Лекция 7

Линейное программирование.

(Конспект: Ф. Александров)

7.1 Метод внутренней точки

Предъявим способ решения общей задачи линейного программирования.

$$\begin{aligned} &\min < c, x >, \qquad (7.1)\\ &Ax = b. \end{aligned}$$

Постановка задачи. Будем решать задачу вида

$$\begin{aligned} &\min < c, x > \stackrel{?}{=} 0,\\ &\Omega : \begin{cases} Ax = \mathbf{0}, \\ \sum_i x_i = 1, \\ x_i \geq 0 \end{cases} \qquad (7.2) \end{aligned}$$

причем известно, что $< c, x > \geqslant 0$ на Ω, и дана внутренняя точка $a \in \Omega$. Иначе говоря, надо выяснить, достигает ли $< c, x >$ нуля на Ω, являющемся пересечением подпространства $\{x | Ax = \mathbf{0}\}$ с *симплексом* $\{x | \sum_i x_i = 1,\ x_i \geq 0\}$. Очевидно, что можно рассматривать только случай, когда $c \in \Omega$.

Задача 7.1. Доказать, что задачи 7.1 и 7.2 эквивалентны.
Указание. Ознакомившись с этой лекцией, проделать необходимые проективные преобразования для того, чтобы загнать все внутрь симплекса и сделать уравнение $Ax = \ldots$ однородным. Добиться поиска конкретного минимума (именно, нуля) можно, комбинируя задачу 7.1 с двойственной.

Схема алгоритма.

Алгоритм 7.1 (Схема)**.** Алгоритм итеративный, при каждом шаге от точки a переходим к $\phi(a)$, и так до тех, пока не подберемся достаточно близко к решению.

Возьмём исходную для алгоритма точку a. Если $< c, a >= 0$, то останавливаемся, искомая точка найдена. Иначе переходим к $\phi(a)$. Проверяем значение $< c, \phi(a) >$, потом $\phi^2(a)$,$\phi^3(a)$,... Так постепенно "подбираемся" к решению. Количество переходов может быть большим, поэтому условия остановки будут приведены позднее.

Подобравшись "достаточно близко" к решению, отправимся, не увеличивая $< c, \cdot >$, к ближайшей вершине. В ней-то и будет решение. □

Условия остановки. Очевидно, что решение задачи 7.2, если оно существует, то находится в вершине области Ω.

Лемма 7.1. *Пусть a_*, b_* — вершины области Ω. Тогда*

$$|< c, a_* > - < c, b_* >| > 2^{-kL},$$

где L — общая длина входа алгоритма, k — вещественная константа.

Задача 7.2. Доказать лемму 7.1.

На основании этой леммы получаем признак того, что "подобрались" достаточно близко к решению. Если значение целевой функции $< c, \cdot >$ в точке a', полученной на данном шаге алгоритма, удовлетворяет неравенству

$$< c, a' > \leqslant 2^{-kL},$$

то "доехав" от a' до ближайшей вершины (не увеличивая при этом $< c, \cdot >$), получим решение (опять же, если оно существует).

Пусть

$$f(x) = \sum_i \ln\left(\frac{< c, x >}{x_i}\right).$$

На каждом шаге алгоритма будем вычислять $f(a)$. Значение этой фунции должно на каждом шаге убывать на некоторую константу. Если на каком-то шаге убывает слабее, то это означает, что задача решения не имеет, завершаем алгоритм.

Функция перехода. Теперь предъявим функцию перехода ϕ. Есть $a = (a_1, ..., a_n)^T$ — внутренняя точка области Ω, построим значение ϕ в этой точке. Рассмотрим матрицу

$$D_a = \begin{pmatrix} a_1 & 0 & \dots\dots\dots & & 0 \\ 0 & a_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & .. & 0 & a_{n-1} & 0 \\ 0 & \dots\dots & & 0 & a_n \end{pmatrix}$$

Преобразование

$$T(x) = \frac{D_a^{-1}x}{< e, D_a^{-1}x >}, \quad e = (1, 1, \dots, 1)$$

переводит внутреннюю точку во внутреннюю, а вершину симплекса — в вершину;

$$T(a) = a_0 = (1/n, \dots, 1/n).$$

В преобразованных координатах наше подпространство имеет вид $\{x' | ADx' = \mathbf{0}\}$, а целевая функция — $< c', \cdot >$, где $c' = Dc$. Наконец, функция f выглядит так:

$$f'(x') = \sum_i \ln(\frac{< c', x' >}{x'_i}) - \sum_i \ln a_i.$$

Далее, в симплекс вписываем шар $B(a_0, r)$ и рассмотрим шар $B(a_0, \alpha r)$, где $\alpha \in (0, 1)$ — параметр алгоритма. *Пересечение* $B' = B(a_0, \alpha r) \cap \{x' | ADx' = 0\}$ *этого шара с подпространством дает содержащийся в множестве, по которому мы оптимизируем, шар меньшей размерности и того же радиуса, ибо наше подпространство содержит его центр:* $ADa_0 = \frac{1}{n}Aa = 0$. В шаре оптимизировать очень просто: из точки a_0 сдвинемся на радиус этого шара в направлении вектора $(-c')$, получив точку a''.

Обратное к $T(x)$ преобразование выглядит так:

$$T^{-1}(x) = \frac{Dx}{< e, Dx >}.$$

Применив его к точке a'' (полученной только что сдвигом из a_0), найдём новую точку, которая и будет искомым $\phi(a)$. Всё, построение отображения ϕ закончили.

Доказательства. Наша цель — показать, что задача имеет решение тогда и только тогда, когда на каждом шаге значение функции f' будет убывать не менее, чем на некоторую константу. Тем самым мы ограничим количество шагов нашего алгоритма и получим оценку на время его работы. Начнем с того, что покажем, что наш шар меньшей размерности действительно содержит точку, в которой f' значительно меньше, чем в a_0.

Лемма 7.2. *Если задача (7.2) имеет решение, то* $\exists b' \in B' : \quad f'(b') \leq f'(a_0) - \delta$, *где* $\delta := \ln(1+\alpha) = \mathrm{const}$.

Доказательство. Пусть x^* — точка $T(\Omega)$, в которой достигается $\min < c', x >= 0$. Предъявим b': проведём отрезок, соединяющий a_0 и x^*. Пересечение этого отрезка и границы шара B' обозначим за b'.

Тогда $\exists \lambda \in (0,1) : \quad < c', b' >= (1-\lambda) < c', a_0 > + \lambda < c', x^* >$. Но $< c', x^* >= 0$, а значит,

$$\frac{< c', a_0 >}{< c', b' >} = \frac{1}{1-\lambda},$$

$$f'(a_0) - f'(b') = \sum_i \ln \frac{< c', a_0 >}{a_{0i}} - \sum_i \ln \frac{< c', b' >}{b'_i} =$$
$$\sum_i \ln \left(\frac{< c', a_0 >}{< c', b' >} \frac{b'_i}{a_{0i}} \right) = \sum_i \ln \left(\frac{1}{1-\lambda} \frac{(1-\lambda)a_{0i} + \lambda x^*_i}{a_{0i}} \right) =$$
$$\sum_i \ln \left(1 + \frac{\lambda}{1-\lambda} \frac{x^*_i}{a_{0i}} \right) \geq \ln \left(1 + \frac{\lambda}{1-\lambda} \sum_{i \in [1..n]} \frac{x^*_i}{a_{0i}} \right) = \ln \left(1 + \frac{\lambda n}{1-\lambda} \right).$$

Далее, так как $b' = (1-\lambda)a_0 + \lambda x^*$, имеем

$$\lambda = \frac{\text{длина отрезка } a_0 b'}{\text{длина отрезка } a_0 x^*} = \frac{\alpha r}{\text{длина отрезка } a_0 x^*} \geq \frac{\alpha r}{R},$$

где R — радиус шара, описанного вокруг симплекса.

Факт 7.1. *Пусть r — радиус вписанного в симплекс шара в $\mathbb{R}^n$, R — радиус описанного вокруг симплекса шара. Тогда*

$$\frac{r}{R} = \frac{1}{n-1}.$$

Воспользовавшись фактом 7.1, получаем

$$\lambda \geq \frac{\alpha}{n-1}.$$

Тогда

$$f'(a_0) - f'(b') \geq \ln\left(1 + \frac{\frac{\alpha n}{n-1}}{1 - \frac{\alpha}{n-1}}\right) = \ln\left(1 + \frac{\alpha n}{n-1-\alpha}\right) \overset{n\to\infty}{\searrow} \ln(1+\alpha).$$

Возьмём $\delta := \ln(1+\alpha)$. □

Показав, что точка с маленьким значением f' действительно имеется, покажем, что в точке, к которой переходит наш алгоритм, значение тоже не слишком велико.

Лемма 7.3. *Пусть* b' *— точка, минимизирующая целевую функцию* $< c', \cdot >$ *на* B'. *Тогда* $f'(b') \leq f'(a_0) - \delta'$, *где* $\delta' = \mathrm{const}$.

Доказательство. Обозначим точку b', полученную в лемме 7.2, как $b_{7.2}$.

$$f'(a_0) - f'(b') = f'(a_0) - f'(b_{7.2}) + f'(b_{7.2}) - f'(b') \overset{\text{лемма 7.2}}{\geq} \delta + f'(b_{7.2}) - f'(b')$$

Пусть

$$\widetilde{f}(x) = n \ln \frac{< c', x >}{< c', a_0 >}.$$

Тогда

$$\begin{aligned} f'(a_0) - f'(b') \quad \geq & \\ & \widetilde{f}(b_{7.2}) - \widetilde{f}(b') + \\ & \Big(f'(b_{7.2}) - \Big(f'(a_0) + \widetilde{f}(b_{7.2})\Big)\Big) - \\ & \Big(f'(b') - \Big(f'(a_0) + \widetilde{f}(b')\Big)\Big), \end{aligned} \tag{7.3}$$

причем

$$\begin{aligned} & f'(x) - \Big(f'(a_0) + \widetilde{f}(x)\Big) = \\ & \sum_i \ln \frac{< c', x >}{x_i} - \sum_i \ln \frac{< c', a_0 >}{a_{0i}} - n \ln \frac{< c', x >}{< c', a_0 >} = \\ & \sum_i \ln \frac{a_{0i}}{x_i} \end{aligned} \tag{7.4}$$

Задача 7.3. Доказать: $\forall x \in B(a_0, \alpha r)$ в $\mathbb{R}^n$

$$\left| \sum_{1 \le i \le n} \ln \frac{a_{0i}}{x_i} \right| \le \frac{\beta^2}{2(1-\beta)}, \qquad \text{где } \beta = \alpha\sqrt{\frac{n}{n-1}}.$$

Пользуясь этим результатом и равенством (7.4), получаем из (7.3):

$$\begin{aligned} f'(a_0) - f'(b') &\ge \ln(1+\alpha) - \frac{\beta^2}{1-\beta} = \\ \ln(1+\alpha) - \frac{\alpha^2 n}{(n-1)(1-\alpha\sqrt{\frac{n}{n-1}})} &\xrightarrow{n\to\infty} \alpha - \frac{\alpha^2}{2} - \frac{\alpha^2}{1-\alpha} > 0. \end{aligned}$$

□

Из леммы 7.3 вытекает, что если f' не уменьшается на каком-то шаге алгоритма, то задача решения не имеет.

Если же мы хотим найти внутреннюю точку, в которой значение целевой функции достаточно мало для того, чтобы найти решение (см. схему алгоритма), сколько итераций должен произвести алгоритм? Это будет ясно из следующей леммы.

Лемма 7.4. *Пусть x — точка, найденная с помощью алгоритма 7.1 за k шагов,*

$$k = O\left(n(q + \ln n)\right) \quad \Rightarrow \quad \frac{< c, x >}{< c, a_0 >} \le 2^{-q}.$$

Доказательство.

$$\sum_i \ln \frac{< c, x >}{x_i} \le \sum_i \ln \frac{< c, a_0 >}{a_{0i}} - k\delta$$

$$n \ln \frac{< c, x >}{< c, a_0 >} \le \sum_i \ln x_i - \sum_i \ln a_{0i} - k\delta \le n \ln n - k\delta$$

$$\ln \frac{< c, x >}{< c, a_0} \le \ln n - \frac{k}{n}\delta \quad \Rightarrow \quad \frac{< c, x >}{< c, a_0 >} \le \frac{n}{e^{\frac{k}{n}\delta}} = \frac{n}{e^{O(\delta(q + \ln n))}} \le \frac{1}{e^{const \cdot \delta q}}$$

□

Легко видеть, что тем самым необходимое количество итераций полиномиально от длины входа, т.е., мы предъявили полиномиальный алгоритм для задачи линейного программирования. Если аккуратно разобраться с вычислениями на каждом шаге (упражнение по линейной алгебре), можно убедиться, что трудоёмкость алгоритма составляет $O\left(n^{3.5} \cdot p(L)\right)$, где $p(L)$ — полином от общей длины входа алгоритма.