Московский государственный университет им. М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по работе на тему

Экспериментальное сравнение методов SAG, SG и FG.

Выполнил студент 216 группы
Измаилов Павел Алексеевич

Москва, 2015

Введение

В данной работе мною произведено сравнение трех методов оптимизации — градиентного спуска (FG), стохастического градиента (SG) и стохастического среднего градиента (SAG) на задачах линейной и логистической регрессии. Работа методов сравнивается на модельных данных а также на наборах данных, расположенных на сайте библиотеки LIBSVM — `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

1. Рассматриваемые задачи

В данной работе рассматриваются задачи линейной и логистической регрессии. В случае линейной регрессии целевая функция имеет вид

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i\theta^T\right)^2,$$

а в случае логистической регрессии

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n}\ln\left(1 + e^{-y_i x_i \theta^T}\right)^2 + \frac{\gamma}{2}\|\theta\|^2.$$

Здесь $y_i \in \mathbb{R}^n$ и $x_i \in \mathbb{R}^{(m,n)}$, $\gamma > 0$ — фиксированные вектор, матрица и число соответственно.

С более общей точки зрения, рассматривается задача оптимизации с целевой функцией, представляющей собой сумму большого числа выпуклых функций:

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + \frac{\gamma}{2}\|\theta\|^2,$$

где все функции f_i выпуклы, $\theta \in \mathbb{R}^m$, а γ — некоторая неотрицательная константа.

2. Рассматриваемые методы

В данной работе рассматриваются следующие методы:

- FG (Full Gradient) — метод градиентного спуска. Итерация этого метода имеет вид
$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k f'(\hat{\theta}_k).$$
В качестве правила выбора длины шага используется правило Армихо.

- SG (Stochastic Gradient) — метод стохастического градиента. Итерация этого метода имеет вид
$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k f_i'(\hat{\theta}_k),$$
где индекс i выбирается случайно. В качестве правила выбора длины шага используется правило вида
$$\alpha_k = \frac{\alpha_0}{k^{0.5+\delta}}.$$
Параметры α_0 и δ выбирается вручную так, чтобы работа метода была наилучшей.

- SAG (Stochastic Average Gradient)— метод стохастического среднего градиента. Итерации метода SAG имеют следующий вид:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} y_k^i,$$

где на каждой итерации индекс i_k выбирается случайным образом из набора $[1, \ldots, n]$, и

$$y_k^i = \begin{cases} f_i'(\hat{\theta}_k), \text{ если } i = i_k \\ y_{k-1}^i, \text{ иначе.} \end{cases}$$

В качестве длины шага используется $\alpha_k = \frac{1}{\ell}$, где ℓ — оценка константы Липшица целевой функции. Эта оценка обновляется на каждой итерации по следующему правилу: если нарушается неравенство

$$f_{i_k}(\hat{\theta}_k - \frac{1}{\ell^k} f_{i_k}'(\hat{\theta}_k)) \leq f_{i_k}(\hat{\theta}_k) - \frac{1}{2\ell^k} ||f_{i_k}'(\hat{\theta}_k)||^2,$$

то ℓ^k удваивается до тех пор, пока неравенство не начнет выполняться.

Все методы реализованы на языке Python.

Чтобы сделать сравнение по времени работы более честным, методы SG и SAG реализованы с использованием вычислений сериями — на каждой итерации этих методов вычисляется градиент не для одного слагаемого, а для суммы нескольких слагаемых. Это решение связано с тем, что в языке Python использование векторных вычислений позволяет значительно ускорить работу методов. В случае отказа от вычислений сериями метод FG побеждает методы SG и SAG по времени работы практически всегда. Размеры серий выбираются так, чтобы скорость работы методов была оптимальной.

3. Экспериментальные результаты

В этом разделе приводятся результаты работы методов на различных наборах данных. Методы сравниваются по эпохам (сравнивается значение целевой функции после каждого прохода методов по набору данных) и по времени работы (сравниваются результаты работы методов за равные промежутки времени). В каждом случае методы запускается с ограничением на максимальное число итераций, и достижение этого максимального числа является критерием остановки (другие правила остановки не используются, так как они оказали бы влияние на время работы методов).

3.1. Модельные данные

В данном разделе методы сравниваются на модельных данных, сгенерированных программным образом.

3.1.1. Линейная регрессия

Здесь будет рассмотрена работа методов на задаче линейной регрессии с данными, сгенерированными программным образом. Данные генерируются следующим образом:

1. Генерируется случайный массив $X \in \mathbb{R}^{n,m}$, этот массив отшкалирован так, что все его элементы лежат в заданном интервале.
2. К массиву X приписывается столбец, состоящий из единиц.
3. Генерируется значение $\theta \in \mathbb{R}^m$.
4. По X и θ вычисляется вектор $y = X\theta^T + \varepsilon$, где ε — случайный шум.

Таким образом, у этой задачи два параметра: количество данных n и размерность задачи m.

Для данной задачи получены следующие результаты.

- n порядка не превосходящего 10^4. В этом случае размер серий, которыми оперируют стохастические методы (при меньших размерах сравнение по времени теряет смысл).

  Даже несмотря на использование вычислений сериями итерация метода FG происходит значительно быстрее, чем итерация SAG, потому что значение n мало. Итерация метода SG также происходит несколько быстрее метода SAG. В результате (см. рис. 1) в сравнении по времени на ранних эпохах лучше всех показывает себя метод SG. Довольно скоро его обгоняет FG, который затем так и остается в лидерах. Метод SAG через какое-то время обгоняет SG и имеет скорость сходимости близкую к FG.

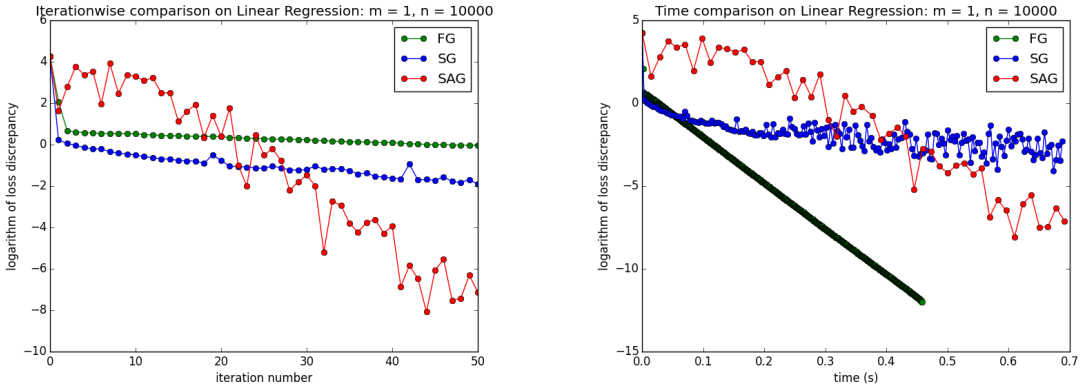

Рис. 1: Линейная регрессия, $n = 10^4, m = 1$

В сравнении по эпохам на ранних эпохах лидирует метод SG, затем его обгоняет SAG.

Таким образом, при малых объемах данных метод на ранних эпохах при правильном выборе длины шага выигрывает метод SG, но вскоре FG и SAG вырываются вперед. Они имеют приблизительно равную скорость сходимости по времени работы, а по эпохам SAG существенно выигрывает.

При возрастании размерности задачи m FG постепенно теряет преимущество перед SAG, заключающееся в более быстрой эпохе. Это приводит к тому (см. рис. 2), что метод SAG начинает работать быстрее FG и SG. Однако достигнув определенного предела методы фактически перестают продвигаться в сторону решения.

- n порядка 10^5 и более. При увеличении размеров выборки соотношение производительности методов по эпохам в общем сохраняется. Однако FG теряет преимущество в скорости на каждой итерации, которое он имел при меньших размерах выборки. В результате (см. рис. 3, 4) SAG все более явно выигрывает у FG. Кроме того, FG требуется все больше итераций, чтобы догнать SG.

Таким образом, при малых объемах данных n для данной задачи целесообразно использовать метод FG, хотя метод SAG показывает себя не многим хуже (при использовании вычислений сериями). При увеличении размеров выборки n или размерности задачи m метод SAG быстро обгоняет FG, а SG становится лучше FG на ранних эпохах.

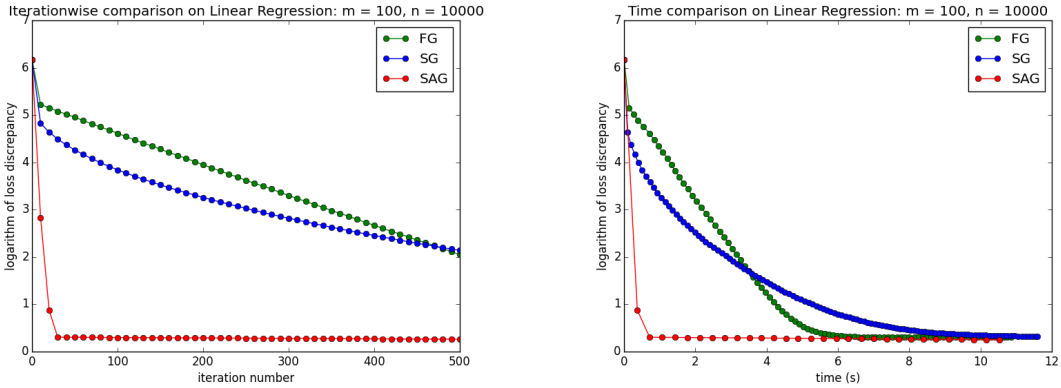

Рис. 2: Линейная регрессия, $n = 10^4, m = 100$

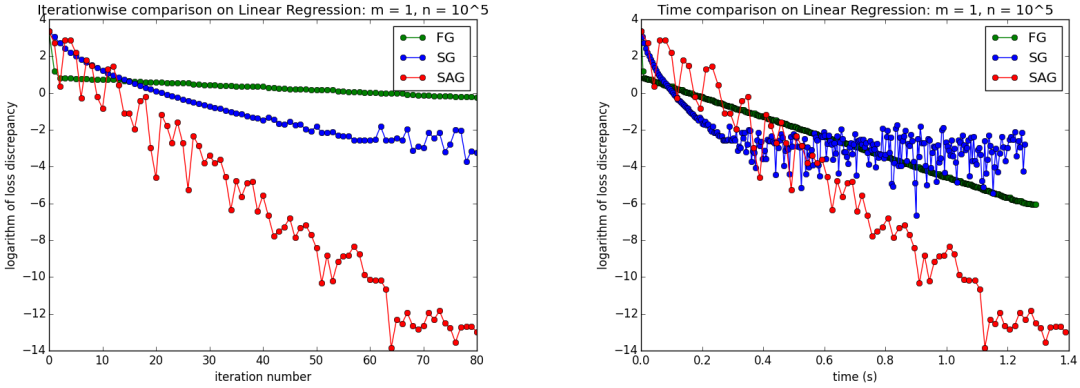

Рис. 3: Линейная регрессия, $n = 10^5, m = 1$

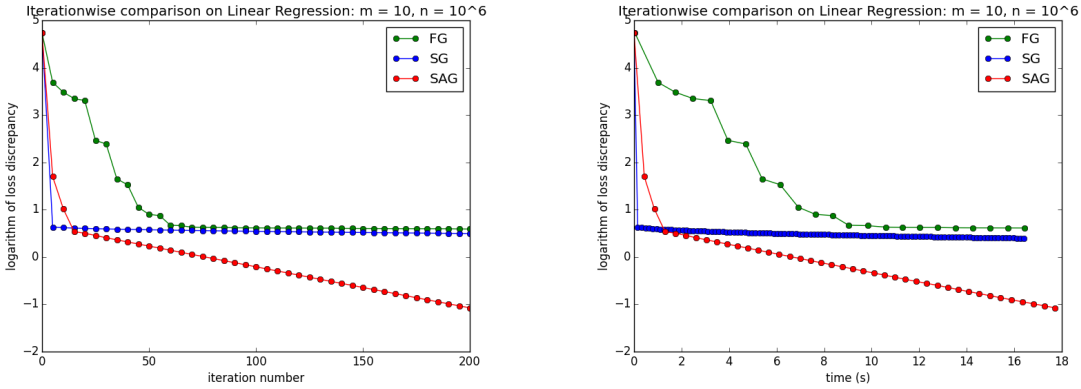

Рис. 4: Линейная регрессия, $n = 10^6, m = 10$

3.1.2. Логистическая регрессия

В этом разделе будет рассмотрена работа методов на задаче логистической регрессии с данными, сгенерированными программным образом. Данные генерируются следующим образом:

1. Генерируется случайный массив $X \in \mathbb{R}^{(n,m)}$, этот массив отшкалирован так, что все его элементы лежат в заданном интервале.

2. К этому массиву добавляется столбец, заполненный единицами.

3. Генерируется значение $\theta \in \mathbb{R}^m$.

4. По X и θ вычисляется вектор $y = \text{sgn}(X\theta^T)$.

Таким образом, у данной задачи, как и у рассмотренной ранее задачи линейной регрессии, два параметра — объем выборки n и размерность задачи m.

Рассуждения в общем совпадают с рассуждениями, приведенными в предыдущем пункте, поэтому ограничимся здесь графиками по времени работы (см. рис. 5, 6).

Основное отличие между этой задачей и задачей линейной регрессии заключается в том, что в данном случае целевая функция является сильно выпуклой. В этом случае теоретическая оценка скорости сходимости метода SAG лучше, чем в общем случае.

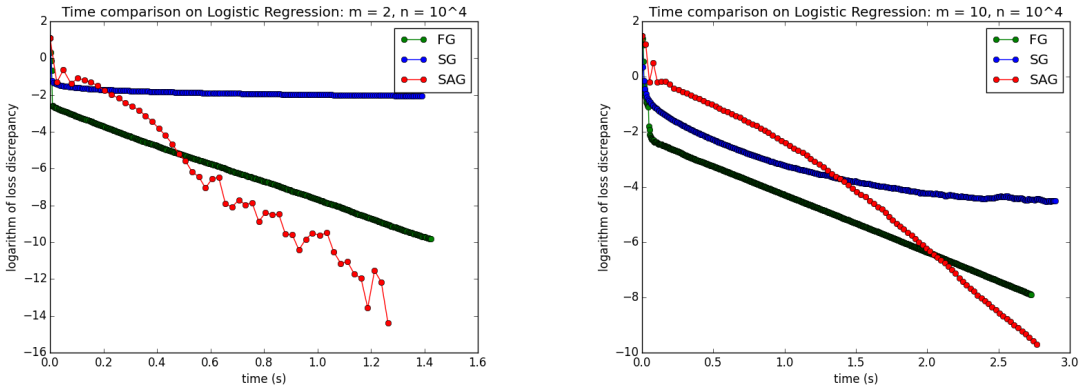

Рис. 5: Логистическая регрессия, $n = 10^4, m = 2$ и $m = 10$

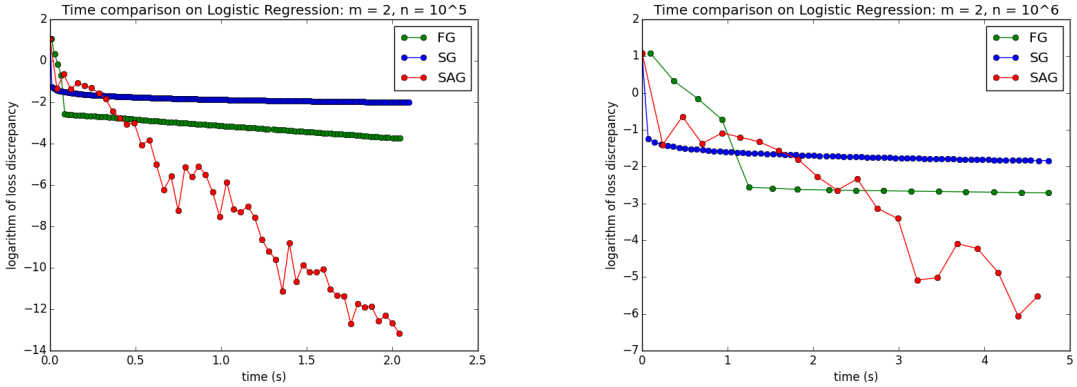

Рис. 6: Логистическая регрессия, $n = 10^5$ и $n = 10^6$, $m = 2$

3.2. Наборы данных небольших размеров

В этом разделе рассматриваются результаты работы методов на наборах данных небольших объемов — порядка 10^3 точек.

3.2.1. Набор данных Fourclass

Данный набор данных предназначен для задачи классификации, он состоит из $n = 861$ точки в $m = 2$-мерном пространстве.

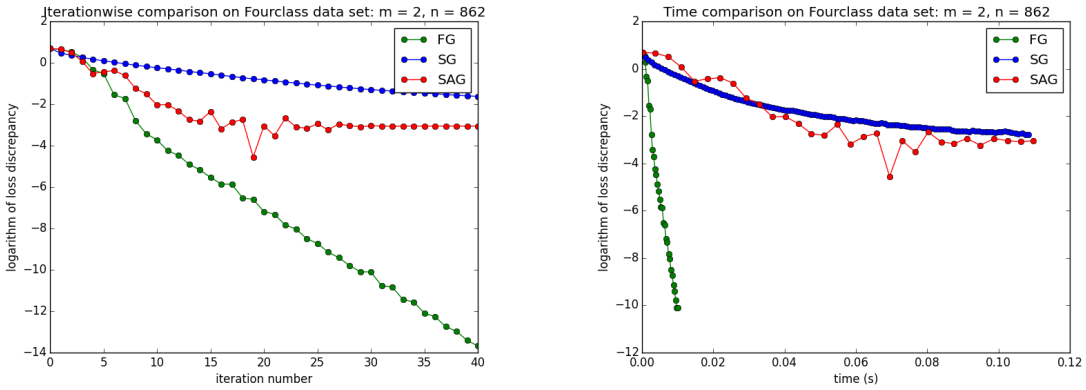

Рис. 7: Набор данных fourclass, $n = 861$, $m = 2$

Графики, приведенные на рис. 7 построены для методов SG и SAG, использующих вычисления сериями, размеры которых составляют порядка 5% объема данных. Видно, по времени ожидаемо выигрывает метод FG, а метод SAG показывает себя неожиданно плохо — SAG и SG показали примерно одинаковые результаты. По эпохам FG также обгоняет SG и SAG, а SG и SAG показывают схожие результаты.

В случае отказа от вычислений сериями (см. рис. 8) SAG в сравнении по эпохам показывает себя лучше, однако по времени работы проигрывает существенно сильнее.

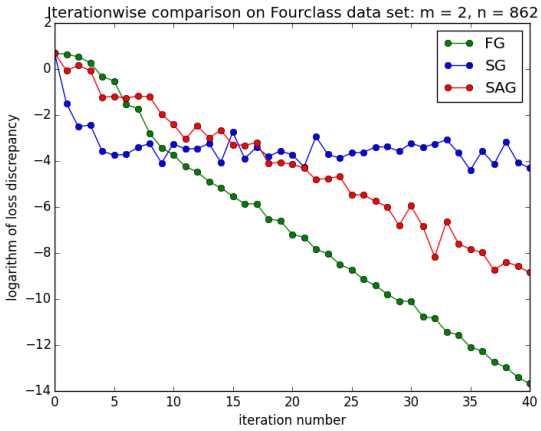

Рис. 8: Набор данных fourclass без использования вычислений сериями $n = 861$, $m = 2$

Таким образом, на данном наборе данных при использовании вычислений сериями SAG показывает себя неожиданно плохо.

3.2.2. Набор данных Australian

Данный набор данных состоит из $n = 690$ точек в $m = 14$-мерном пространстве и предназначен для задачи классификации.

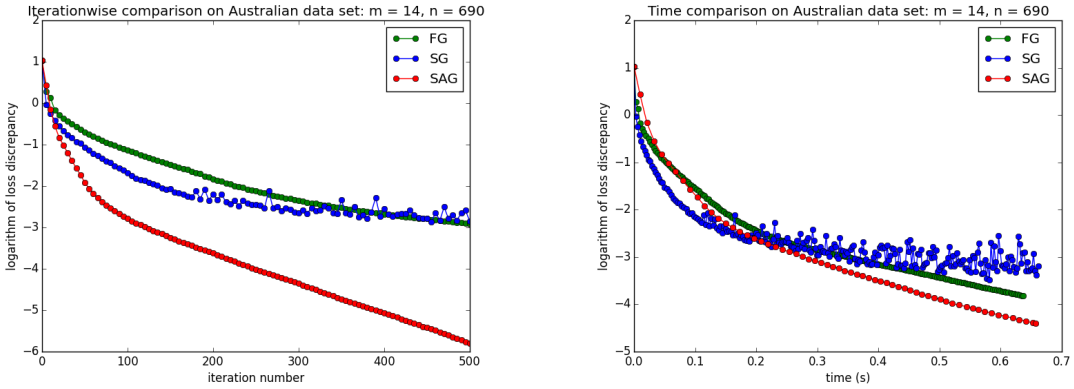

Рис. 9: Набор данных Australian, $n = 690$, $m = 14$

Результаты работы методов приведены на рис. 9. Видно, что SAG ожидаемо выигрывает у FG как по эпохам, так и по времени. По времени на ранних эпохах лидирует SG, но затем его обгоняют как SAG, так и FG.

3.2.3. Набор данных Abalone

Данный набор данных состоит из $n = 4177$ точек в $m = 8$-мерном пространстве, и предназначен для задачи регрессии. Результаты работы методов приведены на рис. 10

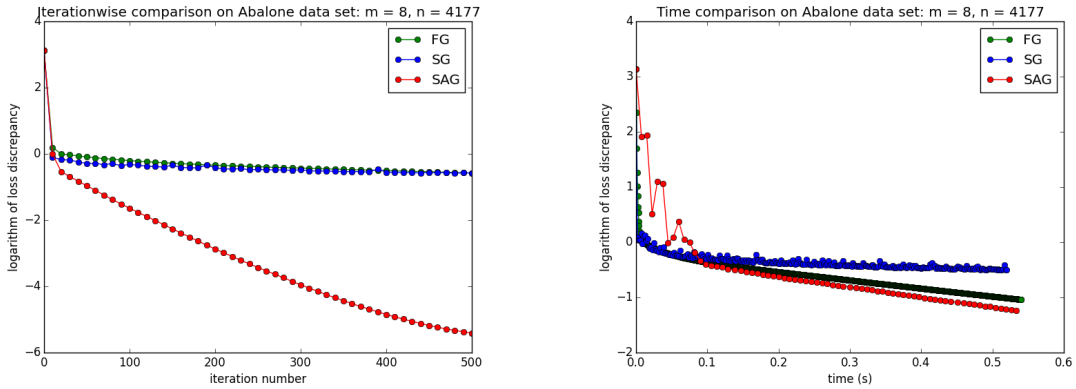

Рис. 10: Набор данных Abalone, $n = 4177$, $m = 8$

Таким образом, мы видим, что на данной задаче регрессии методы качественно ведет себя так же, как и на модельных задачах. На ранних эпохах хорошо себя показывает SG, но вскоре уступает FG и SAG. SAG побеждает FG как по эпохам, так и по времени.

3.3. Наборы данных средних размеров

В этом разделе рассматриваются результаты работы методов на наборах данных средних объемов — порядка 10^4 – 10^5 точек.

3.3.1. Набор данных IJCNN1

Данный набор данных состоит из $n = 49990$ точек в $m = 22$-мерном пространстве, и предназначен для задачи классификации. При таких размерах данных метод SAG начинает существенно выигрывать у FG по времени за счет более быстрых эпох. На данном наборе данных в качестые размера серии для методов SG и SAG использовалось 5% от n. Из графиков (см. рис. 11) видно, что при таких размерах данных SAG значительно обыгрывает FG и лишь на первых эпохах уступает SG. FG же после первых нескольких эпох обгоняет SG.

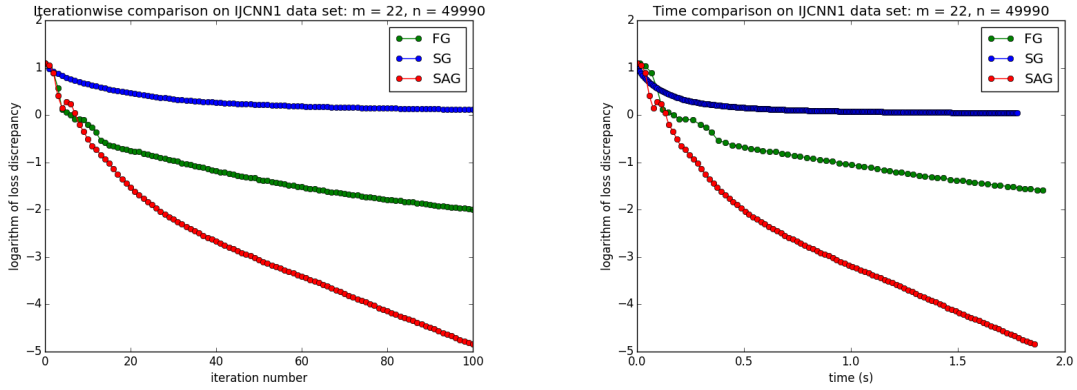

Рис. 11: Набор данных IJCNN1, $n = 49990$, $m = 22$

3.3.2. Набор данных mnist

Данный набор данных состоит из $n = 60000$ точек в $m = 780$-мерном пространстве, и предназначен для задачи классификации. Результаты работы методов приведены на рис. 12.

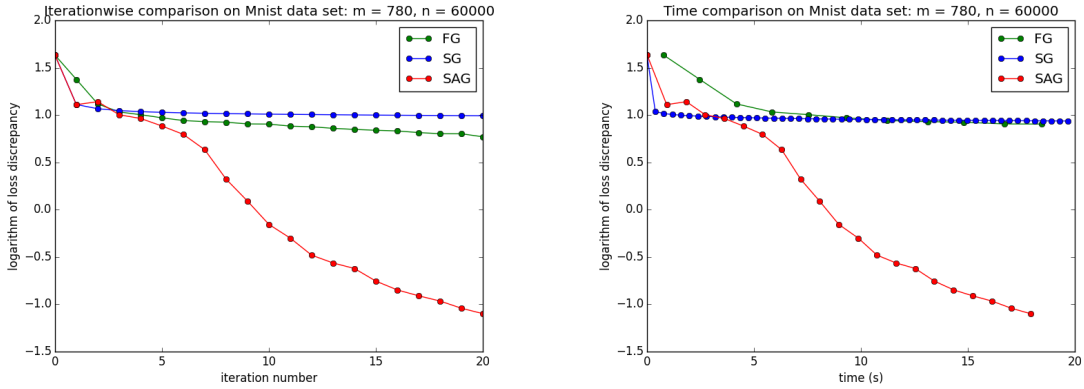

Рис. 12: Набор данных mnist, $n = 60000$, $m = 780$

Как видно из графиков, на ранних эпохах лидирует метод SG, но почти сразу он перестает прогрессировать. Метод SAG быстро обгоняет SG и FG как по времени, так и по эпохам. Метод FG после нескольких итераций догоняет SG, но обогнать его не может.

3.4. Выводы

Из проведенных экспериментов можно заключить следующее:

- При малой размерности задач и малых объемах данных методы FG и SAG показывают близкую производительность. Это обусловлено в частности тем, что при малых размерах данных из-за особенностей языка Python, итерация метода FG происходит приблизительно так же быстро, как итерация метода SAG. В то же время итерация метода SAG примерно так же информативна, как и итерация FG, потому что из-за небольшого количества данных градиенты не успевают сильно устареть.

- При повышении размерности задачи при малых объемах данных все методы ведут себя хуже, чем в случае малых размерностей. Методы SAG и FG ведут себя сходным образом. Во многом это диктуется тем, что итерации методов SAG и FG близки по информативности и трудоемкости, так как последняя определяется в большей степени размерностью задачи, а не количеством рассматриваемых на каждой итерации функций. Метод SG неплохо проявляет себя на ранних итерациях, но быстро перестает прогрессировать.

- При возрастании объема данных метод SAG начинает существенно выигрывать у SG и FG. Это обусловлено тем, что объем серии, используемой при обновлении градиента в методах SAG и SG можно подобрать так, чтобы соотношение времени работы одной итерации и ее информативности было оптимальным, в то время как итерации FG становятся слишком долгими. Метод SG снова хорошо показывает себя на ранних итерациях, но SAG быстро его обгоняет. FG также обгоняет SG засчет линейной скорости сходимости, но происходит это не сразу, и при больших объемах данных в выборе между этими двумя методами может иметь смысл выбирать SG.