Аннотация

Алгоритм lbfgs расшифровывается как limited memory Broyden–Fletcher–Goldfarb–Shanno и является вычислительно улучшенной версией простого bfgs.

Разбор этого алгоритма будет идти серией из трёх частей. В начале мы рассмотрим многие определения и свойства для теоретического обоснования lbfgs, придём к методу Ньютона. Во второй части будет описан сам bfgs, а в заключении рассмотрим хаки, чтобы он стал limited memory и практическую реализацию

1 Первая часть

Согласно википедии, L-BFGS - оптимизационный алгоритм из семейства квазиньютоновских, который приближает алгоритм BFGS, используя ограниченное количество памяти [1]. Проще говоря, он предназначен для максимизации или минимизации функции. В ньютоновских методах необходимо напрямую вычислять гессиан, матрицу вторых производных функции, в то время как квазиньютоновские методы обходятся каким-то его приближением.

Чтобы прийти к ньютоновским методам, в начале нам необходимо рассмотреть метод простой итерации и сжимающее отображение. Сжимающее отображение $\mathcal{A}$ - отображение метрического пространства $\mathcal{M}$, с функцией расстояния ρ, самого в себя, которое уменьшает расстояние между любыми точками. Нам понадобятся три свойства сжимающего отображения.

$\mathcal{A}$ - непрерывное на $\mathcal{M}$, т.е определено во всех точках.

У сжимающего отображения существует единственная неподвижания точка $\hat{x}$: $\mathcal{A}\hat{x} = \hat{x}$.

Итерационная последовательность $x, \mathcal{A}x, \mathcal{A}^2x, \mathcal{A}^3x, ...$ сходится к неподвижной точке $\mathcal{A}$.

Зная это, можем приступить к методу простой итерации. Метод простой итерации предназначен для решения уравнения, а значит мы можем использовать его для решения нашей задачи оптимизации поиском корней уравнения $\nabla f(x)$. Идея метода лежит в том, чтобы для решения уравнения $f(x) = 0$ использовать эквивалетное уравнения $x = \phi(x)$, где ϕ - сжимающее отображение. Из свойств сжимающего отображения следует, что последовательное применение ϕ как $x_i = \phi(x_{i-1})$ сходится к неподвижной точке, что в данном случае равно решению уравнения. Одним из преобразований, которое можно использовать является $\phi(x) = x - l(x)f(x)$, где $l(x) \neq 0$. Понять, почему такое преобразование сохраняет корни можно подставив $\phi(x)$ в $f(x) = 0$.

$f(x) = 0 => xf_1(x) = 0$

$f(\phi(x)) = f(x - l(x)f(x)) = f(x(1 - l(x)f_1(x))) = xf_1(1 - l(xf_1(x))$

Использование метода простой итерации для решения уравнения $\nabla f(x)$ и является методом Ньютона. Для максимальной сходимости необходимо, чтобы для очередного приближение $\hat{x}$ отображение соответствовало условиям $\phi'(\hat{x}) = 0$, т.е чтобы после применения ϕ мы находились в неподвижной точке (производная равная нулю соответствует отсутствию изменения) - а значит в решении нашего уравнения.

Воспользуемся предыдущим выражением для ϕ из метода простых итераций с заменой знака и данным условием.

$\phi(\hat{x}) = \hat{x} - l(\hat{x})f(\hat{x})$

$\phi'(\hat{x}) = (\hat{x})' - (l(\hat{x})f(\hat{x}))'$
$\phi'(\hat{x}) = 1 - l'(\hat{x})f(\hat{x}) - l(\hat{x})f'(\hat{x}) = 0$

В предположении, что точка приближения $\hat{x}$ достаточно близка к корню $\overline{x}$ а значит $f(\hat{x}) \approx f(\overline{x}) = 0$.

Подставив в наше равенство, можем выразить $l(x) = \frac{1}{f'(\hat{x})}$. Тогда итерецпонное преобразование выглядит как $x_i = x_{i-1} - \frac{f(x_{i-1})}{f'x_{i-1}}$.

Для задачи оптимизации функции $f(\vec{x}) : R^n \to R$ будем решать уравнение $\nabla f(\vec{x}) = 0$. Итерационное преобразование же будет выглядеть как $\vec{x^i} = \vec{x}^{i-1} - \frac{\nabla f(\vec{x}^{i-1})}{H(\vec{x}^{i-1})} = \vec{x}^{i-1} - H^{-1}(\vec{x}^{i-1})\nabla f(\vec{x}^{i-1})$, где H - гессиан, матрица вторых производных. Вычисление гессиана с дальнейшим инвертированием матрицы является вычислительно дорогой операцией, в таком случае можно использовать квазиньютоновские методы, которые не буду вычислять его напрямую.

2 Вторая часть

Список литературы

[1] Broyden–Fletcher–Goldfarb–Shanno algorithm