Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

Отчет по практическому домашнему заданию 1

Курс "методы оптимизации в машинном обучении"

Выполнил Петров Олег Евгеньевич, 193 группа

Содержание

1 Вывод формул для логистической регрессии

Attention: в данном решении у матрицы объектов выборки A объекты расположены по столбцам. Таким образом, если ℓ - размер выборки, а d - количество признаков, тогда $A \in \mathbb{R}^{d \times l}$. В программной реализации используется транспонирование для приведения матрицы к стандартному виду, то есть A переходит в A^T и наоборот.

1.1 Функционал

$L(x) = \frac{1}{\ell} \sum_{i}^{l} \ln(1 + \exp(-b_i \langle a_i, x \rangle)) + \frac{\lambda}{2} \|x\|_2^2$

Сумма усреднена, значит ее можно представить в виде вектора. Обозначим этот вектор как v и проведем преобразования:

$$v = \begin{bmatrix} \ln(1 + \exp(-b_1 \langle a_1, x \rangle)) \\ \dots \\ \ln(1 + \exp(-b_\ell \langle a_\ell, x \rangle)) \end{bmatrix} = \ln \begin{bmatrix} 1 + \exp(-b_1 \langle a_1, x \rangle) \\ \dots \\ 1 + \exp(-b_\ell \langle a_\ell, x \rangle) \end{bmatrix} =$$

$$\ln(1_\ell + \exp \begin{bmatrix} -b_1 \langle a_1, x \rangle \\ \dots \\ -b_\ell \langle a_\ell, x \rangle \end{bmatrix}) = \ln(1_\ell + \exp(-u)),$$

где $u_i = b_i \langle a_i, x \rangle$

Вектор u можно представить в виде адамарова произведение вектора скалярных произведений столбцов матрицы A и вектора x с вектором b. Тогда $u = A^T x \circ b$. Тогда $v = \ln(1_\ell + exp(-A^T x \circ b)$. Далее компоненты вектора усредняются: $\frac{\langle v, 1_\ell \rangle}{\ell} = \frac{v^T \cdot 1_\ell}{\ell}$. Таким образом, формула функционала логистической регрессии в матричном виде:

$$L(x) = \frac{1}{\ell} [\ln(1_\ell + exp(-A^T x \circ b)]^T \cdot 1_\ell + \frac{\lambda}{2} \|x\|_2^2$$

1.2 Градиент

Вывод градиента в векторно-скалярном виде, затем итоговая формула переводится в матрично-векторный. Считаем dL, обозначив $u_i = b_i\langle a_i, x\rangle$ и $L_i = \ln(1 + e^{-u_i})$:

$$dL_i(x) = \frac{1}{1+e^{-u_i}} d(1+e^{-u_i}) = -\frac{e^{-u_i}}{1+e^{-u_i}} d(u_i) = -e^{-u_i}\frac{1}{1+e^{-u_i}} b_i\langle a_i, dx\rangle$$

Заметим, что это можно привести к более простому виду:

$$-e^{-u_i}\frac{1}{1+e^{-u_i}} = \frac{1-(1+e^{-u_i})}{1+e^{-u_i}} = \frac{1}{1+e^{-u_i}} - 1 = \sigma(u_i) - 1,$$

σ – логистическая сигмоида. Таким образом,

$$dL_i(x) = \sigma(u_i) b_i\langle a_i, dx\rangle - b_i\langle a_i, dx\rangle \Longrightarrow \nabla L_i = \sigma(u_i) b_i a_i - b_i a_i$$

Дифференциал регуляризатора: $\frac{\lambda}{2} d\|x\|^2 = \lambda\langle x, dx\rangle$, его градиент – λx.
Градиент функционала предварительно будет выглядеть так:

$$\nabla L = \sum_i^l L_i + \lambda x = \sum_i^l \sigma(u_i) b_i a_i - \sum_i^l b_i a_i + \lambda x$$

Заметим, во-первых, что $\sum\limits_i^l b_i a_i = Ab$ (линейная комбинация векторов-столбцов).
Во-вторых, заметим, что $\sum\limits_i^l \sigma(u_i) b_i a_i$ - тоже линейная комбинация столбцов, тогда ее можно представить в виде Az, где вектор $z = \begin{bmatrix} \sigma(u_1) b_1 \\ \dots \\ \sigma(u_\ell) b_\ell \end{bmatrix}$.
Его можно разложить через адамарово произведение:

$$z = \begin{bmatrix} \sigma(u_1) \\ \dots \\ \sigma(u_\ell) \end{bmatrix} \circ b = \sigma(u) \circ b = \sigma(A^T x \circ b) \circ b$$

.

Таким образом, формула градиента:

$$\nabla L = \frac{1}{\ell} A\Big(\sigma(A^T x \circ b) \circ b\Big) - \frac{1}{\ell} Ab + \lambda x$$

1.3 Гессиан

Для вывода гессиана считается второй дифференциал функционала в компонентном виде:
$d^2 L_i[dx_1, dx_2] = d\Big(\sigma(u_i) b_i \langle a_i, dx_1\rangle - b_i \langle a_i, dx_1\rangle\Big) = d\sigma(u_i) b_i \langle a_i, dx_1\rangle - 0 = \sigma(u_i)(1-\sigma(u_i)) du_i b_i \langle a_i, dx_1\rangle = \sigma(u_i)(1-\sigma(u_i)) b_i \langle a_i, dx_2\rangle b_i \langle a_i, dx_1\rangle =$
$dx_1^T \Big[a_i \sigma(u_i)(1-\sigma(u_i)) b_i^2 a_i^T\Big] dx_2$, тогда:

$$\nabla^2 L_i = \underbrace{a_i}_{\mathbb{R}^{d\times 1}} \underbrace{\sigma(u_i)(1-\sigma(u_i)) b_i^2}_{\mathbb{R}} \underbrace{a_i^T}_{\mathbb{R}^{1\times d}} \in \mathbb{R}^{d\times d}$$

Использовалась формула производной логистической сигмоиды:

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

Отметим, что гессиан регуляризатора равен λI_d.

Обозначим $v_i = \sigma(u_i)(1-\sigma(u_i)) b_i^2$, тогда $\sum\limits_i^\ell \nabla_2 L_i = \sum\limits_i^\ell a_i v_i a_i^T$. Наличие в программной реализации функции, вычисляющей значение $A^T \cdot \mathrm{diag}(s) \cdot A$ позволяет предположить, что сумма выше представляется именно так (с учетом того, что в данном теоретическом решении матрица A транспонированная). Пусть сумма выше равна $A \cdot \mathrm{diag}(v) \cdot A^T$. Убедимся в правильности предположениям с помощью прямой проверки:

$$A \cdot \mathrm{diag}(v) \cdot A^T =$$

$$\begin{bmatrix} | & | & \dots & | \\ a_1 & a_2 & \dots & a_\ell \\ | & | & \dots & | \end{bmatrix} \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & v_\ell \end{bmatrix} \begin{bmatrix} -- & a_1 & -- \\ -- & a_2 & -- \\ \vdots & \vdots & \vdots \\ -- & a_\ell & -- \end{bmatrix} =$$

$$\begin{bmatrix} \sum a_i^1 v_i a_i^1 & \dots & \sum a_i^1 v_i a_i^d \\ \vdots & \ddots & \vdots \\ \sum a_i^d v_i a_i^1 & \dots & \sum a_i^d v_i a_i^d \end{bmatrix} = \sum_i^\ell \begin{bmatrix} a_i^1 v_i a_i^1 & \dots & a_i^1 v_i a_i^d \\ \vdots & \ddots & \vdots \\ a_i^d v_i a_i^1 & \dots & a_i^d v_i a_i^d \end{bmatrix} = \sum_i^\ell a_i v_i a_i^T$$

Теперь разложим вектор v.

$$v = \begin{bmatrix} b_1^2 \sigma(u_1)(1-\sigma(u_1)) \\ \vdots \\ b_\ell^2 \sigma(u_\ell)(1-\sigma(u_\ell)) \end{bmatrix} = \begin{bmatrix} \sigma(u_1)(1-\sigma(u_1)) \\ \vdots \\ \sigma(u_\ell)(1-\sigma(u_\ell)) \end{bmatrix} \circ b^2 =$$

$$(\sigma(u) - \sigma(u)^2) \circ b^2 = (\sigma(A^T x \circ b) - \sigma(A^T x \circ b)^2) \circ b^2$$

Здесь возведение в степень 2 - поэлементная операция.

Таким образом, гессиан функционала:

$$\nabla^2 L = \frac{1}{\ell} A \cdot \mathrm{diag}((\sigma(A^T x \circ b) - \sigma(A^T x \circ b)^2) \circ b^2) \cdot A^T + \lambda I_d$$

2 Эксперимент: Траектория градиентного спуска на квадратичной функции

2.1 Дизайн эксперимента

Основная цель данного эксперимента – проанализировать траектории трех методов градиентного спуска (стратегии выбора шага на основе условий Армихо, Вульфа и метод на основе константного шага). Требуется определить взаимосвязь между траекторией спуска и числом обусловленности функции

(имеется в виду число обусловленности матрицы $\text{cond(A)} = \|A\|\|A^{-1}\|$), выбором стартовой точки и выбором стратегии непосредственно.

Предлагается оптимизировать квадратичные векторные функции следующего вида:

$$\frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle,\ A \in \mathbb{S}^n_{++},\ b \in \mathbb{R}^n$$

Для эксперимента были подобраны следующие данные:

1 Матрицы A: $A_1 = \begin{bmatrix} 12 & -\frac{1}{2} \\ -\frac{1}{2} & 5 \end{bmatrix}$, $A_2 = \begin{bmatrix} 25 & -9 \\ -9 & 4 \end{bmatrix}$, $A_3 = \begin{bmatrix} 10 & 7 \\ 7 & 13 \end{bmatrix}$

2 Векторы b: $b_1 = \begin{bmatrix} -12 \\ 6 \end{bmatrix}$, $b_2 = \begin{bmatrix} 17 \\ -7 \end{bmatrix}$, $b_3 = \begin{bmatrix} -15 \\ -10 \end{bmatrix}$

3 Стартовые точки x: $x_1 = \begin{bmatrix} 4 \\ -4 \end{bmatrix}$, $x_2 = \begin{bmatrix} -4 \\ -5 \end{bmatrix}$

4 Константа для константого метода спуска равна 0.05. Остальные гиперпараметры – по умолчанию.

Каждая функция f_i задается матрицей A_i и вектором b_i:

$$f_i(x) = \frac{1}{2}\langle A_i x, x\rangle - \langle b_i, x\rangle$$

Каждый график представляет две траектории спуска из стартовых точек $x_1,\ x_2$ ровно для одной функции и ровно для одного метода. Графики выводятся по три в ряд для каждой функции и нумеруются по правилу строка.столбец. Три графика, выведенные в i-ой по порядку строке представляют собой для функции i траектории методов спуска на основе условий константных, Армихо и Вульфа. соответственно. Таким образом, график под номером 3.2 описывает траектории спуска методом Армихо из двух стартовых точке для f_3. Также подсчитываются числа обусловленность каждой функции в каждой точке и выводятся в виде таблицы. На каждом графике также отображены линии уровней функций.

2.2 Результаты

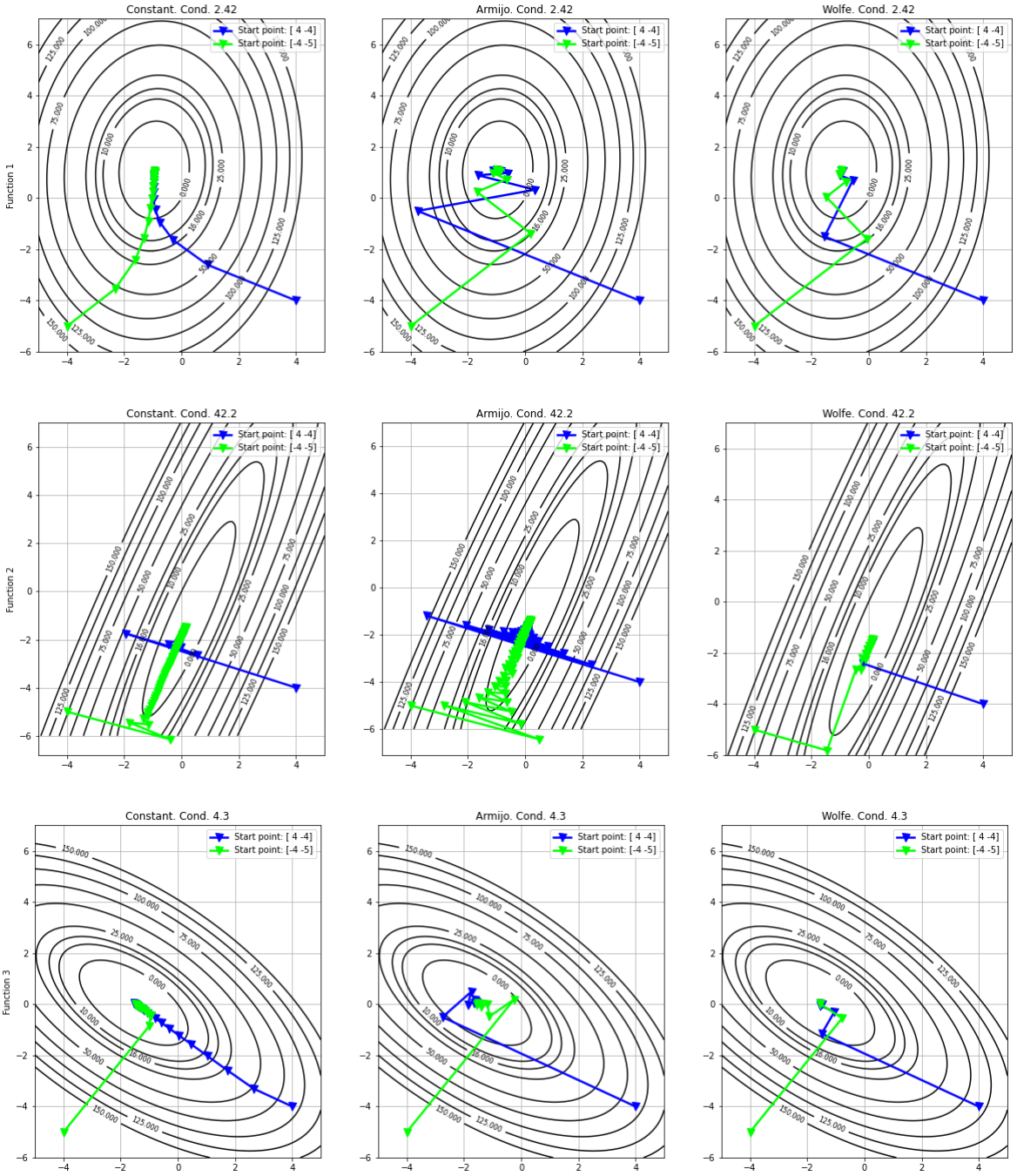

Рис. 1. Траектории спуска для различных методов и функций.

2.3 Выводы

- Любой метод будет вести себя одинаково, если его запустить из разных начальных точек в контексте одной функции;
- Отметим, что в большинстве случаев у всех методов первый шаг - наиболее длинный.

- Константный метод сходится постепенно, за большое число шагов. Метод чувствителен к числу обусловленности – чем оно больше, тем дольше сходится метод. Если плохо подобрать длину шага, то будем долго сходиться. Метод сходится медленнее остальных (по числу шагов);

- Метод на основе условии Амирхо по числу шагов сравним с константным,. Если сравнивать графики 1.1, 1.2 и 3.1, 3.2, то можно заметить, что метод старается делать большие шаги, поэтому его траектория в сравнении с траекторией константного метода выглядит менее плотной и "структурированной". Графики 2.1 и 2.2 примечательны тем, что имеют схожие траектории для обеих стартовых точек. Заметко, что шаг метода Амирхо уменьшается плавно, последовательно, в то время как у константного методы сначала большой шаг, затем резко маленький. Можно сказать, что про своим траекториям метод Амирхо является чем-то средним между константным методом и методом Вульфа;

- Метод Вульфа сходится быстрее всех других (за меньшее число шагов). Он делает достаточно большие и эффективные шаги, и на приведенных графиках видно, что он не кружит вокруг минимума;

- Расположение начальной точки не принципиально, однако чем ближе точка к локальному минимуму, тем лучше мы будем сходится с точки зрения длины шага – нам будет сложнее перескочить минимум, если первые шаги были не очень большими;

- Приведенные матрицы имеют разные числа обусловленности. Самое большое – у A_2. Можно сказать, что чем больше число обусловленности матрицы, тем более вытянутые линии уровней у соответствующей функции. Это усложняет сходимость (графики 2.1, 2.2, 2.3) – в окрестностях минимума любой метод совершает много мелких колебаний. В целом, можно с уверенностью сказать, что число обусловленности матрицы прямо пропорционально числу шагов алгоритма.

3 Эксперимент: Зависимость числа итераций градиентного спуска от числа обусловленности и размерности пространства

3.1 Дизайн эксперимента

Основная цель данного эксперимента – проанализировать траекторию метода градиентного спуска и определить ее зависимость от размерности пространства и числа обусловленности матрицы. Для этого мы для каждой размерности пространства n на всех доступных числах обусловленности k запустим градиентный спуск на сгенерированных данных (seed фиксирован) и посчитаем $T(n, k)$ - число итераций. Повторим процедуру 3 раза и отобразим на графике $T(n, k)$ против k семества функий для каждого n (выделим каждое семейство разным цветом). Самый первый запуск процедуры характеризуется непрозначными линиями, прозрачность уменьшается по мере повторения процедуры.

Рассматриваем следующие данные:

- $n = 10^i \in \{1, 2, 3, 4, 5, 6\}$
- k – натуральное число от 10 до 1060 с шагом 50
- Матрица A_k генерируется из случайного вектора $a \sim U_n[1; k]$ размера n, при этом после генерации вектора двум различным позициями присваиваются значения $1,\ k$. После $A_k = \mathrm{diag}(a)$
- Вектор b_k генерируется из многомерного равномерного распределения $U[-k; k]$
- Стартовая точка x генерирутся из симметричного равномерного распределение (см. результаты)

Градиентный спуск использует гиперпараметры по умолчанию, однако максимально допустимое чисто итераций увеличено до 1000. Если для какоу-нибудь пары $n,\ k$ метод не сошелся, тогда $T(n,k) = 0$.

3.2 Результаты

Результаты эксперимента, как оказалось, сильно зависят от стартовой точки. На рисунке 2 мы берем $x \sim U_n[-0.05; 0.05]$. На рисунке 3 берем $x \sim U_n[-50; 50]$. На первом рисунке 2 кажется, что при росте n число обусловленности k матрицы A не влияет на сходимость – практически на всем отрезке при $n \geqslant 1000$ число итераций не превышает 50, но и не достигает нуля – значит, метод сходится. На рисунке 3 при любом n имеется тенденция к увеличению числу итераций при параллельном увеличении числа обусловленности. Интересно то, что при росте размерности пространства n графики семейств функций имеют более сглаженный характер. Например, этот эффект хорошо заметен, если сравнивать первые два порядка размерности с двумя последними.

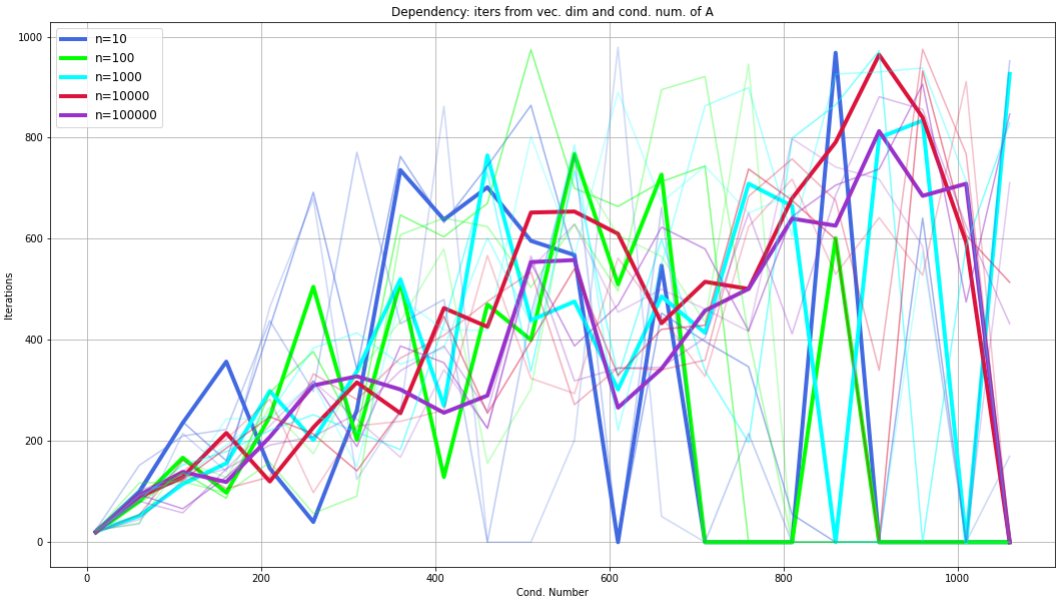

Рис. 2. Зависимость числа итераций от числа обусловленности. x_0 близок к нулю.

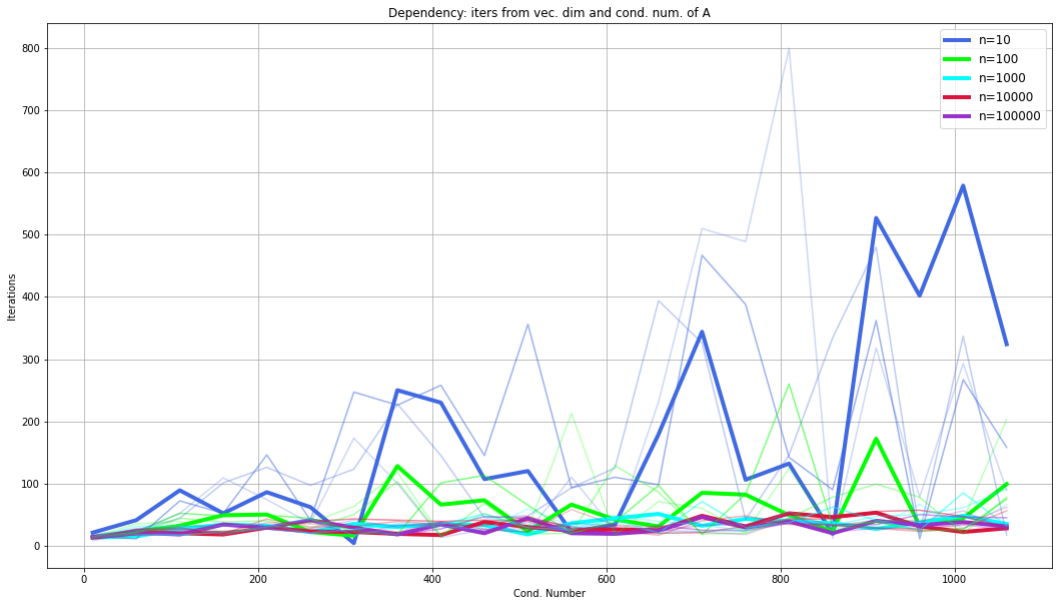

Рис. 3. Зависимость числа итераций от числа обусловленности.

x_0 далек от нуля.

3.3 Выводы

- Увеличение размерности пространства сглаживает закономерности роста числа итерация при росте числа обусловленности;
- В конкретом эксперименте далекая от нуля стартовая точка способствует неувеличению числа итераций при больших порядках размерности. Противоположная ситуация с близкой к нулю стартовой точкой – явная тенденция на увеличение числа итераций.

Такая разница может быть обоснована тем, что у генерируемых нами функций точка минимума близка к нулевому вектору. Поэтому когда берется стартовая точка, близкая к нулю (то есть близкая к минимуму), то методу приходится делать много итераций, чтобы найти минимум. Высокое число обусловленности говорит о том, что функция имеет вытянутые линии уровня, что ухудшает сходимость.

Если же мы берем далекую от нуля точку, то метод начинает сходиться с больших шагов, постепенно их уменьшая, вследствие чего минимум достигается более уверенно.

4 Эксперимент: Какая точность оптимизации нужна в реальных задачах для метода Ньютона?

4.1 Дизайн эксперимента

Основная цель эксперимента – оценить влияние точности оптимизации на качество решения с помощью метода Ньютона. Предлагается оценить error rate (доля неправильных ответов) для каждого значения точности на разных данных.

Мы рассматриваем следующие наборы данных:

1 Wine Quality Classification;

2 Fraud detection;

3 Hospital Readmissions.

Все наборы данных содержат в себе только числовые признаки и отличаются по размерам – порядка 1500, 20000, 60000 объектов соответственно. Последний оказался слишком большим для оперативной памяти, поэтому он был усечен на 30% с использованием stratified fashion (метод разделения на фолды таким образом, чтобы каждый содержал примерно одинаковый процент объектов для каждого класса).

Затем данные были разделены на тренировочные и валидационные выборки в соотношении 3 к 1 – на тренировочных обучалась модель, на валидационных проводилась оценка. Таким образом, error rate, фигурирующий на графиках, был получен на валидационной выборке.

Метод Ньютона использовался со стандартными гиперпараметрами, максимально допустимое число итераций было увеличено до 500.

4.2 Результаты

На графиках изображены зависимости относительно наборов данных в том же порядке, что и в нумерации выше, т.е. первый график соответствует первому набору, второй – второму и т.д.

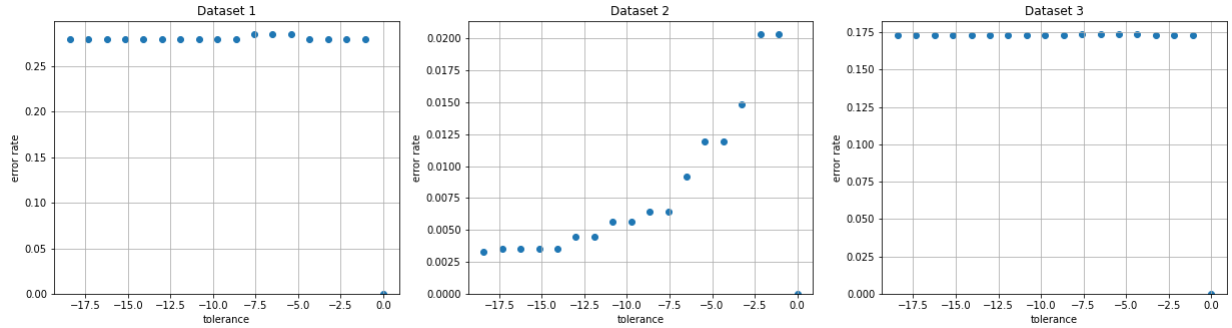

Рис. 4. Зависимость качества от точности.

Первое, на что следует обратить внимание – необычный вид графика для второго набора данных. На нем получится очень низкий error rate на валидации. Это может показаться ошибкой на первый взгляд, однако такое качество обусловенно особенностями выборки. На том же наборе была запущена библиотечная логистическая регрессия с солвером 'newton-cg' и точностью $1e-12$ – на валидации error rate был равен 0.017. Тем не менее, заметно, что при уменьшении точность качество модели незначительно ухудшается (речь идет о доли в пределах 1-2%).

Сильно отличаются от второго первый и третий графики. На них видны незначительные колебания доли неверных ответов, error rate практически константный.

Стоит отметить, что крайняя правая точка для $\ln(\text{tolerance}) = 0$ является особенностью реализации и подсчета метрики. При такой точности алгоритм завершается на первой же итерации, и возвращается вектор нулей (так как из этой точки мы запустили спуск, и она не успела обновиться). Учитывая то, как мы посчитываем error rate, при умножении на нулевой вектор мы всегда получим error rate равный нулю.

4.3 Выводы

Значение точность крайне незначительно влияет на итоговое качество решения. Возможны колебания в пределах сотых и даже тысячных долей, но существенной закономерности не наблюдается. Тем не менее, значение точности может влиять на качество, но это целиком и полностью зависит от данных, на которых обучается модель.

5 Эксперимент: Сравнение методов на реальной задаче логистической регрессии

5.1 Дизайн эксперимента

Основная цель эксперимента – взять несколько реальных наборов данных для бинарной классификации, запустить на них стандартный градиентный спуск и спуск по методу Ньютона, отрисовать и проанализировать графики зависимости значения функционала от времени и итерации и между отношением $\frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_0)\|^2}$ и временем. Время работы отображает число секунд, прошедших с момента запуска обучающего цикла.

Предлагаются использовать следующие наборы: w8a, gisette, real-sim, news20.bina rcv1.binary. Их особенность заключается в громоздкости, обычно эти данные представляются в разреженном формате.

Мы будем поочередно запускать градиентный спуск и метод Ньютона (со стандартными гиперпараметрыми) из нулевой точки на каждом наборе данных и отрисовывать соответствующие графики.

5.2 Результаты

К сожалению, попытка получить графики для предлагаемых наборов данных обернулась неудачей – оперативная память машины (как локальный, так и облачной) не позволяет решить задачу. Вероятно, это связано с тем, что

сложение разреженной матрицы с другой (необязательно плотной) приводит к формированию плотной матрицы, которая не помещается в память. Это означает, что реализуемые нами методы не подходят для решения задач с большими наборами данных.

Для того, чтобы завершить эксперимент и сделать некоторые выводы, было принято решение выполнить эксперимент, используя данные из пункта 4.1 отчета. Каждая строка представленных графиков соответствует одному датасету (в том же порядке, что и в эксперименте 4).

Сразу же заметно, что метод Ньютона сходится лучше с точки зрения минимального значения функции на любом наборе данных.

Мы видим существенное превосходство метода Ньютона над стандартным градиентным спуском на первом наборе данных: метод Ньютона сошелся очень быстро, в то время так конкурирующему методу не хватило и 10000 итераций. Аналогичная картина относительно зависимости между значением функции и временем. Примечательно, что отношение квадратов норм у стандартного градиентного спуска колеблется очень сильно, в то время как у метода Ньютона отношение норм убывает практически мгновенно.

На втором наборе данных стандартный градиентный спуск попадает на плато, и с какого-то момента значение функционала прекращает существенно убывать. Метод Ньютона легко справляется и преодолевает плато, несмотря на то, что стандартный градиентный спуск достигает своего минимального значения быстрее по времени, однако по числу итераций метод Ньютона оказывается лучше. Это говорит о том, что на данном наборе данных одна итерация метода Ньютона занимает больше времени. Отношение квадратов норм градиентов также, как и на предыдущем наборе данных колеблется у стандартного градиентного спуска, а у метода Ньютона убывает монотонно.

На последнем наборе данных метод Ньютона сошелся быстрее по итерациям, а стандартный градиетный спуск – по времени, выйдя на плато как и в предыдущем случае. Абсолютная разница в минимально достигнутых значениях функионала составляет менее 0.04, что может существенно сказаться на

качестве решения (в таких случаях нужно считать значение оценивающей качество метрики). Примечательно, что отношение квадратов норм длительно монотонно убывает у метода Ньютона и практически мгновенно у стандартного градиентного спуска.

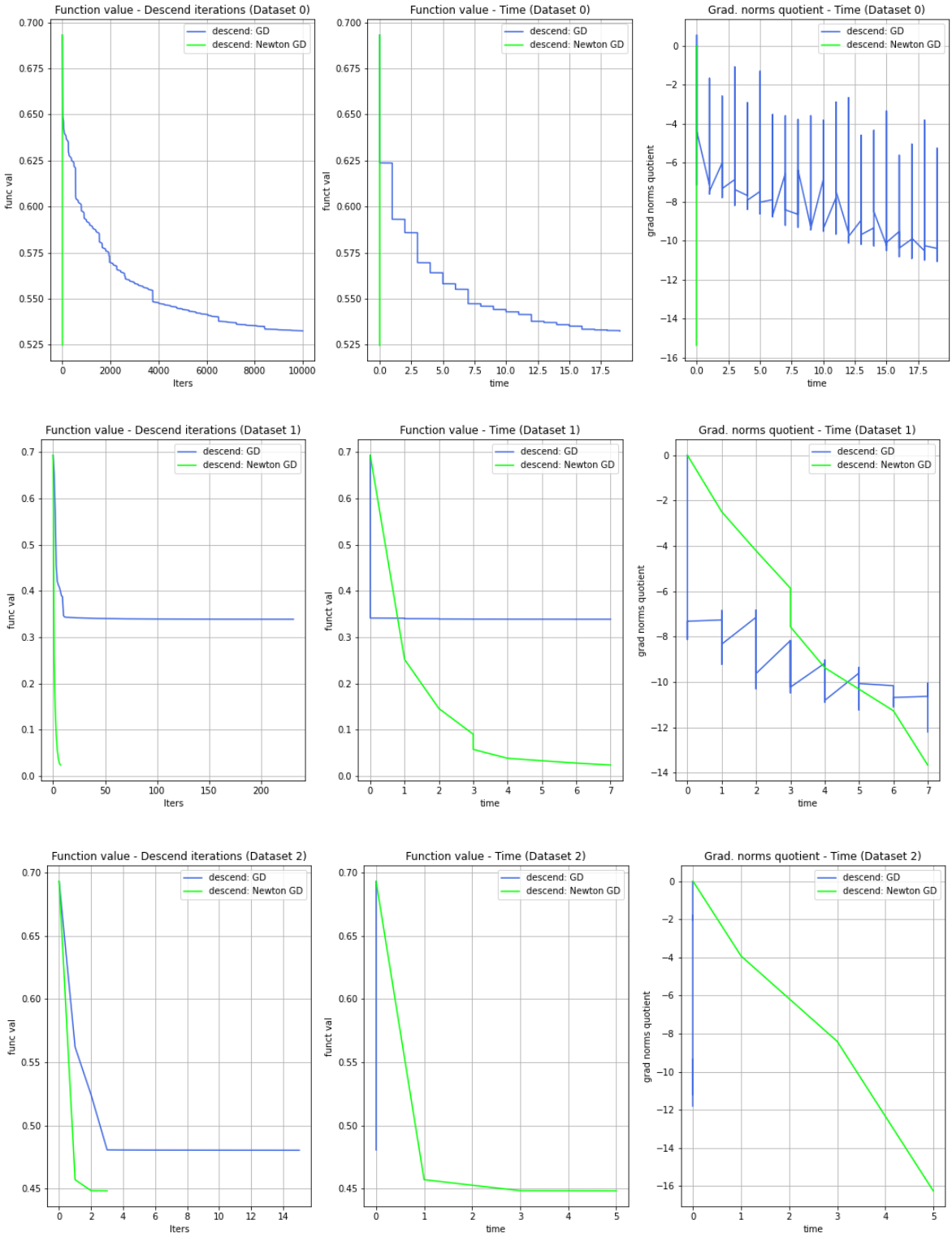

Рис. 5. Зависимость качества от error rate

5.3 Выводы

На разных наборах данных оба метода могут вести себя по-разному, однако в данном эксперименте метод Ньютона показал себя лучше, так как во всех случаях он позволяет достичь меньшее значение функционала, а значит, лучше его оптимизирует. Кроме того, отмечается, что как правило отношение квадратов норм градиентов монотонно убывает, а одна итерация метода Ньютона требует больше времени, чем итерация стандартного градиентного спуска. Кроме того, в нашей реализации метод Ньютона не способен работать с большими выборками, так как имеет множество матричных операций внутри, что не позволяет ему использовать разреженность.

Хоть метод Ньютона и является более продвинутым, все равно нельзя точно сказать, что тот или иной метод лучше в некоторой определенной ситуации – выбор метода зависит от конкретной выборки и гиперпараметров. Однако метод Ньютона не стоит использовать, если выборка очень большая. В любом случае, чтобы достичь хорошего качество модели необходимо экспериментировать.