Практическое задание 3: Градиентные методы для композитной оптимизации и метод барьеров.

Калугин Дмитрий

13 мая 2018 г.

1 Вспомогательная функция, минимизируемая в методе барьеров

Вспомогательная функция, минимизируемая в методе барьеров:

$$\phi_t(x, u) = t \cdot (\frac{1}{2}\|Ax - b\|_2^2 + \lambda\langle 1_n, u\rangle) - \sum_i \left(\log(u_i + x_i) + \log(u_i - x_i)\right).$$

Чтобы выписать линейную систему, задающую ньютоновское направление, найдем градиент и гессиан этой функции(здесь $1/x$ и x^2, где x — вектор, обозначает покоординатное деление и возведение в квадрат соответсвенно). Градиент:

$$\nabla_x \phi_t(x, u) = tA^T(Ax - b) - \frac{1}{u + x} + \frac{1}{u - x}$$
$$\nabla_u \phi_t(x, u) = t\lambda \cdot 1_n - \frac{1}{u + x} - \frac{1}{u - x}$$

Гессиан представим в следующем виде:

$$\nabla^2 \phi_t(x, u) = \begin{bmatrix} \nabla^2_{xx}\phi_t & \nabla^2_{xu}\phi_t \\ \nabla^2_{xu}\phi_t & \nabla^2_{uu}\phi_t \end{bmatrix},$$

где:

$$\nabla^2_{xx}\phi_t = tA^TA + Diag(\frac{1}{(u + x)^2} + \frac{1}{(u - x)^2})$$
$$\nabla^2_{uu}\phi_t = Diag(\frac{1}{(u + x)^2} + \frac{1}{(u - x)^2})$$
$$\nabla^2_{xu}\phi_t = Diag(\frac{1}{(u + x)^2} - \frac{1}{(u - x)^2})$$

В методе Ньютона решается система:

$$\nabla^2 \phi_t \cdot d = -\nabla \phi_t,$$

где $d = (d_x, d_u)$. Это уравнение можно разбить на два:

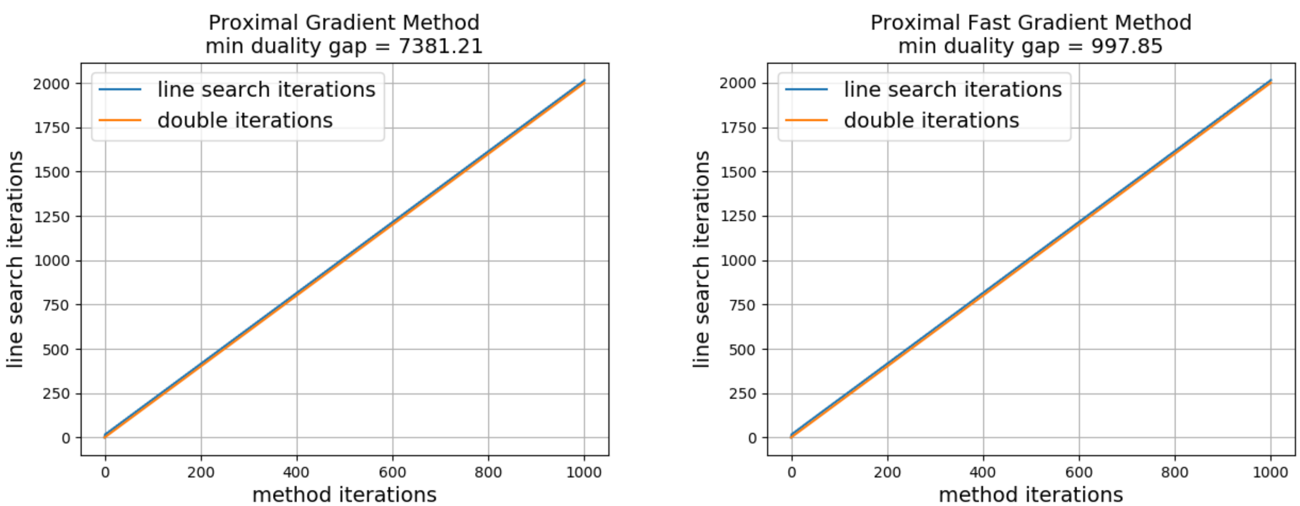
$$\begin{cases} \nabla^2_{xx}\phi_t \cdot d_x + \nabla^2_{xu}\phi_t \cdot d_u = -\nabla_x \phi_t, \\ \nabla^2_{xu}\phi_t \cdot d_x + \nabla^2_{uu}\phi_t \cdot d_u = -\nabla_u \phi_t. \end{cases}$$

При этом градиенты, стоящие во второй строчке — диагональные матрицы, значит уравнение разбивается на пары уравнений по переменным x_i, u_i. Отсюда можно выразить u_i и подставить в первую строчку, это вдвое сократит размерность задачи.

Опишем алгоритм определения максимального шага α. Должны быть выполнены ограничения: $-u_i \leq x_i \leq u_i$. Среди компонент d_x, d_u, если в компоненте $d_{x,i} > d_{u,i}$, то $\alpha \leq 0.99 \cdot \frac{u_i - x_i}{d_{x,i} - d_{u,i}}$. А если $d_{x,i} < -d_{u,i}$, то $\alpha \leq 0.99 \cdot \frac{x_i + u_i}{-d_{x,i} - d_{u,i}}$. Кроме всех этих ограничений, необходимо чтобы было выполнено $\alpha \leq 1$. Наибольшее α, удовлетворяющее всем этим ограничениям и есть искомое.

2 Эксперимент: Среднее число итераций одномерного поиска в градиентных методах

Для градиентного метода и быстрого градиентного метода построим график зависимости суммарного числа итераций одномерного поиска от номера итерации метода.



(a) Проксимальный градиентный метод

(b) Быстрый проксимальный градиентный метод Нестерова

Рис. 1: Число итераций одномерного поиска

На графиках видно, что количество итераций линейного поиска почти точно совпадает с удвоенным количеством итераций. Это означает, что несмотря на адаптивный подбор константы Липшица количество итераций меняется всего примерно в 2 раза.

При этом видно, что наименьший двойственный зазор, достигаемый ускоренным методом Нестерова в несколько раз меньше двойственного зазора, достигаемого обычным градиентным спуском.

Также видно, что оба метода делают 1000 итераций, это максимальное количество итераций в этих методах. Это связано с тем, что проксимальные методы не сходятся абсолютно точно, а в течение многих итераций колеблются вокруг некоторых точек оптимума(это хорошо видно на графиках следующего эксперимента). Из-за этого алгоритм тратит весь запас итераций.

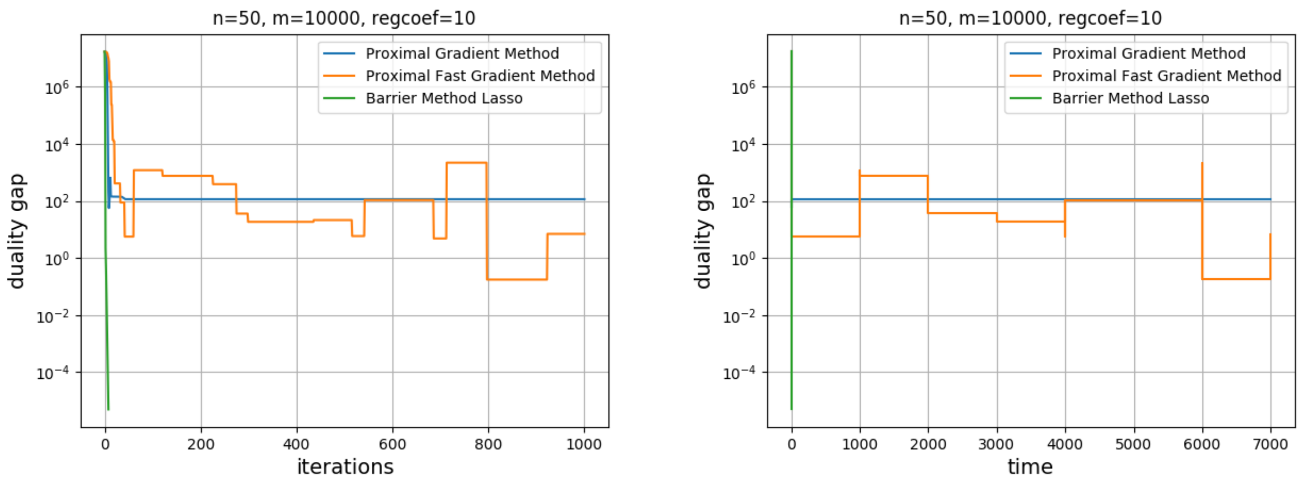
3 Эксперимент: Сравнение методов

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

Данные (матрицу A и вектор b) для задачи Lasso будем генерировать случайно. Проводится полный перебор по сетке:

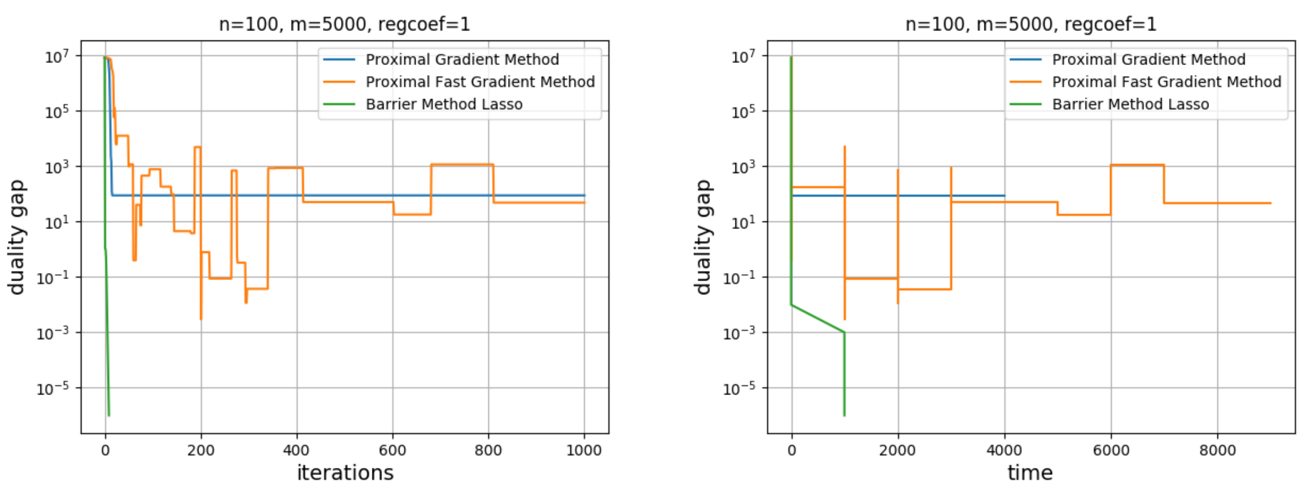
$$n \in \{50,\ 100,\ 500\},\ m \in \{1000,\ 5000,\ 10000\},\ \lambda \in \{0.1,\ 1,\ 10\}.$$

Для каждой тройки параметров строятся зависимости логарифма двойственного зазора от количества итераций и от времени. Итого 54 графика. Приведем здесь некоторые из них, остальные будут в архиве, приложенном к отчету. (проводить эксперименты при больших значениях n я побоялся, мой ноутбук не умеет в метод Ньютона на больших датасетах)



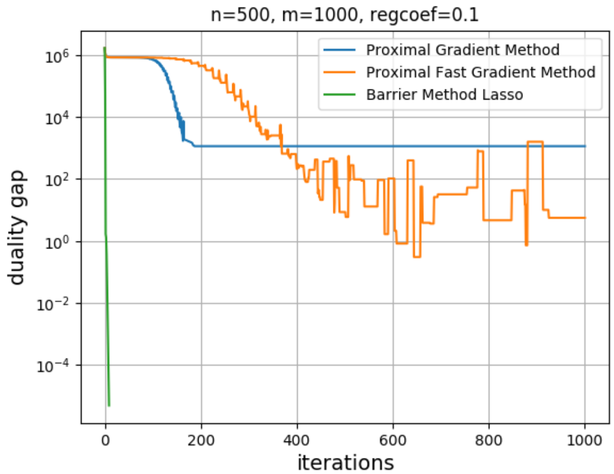
(a) Зависимость двойственного зазора от номера итерации (b) Зависимость двойственного зазора от реального времени

Рис. 2: Эксперименты при $n = 50, m = 10000, \lambda = 10$

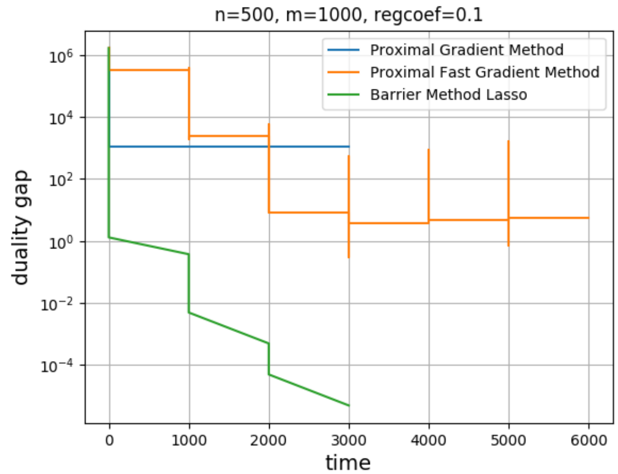


(a) Зависимость двойственного зазора от номера итерации (b) Зависимость двойственного зазора от реального времени

Рис. 3: Эксперименты при $n = 100, m = 5000, \lambda = 1$



(a) Зависимость двойственного зазора от номера итерации



(b) Зависимость двойственного зазора от реального времени

Рис. 4: Эксперименты при $n = 500, m = 1000, \lambda = 0.1$

Время на графиках приведено в миллисекундах.

Из графиков можно сделать следующие выводы.

Метод барьеров всегда показывает лучше качество и скорость сходимости, чем остальные методы. Это логично, так как это более продвинутый метод решения задачи Lasso, плюс к тому же внутри он содержит метод Ньютона, который, как метод второго порядка, гораздо эффективнее методов первого порядка в случаях малой размерности.

Ускоренный метод Нестерова сходится немонотонно. Это логично, так как при построении алгоритма мы этого нигде не требовали, и поэтому в качестве точки ответа мы возвращаем ту точку, в которой было достигнуто наилучшее качество, а не просто последнюю.

Во втором третьем случае видно, что ускоренный градиентный спуск требует примерно в 2 раза больше времени, чем обычный, при том же количестве итераций. Это можно связать с тем, что самые затратные по времени операции, такие как обращение к оракулу, в ускоренном методе делаются дважды за итерацию, против одного раза в обычном градиентном спуске.