

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра Исследования Операций

Кольцов Максим Алексеевич

Построение минимального эллипсоида

Курсовая работа

Научный руководитель:
д. ф.-м. н., профессор Малоземов В. Н.

Санкт-Петербург
2015

Оглавление

Введение	3
1. Задача Сильвестра	4
1.1. Постановка задачи	4
1.2. Общий вид задачи квадратичного программирования и двойственной к ней	4
1.3. Двойственная задача для задача Сильвестра и её решение	5
1.4. Решение задачи в среде MATLAB	7
2. Задача о минимальном эллипсоиде	9
2.1. Оператор сжатия пространства	9
2.2. Итеративный алгоритм построения минимального эллипсоида	11
2.3. Практическое испытание алгоритма	15
Список литературы	17

Введение

В данной работе рассматривается задача построения эллипсоида минимального объёма, содержащего заданное множество точек в \mathbb{R}^n . Как её частный случай также рассмотрена задача Сильвестра — задача построения минимального шара.

Для задачи Сильвестра описано точное решение с помощью квадратичного программирования, реализованное в MATLAB, и продемонстрированы результаты работы программы на тестовых данных.

Для задачи о минимальном эллипсоиде описан итеративный приближённый алгоритм, использующий вспомогательную задачу в пространстве \mathbb{R}^{n+1} , которая решается с помощью сжатия пространства. Рассмотрен и подробно описан оператор сжатия с коэффициентом γ в направлении ξ . Работа оператора сжатия проиллюстрирована на нескольких примерах с шарами и эллипсами на плоскости. Подробно описан переход от решения вспомогательной задачи к основной.

Алгоритм построения минимального эллипсоида реализован в MATLAB и проверен на тестовых данных, для которых минимальный эллипсоид известен. Продемонстрирован процесс сходимости алгоритма к решению.

1. Задача Сильвестра

1.1. Постановка задачи

Будем рассматривать *задачу Сильвестра* — по заданному набору точек $a_i \in \mathbb{R}^n$, $i \in M = 1 : m$, найти наименьший по объёму шар, их содержащий. Чтобы формально поставить задачу, введем переменную $x \in \mathbb{R}^n$ — центр искомого шара. Очевидно, что для того, чтобы шар $B_{\mathbb{R}^n}(x, r)$ содержал все заданные точки, необходимо, чтобы его радиус r удовлетворял следующему условию:

$$\forall i \in M \quad \|a_i - x\| \leq r$$

То есть, в задаче Сильвестра требуется найти точку минимума функции $\max_{i \in M} \{\|a_i - x\|\}$ на \mathbb{R}^n , при этом радиус искомого минимального шара будет равен значению функции в этой точке. Заметим, что выражение под знаком максимума можно заменить на $1/2\|a_i - x\|^2$, что не меняет точку минимума. Теперь можно сказать так: решение задачи Сильвестра — это точка x_* , являющаяся решением экстремальной задачи

$$\varphi(x) := \max_{i \in M} \{1/2\|a_i - x\|^2\} \rightarrow \min_{x \in \mathbb{R}^n} \quad , \quad (1)$$

а радиус соответствующего минимального шара равен $\sqrt{2\varphi(x_*)}$. Доказательство существования и единственности такого решения имеется в книге [1].

Задача 1 сводится к задаче квадратичного программирования. Раскроем по определению нормы:

$$\varphi(x) = 1/2\|x\|^2 + \max_{i \in M} \{-\langle a_i, x \rangle + 1/2\|a_i\|^2\}$$

Теперь выражение под максимумом линейно по x , а значит сам максимум можно обозначить за t и ввести набор ограничений. То есть, исходная задача эквивалентна задаче квадратичного программирования

$$\begin{aligned} \frac{1}{2}\langle Ex, x \rangle + t &\rightarrow \min \\ \langle a_i, x \rangle + t &\geq \frac{1}{2}\|a_i\|^2, \quad i \in M \end{aligned}$$

1.2. Общий вид задачи квадратичного программирования и двойственной к ней

Чтобы двигаться дальше, необходимо сначала рассмотреть понятия задачи квадратичного программирования и двойственной задачи. Задача квадратичного програм-

мирования — это экстремальная задача вида

$$\begin{aligned}
 Q(x) &= \frac{1}{2} \langle Dx, x \rangle + \langle c, x \rangle \rightarrow \min \\
 A[M_1, N] \times x[N] &\geq b[M_1] \\
 A[M_2, N] \times x[N] &= b[M_2] \\
 x[N_1] &\geq \mathbb{O}[N_1]
 \end{aligned}$$

где $D = D[N, N]$ — симметричная неотрицательно-определённая матрица. Как и в случае линейного программирования, можно ввести двойственную задачу и получить теорему, связывающую её с исходной задачей — первую теорему двойственности. При этом двойственная задача выглядит так:

$$\begin{aligned}
 -\frac{1}{2} \langle Dv, v \rangle + \langle b, u \rangle &\rightarrow \max \\
 -v[N] \times D[N, N_1] + u[M] \times A[M, N_1] &\leq c[N_1] \\
 -v[N] \times D[N, N_2] + u[M] \times A[M, N_2] &= c[N_2] \\
 u[M_1] &\geq \mathbb{O}[M_1]
 \end{aligned}$$

Для удобства записи двойственной задачи можно использовать таблицу

	c	
v	$-D$	$-\frac{1}{2}Dv$
u	A	b

Тогда целевая функция получается скалярным умножением первого и последнего столбца таблицы, ограничения — первого и второго, а правые части ограничений стоят в первой строке, причём неравенствами являются ограничения, соответствующие знаковым ограничениям исходной задачи. Знаковые ограничения накладываются на двойственные переменные (компоненты вектора u), соответствующие исходным ограничениям-неравенствам.

1.3. Двойственная задача для задача Сильвестра и её решение

Вернёмся к задаче Сильвестра. Из векторов-строк a_i составим матрицу A_0 и введём стандартные для квадратичной задачи обозначения: матрицу квадратичной формы D , вектор коэффициентов линейной формы c , матрицу ограничений A и столбец

правых частей b . Получаем следующее:

$$D = \begin{pmatrix} & 0 \\ E[N, N] & \vdots \\ & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad c = (0 \quad \dots \quad 0 \quad 1) \quad A = \begin{pmatrix} & 1 \\ A_0 & \vdots \\ & 1 \end{pmatrix} \quad b[i] = \frac{1}{2} \|a_i\|^2$$

Переменной в такой задаче является пара $y = (x, t)$, а в двойственной вектор $z = ((v, s), u)$. Чтобы записать двойственную задачу, воспользуемся таблицей

	0	...	0	1	
$v[N]$	$-E[N, N]$		\vdots	0	$-\frac{1}{2}v[N]$
s	0	...	0	0	0
u	A_0		\vdots	1	b
				1	

Тогда по описанному общему алгоритму имеем

$$g(z) := -\frac{1}{2} \langle v, v \rangle + \langle b, u \rangle \rightarrow \max$$

$$-v + uA_0 = 0$$

$$\sum_{i \in M} u[i] = 1$$

$$u[M] \geq \mathbb{O}[M]$$

Видно, что здесь можно исключить вектор v и получить задачу минимизации на стандартном симплексе

$$h(u) := \frac{1}{2} \|uA_0\|^2 - \langle b, u \rangle \rightarrow \min$$

$$\sum_{i \in M} u[i] = 1$$

$$u[M] \geq \mathbb{O}[M]$$

Для последней задачи существует оптимальный план u_* . В [1] доказано, что в таком случае вектор $x_* = u_*A_0$ является решением задачи Сильвестра. Так как два оптимальных плана пары двойственных задач имеют одно и то же значение целевой функции, то $\varphi(x_*) = -h(u_*)$ (на последнем шаге целевая функция была умножена на -1 чтобы перейти от задачи максимизации к задаче минимизации). Значит, радиус искомого минимального шара равен $\sqrt{-2h(u_*)}$.

1.4. Решение задачи в среде MATLAB

Для удобства решения задачи Сильвестра была написана функция `sylvester`, принимающая единственный аргумент — матрицу A , строки которой образованы исходным множеством точек. Алгоритм, реализованный в функции, следующий:

1. Вычислить матрицу $D = AA^T$ и вектор b , соответствующие двойственной задаче.
2. Вызвать встроенную функцию `quadprog` для решения двойственной задачи и получить её решение — вектор u_* , и значение целевой функции f на нём.
3. Вычислить радиус минимального шара по формуле $r = \sqrt{-2f}$.

Написанная функция работает с пространствами любых размерностей, но для наглядности она была проверена на размерности 2. Для этого генерировались случайные множества точек, лежащих внутри и на границе круга $(x - 4)^2 + (y - 3)^2 \leq 4$ и для них решалась задача Сильвестра. Одно из таких решений с количеством точек 100 представлено на рис. 1.

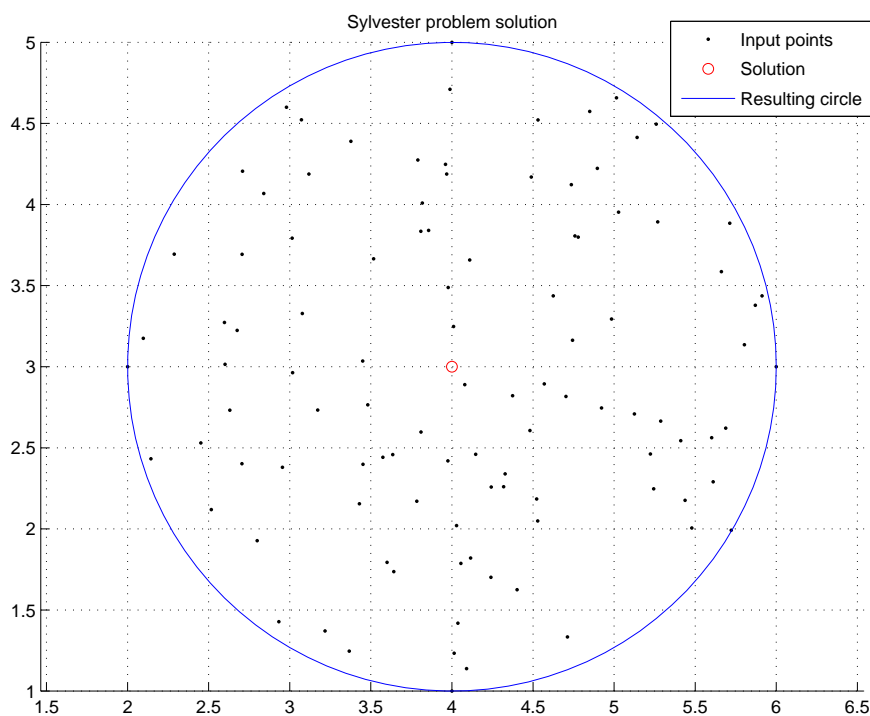


Рис. 1: Решение задачи Сильвестра для $m = 100$

Стоит отметить, что уже при попытке решить задачу для 200 точек, мы получаем ошибку:

```
Maximum number of iterations exceeded; increase options.MaxIter.  
To continue solving the problem with the current solution as the  
starting point, set x0 = x before calling quadprog.
```

Это связано с тем, что по умолчанию количество итераций, которые делает функция `quadprog`, ограничено числом 200. Чтобы решить эту проблему, можно поднять ограничение. При этом оказывается, что количество итераций зависит от количества точек примерно линейно (см рис. 2).

Такой результат можно объяснить тем, что по умолчанию MATLAB выбирает для решения этой задачи алгоритм «active-set». Если же задать использование алгоритма «interior-point-convex», пригодного для решения выпуклых задач, то для 100 точек требуется 6 итераций, а для 500 — всего 9.

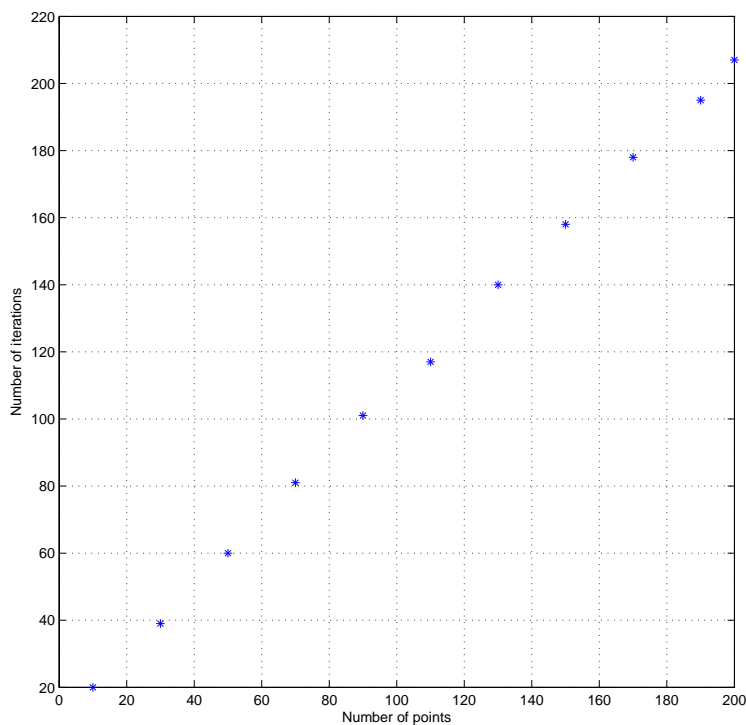


Рис. 2: Зависимость числа итераций от числа точек

Также можно протестировать программу в размерности 3. Для этого было сгенерировано 100 случайных точек внутри и на границе стандартного куба $[-1, 1]^3$. На этом наборе точек программа выдает ожидаемый ответ — центр шара близок к 0, а его радиус — к $\sqrt{3}$.

2. Задача о минимальном эллипсоиде

Рассмотренную выше задачу можно обобщить: искать не шар, а *эллипсоид* минимального объёма. В отличие от задачи Сильвестра, которую можно решить точно с помощью квадратичного программирования, задачу нахождения минимального эллипсоида удаётся решить лишь приближённо. Этот раздел посвящён одному алгоритму, строящему такое приближённое решение.

Теперь поставим задачу формально. Дано множество точек $c_j \in \mathbb{R}^n$, $j \in 1 : m$. Требуется построить эллипсоид $E \subset \mathbb{R}^n$ минимального объёма, который содержит все точки c_j . Эллипсоид E будем задавать таким способом:

$$E = \{x \in \mathbb{R}^n \mid \langle M(x - c), x - c \rangle \leq 1\} ,$$

где $c \in \mathbb{R}^n$ — центр эллипсоида, а M — симметричная положительно-определённая матрица порядка n .

Известно, что объём эллипсоида вычисляется по формуле

$$V = w_n (\det M)^{-1/2} ,$$

где w_n — объём n -мерного шара. Таким образом, задачу нахождения минимального эллипсоида можно сформулировать как экстремальную задачу

$$\begin{aligned} V &:= w_n (\det M)^{-1/2} \rightarrow \min \\ \langle M(c_j - c), c_j - c \rangle &\leq 1, \quad j \in 1 : m \\ M &\text{ положительно определена} \end{aligned}$$

Отбросим множитель w_n , возведём целевую функцию в степень -2 и перейдём к задаче максимизации определителя матрицы M . В итоге получается экстремальная задача

$$\begin{aligned} f &:= \det M \rightarrow \max \\ \langle M(c_j - c), c_j - c \rangle &\leq 1, \quad j \in 1 : m \\ M &\text{ положительно определена} \end{aligned} \tag{2}$$

Если среди c_j найдутся $n + 1$ аффинно независимых точек, то решение задачи (2) существует и единственно (см статью [2]). Нас же пока интересует только алгоритм построения.

2.1. Оператор сжатия пространства

Алгоритм построения минимального эллипсоида использует оператор сжатия пространства, поэтому рассмотрим сначала его свойства.

ОПРЕДЕЛЕНИЕ. Пусть $\xi \in \mathbb{R}^n$, $\|\xi\| = 1$ и $\gamma \in (0, 1)$. Тогда оператор сжатия R_γ

□

К обратному оператору применимы те же рассуждения о собственных числах и собственных векторах, в частности, вектор ξ собственный с собственным числом $\frac{1}{\gamma}$.

Теперь поймём, как оператор сжатия действует на шары и эллипсоиды в \mathbb{R}^n . Пусть, для начала, имеется шар с центром в точке c и радиусом r , который задан уравнением

$$\langle x - c, x - c \rangle \leq r^2$$

Подействуем на него оператором R_γ , получив новые точки $y = R_\gamma x$ и новый центр $a = R_\gamma c$. Так как R_γ обратим, то $x = R_\gamma^{-1}y$ и $c = R_\gamma^{-1}a$. Подставим эти выражения в уравнение шара

$$\begin{aligned} \langle x - c, x - c \rangle &\leq r^2 \\ \langle R_\gamma^{-1}(y - a), R_\gamma^{-1}(y - a) \rangle &\leq r^2 \\ \langle (R_\gamma^{-2})(y - a), y - a \rangle &\leq r^2 \\ \left\langle \frac{R_\gamma^{-2}}{r^2}(y - a), y - a \right\rangle &\leq 1 \end{aligned}$$

Получилось уравнение эллипсоида с центром в точке $a = R_\gamma c$ и матрицей $M := \frac{R_\gamma^{-2}}{r^2}$. Как было установлено выше, матрица M имеет собственный вектор ξ , которому соответствует собственное число $\frac{1}{r^2\gamma^2}$, а остальные собственные векторы ортогональны ξ и имеют собственные числа $\frac{1}{r^2}$. Таким образом, оператор сжатия превращает шар радиуса r в эллипсоид, одной из полуосей которого является ξ с длиной $r\gamma$, а длины остальных полуосей равны r — шар «сплющивается» в направлении ξ .

Разобравшись с шаром, подействуем теперь оператором R_γ на эллипсоид, заданный уравнением

$$\langle M(x - c), x - c \rangle \leq 1$$

Положим аналогично $x = R_\gamma^{-1}y$, $c = R_\gamma^{-1}a$ и получим новое уравнение эллипсоида в виде

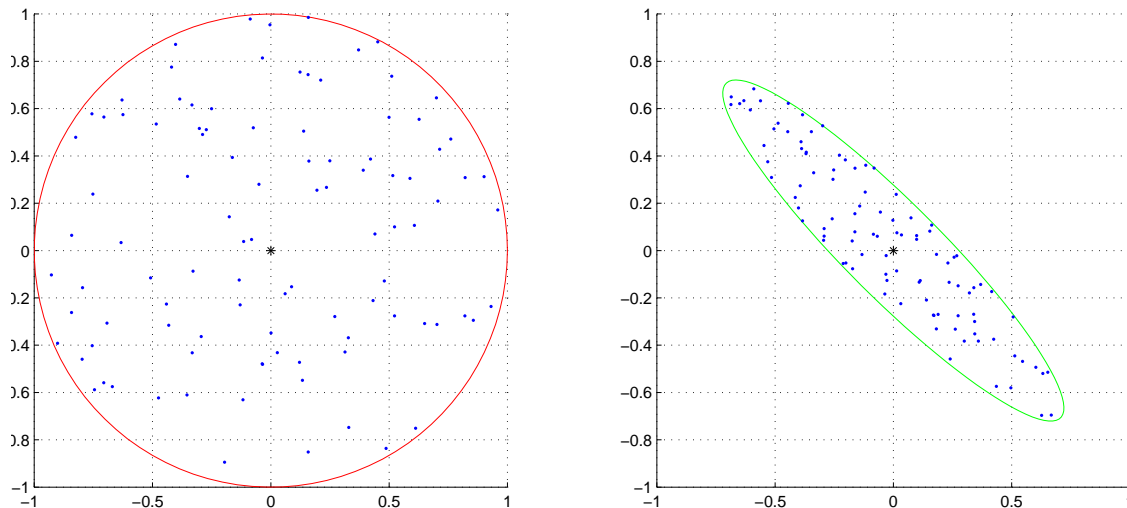
$$\langle R_\gamma^{-1}MR_\gamma^{-1}(y - a), y - a \rangle \leq 1$$

Наглядное представление о работе оператора сжатия даёт рис. 4.

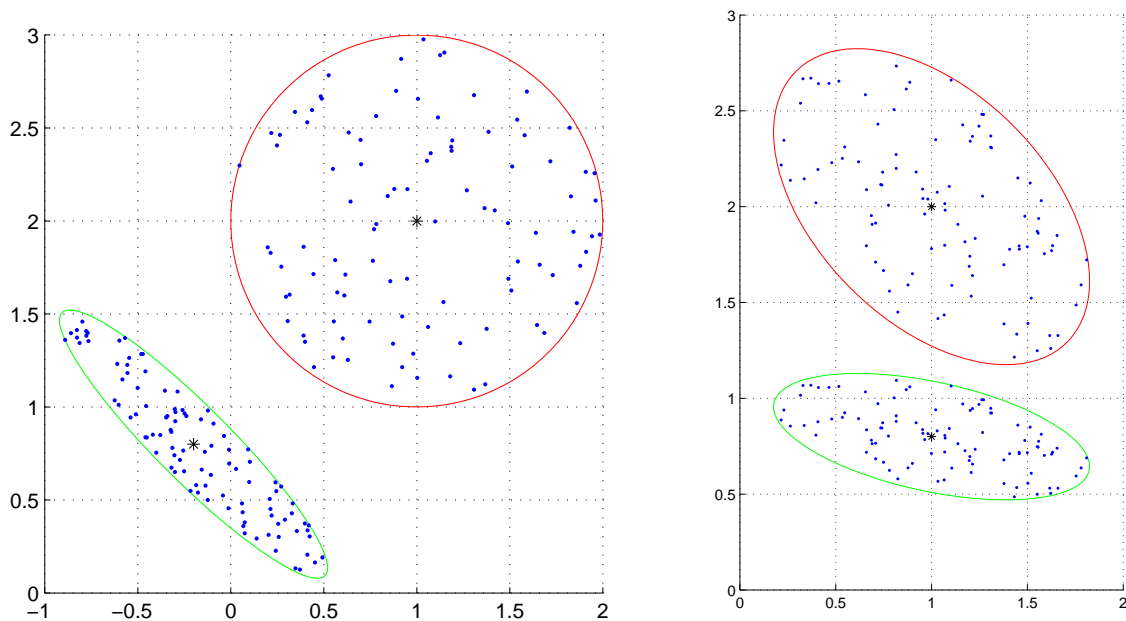
2.2. Итеративный алгоритм построения минимального эллипсоида

Для задачи о минимальном эллипсоиде известен итеративный алгоритм, строящий последовательные приближения к ответу. Этот алгоритм, описанный Н.З. Шором в статье [3], основан на достаточно простых идеях.

Введём вспомогательную задачу: вложим множество точек c_j в гиперплоскость



(а) Простейший случай — сжатие единичного круга в направлении $(1, 1)$ с $\gamma = 0.2$



(b) Сжатие круга в направлении $(1, 1)$ с $\gamma = 0.2$

(c) Сжатие эллипса в направлении $(0, 1)$ с $\gamma = 0.4$

Рис. 4: Действие оператора сжатия на множество, выделенное красной линией. Результат выделен зелёным цветом

$x_{n+1} = 1$ пространства \mathbb{R}^{n+1} . Обозначим $\tilde{c}_j = \begin{pmatrix} c_j \\ 1 \end{pmatrix}$. В новом пространстве будем искать минимальный эллипсоид с фиксированным центром в нуле. В статье [3] утверждается, что сечение такого эллипсоида плоскостью $x_{n+1} = 1$ и будет решением исходной задачи (см рис. 5).

Поставим формально вспомогательную задачу в терминах матрицы \tilde{B} порядка $n + 1$:

$$\begin{aligned} f &:= \det \tilde{B} \rightarrow \max \\ \langle \tilde{B}\tilde{c}_j, \tilde{c}_j \rangle &\leq 1, \quad j \in 1 : m \\ \tilde{B} &\text{ положительно определена} \end{aligned} \quad (3)$$

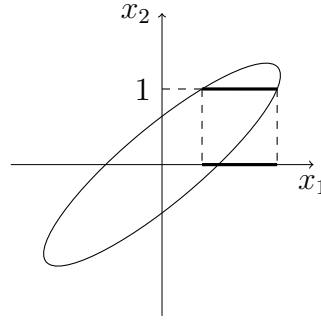


Рис. 5: Сечение эллипса с центром в нуле прямой $x_2 = 1$

Рассмотрим вопрос построения сечения подробнее. Итак, пусть

$$\tilde{E} = \left\{ \tilde{x} \in \mathbb{R}^{n+1} \mid \langle \tilde{B}\tilde{x}, \tilde{x} \rangle \leq 1 \right\}$$

— минимальный эллипсоид, содержащий точки \tilde{c}_j , $j \in 1 : m$. Возьмём две различные точки \tilde{c}_{j_0} и \tilde{c}_{j_1} . Отметим, что

$$\langle \tilde{B}(\tilde{c}_{j_0} + \tilde{c}_{j_1}), \tilde{c}_{j_0} + \tilde{c}_{j_1} \rangle + \langle \tilde{B}(\tilde{c}_{j_0} - \tilde{c}_{j_1}), \tilde{c}_{j_0} - \tilde{c}_{j_1} \rangle = 2\langle \tilde{B}\tilde{c}_{j_0}, \tilde{c}_{j_0} \rangle + 2\langle \tilde{B}\tilde{c}_{j_1}, \tilde{c}_{j_1} \rangle \quad (4)$$

Обозначим

$$\tilde{x}_0 = \frac{1}{2}(\tilde{c}_{j_0} + \tilde{c}_{j_1}) = \begin{pmatrix} x_0 \\ 1 \end{pmatrix}, \quad \tilde{x}_1 = \frac{1}{2}(\tilde{c}_{j_0} - \tilde{c}_{j_1}) = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}$$

Поделив равенство (4) на 4, получим

$$\langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle = \frac{1}{2}\langle \tilde{B}\tilde{c}_{j_0}, \tilde{c}_{j_0} \rangle + \frac{1}{2}\langle \tilde{B}\tilde{c}_{j_1}, \tilde{c}_{j_1} \rangle - \langle \tilde{B}\tilde{x}_1, \tilde{B}\tilde{x}_1 \rangle$$

По определению точки \tilde{c}_{j_0} и \tilde{c}_{j_1} принадлежат \tilde{E} и $\tilde{x}_1 \neq \mathbb{O}$, поэтому

$$\langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle < 1 \quad (5)$$

Представим матрицу \tilde{B} в виде

$$\tilde{B} = \begin{pmatrix} B & b \\ b^T & \hat{b} \end{pmatrix}$$

где B — главный минор порядка n . Очевидно, что B симметрична и положительно определена (по критерию Сильвестра). Аккуратно раскроем скобки в (5):

$$\begin{aligned} \langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle &= \left\langle \begin{pmatrix} Bx_0 + b \\ \langle b, x_0 \rangle + \hat{b} \end{pmatrix}, \begin{pmatrix} cx_0 \\ 1 \end{pmatrix} \right\rangle = \langle Bx_0 + b, x_0 \rangle + \langle b, x_0 \rangle + \hat{b} = \\ &= \langle B(x_0 + B^{-1}b), x_0 + B^{-1}b \rangle - \langle B(x_0 + B^{-1}b), B^{-1}b \rangle + \langle b, x_0 \rangle + \hat{b} = \\ &= \langle B(x_0 + B^{-1}b), x_0 + B^{-1}b \rangle - \langle B^{-1}b, b \rangle + \hat{b} < 1 \end{aligned} \quad (6)$$

ЛЕММА 2. *Справедливы неравенства*

$$0 < -\langle B^{-1}b, b \rangle + \hat{b} < 1$$

Доказательство. Пусть, во-первых, $\tilde{x} = \begin{pmatrix} -B^{-1}b \\ 1 \end{pmatrix} \neq \mathbb{O}$. Матрица \tilde{B} положительно определена, так что $\langle \tilde{B}\tilde{x}, \tilde{x} \rangle > 0$. Рассуждения, аналогичные (6), приводят к равенству $\langle \tilde{B}\tilde{x}, \tilde{x} \rangle = -\langle B^{-1}b, b \rangle + \hat{b}$, что доказывает левое неравенство.

Для доказательства правого неравенства заметим, что в неравенстве (6) $\langle B(x + B^{-1}b), x + B^{-1}b \rangle \geq 0$ в силу положительной определённости B . Тогда имеем

$$1 > \langle B(x + B^{-1}b), x + B^{-1}b \rangle - \langle B^{-1}b, b \rangle + \hat{b} \geq -\langle B^{-1}b, b \rangle + \hat{b}$$

Неравенство доказано. □

Теперь последнее неравенство из (6) поделим на положительное число $1 + \langle B^{-1}b, b \rangle - \hat{b}$, введём обозначения

$$c := -B^{-1}b, \quad M := B / (1 + \langle B^{-1}b, b \rangle - \hat{b}) \quad (7)$$

и окончательно получим

$$\langle M(x - c), x - c \rangle \leq 1 \text{ — уравнение минимального эллипсоида в } \mathbb{R}^n$$

Алгоритм решения задачи (3) устроен следующим образом. Текущее множество точек $\tilde{c}_j^{(k)}$ хранится в виде столбцов матрицы X_k , текущая матрица последовательности операторов сжатия обозначается A_k . Изначально X_0 состоит из исходных точек $\tilde{c}_j \in \mathbb{R}^{n+1}$, а $A_0 = E_{n+1}$. Один шаг работы алгоритма состоит из нескольких простых действий:

- Найти среди векторов $\tilde{c}_j^{(k)}$ вектор $\tilde{c}_{j_k}^{(k)}$ с максимальной нормой $\tau_{k+1} = \|\tilde{c}_{j_k}^{(k)}\|$

- Вычислить единичный вектор направления сжатия $\xi_{k+1} = \frac{\tilde{c}_j^{(k)}}{\tau_{k+1}}$
- Построить оператор сжатия R_{k+1} вдоль вектора ξ_{k+1} с коэффициентом α_{k+1}
- Вычислить новые точки $\tilde{c}_j^{(k+1)}$, умножив матрицу R_{k+1} на матрицу X_k , и сохранить результат в X_{k+1}
- Добавить матрицу R_{k+1} к последовательности операторов сжатия, умножив её на матрицу A_k , и сохранить результат в A_{k+1}

Коэффициенты α_k в соответствии со статьей [3] выбираются из условий

$$\alpha_k = 1 - \beta_k, \quad \beta_k > 0, \quad \beta_k \xrightarrow[k \rightarrow \infty]{} 0, \quad \sum_{k=1}^{\infty} \beta_k = \infty$$

Пусть вычисления остановлены после шага m . Тогда из описания алгоритма следует, что

$$X_m = R_m \cdot R_{m-1} \cdots R_1 \cdot X_0 = A_m \cdot X_0$$

Также известно, что точки $\tilde{c}_j^{(m)}$ лежат внутри сферы с радиусом τ_m и центром в начале координат. С учётом этих двух фактов можно получить матрицу искомого эллипсоида:

$$\begin{aligned} \langle \tilde{c}_j^{(m)}, \tilde{c}_j^{(m)} \rangle &\leq \tau_m^2 \\ \langle A_m \tilde{c}_j, A_m \tilde{c}_j \rangle &\leq \tau_m^2 \\ \langle A_m^T A_m \tilde{c}_j, \tilde{c}_j \rangle &\leq \tau_m^2 \\ \langle \frac{A_m^T A_m}{\tau_m^2} \tilde{c}_j, \tilde{c}_j \rangle &\leq 1 \end{aligned}$$

Матрица $\tilde{B} := A_m^T A_m / \tau_m^2$ симметрична, положительно определена и является (приближённым) решением вспомогательной задачи о минимальном эллипсоиде. Окончательное решение основной задачи получается по формулам (7).

2.3. Практическое испытание алгоритма

Алгоритм был реализован в среде MATLAB и испытан на тестовых данных. Коэффициенты β_k были равны $1/(k+1)$. В качестве тестовых данных было сгенерировано множество случайных точек внутри эллипсоида с центром в точке $(1, 2)$ с полуосями $2, 1$, который вытянут в направлении $(-1, 1)$.

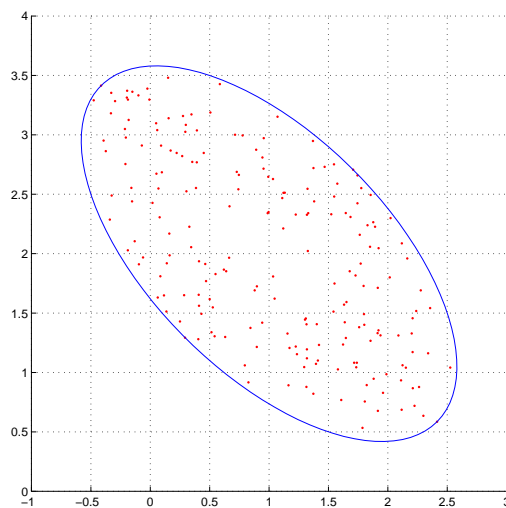


Рис. 6: Тестовое множество точек. Синим отмечена граница минимального эллипсоида

Алгоритм сходится к решению достаточно хорошо, уже после 50-ти шагов построенный эллипсоид близок к искомому.

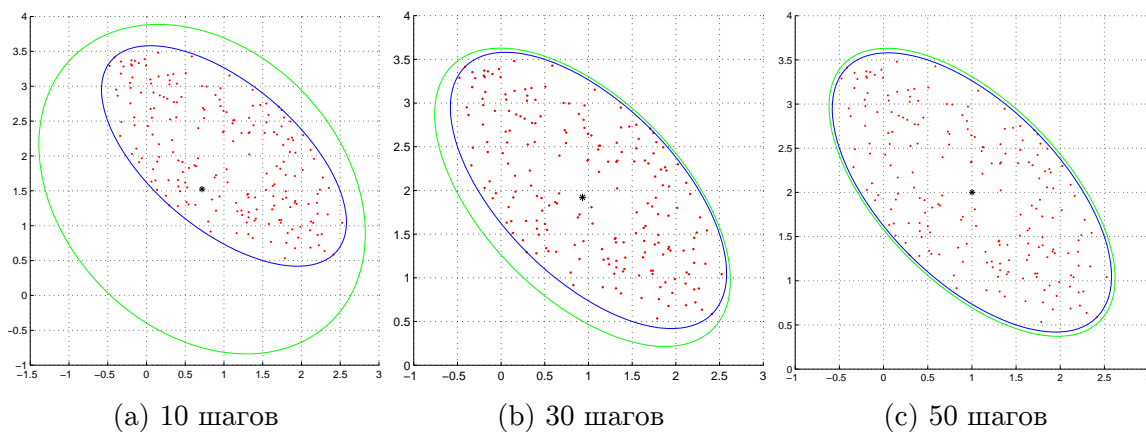


Рис. 7: Результат работы алгоритма. Зелёным обозначен построенный эллипсоид

Список литературы

- [1] Гавурин М.К., Малозёмов В.Н. Экстремальные задачи с линейными ограничениями. — Л. : Изд-во ЛГУ, 1984. — С. 176.
- [2] Danzer L., Laugwitz D., Hanfried L. Über das Löwnersche Ellipsoid und sein Analogon unter den einem Eikörper eingeschriebenen Ellipsoiden // Arch. Math. — 1957. — Vol. 8, no. 3.
- [3] Шор Н.З., Стеценко С.И. Алгоритм последовательного сжатия пространства для построения описанного эллипсоида минимального объёма // Исследование методов решения экстремальных задач. — 1990.