1. Метод градиентного спуска

Градиентные методы довольно часто используются для решения задач многомерной безусловной оптимизации. Пусть стоит задача найти минимум функции:

$$f(\theta) \to \min_{\theta}, \tag{1}$$

где θ – **вектор** переменных. Алгоритм метода градиентного спуска состоит в последовательном движении в направлении наискорейшего спуска, то есть в направлении антиградиента $-\nabla_\theta f(\theta)$.

Метод градиентного спуска

Вход: функция $f(\theta)$, начальное приближение θ^0, шаг градиентного спуска η, число итераций n, ϵ.

1. Определить $\nabla_\theta f$.
2. На каждом шаге $t = 1, 2, \ldots, n$:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta f\left(\theta^{(t)}\right).$$

Возможные критерии останова:

a. $|| \theta^{(t+1)} - \theta^{(t)} || \le \epsilon$,

b. $|| f\left(\theta^{(t+1)}\right) - f\left(\theta^{(t)}\right) || \le \epsilon$.

Если взять чересчур большим шаг градиентного спуска, то есть риск «перескочить» минимум. Чтобы этого избежать на каждой итерации алгоритма шаг градиентного спуска можно менять:

1. Обратно пропорционально номеру итерации:

$$\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)} \nabla_\theta f\left(\theta^{(t)}\right)$$

$$\eta^{(t+1)} = \frac{\eta^{(0)}}{t}.$$

2. Недостатком первого способа является то, что с увеличением номера итерации дальнейшие шаги будут чересчур маленькими, и есть риск не дойти до минимума. Чтобы этого избежать, можно использовать экспоненциально затухающий шаг:

$$\eta^{(t+1)} = \eta^{(0)} e^{\frac{1-t}{t}}.$$

Выбирать шаг градиентного спуска можно таким образом, чтобы значение функции $f\left(\theta^{(t+1)} - \eta^{(t+1)} \nabla_\theta f\left(\theta^{(t+1)}\right)\right)$ было наименьшим. Для нахождения оптимального шага $\eta^{(t+1)}$ используются любые методы одномерной оптимизации. Полученный алгоритм имеет название **метод наискорейшего спуска**.

Задание 1

1. Сгенерировать 1000 точек по следующему правилу:

$y = \cos(1.5\,\pi x) + \mathcal{N}(0,\ 1)$.

2. Методом градиентного спуска определить параметры θ в модели полиномиальной регрессии.

3. Отобразить на графике изменения значения функционала качества с номером итерации.

4. Реализовать метод кросс-валидации для оценки качества полученной модели.

5. Определить степень полинома, при которой достигается наилучшее качество модели на кросс-валидации.

6. Полученную функцию отобразить на графике.

7. Дополнительно. Реализовать метод наискорейшего спуска. Для этого необходимо реализовать один из методов одномерной оптимизации, например, метод золотого сечения.

---

2. Метод импульсов

Приведенные выше методы не учитывают характер и форму целевой функции. Метод импульсов помогает ускорить градиентный спуск в нужном направлении.

Согласно методу импульсов (методу моментов) точка обладает массой, соответствующей текущим значениям вектора переменных, а значит в тот момент, когда точка начнет движение в сторону, противоположную направлению градиента функции, у нее появится ненулевая скорость. Если точка пришла в новое положение с некоторой ненулевой скоростью, то ускорение направлено по градиенту и точка не может резко изменить направление движения. Идея метода состоит в том, чтобы на каждом шаге учитывать направление движения на предыдущем шаге. Обозначим направление движения на предыдущем шаге как $v^{(t-1)}$, тогда:

$$v^{(t)} = \gamma\, v^{(t-1)} + \eta\, \nabla_\theta f\left(\theta^{(t)}\right),$$

$$\theta^{(t+1)} = \theta^{(t)} - v^{(t)},$$

где $\gamma < 1$ – параметр, определяющий скорость изменения направления движения.

---

3. Метод Нестерова

Согласно методу Нестерова градиент необходимо считать не в текущей точке, а в той, в которую она придет после шага градиентного спуска. Таким образом алгоритм заглядывает вперед по вектору обновления:

$$v^{(t)} = \gamma\, v^{(t-1)} + \eta\, \nabla_\theta f\left(\theta^{(t)} - \gamma\, v^{(t-1)}\right),$$

$$\theta^{(t+1)} = \theta^{(t)} - v^{(t)}.$$

Значение параметра γ рекомендуют выбирать равным 0.9.

4. Метод стохастического градиентного спуска

Отличие метода стохастического градиентного спуска состоит в том, что на каждом шаге градиентного спуска вектор переменных меняется не по всем наблюдениям выборки, а лишь по одному случайно взятому объекту. Таким образом, для сходимости алгоритма потребуется гораздо большее количество итераций, однако нет необходимости на каждом шаге вычислять сумму отклонений всех наблюдений от истинных значений.

Метод стохастического градиентного спуска

Вход: функция $f(\theta)$, начальное приближение θ^0, шаг градиентного спуска η, число итераций n, ϵ.

1. Определить $\nabla_\theta f$.
2. На каждом шаге $t = 1, 2, \ldots, n$:

   $x_i \in X$:

   $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta f\left(\theta^{(t)}, \{x_i\}\right)$.

Задание 2

1. Реализовать метод стохастического градиентного спуска для решения задания 1.
2. Дополнительно. Реализовать метод покоординатного спуска.

4*. Метод стохастического градиентного спуска по мини-батчам

Главное преимущество метода стохастического градиентного спуска состоит в том, что для обучения предсказательной модели нет необходимости хранить в памяти всю выборку. Однако можно заметить, что для сходимости данного алгоритма требуется большое число итераций, кроме того алгоритм чувствителен к выбросам. Для устранения указанных недостатков применяют метод стохастического градиентного спуска по мини-батчам.

Метод стохастического градиентного спуска по мини-батчам

Вход: функция $f(\theta)$, начальное приближение θ^0, шаг градиентного спуска η, число итераций n, ϵ, размер мини-батча.

1. Определить $\nabla_\theta f$.
2. На каждом шаге $t = 1, 2, \ldots, n$:

   $X^b \in X$:

   $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta f\left(\theta^{(t)}, X^b\right)$.

Домашнее задание

Реализовать метод стохастического градиентного спуска по мини-батчам для решения задания 1.