

Другими словами, выполняется следующее неравенство:

$$(z - x)^T g \leq 0, \quad z \in X$$

и $(z - x)^T g = 0$ для $z = x$. Это показывает, что $(z - x)^T g \leq 0$ - нейтральная разделяющая гиперплоскость для (4) в точке x

Интерпретация следующая: во время поиска точки минимума можно удалить полупространство $\{z | (z - x)^T g > 0\}$ из рассмотрения, поскольку все точки в нем имеют значение больше, чем в точке x , и поэтому не могут быть оптимальными. Это показано на рисунке 2.

Можно генерировать глубокий разрез, если мы знаем номер \bar{f} , который удовлетворяет

$$f_0(x) > \bar{f} \geq f^*,$$

где $f^* = \inf_x f_0(x)$ - решение задачи (4). В этом случае любая точки минимума x^* должна удовлетворять

$$\bar{f} \geq f^* \geq f_0(x) + (x^* - x)^T g,$$

поэтому глубокий разрез имеет следующий вид:

$$(z - x)^T g + f_0(x) - \bar{f} \leq 0$$

Когда задача оптимизации (4) квазивыпуклая (то есть f_0 = квазивыпуклая), можно найти отсекающую гиперплоскость с помощью ненулевого квазиградиента f_0 в точке x . По сути, по определению неравенство $(z - x)^T g \leq 0$ является разделяющей гиперплоскостью, если g - ненулевой квазиградиент f в точке x

Задача существования (feasibility problem)

Определяется как

$$\text{find } x \quad (5)$$

$$\text{subject to } f_i(x) \leq 0, \quad i = 1, \dots, m \quad (6)$$

где f_i - выпуклые. Здесь целевое множество X - допустимое множество. Для того, чтобы найти отсекающую гиперплоскость в этой задаче в точке x проверим, что x - допустимая точка (то есть удовлетворяет $f_i(x) \leq 0$ для $i = 1, \dots, m$, тогда $x \in X$. Теперь предположим, что x - не допустимая. Это означает, что существует как минимум один индекс j для которого $f_j(x) > 0$ (то есть x нарушает j -ое ограничение). Пусть $g_j \in \partial f_j(x)$. Из неравенства

$$f_j(z) \geq f_j(x) + (z - x)^T g_j$$

можно заключить, что если

$$f_j(x) + (z - x)^T g_j > 0,$$

тогда $f_j(z) > 0$, и поэтому z также нарушает j -ое ограничение. Отсюда следует, что любое допустимое z удовлетворяет неравенству

$$f_j(x) + (z - x)^T g_j \leq 0,$$

из чего получается необходимая разделяющая гиперплоскость. Так как $f_j(x) > 0$, то это глубокий разрез.

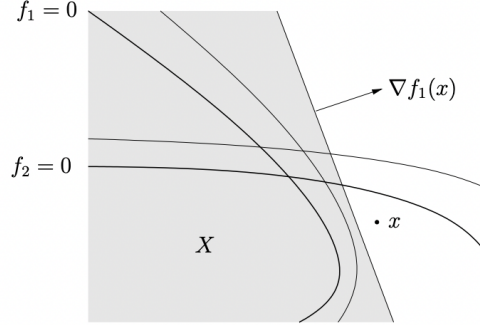


Рис. 3: Линии уровня двух выпуклых функций f_1 и f_2 . Здесь оптимальное множество X определено через $f_1 \leq 0$ и $f_2 \leq 0$, и в точке x одно из условий нарушается

В этом случае можно удалить из рассмотрения полупространство, полученное через неравенство $f_j(x) + (z - x)^T g_j > 0$, потому что все точки в нем нарушают j -ое ограничение (как и x), и, следовательно, недопустимы. Такая ситуация проиллюстрирована на рисунке 3.

Также, можно найти разделяющую гиперплоскость для каждого нарушенного ограничения, поэтому, если в точке x нарушено более одного ограничения, можно сгенерировать несколько отсекающих гиперплоскостей, отделяющих x от X .

Задача минимизации с условиями (Inequality constrained problem)

Объединяя два метода, описанных выше, можно найти разделяющую гиперплоскость в следующей задаче

$$\text{minimize } f_0(x) \quad (7)$$

$$\text{subject to } f_i(x) \leq 0, \quad i = 1, \dots, m \quad (8)$$

где f_0, \dots, f_m - выпуклые. Как и выше, целевое множество - оптимальное множество.

Для данной точки x сначала нужно проверить допустимость для ограничений. Если x недопустима, тогда можно построить разделяющую гиперплоскость

$$f_j(x) + (z - x)^T g_j \leq 0, \quad (9)$$

, где $f_j(x) > 0$ (то есть j - это индекс какого то нарушенного условия) и $g_j \in \partial f_j(x)$. Это определяет гиперплоскость для задачи (7-8), поскольку любая оптимальная точка должна удовлетворять j -му неравенству, а значит, и линейному неравенству (9). Разрез (9) называется допустимым разрезом для задачи (7-8), поскольку мы выкидываем из рассмотрения полуплоскость точек, которые недопустимы (поскольку они нарушают j -ое

ограничение).

Теперь предположим, что искомая точка x допустима. Найдем $g_0 \in \partial f_0(x)$. Если $g_0 = 0$, то x - точка минимума, и мы закончили. Предположим, что $g_0 \neq 0$. В этом случае мы можем построить разделяющую гиперплоскость с помощью неравенства:

$$(z - x)^T g_0 \leq 0$$

которая называется целевой разделяющей гиперплоскостью для проблемы (7-8). Здесь можно отрезать полупространство $\{z | (z - x)^T g_0 > 0\}$, потому что известно, что во всех таких точках значение функции больше, чем в x , и, следовательно, они не могут быть точками минимума.

5 Основная идея

В этом разделе рассматривается идея решения задачи оптимизации следующего типа (то есть минимизации без ограничений):

$$\min_x \{f(x) : x \in G \subset \mathbb{R}^n\} \quad (10)$$

- минимизация выпуклой функции f на замкнутом и ограниченном выпуклом непустом множестве G . В дальнейшем можно легко показать, как от такой задачи легко перейти к задаче минимизации с условиями.

Идея решения

Идея очень проста: возьмем произвольную внутреннюю точку $x_1 \in G$ и вычислим субградиент $f'(x_1)$ в этой точке, так что

$$f(x) - f(x_1) \geq (x - x_1)^T f'(x_1) \quad (11)$$

Если $f'(x_1) = 0$, тогда x_1 - точка глобального минимума функции f на \mathbb{R}^n , и так как этот минимум принадлежит G , то эта точка является решением задачи (10), и алгоритм завершается. Теперь пусть $f'(x_1) \neq 0$ и искомое оптимальное множество обозначается через X^* .

Из (11) следует, что если x принадлежит открытому полупространству

$$\Pi_1^+ = \{x | (x - x_1)^T f'(x_1) > 0\},$$

тогда правая часть неравенства (11) положительная в точке x , и x - не является оптимальным. (в (11) показывается, что $f(x) > f(x_1)$, поэтому значение в x больше чем в допустимом решении x_1) Следовательно, для оптимального множества (о котором мы изначально знаем только то, что $X^* \subset G$) мы можем определить новое локализирующее множество:

$$G_1 = \{x \in G | (x - x_1)^T f'(x_1) \leq 0\}$$

Это множество опять является замкнутым и ограниченным выпуклым непустым множеством (так как пересечение G и закрытого полупространства с границей, проходящей через внутреннюю точку x_1 из G) и меньше, чем G (так как внутренняя точка x_1 множества G - это граничная точка G_1)

Таким образом, выбрав каким-то образом внутреннюю точку x_1 в «начальном локализаторе оптимального множества» $G \equiv G_0$ и смотря на $f'(x_1)$, мы либо завершаем работу алгоритма с решением x_1 , либо можем разрезать G_0 с помощью отсекающей гиперплоскостью и работать уже с меньшим множеством G_1 , которое также содержит оптимальное множество. В этом последнем случае мы можем повторить те же действия - выбрать каким-то образом внутреннюю точку x_2 в G_1 , вычислить $f'(x_2)$ и, в случае $f'(x_2) \neq 0$, определить разделяющую гиперплоскость и заменить с помощью нее G_1 на меньшее локализирующее множество G_2 :

$$G_2 = \{x \in G_1 | (x - x_2)^T f'(x_2) \leq 0\},$$

и так далее. С помощью полученной рекурсии мы либо завершаем работу на определенном шаге с точным решением, либо генерируем последовательность множеств такую, что

$$G = G_0 \supset G_1 \supset G_2 \supset \dots$$

и каждое из этих множеств содержит искомое оптимальное множество X^* . Можно предположить, что если x_t выбраны правильно, тогда локализирующие множества G_t стремятся к X^* с определенной «разумной скоростью», и G_t сходятся к X^* в итоге.

В многомерном случае ситуация гораздо сложнее:

- Неясно, что означает «центр» G_t , т.е. как выбрать $x_{t+1} \in G_t$, чтобы добиться «значительного сужения» множества на шаге (кстати, как измерить это сужение?) С другой стороны, ясно, что если x_{t+1} выбран неудачно (например, близко к границе G_t) и новый разрез «плохо ориентирован», то следующее множество G_{t+1} может быть почти таким же большим, как G_t .
- В одномерном случае линейные размеры локализирующих множеств стремятся к 0 линейно с соотношением $\frac{1}{2}$, и это и есть скорость сходимости x_t к минимизатору f . Совершенно ясно, что в многомерном случае мы не можем заставить линейные размеры G_t стремиться к 0 (например, если $G = G_0$ - единичный квадрат на плоскости и f не зависит от первой координаты, в этом случае все множества G_i будут прямоугольниками одинаковой ширины, и, следовательно, мы не можем заставить их линейные размеры сходить к 0). Таким образом, мы должны тщательно продумать, как измерять размеры множеств локализаторов - при плохом выборе понятия размера мы не сможем привести его к 0.

Итоговый алгоритм выглядит следующим образом:

На входе некоторое замкнутое и ограниченное выпуклое непустое множество $G \equiv G_0$, содержащее искомое оптимальное множество X^*

$t = 0$

repeat

 Выбираем точку x_{t+1} в G_t

 Определяем новую разделяющую гиперплоскость и новое локализирующее множество $G_{t+1} = \{x \in G_t | (x - x_{t+1})^T f'(x_{t+1}) \leq 0\}$

$t = t + 1$

until $f'(x_{t+1}) \neq 0$ или $G_{t+1} \neq \emptyset$

6 Метод центров тяжести

Самое нетривиальное в алгоритме выше - выбрать новую точку x_{t+1} и способ оценивания размера множества для того, чтобы получить адекватную скорость сходимости. В этом разделе будет рассмотрен метод центров тяжести, который помогает ответить на предыдущие вопросы.

В методе центров тяжести размеры локализующих множеств определяются как их n -мерные объемы $Vol(G_t)$, а в качестве x_{t+1} используется центр тяжести G_t :

$$x_{t+1} = \frac{1}{Vol(G_t)} \int_{G_t} x dx$$

Геометрический факт (очень нетривиальный) состоит в том, что при таком методе мы получаем линейную сходимость объемов G_t к 0:

$$Vol(G_t) \leq \left[1 - \left(\frac{n}{n+1} \right)^n \right]^t Vol(G_0),$$

что, в свою очередь, подразумевает линейную сходимость к минимуму функции f : если x^t является наилучшим (функция f принимает наименьшее значение в этой точке) из точек x_1, \dots, x_t , то

$$f(x^t) - \min_G f \leq \left[1 - \left(\frac{n}{n+1} \right)^n \right]^{\frac{t}{n}} \left[\max_G f - \min_G f \right] \quad (12)$$

Неравенство выше демонстрирует глобальную линейную сходимость метода центров тяжести с коэффициентом, не зависящим от x :

$$\kappa(n) = \left[1 - \left(\frac{n}{n+1} \right)^n \right]^{\frac{1}{n}} \leq (1 - \exp(-1))^{\frac{1}{n}}$$

Следовательно, чтобы получить ϵ -решение задачи - найти точку $x^t \in G$ такую, что

$$f(x^t) - \min_G f \leq \epsilon [\max_G f - \min_G f]$$

- требуется выполнить не более

$$\left\lceil \frac{\ln \frac{1}{\epsilon}}{\ln \frac{1}{\kappa(n)}} \right\rceil \leq 2.13n \ln \frac{1}{\epsilon} + 1 \quad (13)$$

шагов за метод.

В результате, глобальная оценка сходимости не зависит от значений функции в точках: не важно насколько может измениться выходное значение функции при небольшом изменении входного аргумента. f может быть негладкой, и т.д.

В общем случае, метод центров тяжести в определенных ситуациях является оптимальным методом для выпуклой оптимизации.

Слабым местом метода является необходимость на каждом шаге находить

центр тяжести предыдущего локализирующего множества. Нахождение центра тяжести n -мерного множества G - сложная с вычислительной точки зрения задача. Конечно, если G (а затем и каждый G_t) является многогранником, эта проблема «алгоритмически разрешима» - можно легко предъявить метод, решающий проблему за конечное число арифметических операций. К сожалению, количество арифметических операций требуемых всеми известными методами для нахождения центра тяжести, растет экспоненциально с n , так что уже в размерности 5-10 сложно вычислить центр тяжести за разумное время. В результате, метод центров тяжести представляет только «академический интерес» - его нельзя использовать в качестве вычислительного инструмента.

7 Метод эллипсоидов

Метод эллипсоидов можно рассматривать как некоторую вычислительную аппроксимацию метода центров тяжести. Идея заключается в том, представить все локализирующие множества через красивую геометрию, удобную для нахождения центров тяжести. Для такой цели как раз подходят эллипсоиды. Предположим, что $G_0 = G$ - эллипсоид. Тогда, не сложно найти центр G_0 , что упрощает первый шаг метода центров тяжести. К сожалению, полученное после первого шага локализирующее множество, пусть оно называется G_1^+ не является эллипсоидом, а будет «полуэллипсоидом» - то есть пересечением эллипсоида G_0 и полупространства с граничной гиперплоскостью, проходящей через центр G_0 . Такое множество получается не очень удобным для дальнейших шагов. Чтобы восстановить красивую геометрию множеств, подберем для полуэллипсоида G_1^+ эллипсоид меньшего объема, чем G_0 , назовем его G_1 (он будет содержать G_1^+), и далее будем использовать G_1 . Заметим, что G_1 действительно является локализирующим множеством оптимального множества X^* (поскольку последнее уже содержится в G_1^+ , а G_1 содержит в себе G_1^+). Теперь мы находимся в той же ситуации, как в самом начале, но заменили G_0 на более меньшее по объему множество G_1 , и можем повторить те же действия: взять центр x_2 эллипсоида G_1 , получить новую разделяющую гиперплоскость и через нее выразить полуэллипсоид G_2^+ , а далее получить эллипсоид G_2 меньшего объема. Таким образом, мы фактически предоставляем своеобразный компромисс между эффективностью алгоритма и вычислительной сложностью шага: увеличивая в объеме локализирующие множества до эллипсоидов, мы добавляем точки, которые наверняка не будут точками минимум, и тем самым замедляем работу алгоритма. Но благодаря такому замедлению уменьшается вычислительная сложность шага за счет работы с простыми геометрическими множествами - эллипсоидами.

Но есть пару моментов, которые следует уточнить:

- Во-первых, заранее неясно, действительно ли мы можем линейно уменьшать объемы последовательных эллипсоидов и получить сходящийся метод - может случиться так, что при расширении полуэллипсоида G_{t+1}^+ до эллипсоида G_{t+1} , мы вернемся к предыдущему эллипсоиду G_t , и сходимости не будет. К счастью, это не так: во время алгоритма с каждым шагом объем эллипсоидов уменьшается на коэффициент $\kappa^*(n) \leq \exp(-\frac{1}{2n})$, следовательно, объемы эллипсоидов стремятся к 0 с линейной скоростью (такое получается благодаря геометрическим свойствам эллипсоидов). Этот коэффициент хуже, чем в методе центров тяжести), но все равно имеется линейная сходимость
- Во-вторых, предполагается, что G - эллипсоид. Что делать, если это не так? Можно выбрать в качестве G_0 произвольный эллипсоид, который содержит исходное множество G , и, соответственно, будет содержать X^* . Но есть нюанс: может случиться, что центр x_1 эллипсоида G_0 находится вне G , тогда непонятно как выбрать отсекающую

гиперплоскость. Более того, эта плохая ситуация - когда центр x_{t+1} текущего эллипсоида G_t находится вне G - может в конечном итоге возникнуть даже тогда, когда $G_0 = G$ - эллипсоид, так как на каждом добавляется какое то множество лишних точек, чтобы в итоге получить новый эллипсоид, и эти точки могут не принадлежать исходному множеству G . В результате, G_t и его центр x_{t+1} , может не содержаться в G .

Можно решить эту проблему следующим образом. Имея предыдущий эллипсоид G_t и его центр x_{t+1} , воспользуемся оракулом, чтобы узнать: $x_{t+1} \in G$. Если да, то нет никаких проблем с разделяющей гиперплоскостью - мы вызываем оракул первого порядка, получаем субградиент f в x_{t+1} и используем его для получения разреза, как и делали раньше. Иначе, если x_{t+1} не находится в G , оракул вернет разделяющую гиперплоскость

$$e^T x_{t+1} > \max_{x \in G} e^T x_t$$

Следовательно, все точки $x \in G$ удовлетворяют неравенству

$$(x - x_{t+1})^T e < 0$$

и можно использовать e для получения следующей разделяющей гиперплоскости и множества

$$G_{t+1}^+ = \{x \in G_t | (x - x_{t+1})^T e \leq 0\}$$

Получаем, что разделяющая гиперплоскость, которая разделяет G_t , выдает в результате множество на G и, соответственно, G_{t+1}^+ содержит X^* .

Как математически представить эллипсоид

Эллипсоид - это геометрический объект, и чтобы начать алгоритм, нужно иметь дело с числовыми представлениями этих фигур. Наиболее удобным является представление эллипсоида как единичный евклидовы шар при несингулярном аффинном отображении:

$$E = E(c, B) = \{x = c + Bu | u^T u \leq 1\} \quad (14)$$

где $c \in \mathbb{R}^n$ - это центр эллипсоида и B - это $n \times n$ несингулярная матрица. Таким образом, говоря в дальнейшем про эллипсоид, мы будем иметь дело со множеством, представленным в виде пары (c, B) , состоящей из вектора $c \in \mathbb{R}^n$ и несингулярной $n \times n$ матрицы B .

Объем такого эллипсоида выражается как

$$Vol(E(c, B)) = |\det B| v(n)$$

где $v(n)$ - объем единичного n -мерного евклидова шара V_n ($E(c, B)$ является образом V_n при аффинном отображении)

Для дальнейшего анализа необходима следующая лемма

Лемма 1. Пусть $n > 1$ и положим

$$E = E(c, B) = \{x = c + Bu | u^T u \leq 1\}$$

- эллипсоид в \mathbb{R}^n и пусть

$$E^+ = \{x \in E | (x - c)^T e \leq 0\} \quad [e \neq 0]$$

- будет полуэллипсоидом (то есть пересечение E и полупространства с границей, проходящей через центр эллипсоида). Тогда E^+ можно вмести в эллипсоид $E' = E(c', B')$ следующего объема:

$$\text{Vol}(E') = \kappa^*(n) \text{Vol}(E)$$

$$\kappa^*(n) = \frac{n^2}{n^2 - 1} \sqrt{\frac{n-1}{n+1}} \leq \exp\left(-\frac{1}{2(n-1)}\right) \quad (15)$$

Параметры c' и B' можно выразить как

$$B' = \alpha(n)B - \gamma(n)(Bp)p^T, \quad c' = c - \frac{1}{n+1}Bp \quad (16)$$

где

$$\alpha(n) = \left(\frac{n^2}{n^2 - 1}\right)^{\frac{1}{4}}, \quad \gamma(n) = \alpha(n) \sqrt{\frac{n-1}{n+1}}, \quad p = \frac{B^T e}{\sqrt{e^T B B^T e}}$$

Доказательство...

Итоговый алгоритм для метода эллипсоидов

Теперь можно предоставить итоговый алгоритм для метода эллипсоидов.

Имеется:

- G - замкнутое и ограниченное выпуклое непустое множество
- оракул первого порядка для f и разделяющий оракул для G
- определяем эллипсоид $G_0 = E(c_0, B_0)$, который содержит G

Инициализация: $t = 1$

Шаг t: На вход получаем $G_{t-1} = E(c_{t-1}, B_{t-1})$, и положим $x_t = c_{t-1}$

1. Обращаемся к разделяющему оракулу, x_t на входе. Если оракул вернул, что $x_t \in G$, шаг называется продуктивным и переходим к пункту 2). Иначе, шаг - непродуктивный и установим e_t через оракул следующим образом:

$$x \in G \Rightarrow (x - x_t)^T e_t < 0$$

и переходим к пункту 3

2. Обращаемся к оракулу первого порядка, x_t на входе. Если $f'(x_t) = 0$, это означает, что алгоритм завершен и $x_t \in G$ - точка минимума функции f на всем \mathbb{R}^n . Иначе, положим

$$e_t = f'(x_t)$$

и переходим к пункту 3

3. Обновляем эллипсоид $E(c_{t-1}, B_{t-1})$ на $E(c_t, B_t)$ по формуле (16), где $e = e_t$, то есть положим

$$B_t = \alpha(n)B_{t-1} - \gamma(n)(B_{t-1}p_t)p_t^T, \quad c_t = c_{t-1} - \frac{1}{n+1}B_{t-1}p_t$$

$$\alpha(n) = \left(\frac{n^2}{n^2-1}\right)^{\frac{1}{4}}, \quad \gamma(n) = \alpha(n)\sqrt{\frac{n-1}{n+1}}, \quad p_t = \frac{B_{t-1}^T e_t}{\sqrt{e_t^T B_{t-1} B_{t-1}^T e_t}}$$

и $t = t + 1$, и переходим к следующему шагу

Аппроксимация решения x_t , которая получена после t шагов метода, по определению получается лучшим решением задачи минимизации по сравнению с x_j , полученными на предыдущих шагах

Скорость сходимости

Скорость сходимости метода эллипсоидов можно выразить через теорему:

Теорема 1. Пусть $n > 1$ и оптимизационная задача (10) решена методом эллипсоидов. Положим

$$N(\epsilon) = \lfloor 2n(n-1) \ln\left(\frac{V}{\epsilon}\right) \rfloor$$

где $0 < \epsilon < 1$ и

$$V = \left[\frac{\text{Vol}(E(c_0, B_0))}{\text{Vol}(G)} \right]^{\frac{1}{n}}$$

Тогда, для любого $\epsilon \in (0, 1)$, аппроксимированное решение $x^{N(\epsilon)}$, найденное за первые $N(\epsilon)$ шагов, определено и является ϵ -решением задачи, то есть принадлежит G и удовлетворяет неравенству

$$f(x^{N(\epsilon)}) - \min_G f \leq \epsilon [\max_G f - \min_G f]$$

Доказательство...

Метод эллипсоидов для задач с ограничениями

Теперь можно обобщить методы, рассмотренные выше, на задачи оптимизации с ограничениями. Тогда можно переписать задачу следующим образом:

$$\min_x \{f(x) \equiv f_0(x) : x \in \hat{G} \equiv \{x \in G | f_i(x) \leq 0, i = 1, \dots, m\}\} \quad (17)$$

Для того, чтобы решить данную задачу, нужно

- переопределить оракул первого порядка и разделяющий оракул:
Оракул первого порядка подходит для новой задачи, а разделяющий оракул для множества \hat{G} можно легко опеределить через оракул для множества G и оракул первого порядка для старой задачи: для данного $x \in \mathbb{R}^n$ проверим сначала принадлежит ли точка G , если нет, то $x \notin \hat{G}$, и разделяющая гиперплоскость (полученная через разделяющий оракул для G) отделит x от \hat{G} . Если $x \in G$, то через оракул первого порядка получим значения и субградиенты для f_1, \dots, f_m в точке x . Если все значения отрицательные, то $x \in \hat{G}$, если одно из них положительное, то $x \notin \hat{G}$ и вектор $f'_i(x)$, где условие нарушается, может быть использован для построения разделяющей гиперплоскости x от \hat{G}
- убедиться, что \hat{G} - замкнутое и ограниченное выпуклое непустое множество

Теорема о сходимости в данной задаче остается справедливой.

8 Реализация

<https://github.com/anyatamax/Cutting-plane-method-optimization/tree/main>

9 Литература

Список литературы

- [1] LECTURE NOTES OPTIMIZATION III. Aharon Ben-Tal[†], Arkadi Nemirovski. Spring Semester 2023. URL: <https://www2.isye.gatech.edu/~nemirovs/OPTIIILN2023Spring.pdf>
- [2] Localization and Cutting-Plane Methods. S. Boyd and L. Vandenberghe. April 13, 2008. URL: https://see.stanford.edu/materials/lsoctee364b/05-localization_methods_notes.pdf
- [3] Выпуклая оптимизация. Е. А. Воронцова, Р. Ф. Хильдебранд, А. В. Гасников, Ф. С. Стонякин. МОСКВА МФТИ 2021. URL: <https://labmmo.ru/upload/000/u8/8/f/2106-01946.pdf>