Ускоренные градиентные методы. Идея метода сопряженных градиентов

Даня Меркулов, Петр Остроухов

1 Квадратичная задача оптимизации

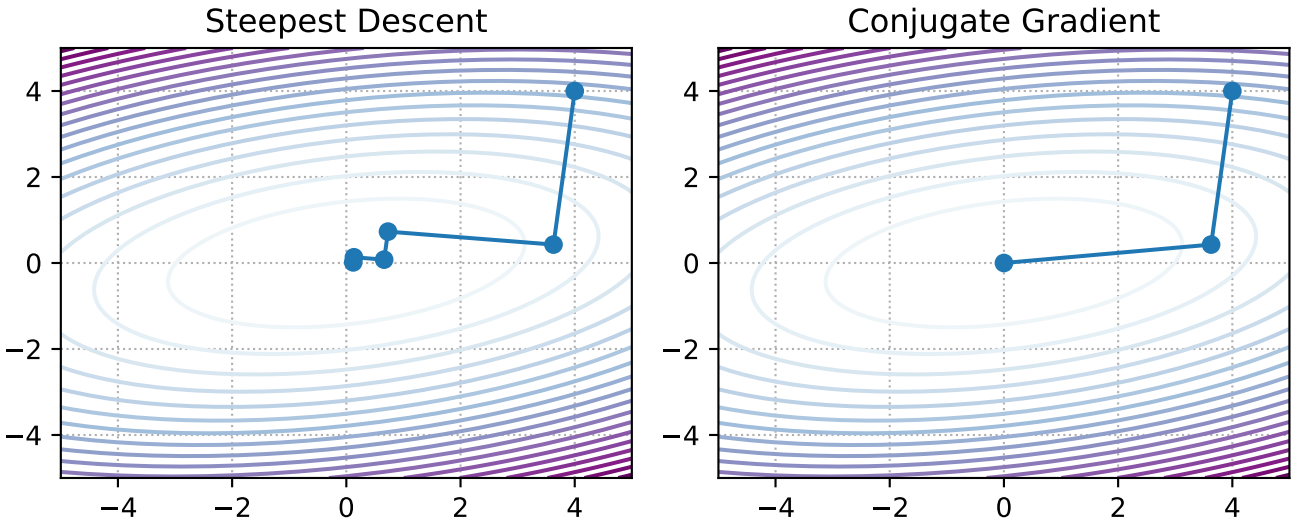
1.1 Сильно выпуклая квадратичная функция

Рассмотрим следующую квадратичную задачу оптимизации:

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x + c, \text{ где } A \in \mathbb{S}^n_{++}. \tag{1}$$

Условия оптимальности

$$Ax^* = b$$



1.2 Наискорейший спуск aka точный линейный поиск

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \arg\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Более теоретический, чем практический подход к выбору шага. Он также позволяет анализировать сходимость, но точный линейный поиск может быть численно сложным, если вычисление функции занимает слишком много времени или требует слишком много ресурсов.

Интересное теоретическое свойство этого метода заключается в том, что каждая следующая итерация метода ортогональна предыдущей:

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Условия оптимальности:

$$\nabla f(x_k)^T \nabla f(x_{k+1}) = 0$$

Оптимальное значение для квадратичных функций

$$\nabla f(x_k)^\top A(x_k - \alpha \nabla f(x_k)) - \nabla f(x_k)^\top b = 0 \qquad \alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T A \nabla f(x_k)}$$

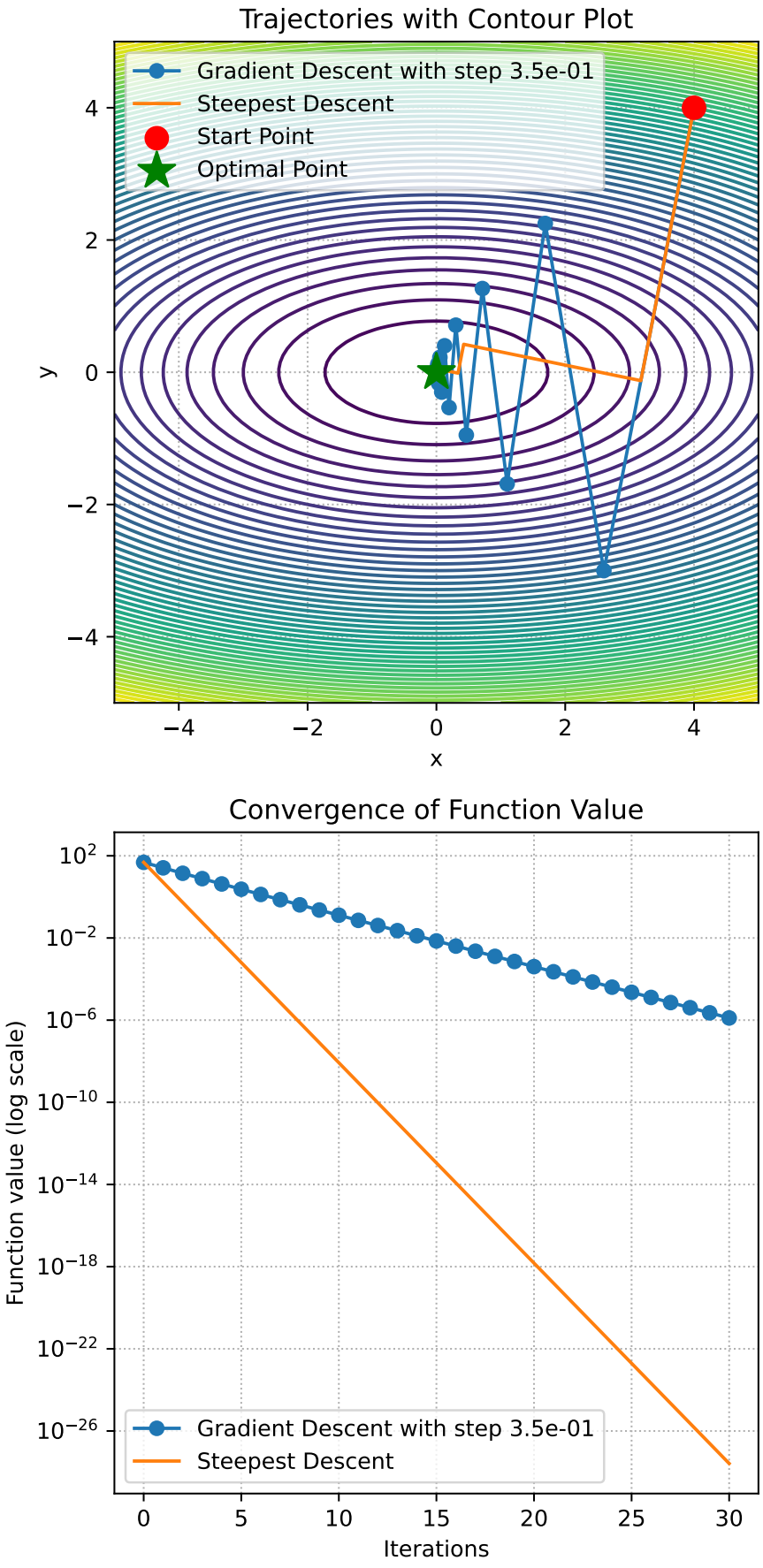


Рисунок 1: Наискорейший спуск

Открыть в Colab ♣

2 Ортогональность

2.1 Сопряженные направления. A-ортогональность.

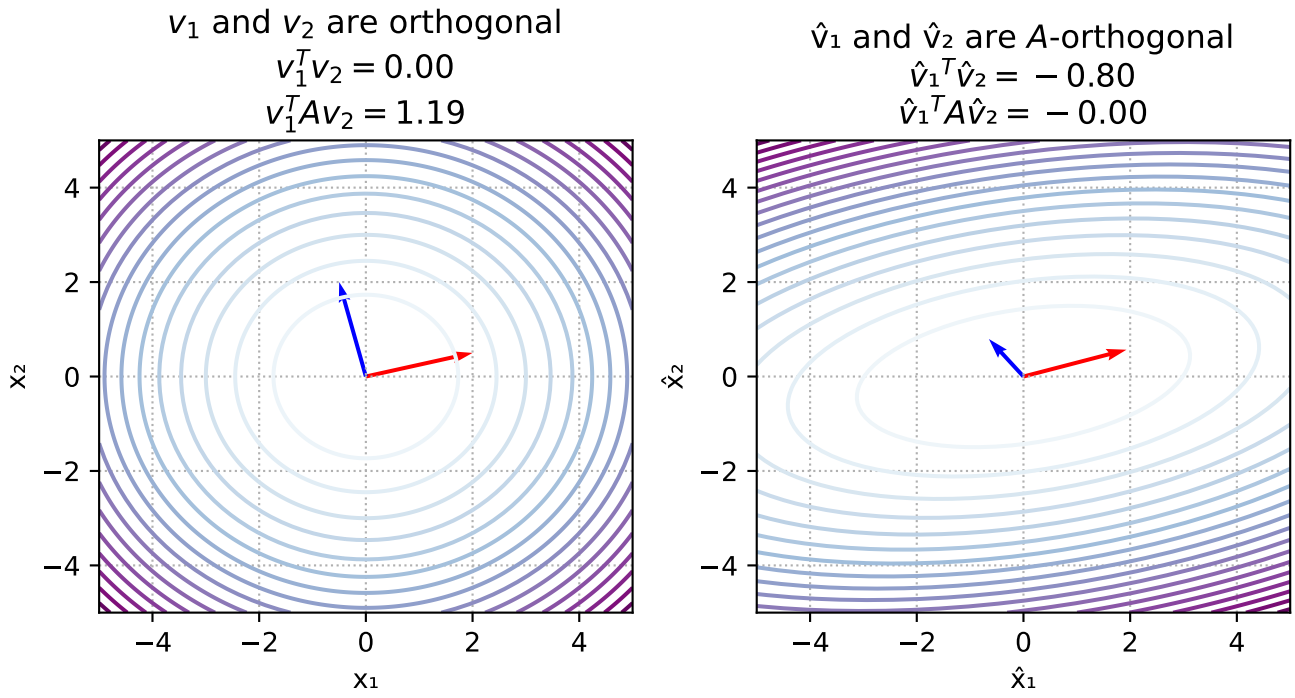


Рисунок 2

Предположим, у нас есть две системы координат и квадратичная функция $f(x) = \frac{1}{2} x^T I x$ выглядит так, как на левой части изображения 2, в то время как в других координатах она выглядит как $f(\hat{x}) = \frac{1}{2} \hat{x}^T A \hat{x}$, где $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2} x^T I x$$

$$\frac{1}{2} \hat{x}^T A \hat{x}$$

Поскольку $A = Q \Lambda Q^T$:

$$\frac{1}{2} \hat{x}^T A \hat{x} = \frac{1}{2} \hat{x}^T Q \Lambda Q^T \hat{x} = \frac{1}{2} \hat{x}^T Q \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} Q^T \hat{x} = \frac{1}{2} x^T I x \text{ и } \hat{x} = Q \Lambda^{-\frac{1}{2}} x$$

A-ортогональные векторы

Векторы $x \in \mathbb{R}^n$ и $y \in \mathbb{R}^n$ называются A-ортогональными (или A-сопряженными), если

$$x^T A y = 0 \qquad \Leftrightarrow \qquad x \perp_A y$$

Когда $A = I$, A-ортогональность превращается в ортогональность.

2.2 Процесс Грама-Шмидта

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

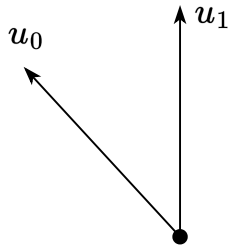


Рисунок 3: Иллюстрация процесса Грама-Шмидта

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

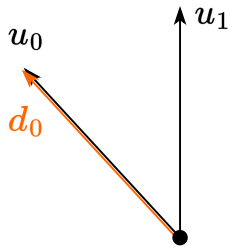


Рисунок 4: Иллюстрация процесса Грама-Шмидта

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

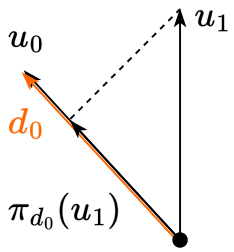


Рисунок 5: Иллюстрация процесса Грама-Шмидта

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

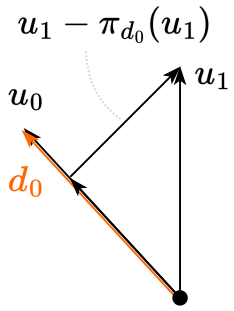


Рисунок 6: Иллюстрация процесса Грама-Шмидта

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

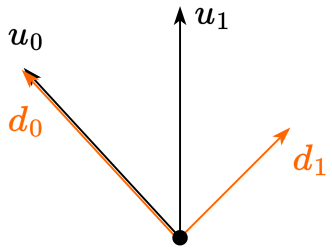
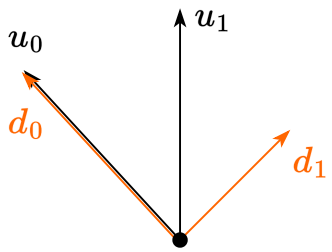
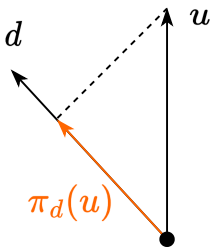


Рисунок 7: Иллюстрация процесса Грама-Шмидта





$$\pi_d(u) = \frac{\langle d, u \rangle}{\|d\|_2^2} d$$

Вход: n линейно независимых векторов $u_0, \dots, u_{n-1}$.

Выход: n линейно независимых векторов $d_0, \dots, d_{n-1}$.

$$\begin{aligned} d_0 &= u_0 \\ d_1 &= u_1 - \pi_{d_0}(u_1) \\ d_2 &= u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2) \\ &\vdots \\ d_k &= u_k - \sum_{i=0}^{k-1} \pi_{d_i}(u_k) \end{aligned}$$

$$d_k = u_k + \sum_{i=0}^{k-1} \beta_{ik} d_i \qquad \beta_{ik} = -\frac{\langle d_i, u_k \rangle}{\langle d_i, d_i \rangle} \tag{2}$$

3 Метод сопряженных направлений (CD)

3.1 Общая идея

- В изотропном случае $A = I$ метод наискорейшего спуска, запущенный из произвольной точки в n ортогональных линейно независимых направлениях, сойдется за n шагов в точных арифметических вычислениях. Мы пытаемся построить аналогичную процедуру в случае $A \neq I$ с использованием концепции A-ортогональности.
- Предположим, у нас есть набор из n линейно независимых A-ортогональных направлений $d_0, \dots, d_{n-1}$ (которые будут вычислены с помощью процесса Грама-Шмидта).

- Мы хотим построить метод, который идет из x_0 в x^* для квадратичной задачи с шагами α_i, который, фактически, является разложением $x^* - x_0$ в некотором базисе:

$$x^* = x_0 + \sum_{i=0}^{n-1} \alpha_i d_i \qquad x^* - x_0 = \sum_{i=0}^{n-1} \alpha_i d_i$$

- Мы докажем, что α_i и d_i могут быть построены очень эффективно с вычислительной точки зрения (метод сопряженных градиентов).

3.2 Идея метода сопряженных направлений (CD)

Таким образом, мы формулируем алгоритм:

Предположим, что нам заранее известны линейно-независимые векторы $u_0, \dots, u_{n-1}$.

1. $d_0 = -\nabla f(x_0)$.
2. С помощью процедуры точного линейного поиска находим оптимальную длину шага. Вычисляем α минимизируя $f(x_k + \alpha_k d_k)$ по формуле

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k} \tag{3}$$

3. Выполняем шаг алгоритма:

$$x_{k+1} = x_k + \alpha_k d_k$$

4. *Обновляем направление:* d_{k+1} *получаем из* u_{k+1} *с помощью модифицированной процедуры Грама–Шмидта в скалярном произведении* $\langle v, w \rangle_A = v^\top A w$ *относительно уже построенных* $d_0, \dots, d_k$*:*

$$d_{k+1} = u_{k+1} - \sum_{i=0}^{k} \beta_{k+1,i}\, d_i, \quad \beta_{k+1,i} = \frac{u_{k+1}^\top A d_i}{d_i^\top A d_i}$$

   что обеспечивает $d_{k+1} \perp_A d_j$ *для всех* $j \le k$.

5. Повторяем шаги 2–4, пока не построим n направлений, где n — размерность пространства (x).

3.3 Метод сопряженных направлений (CD)

> **Лемма 1.** *Линейная независимость A-ортогональных векторов.*
>
> Если множество векторов $d_1, \dots, d_n$ - попарно A-ортогональны (каждая пара векторов A-ортогональна), то эти векторы линейно независимы. $A \in \mathbb{S}^n_{++}$.

Доказательство

Покажем, что если $\sum_{i=1}^{n} \alpha_i d_i = 0$, то все коэффициенты должны быть равны нулю:

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \alpha_i d_i \\
\text{(Умножаем на } d_j^T A\text{)} \qquad &= d_j^\top A \left(\sum_{i=1}^{n} \alpha_i d_i \right) \\
&= \sum_{i=1}^{n} \alpha_i d_j^\top A d_i \\
&= \alpha_j d_j^\top A d_j + 0 + \ldots + 0
\end{aligned}
$$

Таким образом, $\alpha_j = 0$, для всех остальных индексов нужно проделать тот же процесс

3.4 Доказательство сходимости

Введем следующие обозначения:

- $r_k = b - Ax_k$ - невязка
- $e_k = x_k - x^*$ - ошибка
- Поскольку $Ax^* = b$, имеем $r_k = b - Ax_k = Ax^* - Ax_k = -A(x_k - x^*)$

$$r_k = -Ae_k. \tag{4}$$

- Также заметим, что поскольку $x_{k+1} = x_0 + \sum\limits_{i=0}^{k} \alpha_i d_i$, имеем

$$e_{k+1} = e_0 + \sum_{i=0}^{k} \alpha_i d_i. \tag{5}$$

> Лемма 2. Сходимость метода сопряженных направлений.
>
> Предположим, мы решаем n-мерную квадратичную сильно выпуклую задачу оптимизации (1). Метод сопряженных направлений
>
> $$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$
>
> с $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ взятым из точного линейного поиска, сходится за не более n шагов алгоритма.

Доказательство Пусть

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

Докажем, что $\delta_i = -\alpha_i$:

Умножаем обе части слева на $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A e_k = d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) \overset{\perp_A}{=} \delta_k d_k^T A d_k$$

$$\delta_k = \frac{d_k^T A e_k}{d_k^T A d_k} = -\frac{d_k^T r_k}{d_k^T A d_k} \Leftrightarrow \delta_k = -\alpha_k$$

4 Метод сопряженных градиентов (CG)

4.1 Идея метода сопряженных градиентов (CG)

- Это буквально метод сопряженных направлений, в котором мы выбираем специальный набор $d_0, \ldots, d_{n-1}$, позволяющий значительно ускорить процесс Грама-Шмидта.
- Используется процесс Грама-Шмидта с A-ортогональностью вместо Евклидовой ортогональности, чтобы получить их из набора начальных векторов.
- На каждой итерации $r_0, \ldots, r_{n-1}$ используются в качестве начальных линейно-независимых векторов для процесса Грама-Шмидта.
- Основная идея заключается в том, что для произвольного метода CD процесс Грама-Шмидта вычислительно дорогой и требует квадратичного числа операций сложения векторов и скалярных произведений $\mathcal{O}(n^2)$, в то время как в случае CG мы покажем, что сложность этой процедуры может быть уменьшена до линейной $\mathcal{O}(n)$.

CG = CD+$r_0, \ldots, r_{n-1}$ как начальные векторы для процесса Грама-Шмидта+A-ортогональность.

4.2 Леммы для сходимости

Лемма 5. Невязки ортогональны друг другу в методе CG

Все невязки в методе CG ортогональны друг другу:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{6}$$

Лемма 7. Коэффициенты для процесса Грама-Шмидта для CG

В процессе Грама-Шмидта для CG

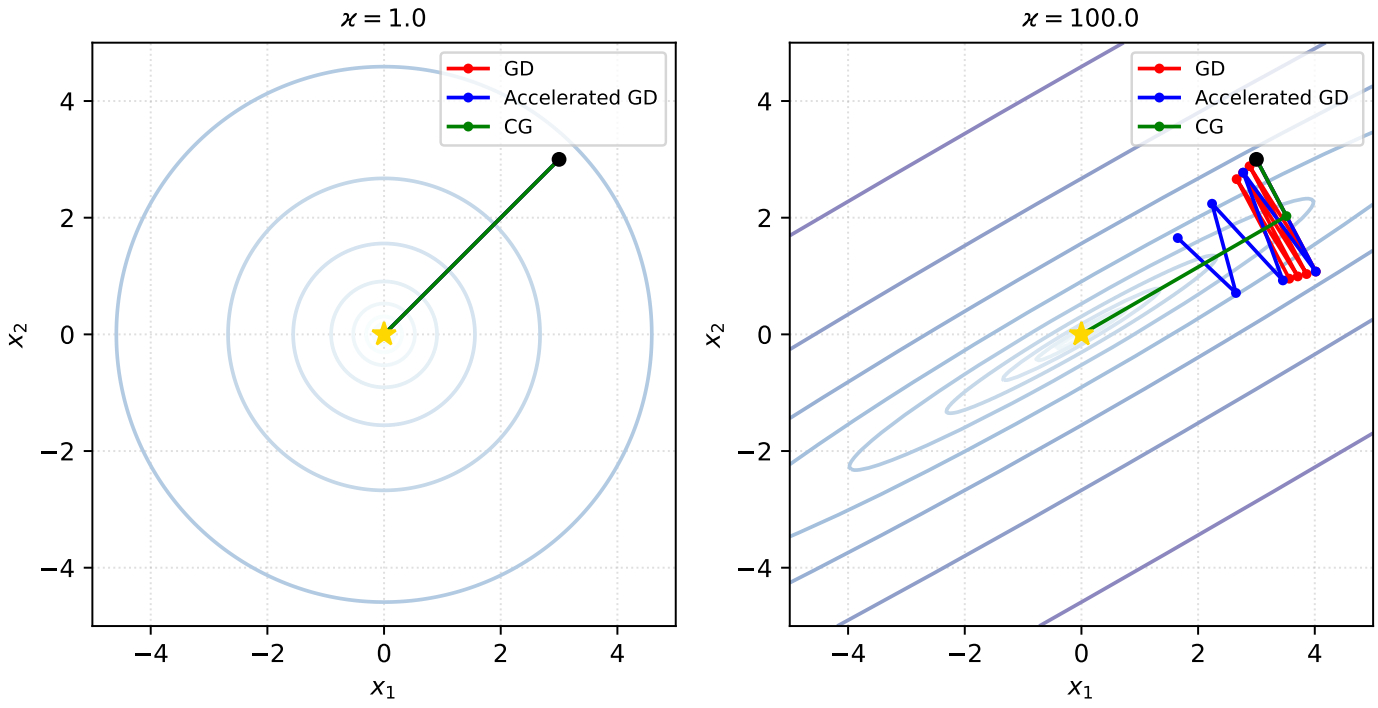
$$\beta_{ji} = \frac{r_i^\top r_i}{r_{i-1}^\top r_{i-1}}, \; i = j + 1.$$

Все остальные коэффициенты равны нулю кроме $i = j$, но этот случай нам неинтересен.

4.3 Метод сопряженных градиентов (CG)

$r_0 := b - Ax_0$
если r_0 достаточно мал, то вернуть x_0 как результат
$d_0 := r_0$
$k := 0$
повторять
  $\alpha_k := \frac{r_k^{\mathrm{T}} r_k}{d_k^{\mathrm{T}} A d_k}$
  $x_{k+1} := x_k + \alpha_k d_k$
  $r_{k+1} := r_k - \alpha_k A d_k$
  если r_{k+1} достаточно мал, то выйти из цикла
  $\beta_k := \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}$
  $d_{k+1} := r_{k+1} + \beta_k d_k$
  $k := k + 1$
конец повторения
вернуть x_{k+1} как результат

4.4 Закрываем квадратичный вопрос



4.5 Сходимость

Теорема 1. Если матрица A имеет только r различных собственных значений, то метод сопряженных градиентов сходится за r итераций.

Теорема 2. Следующая оценка сходимости выполняется для метода сопряженных градиентов, как для итерационного метода в сильно выпуклой задаче:

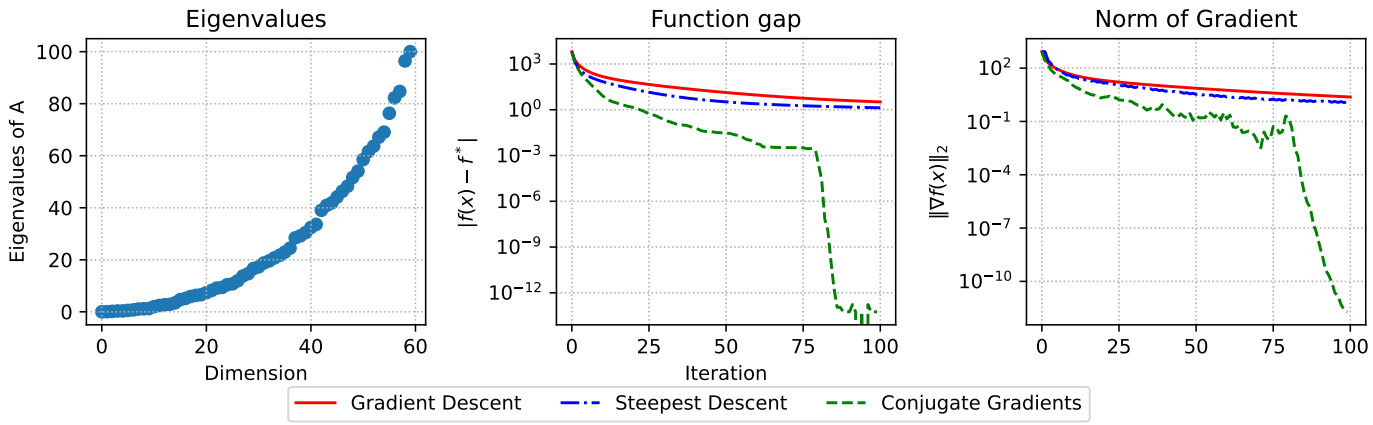
$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|x_0 - x^*\|_A,$$

где $\|x\|_A^2 = x^\top Ax$ и $\varkappa(A) = \frac{\lambda_1(A)}{\lambda_n(A)}$ - это число обусловленности матрицы A, $\lambda_1(A) \geq ... \geq \lambda_n(A)$ - собственные значения матрицы A

Примечание: Сравните коэффициент геометрической прогрессии с его аналогом в методе градиентного спуска.

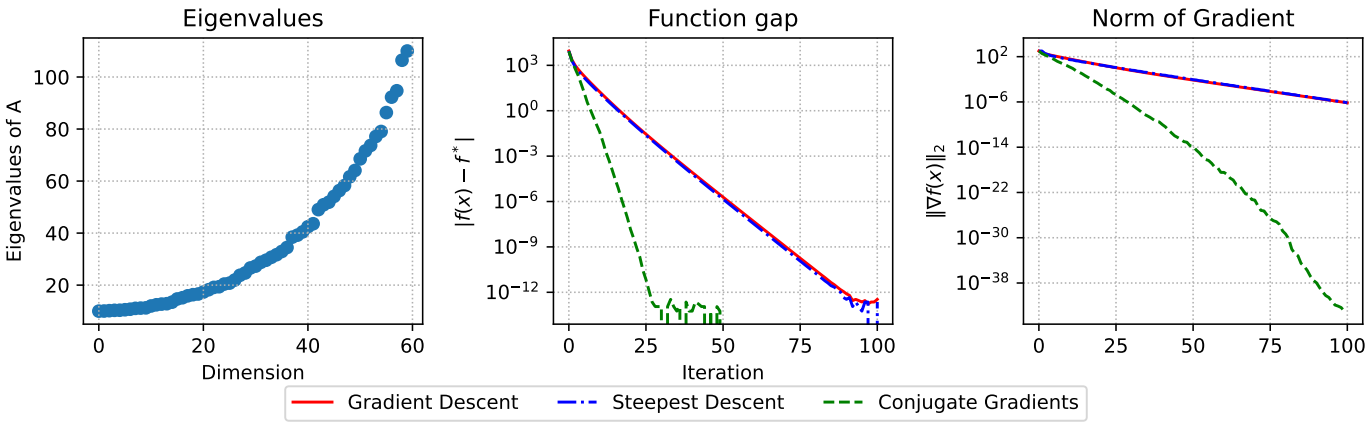
4.6 Численные эксперименты

$$f(x) = \frac{1}{2}x^TAx - b^Tx \to \min_{x\in\mathbb{R}^n}$$



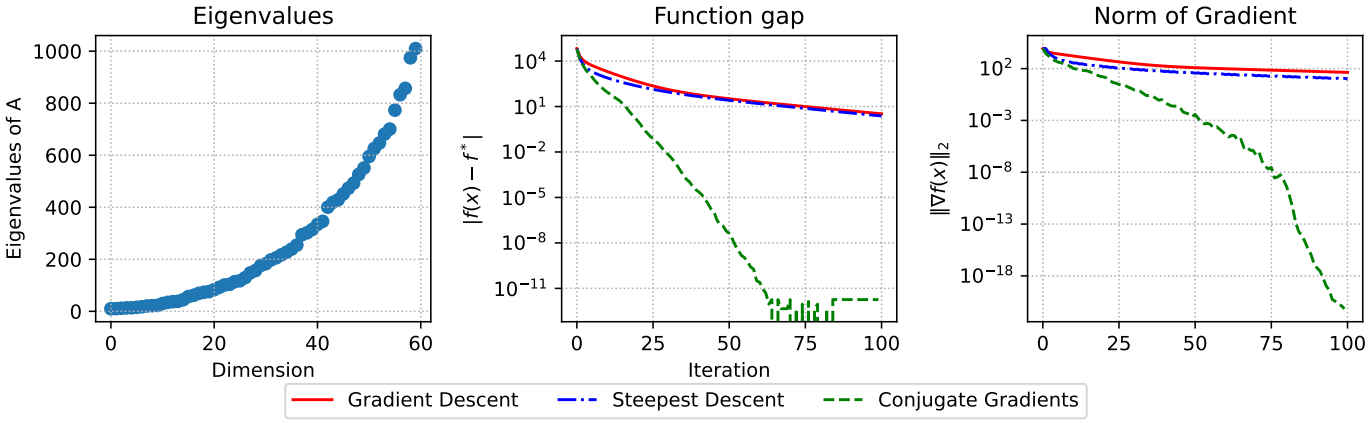
$$f(x) = \frac{1}{2}x^TAx - b^Tx \to \min_{x\in\mathbb{R}^n}$$

Strongly convex quadratics. n=60, random matrix.



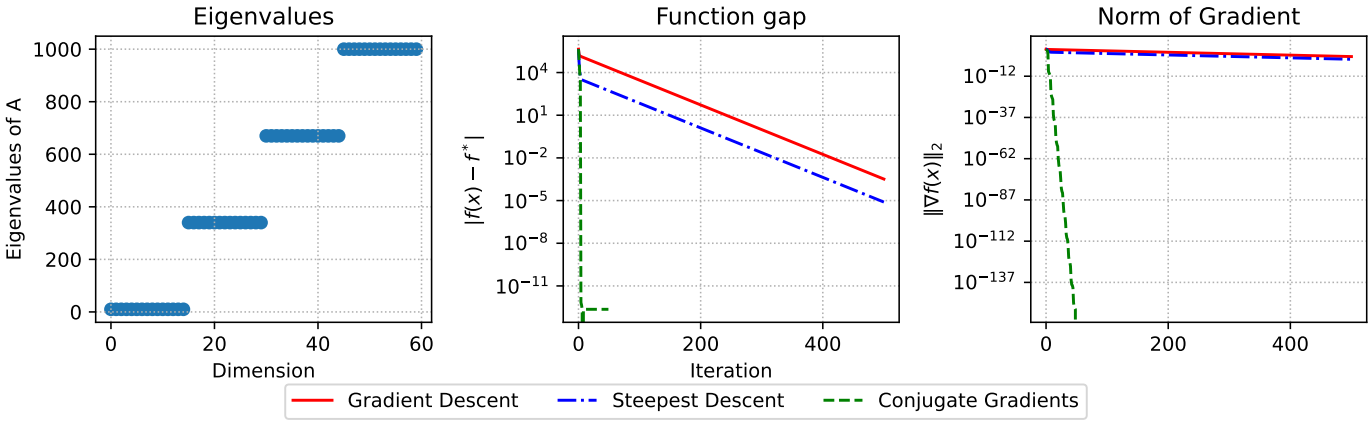
$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

Strongly convex quadratics. n=60, random matrix.



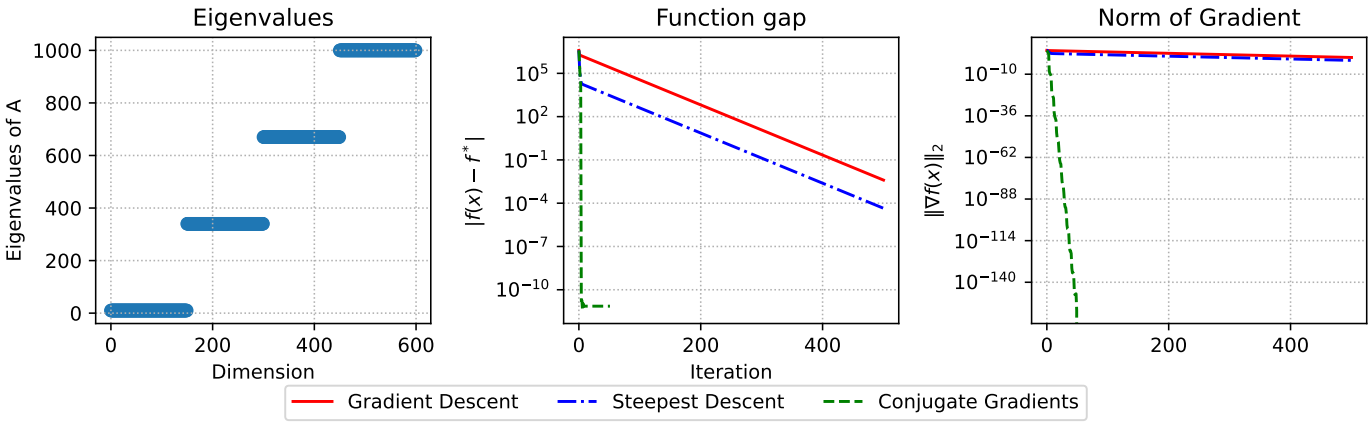
$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

Strongly convex quadratics. n=60, clustered matrix.



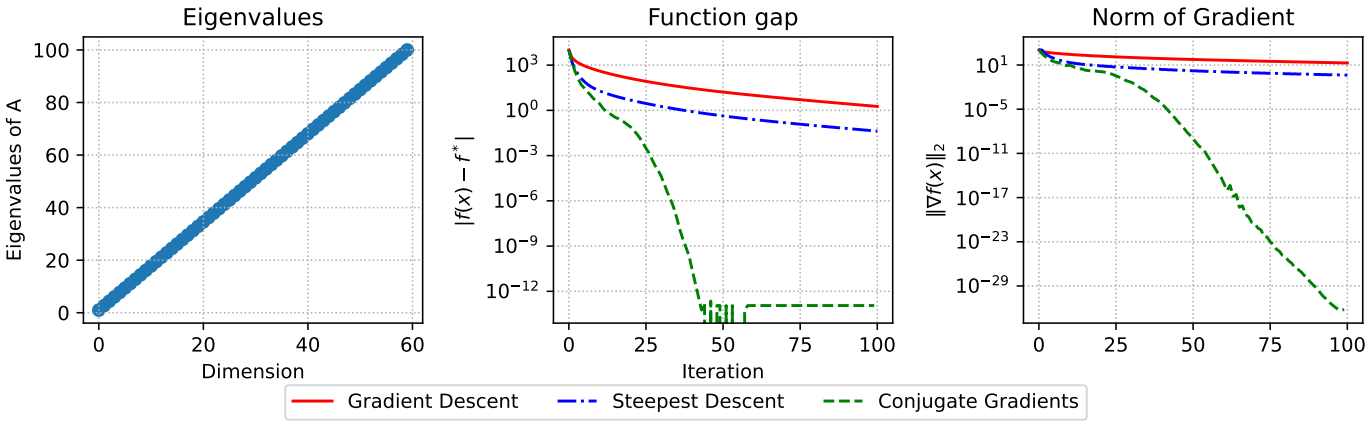
$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

Strongly convex quadratics. n=600, clustered matrix.

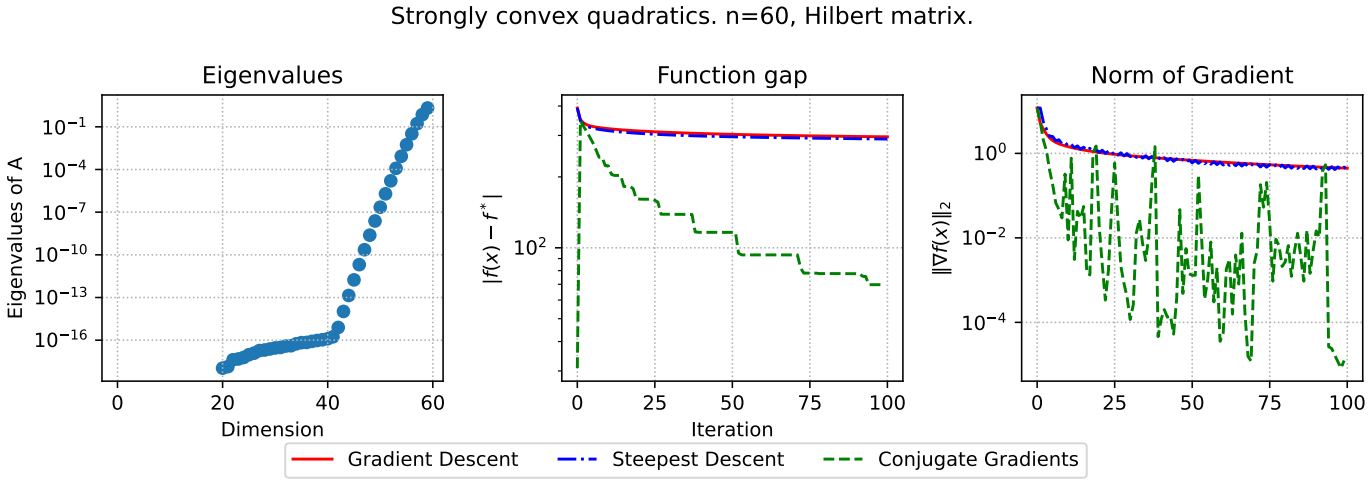


$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

Strongly convex quadratics. n=60, uniform spectrum matrix.



$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$



Посмотрим на видео с экспериментами в VS Code.

5 Метод сопряженных градиентов для неквадратичных задач (Non-linear CG)

В случае, когда нет аналитического выражения для функции или ее градиента, мы, скорее всего, не сможем решить одномерную задачу минимизации аналитически. Поэтому α_k подбирается обычной процедурой линейного поиска. Но для выбора β_k есть следующий математический трюк:

Для двух итераций справедливо:

$$x_{k+1} - x_k = cd_k,$$

где c - некоторая константа. Тогда для квадратичного случая мы имеем:

$$\nabla f(x_{k+1}) - \nabla f(x_k) = (Ax_{k+1} - b) - (Ax_k - b) = A(x_{k+1} - x_k) = cAd_k$$

Выражая из этого уравнения величину $Ad_k = \dfrac{1}{c}\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)$, мы избавляемся от функции в определении β_k, тогда пункт 4 будет переписан как:

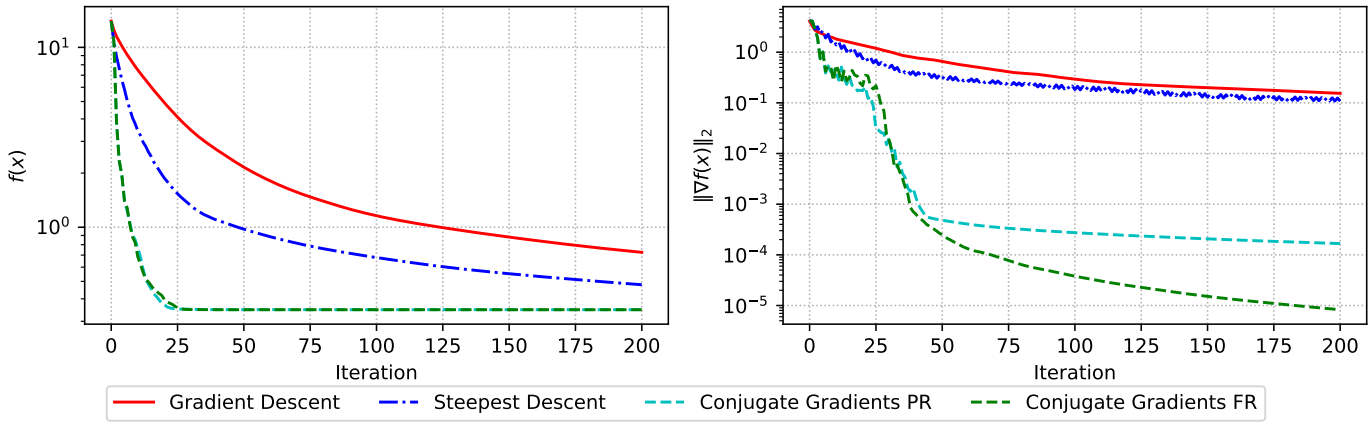
$$\beta_k = \frac{\nabla f(x_{k+1})^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{d_k^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}.$$

Этот метод называется методом Полака-Рибьера.

5.1 Численные эксперименты

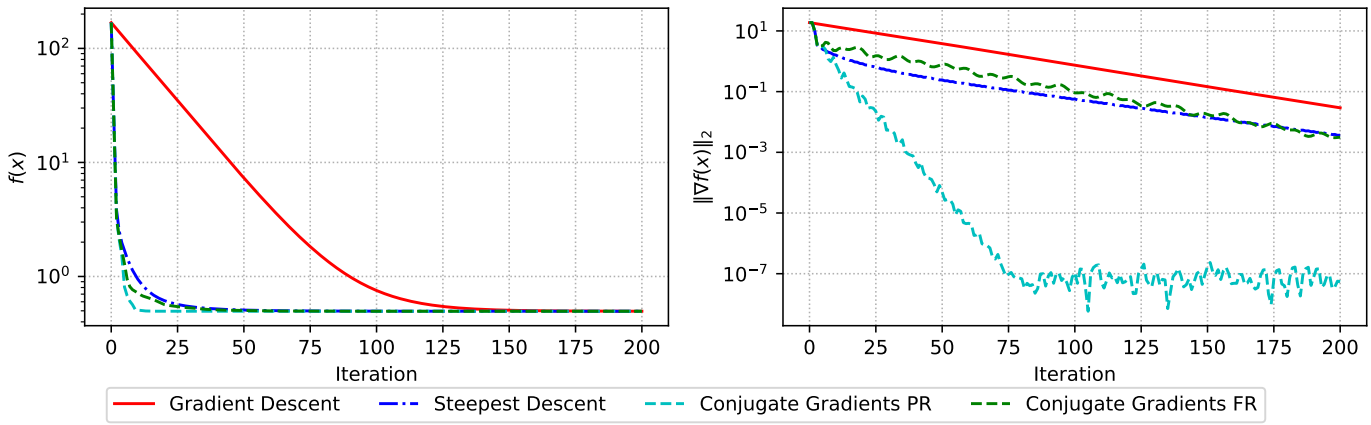
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \log(1 + \exp(-y_i \langle a_i, x\rangle)) \to \min_{x \in \mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=0



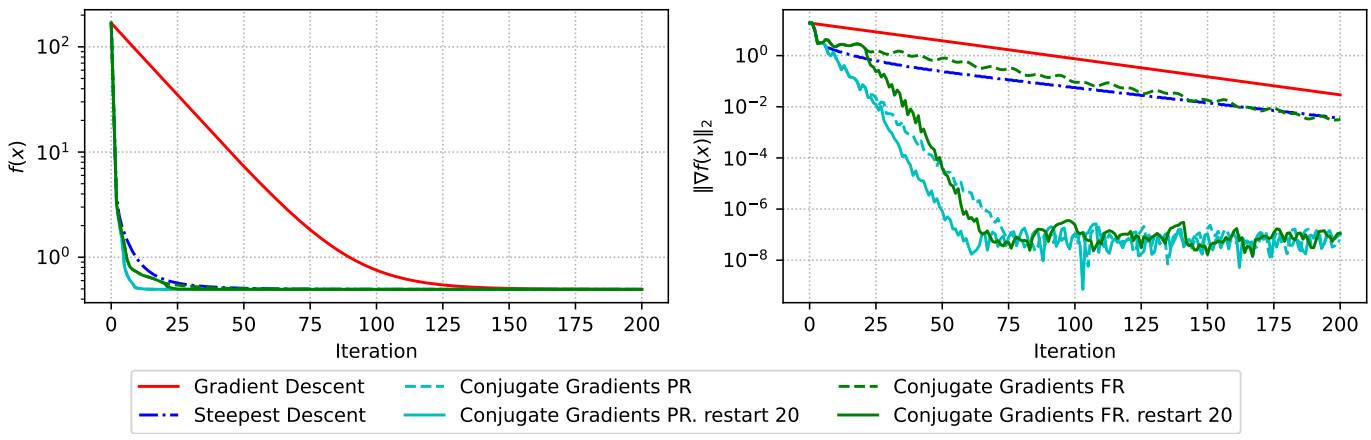
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=1



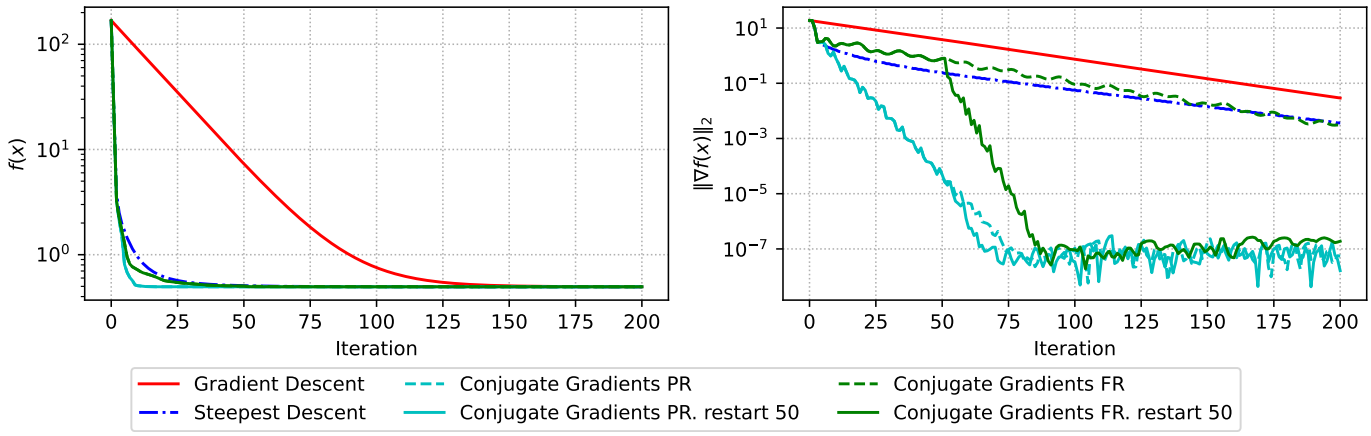
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=1



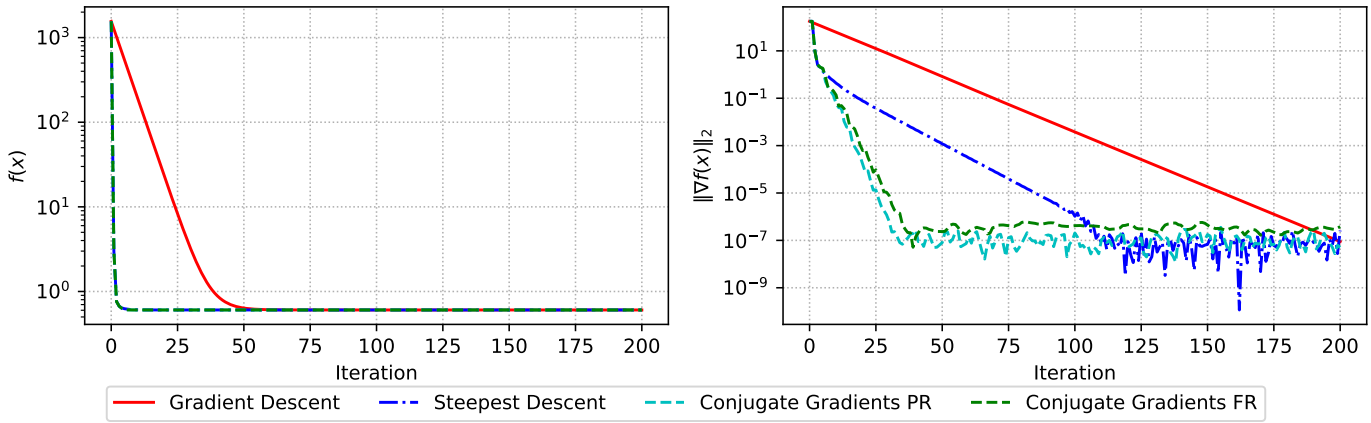
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=1

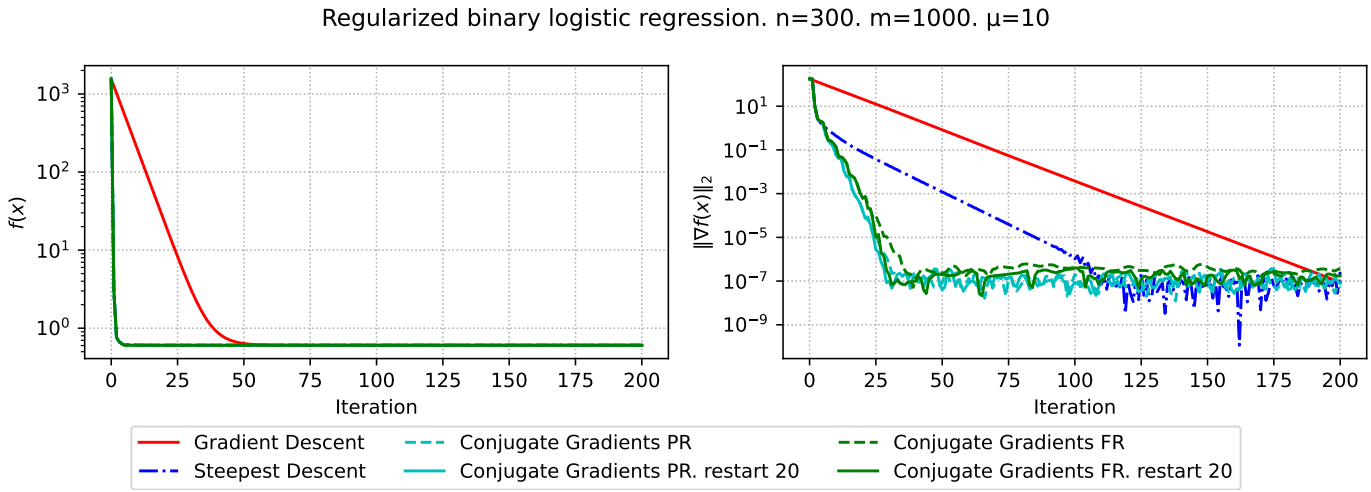


$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=10



$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$



6 Бонус: дополнительные технические леммы и доказательства

6.1 Леммы для сходимости

Лемма 3. Разложение ошибки.

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{7}$$

Доказательство

По определению

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j = x_0 - x^* + \sum_{j=0}^{i-1} \alpha_j d_j = -\sum_{j=0}^{n-1} \alpha_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j = \sum_{j=i}^{n-1} -\alpha_j d_j$$

Лемма 4. Невязка ортогональна всем предыдущим направлениям для CD.

Рассмотрим невязку метода сопряженных направлений на k итерации r_k, тогда для любого $i < k$:

$$d_i^T r_k = 0 \tag{8}$$

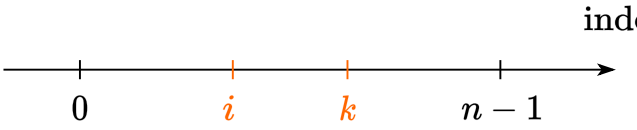
Доказательство

Запишем (7) для некоторого фиксированного индекса k:

$$e_k = \sum_{j=k}^{n-1} -\alpha_j d_j$$

Умножаем обе части на $-d_i^T A$·

$$-d_i^T A e_k = \sum_{j=k}^{n-1} \alpha_j d_i^T A d_j = 0$$



Таким образом, $d_i^T r_k = 0$ и невязка r_k ортогональна всем предыдущим направлениям d_i для метода CD.

> *Лемма 5. Невязки ортогональны друг другу в методе CG*
>
> Все невязки в методе CG ортогональны друг другу:
>
> $$r_i^T r_k = 0 \qquad \forall i \neq k \tag{9}$$

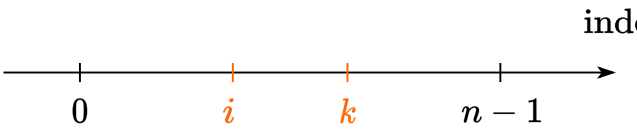
Доказательство

Запишем процесс Грама-Шмидта (2) с $\langle \cdot, \cdot \rangle$ замененным на $\langle \cdot, \cdot \rangle_A = x^T A y$

$$d_i = u_i + \sum_{j=0}^{i-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{10}$$

Тогда, мы используем невязки в качестве начальных векторов для процесса и $u_i = r_i$.

$$d_i = r_i + \sum_{j=0}^{i-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, r_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{11}$$



Умножаем обе части (10) на $r_k^T \cdot$ для некоторого индекса k:

$$r_k^T d_i = r_k^T u_i + \sum_{j=0}^{i-1} \beta_{ji} r_k^T d_j$$

Если $j < i < k$, то имеем лемму 4 с $d_i^T r_k = 0$ и $d_j^T r_k = 0$. Имеем:

$$r_k^T u_i = 0 \;\; \text{для CD} \quad r_k^T r_i = 0 \;\; \text{для CG}$$

Более того, если $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

и мы имеем для любого k (из-за произвольного выбора i):

$$r_k^T d_k = r_k^T u_k. \tag{12}$$

Лемма 6. Пересчет невязки

$$r_{k+1} = r_k - \alpha_k A d_k \tag{13}$$

$$r_{k+1} = -Ae_{k+1} = -A\left(e_k + \alpha_k d_k\right) = -Ae_k - \alpha_k A d_k = r_k - \alpha_k A d_k$$

Наконец, все эти вышеуказанные леммы достаточны для доказательства, что $\beta_{ji} = 0$ для всех i, j, кроме соседних.

Лемма 7. Коэффициенты для процесса Грама-Шмидта для CG

В процессе Грама-Шмидта для CG

$$\beta_{ji} = \frac{\langle r_i, r_i\rangle}{r_{i-1}, r_{i-1}},\ i = j + 1.$$

Все остальные коэффициенты равны нулю кроме $i = j$, но этот случай нам неинтересен.

Рассмотрим процесс Грам-Шмидта в методе CG

$$\beta_{ji} = -\frac{\langle d_j, u_i\rangle_A}{\langle d_j, d_j\rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Рассмотрим скалярное произведение $\langle r_i, r_{j+1}\rangle$ используя (13):

$$\begin{aligned}\langle r_i, r_{j+1}\rangle &= \langle r_i, r_j - \alpha_j A d_j\rangle = \langle r_i, r_j\rangle - \alpha_j\langle r_i, A d_j\rangle\\ \alpha_j\langle r_i, A d_j\rangle &= \langle r_i, r_j\rangle - \langle r_i, r_{j+1}\rangle\end{aligned}$$

1. Если $i = j$: $\alpha_i\langle r_i, A d_i\rangle = \langle r_i, r_i\rangle - \langle r_i, r_{i+1}\rangle = \langle r_i, r_i\rangle$. Этот случай не интересен по построению процесса Грам-Шмидта.
2. Соседний случай $i = j + 1$: $\alpha_j\langle r_i, A d_j\rangle = \langle r_i, r_{i-1}\rangle - \langle r_i, r_i\rangle = -\langle r_i, r_i\rangle$
3. Для любого другого случая: $\alpha_j\langle r_i, A d_j\rangle = 0$, потому что все невязки ортогональны друг другу.

Наконец, мы имеем формулу для $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j}\frac{\langle r_i, r_i\rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j}\frac{\langle r_i, r_i\rangle}{d_j^T A d_j} = \frac{\langle r_i, r_i\rangle}{\langle r_j, r_j\rangle} = \frac{\langle r_i, r_i\rangle}{\langle r_{i-1}, r_{i-1}\rangle}$$

И для направления $d_{k+1} = r_{k+1} + \beta_{k,k+1} d_k, \qquad \beta_{k,k+1} = \beta_k = \dfrac{\langle r_{k+1}, r_{k+1}\rangle}{\langle r_k, r_k\rangle}.$

7 Задачи

7.1 Задача 1

Реализуйте итерации метода сопряжённых градиентов для квадратичной задачи

$$f(x) = \frac{1}{2}x^T Ax - b^T x \longrightarrow \min_{x \in \mathbb{R}^n}$$

и запустите эксперименты для нескольких матриц A. Смотрите код здесь .

7.2 Задача 2

Реализуйте итерации метода Полака—Рибьера и запустите эксперименты для нескольких μ в бинарной логистической регрессии:

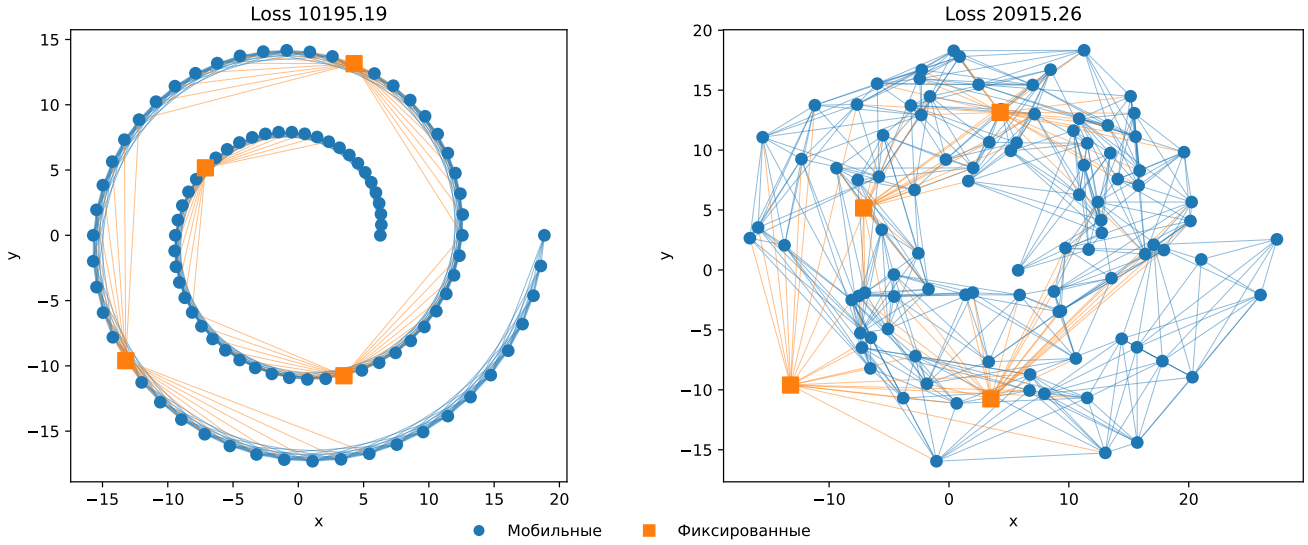
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} log(1 + exp(-y_i\langle a_i, x\rangle)) \longrightarrow \min_{x \in \mathbb{R}^n}$$

Смотрите код здесь .

8 Задачи на дом

8.1 Задача 1. Оптимальная расстановка роботов (19 баллов)

Вы будете решать простую квадратичную задачу оптимизации положения мобильных роботов, сравнивая **градиентный спуск (GD)** и **метод сопряжённых градиентов (CG)** для определения их оптимальных положений.



Есть команда роботов. Некоторые — **неподвижные** (фиксированные), остальные — **мобильные**. Некоторым роботам (на изображении выше между ними нарисованы ребра графа - связи) нужно обмениваться информацией друг с другом.

Для тех, кому нужно обмениваться, важно, чтобы они были близко друг к другу. Цена связи между двумя роботами растёт с расстоянием, поэтому мы минимизируем сумму квадратов расстояний между соединёнными вершинами графа, при этом координаты фиксированных роботов неизменны.

В этой задаче вам нужно будет показать, что поставленная задача может быть записана явно как задача минимизации квадратичной функции. После этого необходимо будет определить свойства этой функции (4 балла + 2 балла), реализовать метод градиентного спуска (3 балла) и метод сопряжённых градиентов (6 баллов) для решения этой задачи. Затем сравнить их работу (4 балла).

Детальное описание задачи с подпунктами и вспомогательным кодом доступно по ссылке.