Вычислительные аспекты оптимизации. Гладкие функционалы и пр. Метод стохастического градиента как метод оптимизаци. Примеры на основе одного из предыдущих методов.
Конспект.

Калина (Бакшинская) Екатерина
Зенкова Наталья
Балагуров Владимир

25 декабря 2019 г.

Содержание

1 Задачи оптимизации в машинном обучении

Пусть есть обучающая выборка

$$X^n = \{(x_i, y_i)\}_{i=1}^n, \quad x_i \in \mathbb{R}^p$$

В задаче регрессии $y_i \in \mathbb{R}$, и ищется параметры ω, такие, что

$$Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i, \omega) - y_i\right)^2 \to \min_{\omega}.$$

В задаче классификации $y_i \in \{-1; 1\}$ и опять же решается задача оптимизации:

$$Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\left[a(x_i, \omega) \cdot y_i > 0\right] \leqslant \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(\langle x_i, \omega\rangle\, y_i\right) \to \min_{\omega},$$

где $\mathcal{L}$ — функция риска от отступа (margin)

2 Общий вид задачи оптимизации. Простой градиентный спуск

В общем виде задача оптимизации в в машинном обучении чаще всего имеет вид:

$$Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(\omega) \to \min_{\omega},$$

где $\mathcal{L}_i$ — функция риска для i-того индивида. Такой вид (сумма по индивидам) встречается наиболее часто, и именно этот вид нам понадобится в будущем.

2.1 Простой алгоритм

Алгоритм с наиболее широкой применимостью является алгоритм градиентного спуска. Данный алгоритм использует свойство антиградиента как "направления наискорейшего убывания целевой функции".

1. Инициализировать начальное значение параметров ω_0
2. Повторять
   (a) Вычислить $\nabla Q(\omega_i)$
   (b) $\omega_{i+1} = \omega_i - \nabla h_i Q(\omega_i)$

   Пока $|\omega_{i+1} - \omega_i| > \varepsilon$

Последовательность h_i используется для затухания длины шагов для того, чтобы алгоритм сходился.

Для этого алгоритма для выпуклых функций гарантируется сходимость при

$$h_i \to 0 \quad \sum_{i=1}^{\infty} h_i = \infty \quad \sum_{i=1}^{\infty} h_i^2 < \infty.$$

Эти условия выполнены,например, для $h_i = \frac{1}{i}$.

Алгоритм, в теории, сойдётся к точке ω^*, такой, что $\nabla Q(\omega^*) = 0$. Условие $\nabla Q(\omega^*) = 0$ не является достаточным для нахождения не только глобального минимума, но и минимума вообще.

2.2 Проблемы и ограничения простого алгоритма

Стоит отдельно упомянуть о недостатках этого алгоритма и способах их решения.

- Задачи оптимизации часто являются задачами условной оптимизации или оптимизации с ограничениями. Решение: посредством теоремы Каруша-Куна-Таккера перейти от задачи условной оптимизации с ограничениями к задаче безусловной оптимизации без ограничений.

- Оптимизационная функция бывает негладкой, негладкой на нужное количество порядков или вообще не непрерывной. Решение: замена оригинальной функции на мажоранту. Такой шаг, вообще говоря, меняет задачу, и нужно дополнительно проверять, что минимум оригинального функционала и минимум мажоранты лежит в одной точке.

- Остановка алгоритма в локальных минимумах. Решение: различные методы "выбивания"из локальных минимумов, будь то одновременное начало из разных точек и последующий выбор наилучшего решения или объединение близких траекторий, или будь то случайные перемещения не в сторону оптимального значения с затухающим радиусом.

- Медленная сходимость для сложных функционалов. Решение: метод скорейшего градиентного спуска с адаптивным шагом. Адаптивный шаг, h^*, находят простейшими методами одномерной оптимизации:
$$h^* = \underset{h}{\operatorname{argmin}}(Q(\omega - h\nabla Q(\omega)))$$

- Вычислительная требовательность по количеству данных: точное вычисление градиента по большой выборке оказывается затратным по времени. Решение: стохастический градиентный спуск

3 Стохастический градиентный спуск

Аддитивный вид функционала в задаче оптимизации,

$$Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(\omega) \to \min_{\omega},$$

позволяет нам упростить трудоёмкие вычисления градиента для многих индивидов следующим образом:

$$\nabla Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\nabla\mathcal{L}_i(\omega).$$

Для более общего вида алгоритма (mini-batch gradient discent) удобнее использовать следующий вид градиента:

$$\nabla Q(\omega) = \frac{1}{n}\sum_{i=1}^{k}\sum_{i\in B_i}\nabla\mathcal{L}_i(\omega),$$

где B_i – "батчи"(подвыборки) обучающей выборки. Так же с

Приведём общий вид такого алгоритма.

3.1 Общий вид mini-batch gradient discent

1. Инициализировать начальное значение параметров ω_0
2. Рассчитать начальное значение $\hat{Q}(\omega_0) = Q(\omega_0)$
3. Для каждой эпохи (всего M эпох) повторять
   (a) Для каждого j – номера батча размера l (обходить батчи случайном порядки) повторять
      i. Вычислить функцию потерь: $\varepsilon_j = \sum\limits_{i\in B_j}\mathcal{L}_i(x_i)$
      ii. Градиентный шаг: $w_{t+1} = w_t + \sum\limits_{i\in B_j}\nabla\mathcal{L}_i(x_i)$
      iii. Оценка нового значения функционала: $\hat{Q}(\omega_{t+1}) = (1-\lambda)Q(\omega_t) + \lambda\varepsilon_j$

Шаг "iii"алгоритма вычисляет оценку на новое значение функцианала в новой точке. Он возможен, опять же, из-за аддитивного вида задачи. Параметр λ ("темп забывания") часто выбирается как $\frac{1}{m}$, где m – номер эпохи.

3.2 Предпосылки и интерпретации

Вычисление аддитивного функционала по всем индивидам

$$Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(\omega) \to \min_{\omega},$$

равно как и вычисление аддинивного градиента по всем индивидам

$$\nabla Q(\omega) = \frac{1}{n}\sum_{i=1}^{n}\nabla\mathcal{L}_i(\omega),$$

может быть проинтерпретировано как нахождение оценки по выборке значения функционала/градиента для случайного индивида. Таким образом появляется не только вычислительная интерпретация стохастического градиента (градиент есть сумма градиентов в аддитивной модели), но и теоретико-вероятностная интерпретация: вместо оценивания мат.ожидания градиента по всей обучающей выборке мы оцениваем его по небольшому количеству индивидов (батчу). Вычислительный аспект алгоритма обязывает нас за одну эпоху пройти всех индивидов по одному разу, однако порядок индивидов не важен, и более того случайный порядок обхода батчей/индивидов повышает скорость сходимости. Сами батчи могут быть произвольного размера в том числе и размеров в одного индивида.

3.3 Модификация для регуляризации

В машинном обучении часто встречаются некорректно поставленные задачи для которых необходимо добавлять к функционалу $Q(\omega)$ различные регуляризационные слагаемые, например:

$$R(\omega) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(\omega) + \frac{\tau}{2}\|\omega\| \to \min_{\omega}$$

Оказывается, что перестроение всего алгоритма, для включения таких добавок не требуется. Тредуется только модификация шага "ii". Для нашего случая с добавкой нормы вектора коэффициентов модифицированный шаг будет выглядеть следующим образом:

$$w_{t+1} = w_t(1 - h\tau) + \sum_{i\in B_j}\nabla\mathcal{L}_i(x_i)$$

4 Преимущества и недостатки стохастического градиентного спуска

Преимущества:

- Легко реализуется
- Функция потери и семейство алгоритмов довольно широко.
- Метод подходит для динамического обучения, когда обучающие объекты поступают потоком и вектор параметров обновляется каждый раз после очередного индивида
- Специально разрабатывался для задач с большими данными, позволет существенно ускорить обучение за счёт перехода к вероятностным оценкам градиента

Недостатки:

- Подбор эвристик (начальные веса, обход индивидов, размеры батчей...) является искусством
- Требуется отдельный расчёт градиента (помимо расчёта функции потерь), или использование методов оценки градиента, что существенно замедляет алгоритм