

Методы оптимизации. Семинар 1. Введение.

Александр Катруца

Московский физико-технический институт,
Факультет Управления и Прикладной Математики

5 сентября 2016 г.

Зачем этот курс?

- Формализация задачи выбора элемента из множества
- Обоснование правильности принятия решения
- Разнообразные приложения:
 - машинное обучение: классификация, кластеризация, регрессия
 - молекулярное моделирование
 - анализ рисков
 - выбор активов (portfolio optimization)
 - оптимальное управление
 - обработка сигналов
 - оценка параметров в статистике
 - и другие¹

¹<http://www.cvxpy.org/en/latest/examples/index.html>

О чём этот курс?

Первый семестр (теоретический):

- Основы выпуклого анализа
- Теория двойственности
- Условия оптимальности

Второй семестр (практический):

- Методы безусловной минимизации первого и второго порядка
- Методы условной оптимизации
- Линейное программирование: симплекс-метод и пр.
- Оптимальные методы
- ...

План на семестр

- Семинар и лекция раз в неделю
- 2 задания
- Итоговая контрольная в конце семестра (и промежуточная в середине семестра)
- Экзамен в конце семестра. Оценка усредняется по оценкам за:
 - 2 задания
 - итоговую контрольную
 - ответ на экзамене
- Миниконтрольные в начале каждого семинара
- Домашнее задание на каждый семинар — \LaTeX

Предварительные навыки

- Линейная алгебра
- Математический анализ
- Программирование: Python (NumPy, SciPy, CVXPY) or MATLAB
- Элементы вычислительной математики

- 1 Определение целевой функции
- 2 Определение допустимого множества решений
- 3 Постановка и анализ оптимизационной задачи
- 4 Выбор наилучшего алгоритма для решения поставленной задачи
- 5 Реализация алгоритма и проверка его корректности

Постановка задачи

$$\begin{aligned} \min_{\mathbf{x} \in X} f_0(\mathbf{x}) \\ \text{s.t. } f_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \\ f_j(\mathbf{x}) \leq 0, \quad j = n + 1, \dots, m, \end{aligned}$$

- $\mathbf{x} \in \mathbb{R}^n$ — искомый вектор
- $f_0(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ — целевая функция
- $f_k(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ — функции ограничений

Пример: выбор активов

- \mathbf{x} — размер инвестиций в каждый актив
- f_0 — суммарный риск или вариация прибыли
- f_k — бюджетные ограничения, min/max вложения в актив, минимально допустимая прибыль

Как решать?

В общем случае:

- NP-полные
- рандомизированные алгоритмы: время vs стабильность

НО определённые классы задач могут быть решены быстро!

- Линейное программирование
- Метод наименьших квадратов
- Малоранговое приближение порядка k
- Выпуклая оптимизация

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{c}^T \mathbf{x} \\ \text{s.t. } \mathbf{a}_i^T \mathbf{x} \leq c_i, \quad i = 1, \dots, m \end{aligned} \tag{1}$$

- нет аналитического решения
- существуют эффективные алгоритмы
- разработанная технология
- симплекс-метод для решения задачи (1) входит в Топ-10 алгоритмов XX века²

²<https://www.siam.org/pdf/news/637.pdf>

Метод наименьших квадратов

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2^2, \quad (2)$$

где $\mathbf{A} \in \mathbb{R}^{m \times n}$ и $\mathbf{b} \in \mathbb{R}^m$.

- имеет аналитическое решение: $\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$
- существуют эффективные алгоритмы
- разработанная технология
- имеет статистическую интерпретацию

Малоранговое приближение ранга k

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \quad & \|\mathbf{A} - \mathbf{X}\| \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \leq k \end{aligned} \tag{3}$$

Theorem (Eckart–Young, 1993)

Пусть $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ — сингулярное разложение матрицы \mathbf{A} , где $\mathbf{U} = [\mathbf{U}_k, \mathbf{U}_{r-k}] \in \mathbb{R}^{m \times r}$, $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_k, \dots, \sigma_r)$, $\mathbf{V} = [\mathbf{V}_k, \mathbf{V}_{r-k}] \in \mathbb{R}^{n \times r}$ и $r = \text{rank}(\mathbf{A})$. Тогда решение задачи (3) можно записать в виде:

$$\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top,$$

где $\hat{\mathbf{U}} \in \mathbb{R}^{m \times k}$, $\hat{\mathbf{\Sigma}} = \text{diag}(\sigma_1, \dots, \sigma_k)$, $\hat{\mathbf{V}} \in \mathbb{R}^{n \times k}$.

Алгоритм вычисления сингулярного разложения и быстрый, и устойчивый.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} f_0(\mathbf{x}) \\ \text{s.t. } f_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, m \end{aligned} \tag{4}$$

- f_0, f_i — выпуклые функции:

$$f(\alpha \mathbf{x}_1 + \beta \mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + \beta f(\mathbf{x}_2),$$

где $\alpha, \beta \geq 0$ и $\alpha + \beta = 1$.

- нет аналитического решения
- существуют эффективные алгоритмы
- часто сложно «увидеть» задачу выпуклой оптимизации
- существуют приёмы для преобразования задачи к виду (4)

Пример

Независимый под-алфавит
максимальной мощности

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^n} \quad & \sum_{i=1}^n x_i \\ \text{s.t.} \quad & x_i^2 - x_i = 0 \quad i = 1, \dots, n \\ & x_i x_j = 0 \quad \forall (i, j) \in \Gamma, \end{aligned}$$

где Γ — множество пар

Truss design

$$\begin{aligned} \min \quad & -2 \sum_{i=1}^k \sum_{j=1}^m c_{ij} x_{ij} + x_{00} \\ \text{s.t.} \quad & \sum_{i=1}^k x_i = 1 \\ & \lambda_{\min}(\mathbf{A}) \geq 0, \end{aligned}$$

где $\mathbf{A} =$

$$\begin{bmatrix} x_1 & \dots & \dots & \sum_{j=1}^m b_{pj} x_{1j} \\ \vdots & \ddots & & \vdots \\ & & x_k & \sum_{j=1}^m b_{pj} x_{kj} \\ \sum_{j=1}^m b_{pj} x_{1j} & \dots & \sum_{j=1}^m b_{pj} x_{kj} & x_{00} \end{bmatrix}$$

Почему выпуклость так важна?

R. Tyrrell Rockafellar (1935 —)

The great watershed in optimization is not between linearity and non-linearity, but convexity and non-convexity.

- Локальный оптимум является глобальным
- Необходимое условие оптимальности является достаточным

Вопросы:

- Любую ли задачу выпуклой оптимизации можно эффективно решить?
- Можно ли эффективно решить невыпуклые задачи оптимизации?

- Организация работы
- Предмет курса по оптимизации
- Общая формулировка оптимизационной задачи
- Классические оптимизационные задачи