Лабораторная работа 3:
Композитная оптимизация

Срок сдачи: 25 июня 2023 (воскресенье), 23:59

1 Алгоритмы

1.1 Композитная оптимизация

Задача композитной минимизации представляет собой задачу минимизации специального вида:

$$\min_{x \in Q} f(x) + h(x)$$

где $f : E \to \mathbb{R}$ - дифференцируемая функция с липшицевым градиентом, определенная на открытом множестве E в $\mathbb{R}^n$, $h : Q \to \mathbb{R}$ - простая выпуклая замкнутая функция, определенная на подмножестве Q множества E.

Под простотой функции h подразумевается то, что возможно эффективно вычислить проксимальный оператор

$$\text{Prox}_{\lambda h}(x) := \underset{y \in Q}{\text{argmin}} \left\{ \lambda h(y) + \frac{1}{2}\|y - x\|^2 \right\}$$

для любого $\lambda > 0$ и любого $x \in \mathbb{R}^n$. Например, для функции $\|\cdot\|_1 : \mathbb{R}^n \to \mathbb{R}$ соответствующее проксимальный оператор может быть вычислено по формуле

$$\text{Prox}_{\lambda\|\cdot\|_1}(x) = \left(\text{sign}\,(x_i)\,[|x_i| - \lambda]_+\right)_{1 \le i \le n}$$

для $x \in \mathbb{R}^n, \lambda > 0$, где $[\cdot]_+ : \mathbb{R} \to \mathbb{R}-$ положительная срезка $[t]_+ := \max\{t, 0\}$.

1.2 Субградиентный метод

Субградиентный метод - это метод решения безусловной негладкой выпуклой задачи оптимизации

$$\min_{x \in Q} \phi(x)$$

где $\phi : Q \to \mathbb{R}-$ выпуклая функция с ограниченными субградиентами, определенная на выпуклом замкнутом множестве Q в пространстве $\mathbb{R}^n$. Заметим, что задача композитной оптимизации формально может рассматриваться как общая задача негладкой выпуклой оптимизации с функцией $\phi = f + h$.

Итерация субградиентного метода заключается в шаге из текущей точки x_k в направлении (произвольного) анти-субградиента $\phi'(x_k)$, затем выполняется проекция на множество Q. При этом, поскольку для негладких задач норма субградиента $\|\phi'(x_k)\|$ не является информативной, субградиентный метод использует в качестве направления нормированный вектор $\phi'(x_k) / \|\phi'(x_k)\|$:

$$x_{k+1} = P_Q\left(x_k - \alpha_k \frac{\phi'(x_k)}{\|\phi'(x_k)\|}\right)$$

где P_Q - оператор ортогональной проекции на множество Q.

Для сходимости метода необходимо, чтобы длины шагов α_k убывали к нулю, но не слишком быстро:

$$\alpha_k > 0, \quad \alpha_k \to 0, \quad \sum_{k=0}^{\infty} \alpha_k = \infty$$

Обычно длины шагов выбирают по правилу $\alpha_k = \alpha/\sqrt{k+1}$, где $\alpha > 0$ - некоторая константа.

Нужно отметить, что последовательность $(x_k)_{k=0}^{\infty}$, построенная субградиентным методом, может не быть (и, как правило, зачастую не является) релаксационной последовательностью для функции ϕ, т. е. неравенство $\phi(x_{k+1}) \leq \phi(x_k)$ может не выполняться. Поэтому в субградиентном методе в качестве результата работы после N итераций метода вместо точки x_N возвращается точка $y_N := \operatorname{argmin}\{\phi(x) : x \in \{x_0, \ldots, x_N\}\}$ (т. е. из всех пробных точек x_k, построенных методом, выбирается та, в которой значение функции оказалось наименьшим[1]

1.3 Градиентный метод

Одним из самых простых методов решения задачи композитной минимизации является градиентный метод. Приведем схему этого метода, в которой используется одномерный поиск для динамической регулировки оценки константы Липшица градиента:

Алгоритм 1 Градиентный метод для композитной минимизации

Вход: Дифференцируемая функция $f : E \to \mathbb{R}$, определенная на открытом множестве E в $\mathbb{R}^n$; выпуклая замкнутая функция $h : Q \to \mathbb{R}$, определенная на подмножестве Q множества E; начальная точка $x_0 \in Q$; начальная оценка константа Липшица $L_0 > 0$ для $\nabla f|_Q$.

1: **for** $k \geq 0$ **do**:
2: Положить $\bar{L}_k := L_k$.
3: Положить $x_{k+1} := \operatorname{argmin}_{x \in Q}\{f(x_k) + \langle \nabla f(x_k), x - x_k\rangle + \frac{\bar{L}_k}{2}\|x - x_k\|^2 + h(x)\}$
4: Если $f(x_{k+1}) > f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{\bar{L}_k}{2}\|x_{k+1} - x_k\|^2$, положить $\bar{L}_k := 2\bar{L}_k$ и вернуться к шагу 3.
5: Положить $L_{k+1} := \bar{L}_k/2$.
6: **end for**

Несмотря на то, что на отдельных шагах этой схемы может выполняться достаточно много итераций одномерного поиска по подбору параметра $\bar{L}_k$, можно показать, что среднее число итераций одномерного поиска за итерацию метода примерно равно двум. Параметр L_0, по сути дела, влияет лишь на число одномерных поисков на самых первых итерациях метода; его всегда можно выбрать равным единице ($L_0 = 1$) без принципиального ущерба

[1] Заметим, что в практической реализации метода для вычисления результата y_N сами точки $x_0, \ldots, x_N$ хранить в памяти не нужно.

для скорости сходимости метода, поскольку в итоговую оценку суммарной трудоемкости метода этот параметр входит под логарифмом.

Заметим, что формула для x_{k+1} может быть переписана в эквивалентной форме с помощью проксимального оператора:

$$x_{k+1} = \mathrm{Prox}_{h/\bar{L}_k}\left(x_k - \frac{1}{\bar{L}_k}\nabla f\left(x_k\right)\right).$$

1.4 Быстрый градиентный метод

Используя технику ускорения Нестерова, градиентный метод можно ускорить:

Алгоритм 2 Быстрый градиентный метод для композитной минимизации

Вход: Дифференцируемая функция $f : E \to \mathbb{R}$, определенная на открытом множестве E в $\mathbb{R}^n$; выпуклая замкнутая функция $h : Q \to \mathbb{R}$, определенная на подмножестве Q множества E; начальная точка $x_0 \in Q$; начальная оценка константа Липшица $L_0 > 0$ для $\nabla f|_Q$.

1: Положить $A_0 := 0$ и $v_0 := x_0$.
2: **for** $k \geq 0$ **do**
3: Положить $\bar{L}_k := L_k$.
4: Положить $a_k := \frac{1+\sqrt{1+4\bar{L}_k A_k}}{2\bar{L}_k}$ и $A_{k+1} := A_k + a_k$.
5: Положить $y_k := \frac{A_k x_k + a_k v_k}{A_{k+1}}$.
6: Вычислить $v_{k+1} := \mathrm{argmin}_{x\in Q}\{\frac{1}{2}\|x - x_0\|^2 + \sum_{i=0}^{k} a_i[f(y_i) + \langle \nabla f(y_i), x - y_i\rangle + h(x)]\}$.
7: Положить $x_{k+1} := \frac{A_k x_k + a_k v_{k+1}}{A_{k+1}}$.
8: Если $f(x_{k+1}) > f(y_k) + \langle \nabla f(y_k), x_{k+1} - y_k\rangle + \frac{\bar{L}_k}{2}\|x_{k+1} - y_k\|^2$, положить $\bar{L}_k := 2\bar{L}_k$ и вернуться к шагу 4.
9: Положить $L_{k+1} := \bar{L}_k/2$.
10: **end for**

Обратите внимание, что для вычисления точки v_{k+1} не нужно каждый раз суммировать по всем $0 \leq i \leq k$; вместо этого достаточно обновлять в итерациях взвешенную сумму градиентов $\sum_{i=0}^{k} a_i \nabla f\left(y_i\right)$.

В отличие от обычного градиентного метода, быстрый градиентный метод работает уже с несколькими последовательностями точек. При этом вдоль каждой из этих последовательностей целевая функция может уменьшаться не монотонно. Поэтому в качестве ответа $\bar{x}_k$ метода следует выдавать ту точку, в которой наблюдалось наименьшее (так называемое рекордное) значение целевой функции среди всех точек x_k, y_k, в которых выполнялись вычисления функции.

2 Задача Lasso

Модель Lasso является одной из стандартных моделей линейной регрессии. Имеется обучающая выборка $((a_i, b_i))_{i=1}^m$, где $a_i \in \mathbb{R}^n-$ вектор признаков i-го объекта, а $b_i \in \mathbb{R}-$ его регрессионное значение. Задача заключается в прогнозировании регрессионного значения b_{new} для нового объекта, представленного своим вектором признаков a_{new} .

В модели Lasso, как и в любой модели линейной регрессии, прогнозирование выполняется с помощью линейной комбинации компонент вектора a с некоторыми фиксированными коэффициентами $x \in \mathbb{R}^n$:

$$b(a) := \langle a, x\rangle$$

Коэффициенты x являются параметрами модели и настраиваются с помощью решения следующей оптимизационной задачи:

$$\phi(x) := \frac{1}{2}\sum_{i=1}^{m}\left(\langle a_i, x\rangle - b_i\right)^2 + \lambda\sum_{j=1}^{n}|x_j| =: \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_1 \to \min_{x\in\mathbb{R}^n}. \quad (1)$$

Здесь $\lambda > 0$ - коэффициент регуляризации (параметр модели). Особенностью Lasso является использование именно l^1-регуляризации (а не, например, l^2-регуляризации). Такая регуляризация позволяет получить разреженное решение. В разреженном решении x^* часть компонент равна нулю. (Можно показать, что при $\lambda \geq \left\|A^T b\right\|_\infty$ все компоненты будут нулевыми). Нулевые веса соответствуют исключению соответствующих признаков из модели (признание их неинформативными).

В этом задании все рассматриваемые методы должны использовать критерий остановки по зазору двойственности (см. ниже).

2.1 Двойственная задача и критерий остановки

Двойственной задачей Фенхеля к задаче (1) является

$$\max_{\mu\in\mathbb{R}^m}\left\{-\frac{1}{2}\|\mu\|^2 - \langle b, \mu\rangle : \left\|A^T\mu\right\|_\infty \leq \lambda\right\} \quad (2)$$

Таким образом, имея в распоряжении допустимую двойственную точку $\mu \in \mathbb{R}^m$, т. е. такую что $\left\|A^T\mu\right\|_\infty \leq \lambda$, можно вычислить следующую оценку для невязки в задаче (1):

$$\phi(x) - \phi^* \leq \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_1 + \frac{1}{2}\|\mu\|^2 + \langle b, \mu\rangle =: \eta(x,\mu).$$

Величина $\eta(x,\mu)$ называется зазором двойственности и обращается в ноль в оптимальных решениях x^* и μ^* задач (1) и(2). Заметим, что решения x^* и μ^* связаны между собой следующим соотношением: $Ax^* - b = \mu^*$. Поэтому для фиксированного $x \in \mathbb{R}^n$ естественным выбором соответствующего μ будет

$$\mu(x) := \min\left\{1, \frac{\lambda}{\|A^T(Ax-b)\|_\infty}\right\}(Ax-b)$$

Такой выбор обеспечивает стремление зазора двойственности $\eta(x,\mu(x))$ к нулю при $x \to x^*$, что позволяет использовать условие $\eta(x,\mu(x)) < \epsilon$ в качестве критерия остановки в любом итерационном методе решения задачи (1).

3 Формулировка задания

1. Даны прототипы функции, которые Вам нужно будет реализовать. Некоторые процедуры уже частично или полностью реализованы.

2. Реализуйте негладкий оракул для функции (1) (классы LeastSquaresOracle и LassoNonsmoothOracle в модуле oracles).

3. Реализуйте субградиентный метод (функция subgradient_method в модуле optimization).

4. Реализуйте композитный оракул для функции (1) (классы L1RegOracle и LassoProxOracle в модуле oracles).

5. Реализуйте градиентный метод для композитной минимизации (функция proximal_gradient_method в модуле optimization).

6. Реализуйте быстрый градиентный метод для композитной минимизации (функция proximal_fast_gradient_method в модуле optimization).

7. Проведите эксперименты, описанные ниже. Напишите отчет.

3.1 Эксперимент: Выбор длины шага в субградиентном методе

Исследуйте работу субградиентного метода в зависимости от выбора константы α_0 в формуле для длины шага. При этом для одной и той же задачи рассмотрите различные начальные точки x_0. Есть ли связь между «наилучшим» коэффициентом α_0 и начальной точкой x_0 ?

3.2 Эксперимент: Среднее число итераций одномерного поиска в градиентных методах

Для градиентного метода и быстрого градиентного метода постройте график зависимости суммарного числа итераций одномерного поиска от номера итерации метода. Действительно ли среднее число итераций линейного поиска примерно равно двум в обоих методах?

3.3 Эксперимент: Сравнение методов

Сравните три реализованных метода на задаче Lasso. При этом рассмотрите различные значения размерности пространства n, размера выборки m и коэффициента регуляризации λ.

Для сравнения методов постройте графики 1) гарантируемая точность по зазору двойственности против числа итераций и 2) гарантированная точность по зазору двойственности против реального времени работы. Для гарантированной точности по зазору двойственности используйте логарифмическую шкалу.

Данные (матрицу A и вектор b) для задачи Lasso можно сгенерировать случайно, либо взять реальные данные с сайта LIBSVM.

4 Оформление задания

Результатом выполнения задания являются

(a) Файлы optimization.py и oracles.py с реализованными методами и оракулами.

(b) Полные исходные коды для проведения экспериментов и рисования всех графиков. Все результаты должны быть воспроизводимыми. Если вы используете случайность - зафиксируйте seed.

(c) Отчет в формате PDF о проведенных исследованиях.

Каждый проведенный эксперимент следует оформить как отдельный раздел в PDF документе (название раздела - название соответствующего эксперимента). Для каждого эксперимента необходимо сначала написать его описание: какие функции оптимизируются, каким образом генерируются данные, какие методы и с какими параметрами используются. Далее должны быть представлены результаты соответствующего эксперимента - графики, таблицы и т. д. Наконец, после результатов эксперимента должны быть написаны Ваши выводы - какая зависимость наблюдается и почему.

Важно: Отчет не должен содержать никакого кода. Каждый график должен быть прокомментирован - что на нем изображено, какие выводы можно сделать из этого эксперимента. Обязательно должны быть подписаны оси. Если на графике нарисовано несколько кривых, то должна быть легенда. Сами линии следует рисовать достаточно толстыми, чтобы они были хорошо видимыми.

Важно: Практическое задание выполняется самостоятельно. Если вы получили ценные советы (по реализации или проведению экспериментов) от другого студента, то об этом должно быть явно написано в отчёте. В противном случае «похожие» решения считаются плагиатом.