Методы оптимизации в машинном обучении
Практическое задание 3
Отчет

Асланов Алишер, БПМИ191

1 Введение

В данной работе рассматривается применение метода логарифмических барьеров к задаче линейной регрессии с ℓ_1-регуляризацией. Как известно, часто различные методы оптимизации можно модифицировать под конкретную задачу и получить более эффективный алгоритм по сравнению с общим случаем. Поэтому сперва мы выведем все необходимые формулы для нашей задачи и воспользуемся ее структурой, чтобы ускорить метод (а точнее, процесс решения СЛАУ в методе Ньютона). Затем мы проведем ряд экспериментов над реализацией нашего метода и выявим зависимости его поведения от параметров и размеров входных данных.

2 Метод лог. барьеров для LASSO-регрессии

После сведения через надграфик наша задача имеет следующий вид:

$$\begin{cases} \frac{1}{2}\|Ax - b\|_2^2 + \lambda\langle 1_n, u\rangle \to \min\limits_{x,u} \\ -u \preceq x \preceq u \end{cases},$$

где $A \in \mathbb{R}^{m\times n}$ — матрица объекты-признаки, $b \in \mathbb{R}^m$ — вектор правильных ответов, $\lambda \in \mathbb{R}$ — коэффициент регуляризации и $x, u \in \mathbb{R}^n$ — переменные, по которым ведется оптимизация.

Выпишем барьерную функцию F:

$$F(x,u) = -\sum_{i=1}^n \ln(u_i + x_i) - \sum_{i=1}^n \ln(u_i - x_i) = -\sum_{i=1}^n \ln(u_i + x_i)(u_i - x_i) = -\sum_{i=1}^n \ln(u_i^2 - x_i^2).$$

Тогда вспомогательная функция f_t будет иметь следующий вид:

$$f_t(x,u) = t\left(\frac{1}{2}\|Ax - b\|_2^2 + \lambda\langle 1_n, u\rangle\right) - \sum_{i=1}^n \ln(u_i^2 - x_i^2).$$

Чтобы применить к этой функции метод Ньютона, необходимо вычислить ее градиент и матрицу Гессе.

$$\nabla_x f_t(x,u) = \frac{t}{2}\nabla_x\|Ax - b\|_2^2 - \sum_{i=1}^n \nabla_x \ln(u_i^2 - x_i^2) = tA^T(Ax - b) + \left[\frac{2x_i}{u_i^2 - x_i^2}\right]^T_{i=1,\dots,n},$$

$$\nabla_u f_t(x,u) = t\lambda\nabla_u\langle 1_n, u\rangle - \sum_{i=1}^n \nabla_u \ln(u_i^2 - x_i^2) = t\lambda\cdot 1_n - \left[\frac{2u_i}{u_i^2 - x_i^2}\right]^T_{i=1,\dots,n}.$$

Чтобы найти матрицу Гессе, необходимо вычислить все частные производные второго порядка. Для начала заметим, что

$$\nabla_x^2 \left(\frac{t}{2} \|Ax - b\|_2^2 \right) = \mathfrak{J}_x \left(tA^T(Ax - b) \right) = tA^T A,$$

где $\mathfrak{J}(\cdot)$ — матрица Якоби отображения. Тогда

$$\frac{\partial^2 f_t}{\partial x_i^2} = \frac{\partial}{\partial x_i} \left(\left[tA^T(Ax - b) \right]_i + \frac{2x_i}{u_i^2 - x_i^2} \right) = t[A^T A]_{ii} + 2\frac{u_i^2 + x_i^2}{(u_i^2 - x_i^2)^2},$$

$$\frac{\partial^2 f_t}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i} \left[tA^T(Ax - b) \right]_j + \underbrace{\frac{\partial}{\partial x_i} \left(\frac{2x_j}{u_j^2 - x_j^2} \right)}_{0} = t[A^T A]_{ij}, \quad (i \neq j)$$

$$\frac{\partial^2 f_t}{\partial u_i^2} = \frac{\partial}{\partial u_i} \left(t\lambda - \frac{2u_i}{u_i^2 - x_i^2} \right) = 2\frac{u_i^2 + x_i^2}{(u_i^2 - x_i^2)^2},$$

$$\frac{\partial^2 f_t}{\partial u_i \partial u_j} = \frac{\partial}{\partial u_i} \left(t\lambda - \frac{2u_j}{u_j^2 - x_j^2} \right) = 0, \quad (i \neq j)$$

$$\frac{\partial^2 f_t}{\partial x_i \partial u_i} = \frac{\partial^2 f_t}{\partial u_i \partial x_i} = \frac{\partial}{\partial x_i} \left(t\lambda - \frac{2u_i}{u_i^2 - x_i^2} \right) = -\frac{4x_i u_i}{(u_i^2 - x_i^2)^2},$$

$$\frac{\partial^2 f_t}{\partial x_i \partial u_j} = \frac{\partial^2 f_t}{\partial u_j \partial x_i} = \frac{\partial}{\partial x_i} \left(t\lambda - \frac{2u_j}{u_j^2 - x_j^2} \right) = 0. \quad (i \neq j)$$

По теореме Шварца, все смешанные частные производные, отличающиеся лишь порядком дифференцирования, совпадают. Действительно, f_t — дважды непрерывно дифференцируемая функция, и знаменатели производных не обращаются в нуль, поскольку $u_i^2 - x_i^2 = 0 \iff |u_i| = |x_i|$, но в нашем методе все точки внутренние, т.е. $|x_i| < u_i$ для всех i.

Таким образом, наш гессиан имеет вид

$$\nabla^2 f_t(x, u) = \begin{bmatrix} tA^T A + C & D \\ D & C \end{bmatrix} \in \mathbb{R}^{2n \times 2n},$$

где

$$C = \operatorname{diag} \left(2\frac{u^2 + x^2}{(u^2 - x^2)^2} \right), \quad D = \operatorname{diag} \left(-\frac{4xu}{(u^2 - x^2)^2} \right),$$

и в аргументе $\operatorname{diag}(\cdot)$ все операции подразумеваются покомпонентными.

Выпишем СЛАУ, задающую ньютоновское направление $d_k = (d_k^x, d_k^u)$:

$$\begin{bmatrix} tA^T A + C & D \\ D & C \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^u \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \tag{1}$$

где $g = -\nabla_x f_t(x_k, u_k)$ и $h = -\nabla_u f_t(x_k, u_k)$. Заметим, что неизвестные d_k^u можно однозначно выразить через d_k^x из второго уравнения системы (1):

$$Dd_k^x + Cd_k^u = h \iff d_k^u = C^{-1}(h - Dd_k^x), \tag{2}$$

причем каждая компонента d_k^u зависит только от соответствующей компоненты d_k^x (в силу диагональности матриц C и D). Тогда решение СЛАУ (1) сводится к решению системы

$$(tA^T A + C)d_k^x + Dd_k^u = g \iff (tA^T A + C)d_k^x + DC^{-1}h - DC^{-1}Dd_k^x = g \iff$$

$$\iff (tA^TA + C - D^2C^{-1})d_k^x = g - DC^{-1}h \tag{3}$$

Исследуем матрицу системы (3) на знакоопределенность. Рассмотрим сперва матрицу $C - D^2C^{-1}$. Она, очевидно, диагональна, причем

$$[C - D^2C^{-1}]_{ii} = 2\frac{u_i^2 + x_i^2}{(u_i^2 - x_i^2)^2} - \frac{16x_i^2u_i^2}{(u_i^2 - x_i^2)^4} \cdot \frac{(u_i^2 - x_i^2)^2}{2(u_i^2 + x_i^2)} = \frac{2(u_i^2 + x_i^2)}{(u_i^2 - x_i^2)^2} - \frac{8x_i^2u_i^2}{(u_i^2 - x_i^2)^2(u_i^2 + x_i^2)} =$$

$$= \frac{2(u_i^4 + 2x_i^2u_i^2 + x_i^4) - 8x_i^2u_i^2}{(u_i^2 - x_i^2)^2(u_i^2 + x_i^2)} = \frac{2(u_i^2 - x_i^2)^2}{(u_i^2 - x_i^2)^2(u_i^2 + x_i^2)} = \frac{2}{u_i^2 + x_i^2},$$

что строго больше нуля для внутренних точек в нашем методе. Теперь, для любого вектора $q \in \mathbb{R}^n \setminus \{0\}$ и любого $t > 0$ имеем

$$q^T(tA^TA + C - D^2C^{-1})q = q^T(tA^TA)q + q^T(C - D^2C^{-1})q = \underbrace{t\|Aq\|_2^2}_{\geq 0} + \underbrace{\sum_{i=1}^n \frac{2}{u_i^2 + x_i^2}q_i^2}_{>0} > 0.$$

Таким образом, $tA^TA + C - D^2C^{-1} \in \mathbb{S}^n_{++}$ (причем, для любой матрицы A), что, в частности, означает существование разложения Холецкого.

Итак, предлагается следующая схема решения системы (1):

1. Вычислить минус градиент функции f_t в точке (x_k, u_k) (т.е. векторы g и h); $\mathcal{O}(nm)$
2. Вычистить вектор правой части $g - DC^{-1}h$ системы (3); $\mathcal{O}(n)$
3. Вычислить матрицу $tA^TA + C - D^2C^{-1}$; $\mathcal{O}(n^2m)$
4. Найти решение d_k^x системы (3) методом Холецкого; $\mathcal{O}\left(\frac{1}{3}n^3\right)$
5. Найти d_k^u по формуле (2). $\mathcal{O}(n)$

Итоговая сложность этого алгоритма составляет $\mathcal{O}\left(n^2m + \frac{1}{3}n^3\right)$. Заметим, что это куда лучше, чем если бы мы напрямую применили метод Холецкого для (1) — тогда решение системы стоило бы нам порядка $\frac{8}{3}n^3$, что в 8 раз дороже решения систем (3) и (2). Однако, у этого метода есть недостаток в виде явного вычисления матрицы A^TA, что, как известно, может быть численно неустойчивой процедурой. Кроме того, эта матрица может быть куда хуже обусловлена по сравнению с A. С другой стороны, кажется, что без явного вычисления A^TA систему (1) решить не удастся.

Теперь разберемся с максимальной допустимой длиной шага α_k. Ограничение на нее задает условие принадлежности следующей точки внутренности доменного множества:

$$-u_k - \alpha_k d_k^u \prec x_k + \alpha_k d_k^x \prec u_k + \alpha_k d_k^u.$$

Все ограничения аффинные, поэтому можно воспользоваться формулами из условия задания и (в предположении, что для подбора шага используется бэктрекинг) положить

$$\alpha_k^{start} = \min\left\{1,\ 0.99 \min_{i:\ d_{ki}^x > d_{ki}^u}\left(\frac{u_{ki} - x_{ki}}{d_{ki}^x - d_{ki}^u}\right),\ 0.99 \min_{i:\ d_{ki}^x < -d_{ki}^u}\left(\frac{-u_{ki} - x_{ki}}{d_{ki}^x + d_{ki}^u}\right)\right\}.$$

Осталось выбрать допустимую начальную точку (x_0, u_0) для нашего метода. Как известно из семинара, разумнее всего брать точку поближе к центру доменного множества. В нашем случае, можно положить, например,

$$x_0 = 0,\ u_0 = 10 \cdot 1_n.$$

Очевидно, эта точка (строго) удовлетворят всем неравенствам.

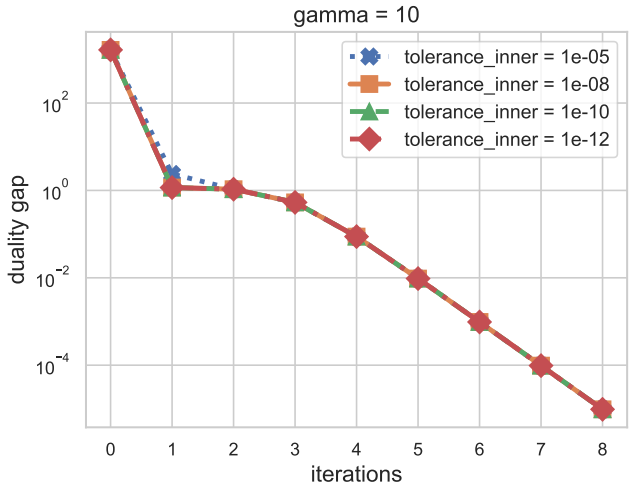
3 Эксперименты

Поэкспериментируем с функцией `barrier_method_lasso`, реализующей описанный выше метод барьеров.

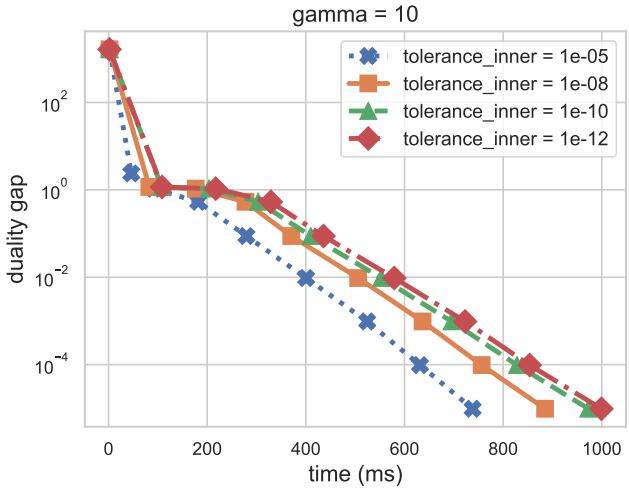
3.1 Чувствительность метода к выбору параметров γ и $\varepsilon_{\text{inner}}$

В данном разделе размер выборки $m = 10000$ и размерность пространства $n = 500$ фиксированы. Данные для задачи (A и b) будем генерировать случайно ($a_{ij}, b_i \overset{\text{i.i.d}}{\sim} U[0, 1]$).

По рекомендации преподавателя, эксперимент будет устроен следующим образом. Сперва мы зафиксируем параметр $\gamma = 10$ (значение по умолчанию) и переберем по сетке $\varepsilon_{\text{inner}} \in \{10^{-5}, 10^{-8}, 10^{-10}, 10^{-12}\}$, и для каждого из этих значений построим требуемые графики. Затем наоборот — зафиксируем в значение по умолчанию $\varepsilon_{\text{inner}} = 10^{-8}$ и выполним перебор параметра $\gamma \in \{10, 100, 1000, 10000\}$. Остальные параметры метода оставим равными их значениям по умолчанию. Результаты эксперимента приведены ниже (время измерялось в миллисекундах):



(a) Зависимость зазора двойственности от номера итерации



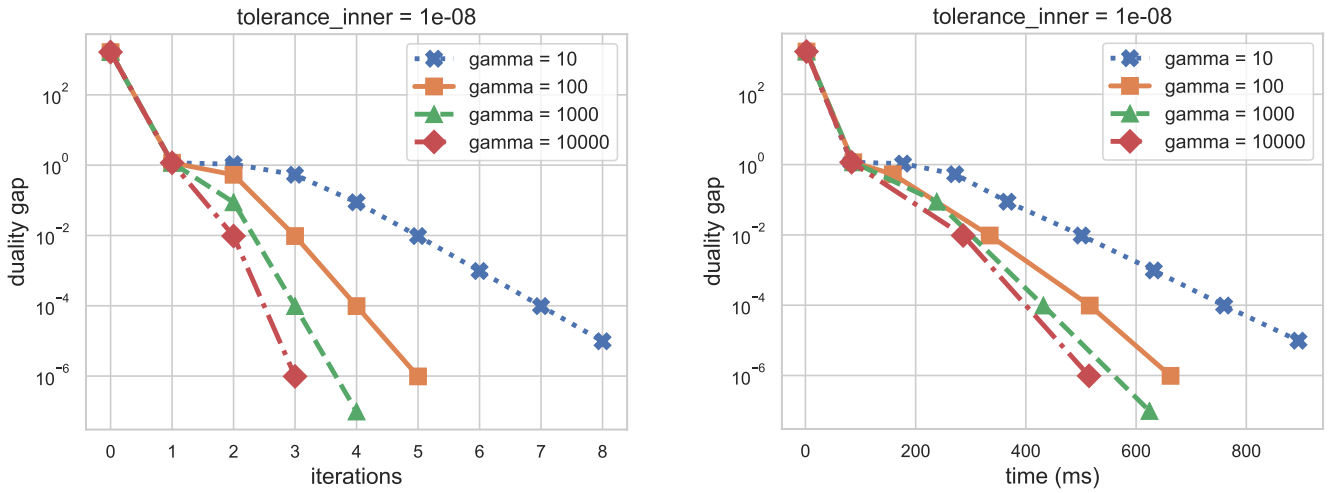
(b) Зависимость зазора двойственности от реального времени работы

Рис. 1: Поведение метода при фиксированном γ и меняющемся $\varepsilon_{\text{inner}}$

Из получившихся графиков можно сделать следующие выводы. Для всех $\varepsilon_{\text{inner}}$ (взятых в разумных пределах) поведение метода с точки зрения внешних итераций ничем не отличается, значения зазоров двойственности примерно одинаковые после каждой итерации. Оно и понятно: ведь этот параметр отвечает за точность решения вспомогательной задачи, и, начиная с некоторого момента, влияние этой точности перестает быть существенным для «глобального» поведения нашего метода. Действительно, решение очередной вспомогательной задачи либо удовлетворяет внешнему критерию остановки, либо используется в качестве начального приближения в следующей вспомогательной задаче, и в обоих этих случаях, эффект при всех достаточно малых различных $\varepsilon_{\text{inner}}$ будет неотличим. С точки же зрения реального времени работы алгоритма, результаты тоже весьма логичны. Чем бóльшую точность мы требуем в методе Ньютона, тем больше итераций ему потребуется, чтобы этой точности достичь $\implies$ время работы увеличивается с уменьшением $\varepsilon_{\text{inner}}$. Однако, в силу локальной квадратичной сходимости, для вычисления очередного десятичного знака в решении методу Ньютона не требуется делать

много дополнительных итераций, поэтому и приросты по времени получились небольшими. Наконец, осталось отметить, что если точность выбрать еще меньше, чем в нашем эксперименте (например, порядка 10^{-14}), то в силу ограничения на максимальное количество итераций, метод Ньютона может разойтись, поэтому рассуждения выше справедливы при условии, что нам хватает итераций, чтобы достичь требуемой точности.

Результаты эксперимента при переборе γ:



(a) Зависимость зазора двойственности от номера итерации

(b) Зависимость зазора двойственности от реального времени работы

Рис. 2: Поведение метода при фиксированном $\varepsilon_{\text{inner}}$ и меняющемся γ

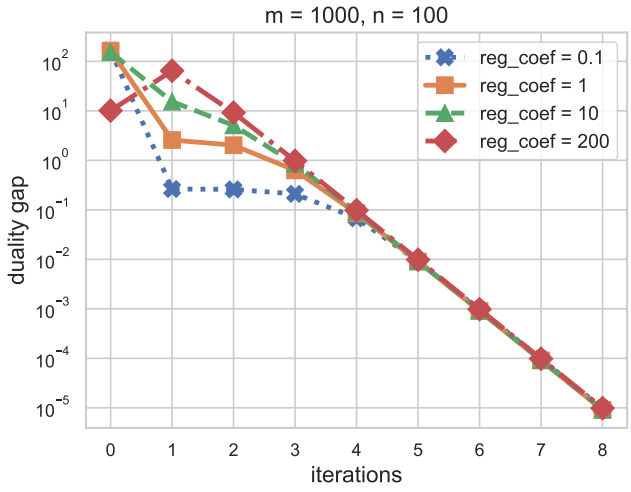
Из рис. 2 (a) можно сделать вывод, что чем больше γ, тем меньше внешних итераций требуется методу, чтобы сойтись. Рис. 2 (b) позволяет сделать предположение, что время работы алгоритма обратно пропорционально γ, если последняя лежит в «разумных» пределах. Имеется в виду следующее: как известно из лекций, γ отвечает за компромисс между количеством внешних и внутренних итераций. То есть пока методу Ньютона будет хватать «сил», чтобы сойтись от очередного начального приближения к следующему решению вспомогательной задачи, то общее реальное время работы будет сокращаться по мере увеличения γ. Однако, если выбрать последнюю слишком большой, то из-за плохой стартовой точки метод Ньютона может разойтись (в силу ограничения на количество внутренних итераций). Если же предположить, что этого ограничения нет, то для перехода от решения вспомогательной задачи для t_k к решению для γt_k может потребоваться много итераций, возможно, гораздо больше, чем суммарно требуется при маленьких γ, чтобы получить искомое ε-приближение для исходной задачи. Поэтому при фиксированном ε и $\gamma \to \infty$, время работы будет расти. Кроме того, при слишком больших γ могут возникнуть вычислительные трудности, например, переполнение или NaN.

3.2 Поведение метода при различных значениях m, n и λ

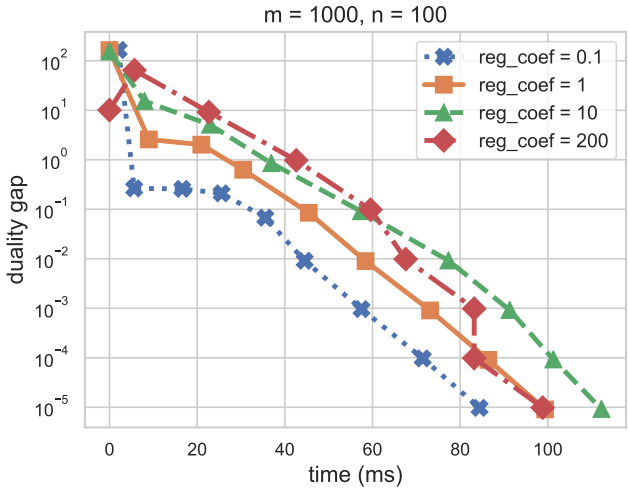
В данном эксперименте мы выполним полный перебор по сетке

$$m \in \{1000, 5000, 10000\},\ n \in \{50, 100, 1000\}$$

и для каждой пары размерностей построим требуемые графики при всех $\lambda \in \{0.1, 1, 10, 200\}$. Остальные параметры метода выставлены в свои значения по умолчанию. Данные генерируются таким же образом, как в предыдущем разделе. Приведем некоторые из получившихся графиков и сделаем по ним выводы:

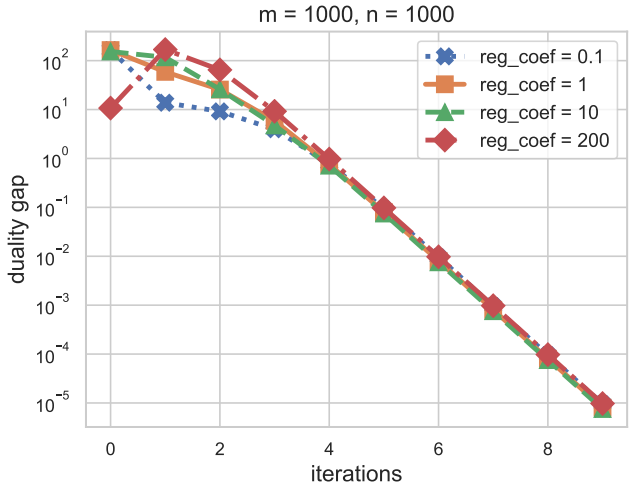


(a) Зависимость зазора двойственности от номера итерации

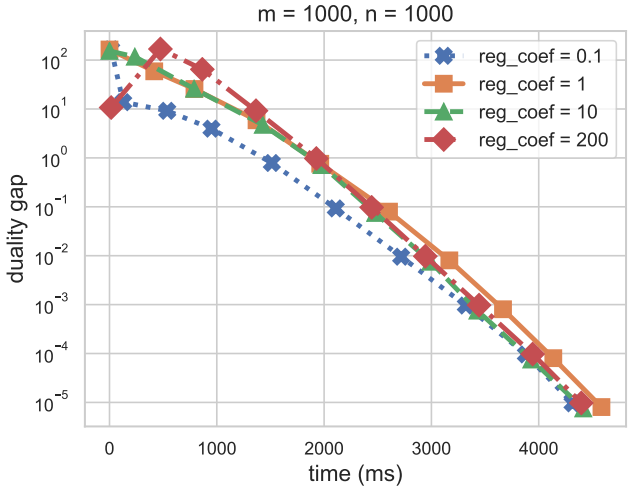


(b) Зависимость зазора двойственности от реального времени работы

Рис. 3: Результаты эксперимента при $m = 1000$ и $n = 100$



(a) Зависимость зазора двойственности от номера итерации



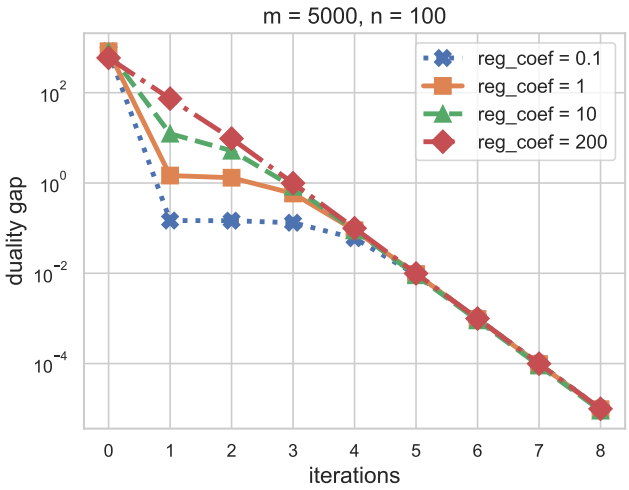
(b) Зависимость зазора двойственности от реального времени работы

Рис. 4: Результаты эксперимента при $m = 1000$ и $n = 1000$

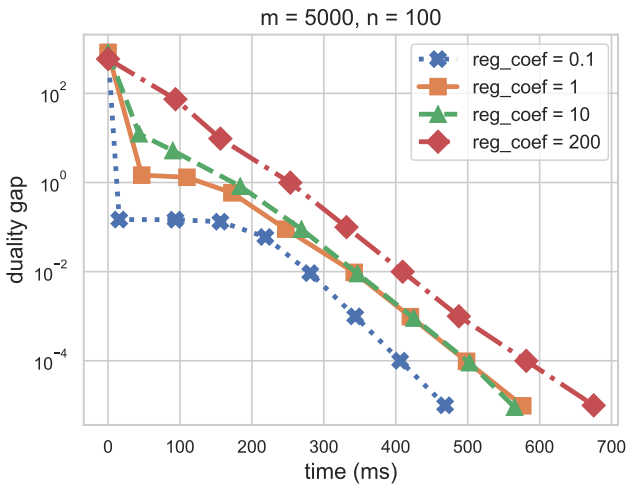
Сперва заметим, что по мере увеличения коэффициента регуляризации λ уменьшение зазора двойственности на первой итерации становится все менее значительным. А в случае $\lambda = 200$ он даже увеличивается по сравнению со стартовой точкой. При этом, начиная с некоторого момента и до сходимости, значения двойственных зазоров примерно одинаковы при всех λ, как и количество итераций, потребовавшихся методу для сходимости. Отсюда вывод, что в нашем случае коэффициент регуляризации существенно не влияет на прогресс метода по итерациям. Однако, это будет выполняться до тех пор, пока $\lambda < \|A^T b\|_\infty$. Иначе, как написано в условии задания, оптимальное решение исходной задачи будет нулевым, а поскольку мы выбрали в качестве начальной точки $x_0 = 0$, то метод сойдется за 0 итераций. Этим же можно объяснить и скачок зазора двойственности на первой итерации при большом λ: в силу того, что оптимальное решение в таком случае разреженное, стартовая точка оказалась к нему ближе, чем точка, полученная после первой итерации. Далее метод «понимает», что нужно идти обратно в сторону нуля, и зазор двойственности начинает стремительно уменьшаться.

Также из рис. 3 и 4 можно сделать вывод, что при увеличении размерности пространства n реальное время работы метода и количество итераций до сходимости увеличиваются. Значения двойственных зазоров на каждой итерации при $n = 1000$ примерно на порядок больше,

чем соответствующие значения при $n = 100$. Это можно объяснить тем, что в методе Ньютона решается СЛАУ с матрицей размера $n \times n$, что, очевидно, будет становиться все дороже с ростом n. Более того, сама оптимизируемая функция становится более сложной при увеличении размерности пространства, поэтому и итераций до сходимости нам будет требоваться все больше.

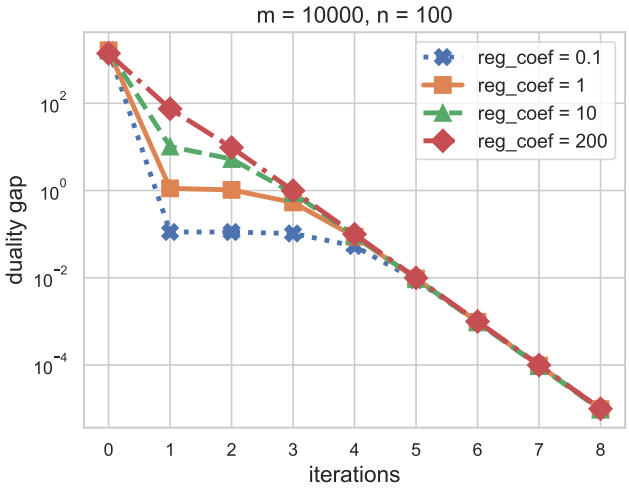


(a) Зависимость зазора двойственности от номера итерации

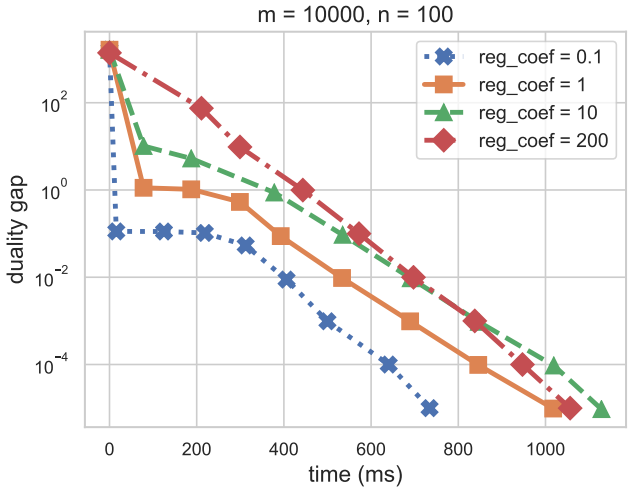


(b) Зависимость зазора двойственности от реального времени работы

Рис. 5: Результаты эксперимента при $m = 5000$ и $n = 100$



(a) Зависимость зазора двойственности от номера итерации



(b) Зависимость зазора двойственности от реального времени работы

Рис. 6: Результаты эксперимента при $m = 10000$ и $n = 100$

Вернемся к влиянию на наш метод коэффициента регуляризации λ. Из рис. 5 (b) и 6 (b), можно сделать вывод, что при всех достаточно малых λ время работы метода пропорционально λ. Интуитивно, по мере уменьшения коэффициента регуляризации негладкая часть исходной функции играет все меньшую роль, и при $\lambda \to 0$ наша задача становится все более «похожа» на обычную задачу линейной регрессии, которая решается проще и, стало быть, быстрее.

Осталось отметить, какое влияние на метод оказывает размер выборки m. В целом, сравнивая рис. 3 (b), 5 (b) и 6 (b), можно сказать, что время работы увеличивается при росте m, однако не так стремительно, как при увеличении n (см. рис 4 (b)). То же самое можно говорить и о влиянии с точки зрения итераций — оно определенно есть, но не столь сильное, как у размерности пространства n. Это тоже логичный результат, если мы вспомним оценку на время

решения СЛАУ в методе Ньютона — $\mathcal{O}\left(n^2m + \frac{1}{3}n^3\right)$. От m она зависит линейно, в то время как от n — кубически.

Отметим, что уже после проведения экспериментов реализация метода была оптимизирована (матрица A^TA сохраняется в памяти, а не вычисляется каждый раз заново), поэтому влияние размера выборки m на время работы метода на самом деле может быть еще меньше.

4 Источники

Д. Кропотов. Курс «Методы оптимизации в машинном обучении», лекция 10, 2022;

Д. Кропотов. Курс «Методы оптимизации в машинном обучении», семинар 10, 2022;

А. Попов, М. Находнов. Подготовка текстовых отчётов в системе TEX, 2020.