# Содержание

# 1 Базовые определения

**Определение 1.1.** Множество $\mathcal{X}$ называется выпуклым, если для любых $x, y \in \mathcal{X}$ и для любого $\lambda \in [0, 1]$ следует, что

$$\lambda x + (1 - \lambda)y \in \mathcal{X}$$

**Определение 1.2.** Произвольная функция называется выпуклой, если для любых $x, y \in \mathbb{R}^d$ и для любого $\lambda \in [0, 1]$ выполнено

$$f(\lambda x + (1 - \lambda)y) \leqslant \lambda f(x) + (1 - \lambda)f(y)$$

**Определение 1.3.** Непрерывно дифференцируемая на $\mathbb{R}^d$ функция называется $\mu$-сильно выпуклой ($\mu > 0$), если для любых $x, y \in \mathbb{R}^d$ выполнено

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|_2^2$$

**Теорема 1.1.** *Критерий сильной выпуклости дважды непрерывно дифференцируемой функции.*

*Пусть функция $f : \mathbb{R}^d \to \mathbb{R}$ дываждны непрерывно дифференцируема на $\mathbb{R}^d$. Тогда функция $f$ является $\mu$-сильно выпуклой тогда и только тогда, когда для любого $x \in \mathbb{R}^d$ выполнено*

$$\nabla^2 f(x) \succeq \mu I$$

**Определение 1.4.** Для $f : \mathbb{R}^n \to \mathbb{R}$ функция $f^* : \mathbb{R}^n \to \mathbb{R}$, определённая как

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \{\langle x, y \rangle - f(x)\}$$

называется сопряжённой функией к $f$.

**Определение 1.5.** Задачей оптимизации стандартной формы называется

$$\min_x f_0(x) \tag{1}$$

$$\text{s.t. } f_i(x) \leqslant 0, i = \overline{1, m} \tag{2}$$

$$h_j(x) = 0, j = \overline{1, n} \tag{3}$$

**Определение 1.6.** Лагранжиан $L$ относительно задачи оптимизации (1) задаётся следующим образом:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{n} \nu_j h_j(x)$$

**Определение 1.7.** Определим двойственную функцию по Лагранжу (или просто двойственную функцию) $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ следующим образом:

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu)$$

**Определение 1.8.** Двойственной к (1) называется следующая задача:

$$\max_{\lambda, \nu} g(\lambda, \nu) \tag{4}$$

$$\text{s.t. } \lambda \succeq 0 \tag{5}$$

**Определение 1.9.** Обозначим оптимальное значение двойственной задачи относительно начальной как $d^*$. А оптимальнное значение исходной как $p^*$.

Заметим, что в силу произвольности выбора двойственных переменных всегда выполняется

$$d^* \leqslant p^*$$

данное неравенство называется слабой двойственностью.

В частности, когда

$$d^* = p^*$$

то будем говорить, что выполняется свойство сильной двойственности.

**Определение 1.10.** Рассмотрим задачу следующего вида:

$$\min_x f_0(x) \tag{6}$$

$$\text{s.t. } f_i(x) \leqslant 0, i = \overline{1,m} \tag{7}$$

$$Ax = b \tag{8}$$

Будем говорить, что для такой задачи выполняется условие Слейтера, если $\exists x \in \text{relint } D$, такой что

$$f_i(x) < 0, i = \overline{1,m}$$

$$Ax = b$$

**Теорема 1.2.** *Теорема Слейтера.*

*Если для задачи* (6) *выполняется условие Слейтера, то тогда при построении двойственной задачи выполняется свойство сильной двойственности.*

**Определение 1.11.** Необходимым условием набора прямых и двойственных переменных в совокупности являются условия Каруша-Куна-Такера:

$$f_i(x^*) \leqslant 0, i = \overline{1,m} \tag{9}$$

$$h_j(x^*) = 0, j = \overline{1,n} \tag{10}$$

$$\lambda_i^* \geqslant 0, i = \overline{1,m} \tag{11}$$

$$\lambda_i^* f_i(x^*) = 0, i = \overline{1,m} \tag{12}$$

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{j=1}^{n} \nu_i^* \nabla h_j(x^*) = 0 \tag{13}$$

В постановке, когда $f_i$ выпуклые, а $h_j$ афинные, это условие является достаточным.

**Определение 1.12.** Заача линейного программирования в общем виде представима в виде:

$$\min_{x \in \mathbb{R}^n} c^T x$$

$$\text{s.t. } Ax = b$$

$$Gx \leqslant h$$

где $A \in \mathbb{R}^{m \times n}, G \in \mathbb{R}^{k \times n}$

**Определение 1.13.** Задачей линейного программирования в стандартной форме называется задача вида:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & c^T x \\ \text{s.t. } & Ax = b \\ & x \geqslant 0 \end{aligned}$$

**Определение 1.14.** Задача квадратичного программирования записывается в виде:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \frac{1}{2} x^T A x + c^T x \\ \text{s.t. } & Ex = f \\ & Gx \leqslant h \end{aligned}$$

где $A \in \mathbb{S}^n_+, E \in \mathbb{R}^{m \times n}, G \in \mathbb{R}^{k \times n}$

**Определение 1.15.** Задача с конусами второго порядка (SOCP) записывается в следующем виде:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & c^T x \\ \text{s.t. } & Ax = b \\ & \|G_i x - h_i\|_2 \leqslant e_i^T x + f_i, i = \overline{1, M} \end{aligned}$$

где $A \in \mathbb{R}^{m \times n}, G_i \in \mathbb{R}^{k_i \times n}, i = \overline{1, M}$. Собственно, последнее ограничение и означает, что пары $(G_i x - h_i, e_i^T x + f_i)$ лежат в конусах второго порядка $K_2 = \left\{(y, t) \in \mathbb{R}^{k_i} \times \mathbb{R}_+ \mid \|y\| \leqslant t\right\}$

**Определение 1.16.** Задача полуопределённого программирования (SDP) в стандартном виде записывается следующим образом:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & c^T x \\ \text{s.t. } & Ax = b \\ & F_0 + \sum_{i=1}^{n} F_i x_i \succeq 0 \end{aligned}$$

где $A \in \mathbb{R}^{m \times n}, F_j \in S^n, j = \overline{0, n}$

**Определение 1.17.** Стандартная форма SDP имеет вид:

$$\begin{aligned} \min_{X \in \mathbb{S}^n} & \text{tr } (CX) \\ \text{s.t. } & \text{tr } (A_i X) = b_i, i = \overline{1, m} \\ & X \succeq 0 \end{aligned}$$

**Определение 1.18.** Субградиентом функции $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ в точке $x_0 \in \text{dom } f$ называется $g \in \mathbb{R}^n$:

$$\forall y \in \mathbb{R}^n : \ f(y) \geqslant f(x_0) + \langle g, y - x_0 \rangle$$

**Определение 1.19.** Множеством всех субградиентов в точке $x_0$ функции $f$ называется субдифференциалом. Обозначение $\partial f(x_0)$.

# 2 О минимумах и свойствах функций

**Теорема 2.1.** *Пусть $x^*$ – локальный минимум функции $f$ на $\mathbb{R}^d$, тогда если $f$ дифференцируема, то $\nabla f(x^*) = 0$.*

*Доказательство.* НА ДВА БАЛЛА.

Пойдём от противного и предположим, что $x^*$ – локальный минимум, но $\nabla f(x^*) \neq 0$. Разложим функцию $f$ в ряд в окрестности локального минимума:

$$f(x) = f(x^*) + \langle \nabla f(x^*), x - x^* \rangle + o(\|x - x^*\|_2), x \to x^*$$

Рассмотрим $x_\lambda = x^* - \lambda \nabla f(x^*)$. Найдём $\lambda_1 > 0$ такое, что для любого $0 < \lambda \leqslant \lambda_1$ можно гарантировать, что $\|x_\lambda - x^*\|_2 \leqslant r$, то есть $x_\lambda$ попадает в нужную окрестность из определения локального минимума.

Тогда по определению локального минимума:

$$\forall \lambda, 0 < \lambda \leqslant \lambda_1 : \ f(x_\lambda) \geqslant f(x^*)$$

При этом разложение в ряд для точек $x_\lambda$ имеет вид:

$$f(x_\lambda) = f(x^*) + \langle \nabla f(x^*), x_\lambda - x^* \rangle + o(\|x_\lambda - x^*\|_2) = f(x^*) - \lambda \|\nabla f(x^*)\|_2^2 + o(\lambda \|\nabla f(x^*)\|_2)$$

Найдём достаточно малое $\lambda_2$:

$$|o(\lambda \|\nabla f(x^*)\|_2)| \leqslant \frac{\lambda}{2} \|\nabla f(x^*)\|_2^2$$

Тогда

$$\forall \lambda \leqslant \min\{\lambda_1, \lambda_2\} : \ f(x_\lambda) \leqslant f(x^*) - \frac{\lambda}{2} \|\nabla f(x^*)\|^2$$

Пришли к противоречию, что $x^*$ – локальный минимум. А значит $\nabla f(x^*) = 0$. □

**Теорема 2.2.** *Пусть дана выпуклая непрерывно дифференцируемая на $\mathbb{R}^d$ функция $f : \mathbb{R}^d \to \mathbb{R}$. Если для некоторой точки $x^* \in \mathbb{R}^d$ верно, что $\nabla f(x^*) = 0$, то $x^*$ – глобальный минимум $f$ на всём $\mathbb{R}^d$.*

*Доказательство.* НА ДВА БАЛЛА.

Достаточно записать определение выпуклости:

$$f(x) \geqslant f(x^*) + \langle \nabla f(x^*), x - x^* \rangle = f(x^*)$$

Получается, что равенство градиента нулю является достаточным условием глобального минимума.

В обратную сторону уже доказывали выше для произвольных функций. □

**Теорема 2.3.** *Пусть дана выпуклая непрерывно дифференцируемая на $\mathbb{R}^d$ функция $f : \mathbb{R}^d \to \mathbb{R}$ и выпуклое множество $\mathcal{X}$. Тогда $x^* \in \mathcal{X}$ – глобальный минимум $f$ на $\mathcal{X}$ тогда и только тогда, когда для всех $x \in \mathcal{X}$ выполнено*

$$\langle \nabla f(x^*), x - x^* \rangle \geqslant 0$$

*Доказательство.* НА ДВА БАЛЛА.

Достаточность.

Пусть $\langle \nabla f(x^*), x - x^* \rangle \geqslant 0$ для $x \in \mathcal{X}$, тогда воспользуемся определением выпуклости:

$$f(x) \geqslant f(x^*) + \langle \nabla f(x^*), x - x^* \rangle \geqslant f(x^*)$$

Откуда следует, что $x^*$ – глобальный минимум на $\mathcal{X}$.

Необходимость. Предположим, что существует $x \in \mathcal{X}$ такой, что $\langle \nabla f(x^*), x - x^* \rangle < 0$. Рассмотрим точки вида

$$x_\lambda = \lambda x + (1 - \lambda)x^*, \lambda \in [0, 1]$$

В силу выпуклости множества $\mathcal{X}$ точки $x_\lambda \in \mathcal{X}$. Посмотрим, как ведёт себя функция $\varphi(\lambda) = f(x_\lambda) = f(\lambda x + (1 - \lambda)x^*)$. В частности:

$$\frac{d\varphi}{d\lambda} = \langle \nabla f(x^* + \lambda(x - x^*)), x - x^* \rangle$$

Заметим, что $\frac{d\varphi}{d\lambda}|_{\lambda=0} = \langle \nabla f(x^*), x - x^* \rangle < 0$. Это значит, что функция $\varphi$ убывает в окрестности нуля. А значит для достаточно малых $\lambda > 0$ выполнено:

$$f(x^* + \lambda(x - x^*)) = \varphi(\lambda) < \varphi(0) = f(x^*)$$

Противоречие. □

**Определение 2.1.** Пусть дана непрерывно дифференцируемая на $\mathbb{R}^d$ функция $f : \mathbb{R}^d \to \mathbb{R}$. Будем говорить, что данная функция имеет $L$-Липшицев градиент (говорить, что она является $L$-гладкой), если для любых $x, y \in \mathbb{R}^d$ выполнено

$$\|\nabla f(x) - \nabla f(y)\|_2 \leqslant L\|x - y\|_2$$

**Теорема 2.4.** *НА ДВА БАЛЛА.*

*Пусть дана $L$-гладкая функция $f : \mathbb{R}^d \to \mathbb{R}$. Тогда для любых $x, y \in \mathbb{R}^d$ выполнено*

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leqslant \frac{L}{2}\|x - y\|_2^2$$

*Доказательство.* Для доказательства будем интегрировать по кривой $r(\tau) = x + \tau(y - x), \tau \in [0, 1]$. Тогда

$$f(y) - f(x) = \int_0^1 \langle \nabla f(x + \tau(y - x)), y - x \rangle d\tau =$$

$$\langle \nabla f(x), y - x \rangle + \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x), y - x \rangle d\tau$$

Переместив скалярное произведение влево и взяв модуль от обоих частей, получим

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| =$$

$$\left| \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x), y - x \rangle d\tau \right| \leqslant \int_0^1 |\langle \nabla f(x + \tau(y - x)) - \nabla f(x), y - x \rangle| d\tau$$

Далее воспользуемся неравенством КБШ, а затем $L$-гладкостью:

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leqslant \int_0^1 \|\nabla f(x + \tau(y - x)) - \nabla f(x)\|_2 \|y - x\|_2 d\tau \leqslant$$

$$L\|y - x\|_2^2 \int_0^1 \tau d\tau = \frac{L}{2}\|x - y\|_2^2$$

□

**Теорема 2.5.** *НА ДВА БАЛЛА.*

*Пусть дана выпуклая функция $f : \mathbb{R}^d \to \mathbb{R}$. Тогда функция $f$ является $M$-Липшицевой тогда и только тогда, когда*

$$\forall x \in \mathbb{R}^d \, \forall g \in \partial f(x) : \ \|g\|_2 \leqslant M$$

*Доказательство.* $\Rightarrow$ Пусть дополнительно к выпуклости функция $f$ ещё и $M$-Липшицева, тогда рассмотрим $g \in \partial f(x)$, по определению субградиента:

$$\forall y \in \mathbb{R}^d : \ f(y) - f(x) \geqslant \langle g, y - x \rangle$$

Из липшицевости $f$:

$$f(y) - f(x) \leqslant M\|y - x\|_2$$

Взяв $y = g + x$ и объединив прошлые неравенства получим:

$$M\|g\|_2 = M\|y - x\|_2 \geqslant \langle g, y - x \rangle = \|g\|_2^2$$

$\Leftarrow$

Пусть дополнительно к выпуклости $f$ все субградиенты равномерно ограничены: $\|g\|_2 \leqslant M$ для любого $x \in \mathbb{R}^d, g \in \partial f(x)$. Тогда рассмотрим $g \in \partial f(x)$, по выпуклости и определению субградиента:

$$\forall y \in \mathbb{R}^d : \ f(y) - f(x) \leqslant \langle g, x - y \rangle$$

КБШ:

$$f(y) - f(x) \leqslant \|g\|_2\|x - y\|_2$$

Пользуемся предположением и получаем:

$$f(y) - f(x) \leqslant M\|x - y\|_2$$

□

# 3 О градиентном спуске

**Определение 3.1.** Итерация метода градиентного спуска:

$$\begin{aligned} \text{compute } \nabla f(x^k) \\ x^{k+1} = x^k - \gamma_k \nabla f(x^k) \end{aligned}$$

Интуиция метода градиентного спуска заключается в том, что мы идём против направления наибольшего роста, то есть в направление наибольшего уменьшения значения целевой функции.

Шаг нужен, чтобы метод вообще сходился, в одном из доказательств было представлено неравенство:

$$\|x^{k+1} - x^*\|_2^2 \leqslant (1 - 2\gamma_k\mu + \gamma_k^2 L^2)\|x^k - x^*\|_2^2$$

Для сходимости нужно, чтобы $0 < (1 - 2\gamma_k\mu + \gamma_k^2 L^2) < 1$. То есть шагом мы контролируем сходимость.

НА ОДИН БАЛЛ: сходимость линейная.

НА ДВА БАЛЛА:

**Теорема 3.1.** *Пусть задача безусловной оптимизации с $L$-гладкой, $\mu$-сильно выпуклой целевой функцией $f$ решается с помощью градиентного спуска. Тогда справедлива следующая оценка сходимости*

$$\|x^K - x^*\|_2^2 \leqslant \left(1 - \frac{\mu}{L}\right)^K \|x^0 - x^*\|_2^2$$

*Более того, чтобы добиться точности $\varepsilon$ по аргументу, необходимо*

$$K = O\left(\frac{L}{\mu} \log \frac{\|x^0 - x^*\|_2}{\varepsilon}\right)$$

*итераций.*

**Теорема 3.2.** *НА ДВА БАЛЛА:*

*В случае $L$-гладких, выпуклых задач количество итераций, необходимых для точности $\varepsilon$ по аргументу увеличивается до*

$$K = O\left(\frac{L\|x^0 - x^*\|_2^2}{\varepsilon}\right)$$

# 4 О методе тяжёлого шарика

**Определение 4.1.** Итерация метода тяжёлого шарика выглядит, как:

$$\begin{aligned} &\text{compute } \nabla f(x^k) \\ x^{k+1} &= x^k - \gamma_k \nabla f(x^k) + \tau_k(x^k - x^{k-1}) \end{aligned}$$

где $\{\tau_k\}_{k=0} \subset [0, 1]$ называются моментумами.

**Определение 4.2.** Итерация ускоренного градиентного метода выглядит, как:

$$\begin{aligned} &\text{compute } \nabla f(y^k) \\ x^{k+1} &= y^k - \gamma_k \nabla f(y^k) \\ y^{k+1} &= x^{k+1} + \tau_k(x^{k+1} - x^k) \end{aligned}$$

Данные методы имеют следующую интуицию, связанную с моментумами: мы запоминаем, как ходили в прошлый раз, чтобы сохранить некую инерцию, которая позволит нам сильно не отклоняться от траектории.

В ускоренном же методе ещё используется стратегия взгляда вперёд – инерцию считаем не по сравнению с прошлым, а будущим.

Типично $\tau_k$ берут близким к единице или устремляем к единице.

Стоит отметить, что характер сходимости для $L$-гладких, $\mu$-сильно выпуклых функций также линейный, как и у градиентного спуска, но на практике работают лучше.

**Теорема 4.1.** *НА ДВА БАЛЛА.*

*Пусть задача безусловной оптимизации с $L$-гладкой, $\mu$-сильно выпуклой целевой функцией $f$ решается с помощью ускоренного градиентного метода. Тогда при $\gamma_k = \frac{1}{L}, \tau_k = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ справедлива следующая оценка сходимости:*

$$f(x^K) - f(x^*) \leqslant \left(1 - \sqrt{\frac{\mu}{L}}\right)^K L\|x^0 - x^*\|_2^2$$

**Теорема 4.2.** *НА ДВА БАЛЛА.*

*Для любого метода из класса, описанного выше, существует безусловная задача оптимизации с $L$-гладкой, $\mu$-сильно выпуклой целевой функцией $f$ такая, что для решения этой задачи методу необходимо*

$$\Omega\left(\sqrt{\frac{L}{\mu}}\log\frac{\|x^0-x^*\|_2}{\varepsilon}\right)$$

*вызовов оракула.*

# 5 О сопряжённых направлениях

**Определение 5.1.** Множество векторов $\{p_i\}_{i=0}^{n-1}$ будем называть сопряжёнными относительно положительно определённой матрицы $A$, если для любых $i \neq j \in \{0, \cdots, n-1\}$ следует

$$p_i^T A p_j = 0$$

**Теорема 5.1.** *Сопряжённые вектора являются линейно независимыми.*

Интуиция метода заключаются в том, что мы сможем быстро находить решение уравнения $Ax-b=0$, решение которого эквивалентно поиску минимума квадратичной задачи.

Как это работает? Начинаем с какого-то $x^0$, ищем "проекцию"того, насколько мы далеко от оптимального значения $(Ax^0-b)$ на один из сопряжённых векторов и прибавляем её. Сделав эту итерацию для всех сопряжённых векторов мы занулимся по всем проекциям, то есть найдём оптимальное значение.

Для поиска проекции используем формулу:

$$\alpha_k = -\frac{p_k^T r_k}{p_k^T A p_k}$$

Здесь введено обозначение $r_k = Ax^k - b$. Как вы можете увидеть, формула очень похожа на классическое взятие проекции в евклидовом пространстве

$$\frac{(e_i, \vec{v})}{(e_i, e_i)}$$

Осталось понять как находить сопряжённые вектора, но по программе нам это не надо знать, поэтому направляю любознательного читателя смотреть лекцию.

**Теорема 5.2.** *Характер сходимости для систем линейных уравнений с положительно определённой матрицей.*

*Метод сопряжённых градиентов для решения системы линейных уравнений с квадратной положительно определённой матрицей размера $d$ находит точное решение не более чем за $r$ итераций, где $r$ – число уникальных собственных значений матрицы.*

**Теорема 5.3.** *НА ДВА БАЛЛА.*

*Метод сопряжённых градиентов для решения системы линейных уравнений с квадратной положительно определённой матрицей размера $d$ имеет следующую оценку сходимости:*

$$\|x^k - x^*\|_A \leqslant 2\left(\frac{\sqrt{k(A)}-1}{\sqrt{k(A)}+1}\right)^k \|x^0 - x^*\|_A$$

*Здесь $\|x\|_A^2 = x^T A x$ и $k(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$*

# 6 О методе Ньютона

**Определение 6.1.** Итерация классического метода Ньютона выглядит, как:

$$x^{k+1} = x^k - \left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k)$$

Интуиция метода заключается в том, что градиентный спуск работает с линейной аппроксимацией в текущей точке, а метод Ньютона – с квадратичной.

**Теорема 6.1.** *Пусть задача безусловной оптимизации с $\mu$-сильно выпуклой целевой функцией $f$ с $M$-Липшецевым гессианом решается методом Ньютона. Тогда справедлива следующая оценка сходимости за 1 итерацию: (ФОРМУЛА НА ДВА БАЛЛА, ТОЛЬКО ХАРАКТЕР СХОДИМОСТИ НА ОДИН)*

$$\|x^{k+1} - x^*\|_2 \leqslant \frac{M}{2\mu}\|x^k - x^*\|_2^2$$

*Мы уже знаем, что такого рода оценки дают квадратичную скорость сходимости.*

Хочется заменить $(\nabla^2 f(x^k))^{-1}$ на что-то более дешёвое с точки зрения вычислений. Выудим свойства присущие гессиану:

- $x^{k+1} - x^k \approx (\nabla^2 f(x^{k+1}))^{-1}(\nabla f(x^{k+1}) - \nabla f(x^k))$
- Гессиан – симметричная матрица

Заменим $(\nabla^2 f(x^{k+1}))^{-1}$ на $H_{k+1}$, введём обозначения $s^k = x^{k+1} - x^k$ и $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$. Получим квазиньютоновское уравнение:

$$s^k = H_{k+1} y^k$$

НА ДВА БАЛЛА: Способы получения глобальной сходимости:

- Ввести шаг – демпфированный метод:

  $$x^{k+1} = x^k - \gamma_k (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$$

  Как выбирать шаг? Как угодно: константа, уменьшащийся, линейным поиском.

- Оценка сверху:

  $$x^{k+1} = \text{argmin}_{\,x \in \mathbb{R}^d} \left( f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2}\langle x - x^k, \nabla^2 f(x^k)(x - x^k)\rangle + \frac{M}{6}\|x - x^k\|_2^3 \right)$$

  Здесь $M$ – константа Липшица гессиана. Такой метод называется кубическим методом Ньютона.

НА ДВА БАЛЛА: Метод SR1 предлагает искать $H_{k+1}$ как одноранговую, используем следующий пересчёт:

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k)(s^k - H_k y^k)^T}{(s^k - H_k y^k)^T y^k}$$

НА ДВА БАЛЛА:

Также можно искать $H$, как решение следующей задачи условной оптимизации:

$$\begin{aligned} H_{k+1} = \text{argmin}_{H \in \mathbb{R}^{d \times d}} \|H - H_k\|^2 \\ \text{s.t. } s^k = Hy^k \\ H^T = H \end{aligned}$$

Норму можно брать любую, выбор взвешенной нормы Фробениуса

$$\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$$

где должно выполняться $Wy^k = s^k$ даёт метод BFGS. Каждый раз решать задачу оптимизации нам не придётся, явный вид метода давно известен:

$$H_{k+1} = (I - \rho_k s^k (y^k)^T) H_k (I - \rho_k y^k (s^k)^T) + \rho_k s^k (s^K)^T$$

где $\rho = \frac{1}{(y^k)^T s^k}$

# 7 Об оптимизации на простых множествах

**Определение 7.1.** Евклидовой проекцией точки $y$ на множество $\mathcal{X}$ является решение следующей задачи оптимизации:

$$\Pi_{\mathcal{X}}(y) = \text{argmin}_{x \in \mathcal{X}} \|x - y\|_2^2$$

**Определение 7.2.** Итерация метода градиентного спуска с проекцией выглядит следующим образом:

$$\begin{aligned} \text{compute } \nabla f(x^k) \\ x^{k+1} = \Pi_{\mathcal{X}}(x^k - \gamma_k \nabla f(x^k)) \end{aligned}$$

Интуиция метода очень простая – после каждого шага старого доброго градиентного спуска применяем проекцию, чтобы не выйти за пределы множества.

**Теорема 7.1.** *Метод градиентного спуска с проекцией для $L$-гладкой и $\mu$-сильно выпуклой целевой функции имеет такую же сходимость, что и метод градиентного спуска для аналогичной безусловной задачи оптимизации.*

**Определение 7.3.** Итерация метода Франка-Вульфа выглядит следующим образом:

$$\begin{aligned} \text{compute } \nabla f(x^k) \\ \text{find } s^k = \text{argmin}_{s \in \mathcal{X}} \langle s, \nabla f(x^k) \rangle \\ \gamma_k = \frac{2}{k+2} \\ x^{k+1} = (1 - \gamma_k) x^k + \gamma_k s^k \end{aligned}$$

Интуиция метода заключается в том, что мы ищем минимум линейной аппроксимации функции на требуемом множестве (уголок). Следующая точка выбирается, как усреднение этих уголков, из-за чего уголки, которые ближе к решению, будут выбираться чаще.

**Теорема 7.2.** *Пусть дана непрерывно дифференцируемая выпуклая $L$-гладкая функция $f : \mathbb{R}^d \to \mathbb{R}$, тогда для метода Франк-Вульфа справедлива следующая оценка сходимости: НА ДВА БАЛЛА ФОРМУЛА, НА ОДИН ПРОСТО ЗНАТЬ ХАРАКТЕР СХОДИМОСТИ*

$$f(x^K) - f(x^*) \leqslant \frac{\max\{2Ldiam\ (\mathcal{X})^2, f(x^0) - f(x^*)\}}{K+2}$$

*где $diam\ (\mathcal{X}) := \max_{x,\, y \in \mathcal{X}} \|x - y\|$ – диаметр множества $\mathcal{X}$.*

*То есть сублинейная сходимость, как и у градиентного спуска для выпуклой $L$-гладкой функции.*

**Теорема 7.3.** *НА ДВА БАЛЛА.*

*Свойства оператора проекции:*

- *Для выпуклого множества $\mathcal{X}$ и любой точки оператор проекции существует и принимает единственное значение.*
- *Пусть $\mathcal{X} \subseteq \mathbb{R}^d$ – выпуклое множество, $x \in \mathcal{X}, y \in \mathbb{R}^d$. Тогда*

$$\langle x - \Pi_{\mathcal{X}}(y), y - \Pi_{\mathcal{X}}(y)\rangle \leqslant 0$$

- *Пусть $\mathcal{X} \subseteq \mathbb{R}^d$ – выпуклое множество, $x_1, x_2 \in \mathbb{R}^d$. Тогда*

$$\|\Pi_{\mathcal{X}}(x_1) - \Pi_{\mathcal{X}}(x_2)\| \leqslant \|x_1 - x_2\|_2$$

- *Для $x^*$ – решения условной задачи минимизации выпуклой непрерывно дифференцируемой функции $f$ на выпуклом множестве $\mathcal{X}$ справедливо*

$$x^* = \Pi_{\mathcal{X}}(x^* - \gamma \nabla f(x^*))$$

# 8 О зеркальном спуске

**Определение 8.1.** Пусть дана дифференцируемая 1-сильно выпуклая относительно нормы $\|\cdot\|$ на множестве $\mathcal{X}$ функция $d$. Дивергенцией Брэгмана, порождённой функцией $d$ на множестве $\mathcal{X}$, называется функция двух аргументов $V(x, y) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ такая, что для любых $x, y \in \mathcal{X}$:

$$V(x, y) = d(x) - d(y) - \langle \nabla d(y), x - y\rangle$$

По сути, дивергенция Брэгмана – это разность значения функции $d$ в точке $x$ с её линейной аппроксимацией по точке $y$.

**Определение 8.2.** Итерация метода зеркального спуска выглядит следующим образом:

$$x^{k+1} = \text{argmin}_{\ x \in \mathcal{X}}\{\langle \gamma \nabla f(x^k), x\rangle + V(x, x^k)\}$$

Для понятия интуиции метода выпишем условие оптимальности для шага метода:

$$\gamma \nabla f(x^k) + \nabla d(x^{k+1}) - \nabla d(x^k) = 0 \Leftrightarrow x^{k+1} = (\nabla d)^{-1}(\nabla d(x^k) - \gamma \nabla f(x^k))$$

То есть $\nabla d$ переносит нас из $E$ в $E^*$, там мы можем оперировать с $\nabla f(x^k)$. Сделаем шаг градиентного спуска в зеркальном пространстве и получим некоторый вектор, из которого с помощью обратного преобразования $(\nabla d)^{-1}$ можно получить $x^{k+1}$.

**Теорема 8.1.** *Пусть задача оптимизации на выпуклом множестве $\mathcal{X}$ с $L$-гладкой относительно нормы $\|\cdot\|$, выпуклой целевой функцией $f$ решается с помощью зеркального спуска с шагом $\gamma \leqslant \frac{1}{L}$. Тогда характер сходимости метода будет сублинейным.*

НА ДВА БАЛЛА:

Давайте найдём явный вид зеркального спуска для $V(x,y) = \sum_{i=1}^d x_i \log\left(\frac{x_i}{y_i}\right)$ дивергенции (KL) на симплексе на $\triangle_d$ (рандомном множестве).

Формальная запись задачи минимизации:

$$\min_{x \in \triangle_d} \langle \gamma \nabla f(x^k), x\rangle + V(x, x^k)$$
$$\text{s.t. } -x_i \leqslant 0$$
$$\sum_{i=1}^d x_i - 1 = 0$$

Выпишем лагранжиан:

$$L(x,\lambda,\nu) = \langle \gamma \nabla f(x^k), x\rangle + V(x,x^k) + \sum_{i=1}^d \lambda_i(-x_i) + \nu\left(\sum_{i=1}^d x_i - 1\right) =$$
$$\sum_{i=1}^d \left(\log\left(\frac{x_i}{x_i^k}\right) + \gamma[\nabla f(x^k)]_i - \lambda_i + \nu\right) x_i - \nu$$

Минимизируя по каждой $x_i$, получим двойственную:

$$\inf_x L(x,\lambda,\nu) = \sum_{i=1}^d -x_i^k \exp(-1 + \lambda_i - \gamma[\nabla f(x^k)]_i - \nu) - \nu$$

Двойственная задача будет иметь вид:

$$\max_{\lambda_i \geqslant 0,\, \nu \in \mathbb{R}} \left[\sum_{i=1}^d -x_i^k \exp(-1 + \lambda_i - \gamma[\nabla f(x^k)]_i - \nu) - \nu\right]$$

Видно, что функция убывает по $\lambda$, поэтому $\lambda_i^* = 0$. Выпишем условие ККТ:

$$\nabla_x \left(\sum_{i=1}^d \left(\log\left(\frac{x_i}{x_i^k}\right) + \gamma[\nabla f(x^k)]_i - \lambda_i + \nu\right) x_i - \nu\right)$$

Откуда

$$\log\left(\frac{x_i^*}{x_i^k}\right) + 1 + \gamma[\nabla f(x^k)]_i + \nu^* = 0$$

Преобразуем и получаем:

$$x_i^* = x_i^k \exp(-\gamma[\nabla f(x^k)]_i) \cdot \exp(1 + \nu^*)$$

Осталось подобрать $\nu^*$, вспомнив об условии симплекса $\sum_{i=1}^d x_i^* = 1$, тогда окончательно

$$x_i^{k+1} = x_i^* = \frac{x_i^k \exp(-\gamma[\nabla f(x^k)]_i)}{\sum_{i=1}^d x_i^k \exp(-\gamma[\nabla f(x^k)]_i)}$$

# 9 О методе экстраградиента

**Определение 9.1.** Точка $(x^*, \lambda^*, \nu^*) \in \mathbb{R}^d \times \mathbb{R}^m_+ \times \mathbb{R}^n$ называется седловой для функции $L(x, \lambda, \nu)$, если для любых $(x, \lambda, \nu) \in \mathbb{R}^d \times \mathbb{R}^m_+ \times \mathbb{R}^n$ выполнено:

$$L(x, \lambda^*, \nu^*) \geqslant L(x^*, \lambda^*, \nu^*) \geqslant L(x^*, \lambda, \nu)$$

Оптимизация функции Лагранжа – седловая задача. Будем рассматривать следующую задачу:

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \in \mathbb{R}^n} L(x, \lambda)$$

где $L$ непрерывно дифференцируема по обеим группам переменных, выпукла-вогнута: выпукла по $x$ и вогнута по $\lambda$, а также градиенты по обеим группам переменных являются $\frac{L}{\sqrt{2}}$-Липшицевыми.

**Определение 9.2.** Итерация экстраградиентного метода для рассматриваемой задачи будет выглядеть следующим образом:

$$\begin{aligned} x^{k+1/2} &= x^k - \gamma \nabla_x L(x^k, \lambda^k) \\ \lambda^{k+1/2} &= \lambda^k + \gamma \nabla_\lambda L(x^k, \lambda^k) \\ x^{k+1} &= x^k - \gamma \nabla_x L(x^{k+1/2}, \lambda^{k+1/2}) \\ \lambda^{k+1} &= \lambda^k + \gamma \nabla_\lambda (x^{k+1/2}, \lambda^{k+1/2}) \end{aligned}$$

Для интуиции метода заметим, что иногда метод спуска-подъёма ведёт себя плохо.

Для задачи $L(x, \lambda) = x\lambda$ с очевидным решением $(0, 0)$: вектор $\begin{pmatrix} \nabla_x L(x^k, \lambda^k) \\ -\nabla_\lambda L(x^k, \lambda^k) \end{pmatrix}$ всегда ортогонален направлению на решение $\begin{pmatrix} x^k - x^* \\ \lambda^k - \lambda^* \end{pmatrix}$.

Это значит, что метод не стремится к решению, а вот экстраградиентный метод будет! (заглядываем немного вперёд, из-за чего не ходим по кругу, как было бы с классическим GD).

**Теорема 9.1.** *Пусть дана непрерывно дифференцируемая по обеим группам переменных выпуклая-вогнутая $L$-гладкая функция $L : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ тогда для экстраградиентного метода справедлива следующая оценка сходимости для любого $u \in \mathbb{R}^d \times \mathbb{R}^n$ и для любого $\gamma \leqslant \frac{1}{L}$: ТОЧНАЯ ФОРМУЛА НА ДВА БАЛЛА, ХАРАКТЕР СХОДИМОСТИ НА ОДИН*

$$L\left(\frac{1}{K}\sum_{k=0}^{K-1} x^{k+1/2}, u_\lambda\right) - L\left(u_x, \frac{1}{K}\sum_{k=0}^{K-1} \lambda^{k+1/2}\right) \leqslant \frac{\|z^0 - u\|_2^2}{2\gamma K}$$

*То есть характер сходимости сублинейный.*

# 10 О симплекс методе

**Определение 10.1.** Угловой точкой называется точка из допустимого множества, лежащая на границе $n$ линейно независимых ограничений.

**Определение 10.2.** Базисом $B$ называется набор индексов $n$ ЛНЗ векторов из матрицы $A$, задающих угловую точку.

**Определение 10.3.** Допустимым базисом $B$ называется базис, если полученная угловая точка $x_B$ лежит в допустимом множестве, то есть $Ax_B \leqslant b$.

**Определение 10.4.** Базис $B$ называется оптимальным базисом, если полученная угловая точка является решением задачи линейного программирования, то есть

$$\forall x \in S: \; c^T x_B \leqslant c^T x$$

Симплекс метод ищет решение задачи линейного программирования, перебирая вершины многогранника (допустимого множества задачи). Перебор вершин производится направленно, то есть алгоритм проходит по рёбрам многогранника, предварительно выбирая на каждой угловой точке ребро, которое больше всего уменьшает значение целевой функции задачи $c^T x$.

**Лемма 10.1.** *Пусть есть допустимый базис $B$, мы можем разложить вектор из целевой функции $c$ по этому базису, а также найти скалярные коэффициенты $\lambda_B$:*

$$c^T = \lambda_b^T A_B \Leftrightarrow \lambda_b^T = c^T A_B^{-1}$$

*где $\lambda_B$ – коэффициенты разложения вектора $c$ в базис $B$. Тогда базис $B$ является оптимальным, если $\lambda_B \leqslant 0$.*

Тогда выпишем шаги алгоритма:

1. Выбрать допустимый базис
$$B_k \Rightarrow x_k = A_{B_k}^{-1} b_{B_k}$$
2. Разложить вектор $c$ в выбранный базис
$$B_k: \; c = A_{B_k}^T \lambda_{B_k}$$
3. Проверить оптимальность базиса. Если $\lambda_{B_k} \leqslant 0$, то алгоритм завершает работу, а $x_k$ является решением задачи. Если $\lambda_{B_k} > 0$, то меняем базис.
4. Заменяем базис
$$x_{k+1} = x_k + \mu_k d_k$$
Возвращаемся на Шаг 2.

Непонятно, как в этом методе говорить о характере сходимости, ведь он не непрерывный, а по сути является перебором вершин. Можно сказать, что мы сойдёмся не более чем за количество угловых точек.

# 11 О штрафах и подъёмах

**Определение 11.1.** Для задачи условной оптимизации вида

$$\begin{aligned} &\min_{x \in \mathbb{R}^d} f(x) \\ \text{s.t. } &h_i(x) = 0, i = \overline{1,m} \\ &g_i(x) \leqslant 0, j = \overline{1,n} \end{aligned}$$

назовём штрафной функцией

$$f_\rho(x) := f(x) + \rho \cdot \frac{1}{2} \sum_{i=1}^{m} h_i^2(x) + \rho \cdot \frac{1}{2} \sum_{j=1}^{n} (g_j^+)^2(x)$$

где $y^+ = \max\{y, 0\}$.

**Определение 11.2.** Методом штрафных функций называется переход от задачи условной оптимизации к безусловной оптимизации штрафной функции.

Интуиция ADMM: давайте перейдём от градиентного спуска к градиентному подъёму для двойственной задачи, а ещё и можно добавить регуляризацию для сходимости метода.

Более строго, будем решать

$$\min_{x \in \mathbb{R}^{d_x},\, y \in \mathbb{R}^{d_y}} f(x) + g(y)$$
$$\text{s.t. } Ax + By = c$$

где $A \in \mathbb{R}^{n \times d_x}, B \in \mathbb{R}^{n \times d_y}, c \in \mathbb{R}^n$.

Аугментация (переход к эквивалентной задаче, которая будет лучше сходиться засчёт квадратичной прибавки):

$$\min_{x \in \mathbb{R}^{d_x},\, y \in \mathbb{R}^{d_y}} f(x) + g(y) + \frac{\rho}{2}\|Ax + By - c\|_2^2$$
$$\text{s.t. } Ax + By = c$$

Легко видеть, что Лагранжиан этой задачи

$$L_\rho(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|_2^2$$

порождает выпукло-вогнутую седловую задачу.

**Определение 11.3.** Итерация Alternating Method of Multipliers (ADMM) для рассматриваемой задачи будет выглядеть, как:

$$x^{k+1} = \operatorname{argmin}_x L_\rho(x, y^k, \lambda^k)$$
$$y^{k+1} = \operatorname{argmin}_y L_\rho(x^{k+1}, y, \lambda^k)$$
$$\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + By^{k+1} - c)$$

**Лемма 11.1.** *НА ДВА БАЛЛА.*

*Свойства решения штрафной задачи:*

- *С увеличением $\rho$ решения штрафной задачи (если существует) гарантировано не ухудшает степень нарушения ограничений, то есть для $\rho_1 > \rho_2$ следует, что*

  $$(f - f_{\rho_2})(x^*_{\rho_2}) \geqslant (f - f_{\rho_1})(x^*_{\rho_1})$$

  *где $x^*_{\rho_1}, x^*_{\rho_2}$ – решения соответствующих штрафных задач.*

- *Пусть функция $f$ и все функции $h_i, i = \overline{1, m}$ являются непрерывными. Пусть $X^*$ – множество решений исходной условной задачи оптимизации и для $x^* \in X^*$ множество*

  $$U = \{x \in \mathbb{R}^d \mid f(x) \leqslant f(x^*)\}$$

  *ограничено. Тогда для любого $\varepsilon > 0$ существует $\rho(\varepsilon) > 0$ такое, что множество решений штрафной задачи $X^*_\rho$ для любых $\rho \geqslant \rho(\varepsilon)$ содержится в*

  $$X^*_\varepsilon = \{x \in \mathbb{R}^d \mid \exists x^* \in X^* : \|x - x^*\|_2 \leqslant \varepsilon\}$$

# 12 О барьерах и внутренней точке

**Определение 12.1.** Функция $F$ называется барьерной к множеству $G$, если она удовлетворяет следующим свойствам:

1. она является непрерывно дифференцируемой на int $G$
2. Для любой последовательности $\{x_i\} \subset \text{int } G$ такой, что $x_i \to x \in \partial G$, выполнено $F(x_i) \to +\infty$

**Определение 12.2.** Барьерной задачей к задаче условной оптимизации

$$\min_{x \in G} f(x)$$

называется задача безусловной оптимизации

$$\min_{x \in \mathbb{R}^d} \left[ F_\rho(x) = f(x) + \frac{1}{\rho} F(x) \right]$$

где $F$ – барьерная функция множества $G$.

В общем случае метод внутренней точки заключается в том, чтобы каждую итерацию увеличить $\rho_k > \rho_{k-1}$ и с помощью некоторого метода решить численно задачу безусловной оптимизации с целевой функцией $F_{\rho_k}$ и стартовой точкой $x_k$ .

Обязательно гарантировать, что выход метода $x_{k+1}$ будет близок к реальному решению $x^*(\rho_k)$.

**Определение 12.3.** НА ДВА БАЛЛА.

Выпуклая трижды непрерывно дифференцируемая на int $G$ функция называется самосогласованной, если выполнены следующие условия:

- $$\forall x \in \text{int } G \, \forall h \in \mathbb{R}^d : \left| \frac{d^3}{dt^3} F(x + th) \right| \leqslant 2[h^T \nabla^2 F(x) h]^{\frac{3}{2}}$$
- Она является барьером

**Определение 12.4.** НА ДВА БАЛЛА.

Функция $F$ является $\nu$-самосогласованным барьером ($\nu$ всегда $\geqslant 1$) на множестве int $G$, если

- $F$ самосогласованным барьером
- Выполнено условие:

$$\forall x \in \text{int } G \, \forall h \in \mathbb{R}^d : \, |h^T \nabla F(x)| \leqslant \sqrt{\nu} \sqrt{h^T \nabla^2 F(x) h}$$

**Лемма 12.1.** *НА ДВА БАЛЛА.*

*В случае самосогласованного барьера, метод внутренней точки имеет известный вид. Для его описания введём дополнительные объекты:*

- $\Phi_\rho(x) = \rho F_\rho(x)$

- $\lambda(\Phi_\rho, x) = \sqrt{[\nabla\Phi_\rho(x)]^T[\nabla^2\Phi_\rho(x)]^{-1}\nabla\Phi_\rho(x)}$

*Из параметров оставим* $e_1, e_2 \in (0,1), \rho_{-1} > 0, x^0 \in int\, G$. *Причём выберем* $e_1$ *так, чтобы*

$$\lambda(\Phi_{\rho_{-1}}, x^0) \leqslant e_1$$

*Увеличим* $\rho$*:*

$$\rho_k = \left(1 + \frac{e_2}{\sqrt{\nu}}\right)\rho_{k-1}$$

*Сделаем шаг демпфированного метода Ньютона:*

$$x^{k+1} = x^k - \frac{1}{1+\lambda(\Phi_{\rho_k}, x^k)}[\nabla^2\Phi_{\rho_k}(x^k)]^{-1}\nabla\Phi_{\rho_k}(x^k)$$

# 13 О методах для негладких задач

Идея субградиентного метода очень проста – вместо градиента используем какой-то субградиент в текущей точке.

**Определение 13.1.** Итерация субградиентного метода имеет вид:

$$\begin{aligned} \text{compute } g^k &\in \partial f(x^k) \\ x^{k+1} &= x^k - \gamma g^k \end{aligned}$$

**Теорема 13.1.** *Пусть задача безусловной оптимизации с* $M$*-Липшицевой, выпуклой целевой функцией* $f$ *решается с помощью субградиентного спуска. Тогда справедлива следующая оценка сходимости: ТОЧНАЯ ФОРМУЛА НА ДВА БАЛЛА, ХАРАКТЕР СХОДИМОСТИ НА ОДИН*

$$f\left(\frac{1}{K}\sum_{k=0}^{K-1} x^k\right) - f(x^*) \leqslant \frac{M\|x^0 - x^*\|_2}{\sqrt{K}}$$

*Более того, чтобы добиться точности* $\varepsilon$ *по функции, необходимо*

$$K = O\left(\frac{M^2\|x^0 - x^*\|_2^2}{\varepsilon^2}\right)$$

*итераций.*

Рассмотрим следующую задачу:

$$\min_{x\in\mathbb{R}^d}[f(x) + r(x)]$$

Такая задача называется композитной.

**Определение 13.2.** Для функции $r : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ проксимальный оператор определяется следующим образом:

$$\text{prox}_r(x) = \text{argmin}_{\hat{x}\in\mathbb{R}^d}\left(r(\hat{x}) + \frac{1}{2}\|x - \hat{x}\|^2\right)$$

Предположим, что $f$ является $L$-гладкой выпуклой функцией, а $r$ – просто выпуклой проксимально дружественной функцией.

**Определение 13.3.** Итерация проксимального градиентного метода для рассматриваемой задачи имеет вид:

$$\begin{aligned} &\text{compute } \nabla f(x^k) \\ x^{k+1} &= \text{prox }_{\gamma r}(x^k - \gamma \nabla f(x^k)) \end{aligned}$$

**Теорема 13.2.** *Проксимальный градиентный спуск для композитной задачи $L$-гладкой выпуклой функцией $f$ и выпуклой проксимально дружественной функцией $r$ имеет такую же сходимость, что и метод градиентного спуска для функции $f$. Свойства гладкости/негладкости $r$ при этом не влияют.*

**Лемма 13.1.** *НА ДВА БАЛЛА.*

*Пусть $r : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ – выпуклая функция, для которой определён prox $_r$. Тогда для любых $x, y \in \mathbb{R}^d$ следующие три условия являются эквивалентными:*

- *prox $_r(x) = y$*
- *$x - y \in \partial r(y)$*
- *$\forall z \in \mathbb{R} : \langle x - y, z - y \rangle \leqslant r(z) - r(y)$*

**Лемма 13.2.** *НА ДВА БАЛЛА.*

*Пусть $r : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ – выпуклая функция, для которой определён prox $_r$. Тогда для любых $x, y \in \mathbb{R}^d$ выполнено следующее:*

- *$\langle x - y,$ prox $_r(x) -$ prox $_r(y) \rangle \geqslant \|$prox $_r(x) -$ prox $_r(y)\|_2^2$*
- *$\|$prox $_r(x) -$ prox $_r(y)\|_2 \leqslant \|x - y\|_2$*

# 14 О стохастике

Будем оптимизировать функцию, которая представляется, как матожидание (интеграл/сумма) по какому-то неизвестному нам распределению (очень часто встречается в машинном обучении):

$$f(x) := \mathbb{E}_\xi[f(x, \xi)]$$

где $\xi \sim \mathcal{D}$ – распределение данных (природа данных), $\xi = (\xi_x, \xi_y)$ – элемент выборки.

Мы хотим подстроиться, чтобы потери модели в среднем по всему распределению были наименьшими.

Проблема в том, что функция $f$ вместе со своими производными любых порядков в общем случае не считается, так как мы не знаем $\mathcal{D}$, да даже если и знаем, то матожидание часто взять не так просто.

Возникает потребность в методе, который может оперировать с $\nabla f(x, \xi)$ – градиент по конкретному сэмплу из распределения данных. То есть хотим работать в онлайн режиме: поступают сэмплы, мы их обрабатываем.

Часто в машинном обучении мы стартуем не с нуля и дана обучающая выборка, тогда задачу обучения записывают в виде минимизации эмпирического риска:

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{n}\sum_{i=1}^{n}[\ell(g(x,\xi_{x,i}),\xi_{y,i})]\right]$$

где $\{\xi_i\}_{i=1}^n$ – выборка из $\mathcal{D}$, $g$ – модель, $\ell$ – функция. Такую постановку называют оффлайн (данные фиксированы, а не поступают в режиме реального времени).

Как вы могли заметить, в оффлайн постановке уже можно считать полный градиент, но зачастую это бывает дорого и долго, поэтому вместо полного градиента вызывают градиент по случайному сэмплу – интуиция SGD.

**Определение 14.1.** Итерация стохастического градиентного спуска (SGD) выглядит следующим образом:

$$\begin{aligned}\text{random } \xi^k\\ \text{compute } \nabla f(x^k,\xi^k)\\ x^{k+1} = x^k - \gamma_k\nabla f(x^k,\xi^k)\end{aligned}$$

**Теорема 14.1.** *Пусть задача безусловной стохастической оптимизации с $L$-гладкой, $\mu$-сильно выпуклой целевой функцией $f$ решается с помощью SGD с $\gamma_k \leqslant \frac{1}{L}$ в условиях насыщенности и ограниченности дисперсии стохастического градиента. Тогда справедлива следующая оценка сходимости: ТОЧНАЯ ФОРМУЛА НА ДВА БАЛЛА, НА ОДИН БАЛЛ ЗНАТЬ ХАРАКТЕР*

$$\mathbb{E}\left[\|x^{k+1}-x^*\|^2\right] \leqslant (1-\gamma_k\mu)\mathbb{E}\left[\|x^k-x^*\|^2\right] + \gamma_k^2\sigma^2$$

*Первый член – линейная сходимость к решению.*

*Второй член говорит о том, что некоторую точность метод преодолеть не может и начинает осциллировать, больше не приближаясь к решению.*

Изначально у SGD наблюдается поведение, как и у градиентного спуска: $x \to x^*$, но потом начинаются осцилляции. Это происходит из-за того, что в градиентном спуске $\nabla f(x) \to \nabla f(x^*) = 0$. Сейчас никто это не гарантирует: $\nabla f(x,\xi)$ может не стремится к нулю.

Интуиция нового метода SAGA – взять метод на подобии SGD:

$$x^{k+1} = x^k - \gamma g^k$$

где $\lim_{x^k\to x^*} g^k = \nabla f(x^*) = 0$.

По возможности хотелось бы, чтобы

$$\mathbb{E}\left[g^k \mid x^k\right] = \nabla f(x^k) \text{ или } \mathbb{E}[g^k] = \nabla f(x^k)$$

В общем онлайн случае это нереализуемо. Но возможно в оффлайн виде:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

**Определение 14.2.** Итерация метода SAGA имеет следующий вид:

$$\text{random } i_k$$
$$g^k = \nabla f_{i_k}(x^k) - y^k_{i_k} + \frac{1}{n}\sum_{j=1}^{n} y^k_j$$
$$\text{update } y^{k+1}_i = \begin{cases} \nabla f_i(x^k), i = i_k \\ y^k_i, \text{else} \end{cases}$$
$$x^{k+1} = x^k - \gamma g^k$$

Данный метод сходится за $\mathcal{O}\left(\left[n + \frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right)$ итераций, и вместе с этим является в $n$ раз дешевле (считаем не полный градиент, а только 1 слагаемое).