ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
ИНСТИТУТ ДИЗАЙНА И УРБАНИСТИКИ
НАПРАВЛЕНИЕ: ЦИФРОВОЕ ЗДРАВООХРАНЕНИЕ

Курсовой проект по дисциплине: «Анализ и разработка алгоритмов»

Тема: Квазиньютоновские методы нелинейной оптимизации без ограничений (Symmetric Rank 1, BHHH algorithm, BFGS algorithm, Limited-memory BFGS)

Выполнили: Кошман Варвара (С4113),
Самарин Антон (С4113),
Гончаров Андрей (С4113)

Санкт-Петербург
2019

Вступление

Класс квазиньютоновских методов часто используется в задачах на нахождение минимума функционала, так как предлагает более оптимальную по времени и памяти альтернативу методу Ньютона. Шаг для обновления координаты в методе Ньютона определяется отношением:
$x_{n+1} = x_n - [H_{x_n} f]^{-1} \nabla a_n f$
, где обратный гессиан $[H_{x_n} f]^{-1}$ на каждой итерации метода считается численно, что затратно и долго, так как нахождение обратной матрицы требует решения системы линейных уравнений. Symmetric Rank 1, BHHH, BFG, Limited-memory BFGS алгоритмы - наиболее часто применяемые квазиньютоновские методы, которые не вычисляют обратную матрицу, а дают ей оценку, пользуясь свойством ее симметричности.

Общий алгоритм для всех квазиньютоновских методов имеет следующий вид:

1) выбираются начальное приближение для точки старта, в этой точке считаются первая и вторая производные ($x_0,\ H_0,\ g_0$) (можно инициализировать Якобиан и Гессиан единичными матрицами)
2) пока соседние точки не буду достаточно близки (с заданной точностью ε):
   находится направление поиска, и в этом направлении делается шаг, величина которого чаще всего определяется соответственно условиям Вольфе - такой шаг, чтобы он оптимально минимизировал функцию в этом направлении:

   $$argmin_{a>0} f(x_k + a * (-H_k * g_k))$$

   $$x_{k+1} = x_k + a_k * (-H_k * g_k)$$
   $$g_{k+1} = \nabla x_{k+1} f$$

   и обновляется значение приближенного Гессиана согласно определенному алгоритму:
   $H_{k+1} = \text{алгоритм}(H_k, x_k, x_{k+1}, g_k, g_{k+1})$

Именно по выбору стратегии для обновления H_k существует разделение квазиньютоновских методов.

Методы

BHHH

Численный метод, имеющий широкое распространение в эконометрических задачах. Целевая функция представляется как функция правдоподобия, которую необходимо максимизировать. Целевая функция должна иметь следующий вид:

$$f(x) = N^{-1} \sum_{i=1}^{N} q(w_i, x),$$

где x это неизвестный параметр нашего распределения а qi - результаты эксперимента. Из нашей функции максимального правдоподобия можно использовать равенство информационных матриц из которого вытекает, что гессиан можно представить как внешнее произведение градиентов. Т.е.

$$\nabla_x^2 f(x) \cong N^{-1} \sum_{i=1}^{N} \nabla_x q(w_i, x)' \nabla_x q(w_i, x),$$

Из основных плюсов данного метода, что он позволяет не высчитывать гессиан. Из минусов: 1) Начальное значение должно быть близко к истинным параметрам 2) Выборка должна быть достаточно большой 3) Правильная целевая функция, необходимо для равенства информационных матриц.

BFGS

Наиболее часто используемый квазиньютоновский алгоритм. Не накладывает ограничения на x, от f требует непрерывность вторых производных.

Вместо точного вычисления гессиана метод предлагает добавлять к текущему значению две симметричные матрицы:

$$B_{k+1} = B_k + U_k + V_k$$

Причем, чтобы сохранить симметричность и положительную определенность получаемой матрицы, нужно гарантировать:

$$B_{k+1} = B_k + \alpha \mathbf{u}\mathbf{u}^\top + \beta \mathbf{v}\mathbf{v}^\top$$

коэффициенты α, β получаются из условий секущих, а для получения из полученного приближения Гессиана приближение его обратной матрицы

используется формула Шермана-Моррисона, давая то значение, которое используется для обновления на каждой итерации:

$$B_{k+1}^{-1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) B_k^{-1} \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}$$

При этом, градиент-вектор y_k хранит всю историю градиентов, в случаях, когда n слишком большое, рекомендуется использовать модификацию с ограниченной памятью.

L-BFGS

L-BFGS - модификация BFGS, которая, во-первых, не хранит в явном виде Гессиан (то есть не выделяет память под хранение n*(n+1)/2 элементов), а, во-вторых, в вычислениях использует не все градиенты, а некоторое небольшое число(меньше 10) последних градиентов, что обеспечивает экономию по времени и памяти, но не всегда на небольших n показывает лучшие результаты, чем BFGS.

SR1

Еще один метод, который, как и BFGS, использует для обновления первые производные, но в отличие от него не гарантирует положительную определенность получаемой матрицы.

$$\boldsymbol{H}_{k+1} \leftarrow \boldsymbol{H}_k + \frac{\boldsymbol{w}_k \boldsymbol{w}_k^T}{\boldsymbol{w}_k^T \boldsymbol{y}_k} \qquad \boldsymbol{w}_k = \boldsymbol{s}_k - \boldsymbol{H}_k \boldsymbol{y}_k$$

По некоторым численным экспериментам SR1 сходится к точному Гессиану быстрее, чем BFGS, однако важный его недостаток в том, что знаменатель может обнулиться из-за слишком маленьких градиентов и привести к расходимости алгоритма. Как вариант решения проблемы, можно обновлять приближение Гессиана только в том случае, если этот множитель не лежит в некоторой заданной малой окрестности:

$$\left|w_k.T * y_k\right| \geq \varepsilon * \left\|w_k\right\| * \left\|y_k\right\|$$

Однако, если использовать линейный поиск для нахождения оптимального шага, удовлетворяющего условиям Вольфе, проблема с исчезающими градиентами все равно всплывает. Фиксированный шаг может быть не эффективный, по крайней мере, не всегда. Поэтому этот метод лучше применять на разряженных данных.

Модель

Эффективность работы всех методов была сравнена на решении задачи линейной регрессии: восстановления некоторых коэффициентов a и b $\sim U(0,\ 1)$ функции $y = a * x + b + d$, где $x = 1/k,\ k = 1..100,\ d \sim N(0,\ 1)$. Линии регрессии для каждого метода были отображены, аппроксимируя исходные зашумленные данные. Сходимость методов была сравнена в терминах итераций и отображена графически.

Модель линейной регрессии выбрана была отчасти потому, что реализованные вручную методы Symmetric Rank 1 и BHHH-алгоритм оказались чувствительны к большому числу неизвестных в целевой функции, в зависимости от начального приближения часто становились численно нестабильны и расходились. Однако проверить BFGS и L-BFGS и сравнить с традиционными методами на реальной задаче машинного обучения все еще интересно: на примере с линейной регрессией с всего двумя неизвестными не заметны преимущества L-BFGS над BFGS.

В качестве примера была выбрана задача бинарной классификации пациентов с сердечными болезнями: на основании 13 признаков принимать решении о наличии какого-либо сердечного заболевания.

Целевая оптимизируемая функция имеет вид:

$$f(\theta) = 1/2 * m * \sum_{i=1}^{n}(-y * ln(sigmoid(\theta * x) - (1 - y) * ln(1 - sigmoid(\theta * x)))$$

Функция минимизировалась BFGS, L-BFGS, методом Ньютона и градиентным спуском с фиксированным шагом. Для каждого метода на тестовой выборке была оценена точность классификатора на полученном им векторе параметров и подсчитано общее число итераций.

Результаты

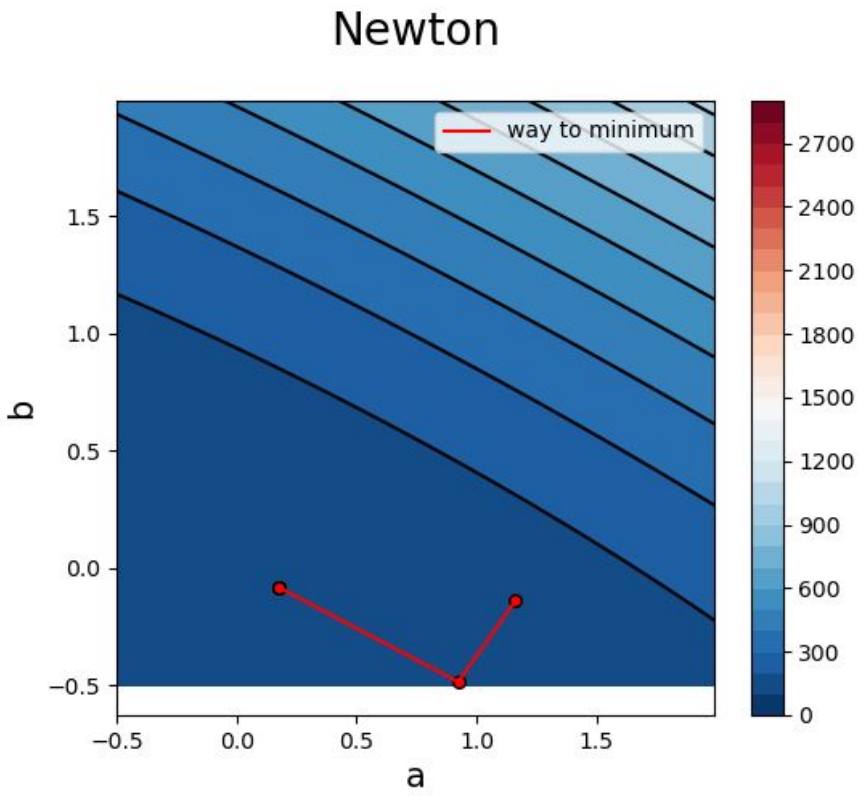
Коэффициенты регрессии, с которыми генерировалась исходная выборка:
a = 0.13977738144051421
b = 0.008890950408530718

method	root (precision = 10e-3)	# of iterations
Newton	[0.17520752 -0.07967756]	4
BFGS	[0.17518778 -0.07966777]	2
L-BFGS	[0.17518773 -0.0796678]	5

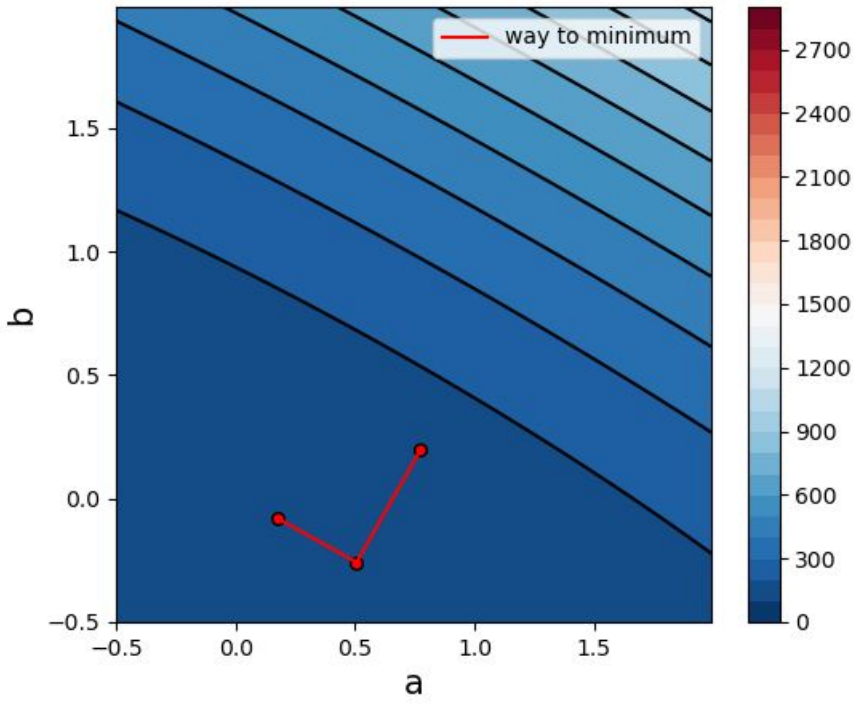
SR1	[0.17499377 -0.07958188]	10
BHHH	[0.1751878 -0.07966781]	10

Таблица 1: восстановленные коэффициенты регрессии (с точностью до шума) всеми методами и соответствующие количества итераций

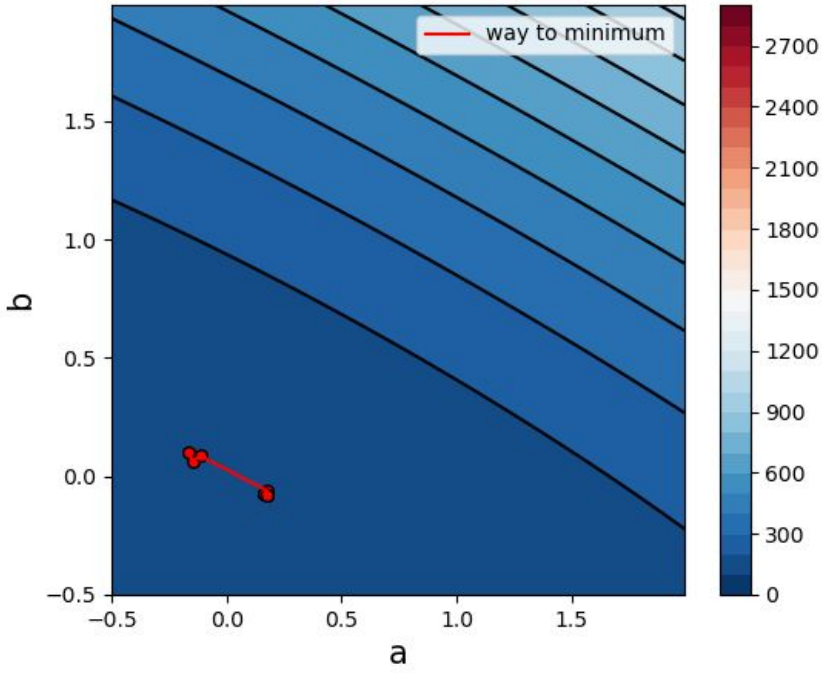
Построение пути сходимости для методов:

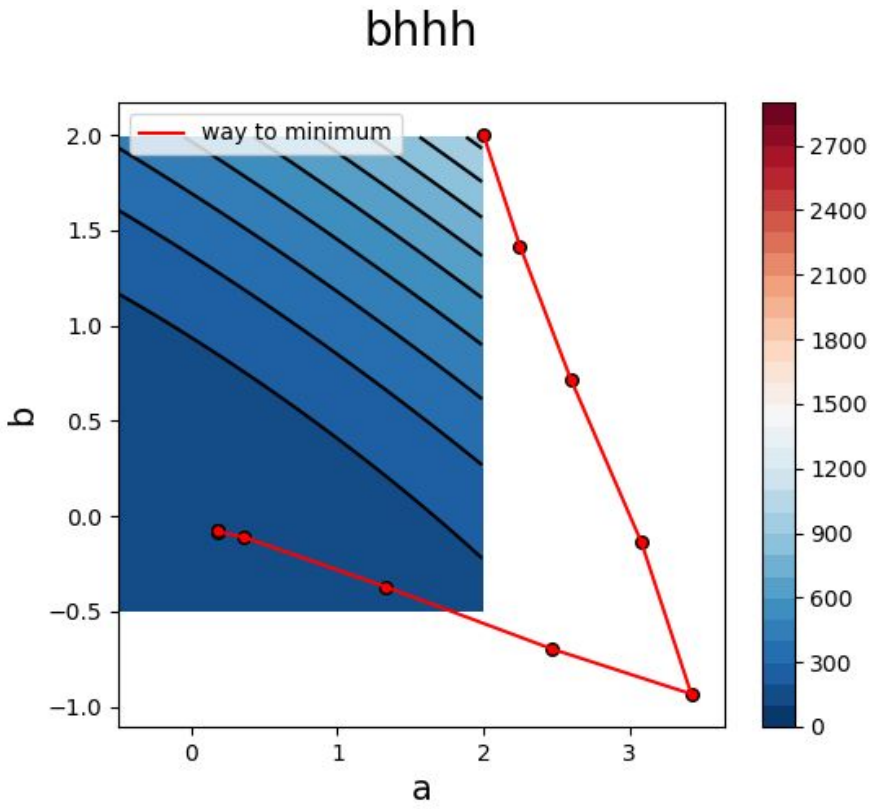


BFGS


way to minimum
2700
2400
2100
1800
1500
1200
900
600
300
0
1.5
1.0
0.5
0.0
−0.5
b
−0.5
0.0
0.5
1.0
1.5
a

SR1


way to minimum
2700
2400
2100
1800
1500
1200
900
600
300
0
1.5
1.0
0.5
0.0
−0.5
b
−0.5
0.0
0.5
1.0
1.5
a



Полученные линии регрессии отображены на зашумленном наборе исходных данных. Видно, что найденные точки минимума совпадают для всех методов.

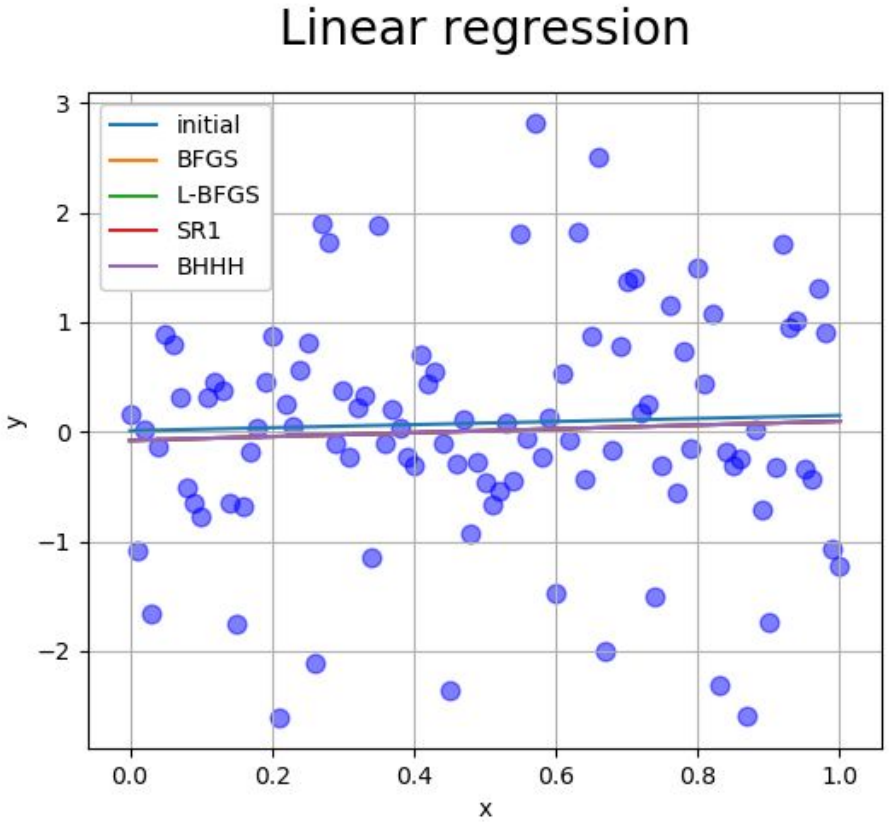


Таблица результатов для задачи классификации пациентов:

method	accuracy	# of iterations
Gradient descent with fixed step	85.25%	17752
Newton	85.25%	9
BFGS	85.25%	79
Limited memory BFGS	85.25%	20

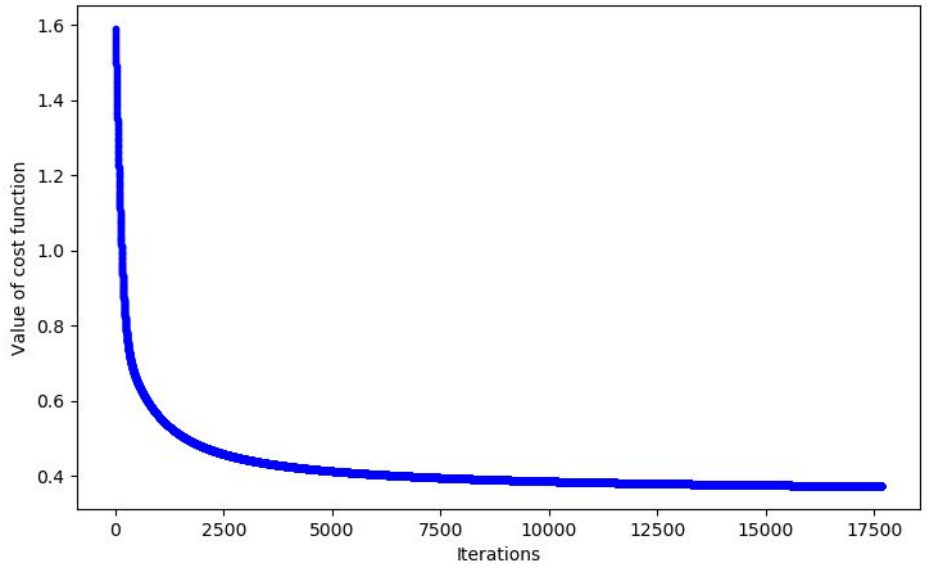


График зависимости минимизируемой целевой функции от числа итераций

Заключение

На примере линейной регрессии видно, что все методы сходятся за небольшое число итераций, хотя BFGS всегда несколько быстрее.

Полученный классификатор в 85.25% случаев правильно определял сердечную болезнь, видно, что метод Ньютона и квазиньютоновские BFGS и L-BFGS по числу итерации сходятся к тем же значениям гораздо быстрее. Ясно также, что метод Ньютона сходится несколько быстрее квазиньютоновских, это понятно, так как его обновления на каждой итерации точны, но по памяти это гораздо затратнее, так что использование квазиньютоновских методов хорошо обусловлено.

Ссылки

GitHub:
https://github.com/vkoschman/heart-disease-classification-with-Quasi-Newton-methods

Dataset: https://www.kaggle.com/ronitf/heart-disease-uci

Quasi-Newton methods for minimization - Lectures for PHD course on Numerical optimization:
http://www.ing.unitn.it/~bertolaz/2-teaching/2011-2012/AA-2011-2012-OPTIM/lezioni/slides-mQN.pdf

On Optimization Algorithms for Maximum Likelihood Estimation. Anh Tien Mai, Fabian Bastin, Michel Toulouse
https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2014-64.pdf

Unconstrained Numerical Optimization An Introduction for Econometricians, Anders Munk-Nielsen
http://web.econ.ku.dk/munk-nielsen/notes/noteOptimization.pdf