Применение проективного покоординатного спуска в задаче Фекете

Ильяс Фатхуллин

Московский физико-технический институт

19.06.2019

Содержание

- Постановка задачи Фекете
- Метод первого порядка
- Метод второго порядка
- Быстрый счет функции и градиента
- Вычислительный эксперимент

Постановка задачи Фекете

Ограничение

$$M = \prod_{i=1}^{N} S_i^2 = \{\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^{3N} : x_i \in S_i^2 \text{- единичная сфера в } \mathbb{R}^3\} \tag{1}$$

Функция

$$U(\mathbf{x}) := \sum_{1 \le i < j \le N} U(x_i, x_j), \text{ где } U(x_i, x_j) - \text{ядро} \tag{2}$$

Задача оптимизации

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in M} U(\mathbf{x}), \tag{3}$$

Постановка задачи Фекете

Возможный вид ядра

$$U(x_i, x_j) = \frac{1}{||x_i - x_j||_2^s}, s > 0 \quad (4)$$

$$U(x_i, x_j) = -\log ||x_i - x_j||_2 \quad (5)$$

Что известно про задачу?

- Это невыпуклая задача с нелинейными ограничениями
- Аналитические решения при s = 1 для $N = 2 - 6$ и 12
- Для различных s и больших N получены приближенные решения с помощью численных методов
- Появляется в списке Смейла - нерешенных задач для 21-го века
- В силу своей сложности стала эталоном для тестирования численных методов оптимизации

Метод первого порядка

Обозначения

Фиксируем все точки $x_i \in S_i^2$, кроме одной k - ой. Обозначим градиент функции (2) в точке $y \in S_k^2$:

$$\nabla U_k(y, \boldsymbol{\alpha}_k), \tag{6}$$

где $\boldsymbol{\alpha}_k$ - вектор параметров, определяющий положение остальных точек.

Описание метода

Обновим k - ую компоненту x_k, сделав шаг:

$$x_k = \frac{x_k - h\nabla U_k(x_k, \boldsymbol{\alpha}_k))}{||x_k - h\nabla U_k(x_k, \boldsymbol{\alpha}_k)||}, \tag{7}$$

где $h > 0$ - длина шага.

Метод первого порядка

Общая структура метода

- Выбрать начальное приближение $\mathbf{x}^0$
- Пока не достигнут критерий останова на n-ой итерации:
- Выбрать индекс k точки, которую будем двигать
- Вычислить шаг h_n
- Вычислить градиент $\nabla U_k(x_k^n, \alpha_k^n))$ относительно k -ой точки
- Сделать шаг (7), обновив k-ую компоненту $\mathbf{x}^n$

Метод второго порядка

Идея

Разложим функцию $U_k(z, \alpha_k) : \mathbb{R}^3 \to \mathbb{R}$ зависящую только от координаты одной точки (остальные фиксированы) до второго порядка в окрестности точки x_k. Разложение будет иметь вид:

$$U_k(z, \alpha_k) = \mathbf{c}(\mathbf{x}) + \frac{1}{2}(\mathbf{A}(\mathbf{x})z, z) - \mathbf{b}^T(\mathbf{x})z \tag{8}$$

Подзадача

На каждом шаге обновляем одну компоненту, решая подзадачу:

$$x_i^{n+1} = \arg\min_{z \in S^2}\{\frac{1}{2}z^T\mathbf{A}(\mathbf{x}^n)z - \mathbf{b}^T(\mathbf{x}^n)z\} \tag{9}$$

Быстрый счет функции и градиента

Функция

Организуем подсчет функции так, что на каждом шаге это занимает линейное время (от N). Хранить в памяти матрицу (расстояний) значений ядра на парах $x_i, x_j, \forall i, j \in \overline{1, N}$. На каждой итерации (для i -ой точки) будем обновлять энергию следующим образом:

- Вычтем $N - 1$ слагаемое (те которые содержат индекс i) до выполнения шага
- Прибавим соответствующие слагаемые после выполнения шага.

Быстрый счет функции и градиента

Градиент

На каждой итерации храним и обновляем N градиентов $\nabla U_k(y, \alpha_k)$:
Пусть на предыдущей итерации двигалась точка с индексом j, тогда для обновления градиентов достаточно:

- Из каждого градиента $\nabla U_k(x_k, \alpha_k)$, $k \neq j$ вычесть $-\frac{s}{r_{jk}^{s+2}}(x_k - x_j)$ до выполнения итерации
- Прибавить такое же слагаемое после выполнения итерации.
- Обновить градиент для j-ой точки.

Также за линейное время (по N) можно выбрать компоненту с максимальной проекцией на касательную плоскость, то есть по критерию:

$$||(I - x_k x_k^T)\nabla U_k(x_k, \alpha_k)|| \quad (10)$$

Вычислительный эксперимент

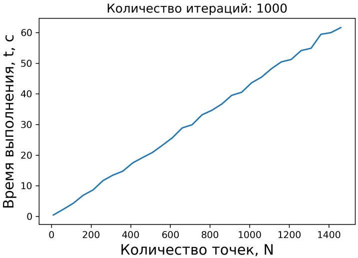

Рис.: Линейное время выполнения

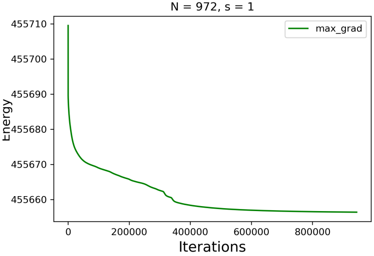

Рис.: Скорость сходимости, $N = 972, s = 1$

Вычислительный эксперимент

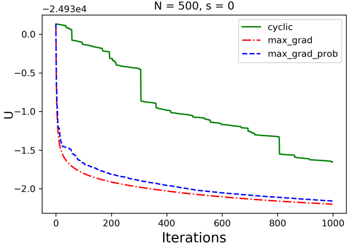

Рис.: Сравнение способов выбора точек для выполнения шага, $N = 500$, $s = 0$

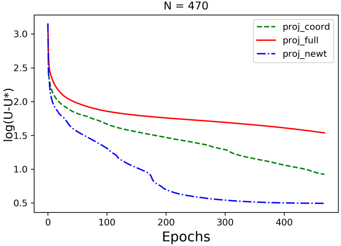

Рис.: Сравнение полноградиентного метода с покоординатными методами с градиентным и Ньютоновским шагами

Вычислительный эксперимент

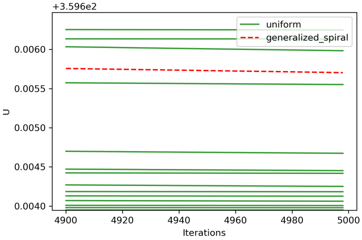

Рис.: Локальные минимумы для $N = 30, s = 1$, Метод 1 порядка

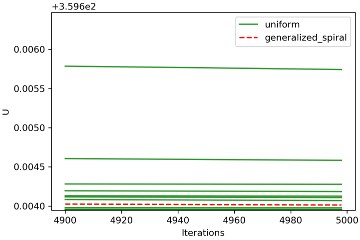

Рис.: Локальные минимумы для $N = 30, s = 1$, Метод 2 порядка

Вычислительный эксперимент

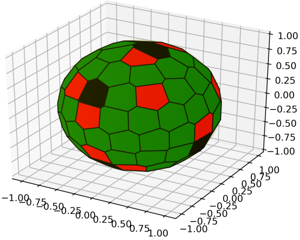

Рис.: Диаграмма Вороного до оптимизации, N = 100

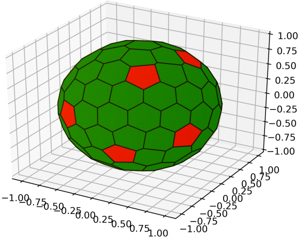

Рис.: Диаграмма Вороного после оптимизации, N = 100

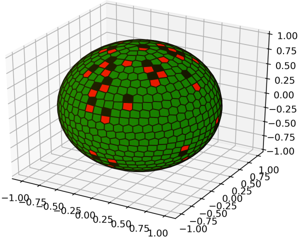

Рис.: Диаграмма Вороного до оптимизации, N = 972

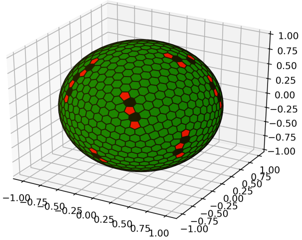

Рис.: Диаграмма Вороного после оптимизации, N = 972

Вычислительный эксперимент

Таблица: Результаты экспериментов при $s = 1$ для больших N

N	$U(\mathbf{x}^*)$	Минимум (метод 1)	Минимум (метод 2)	Итераций (1, 2)	Время (1), мин.	Время (2), мин.
100	4448.351	4448.493	4448.450	10000	0.7	2.0
200	18438.843	18439.644	18439.156	40000	5.3	9.6
312	45629.314	45630.736	45629.926	97344	20	42
400	75582.449	75584.100	75583.854	160000	42	85
470	104822.886	104825.398	104824.527	220900	85	150

$U(\mathbf{x}^*)$ - минимально известное значение из литературы

Заключение

Резюме

- Предложены два проективных покоординатных метода для поиска локального минимума в задаче Фекете.
- Алгоритмы используют быстрый счет функции, градиента и выбор наилучшей компоненты для оптимизации.
- Наши численные эксперименты показывают превосходство предложенных методов перед полноградиентным методом при количестве точек $N < 1000$.
- Можно ожидать, что предложенные методы окажутся предпочтительными для задач больших размерностей.