Методы стохастической оптимизации

Денис Золотухин

НИУ ВШЭ

28 сентября, 2018

Что такое стохастическая оптимизация?

Оптимизация: Минимизация функции потерь

$$L = L(\theta)$$

по параметру $\theta \in \Theta$. Мы считаем, что $\Theta \subseteq \mathbb{R}^p$, соответственно, θ - p-мерный вектор параметров. θ^* - искомый оптимальный параметр.

Что такое стохастическая оптимизация?

Два вида стохастичности:

- Неточное измерение функции потерь (примесь шума).
- Случайность в алгоритме поиска оптимума.

Что такое стохастическая оптимизация?

Два вида стохастичности:

- Неточное измерение функции потерь (примесь шума).
- Случайность в алгоритме поиска оптимума.

В этом докладе будет рассмотрено два метода, каждый про один из этих видов:

- Стохастическая аппроксимация.
- Генетические алгоритмы.

Стохастический градиентный спуск (SGD)

При большой выборке вычисление градиента на каждом шаге спуска - затратная операция. В алгоритме **SGD** на каждом шаге считается градиент не для всей выборки, а для случайного набора (batch) элементов небольшого размера:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}(\hat{\theta}_k)$$

$\hat{g}$ - градиент на batch-е.

Про это будет подробно рассказано во втором докладе.

Шум

На практике бывают ситуации, когда мы знаем не точное значение $L(\theta)$, а лишь его приближение, например если к L примешивается случайный шум:

$$y(\theta) = L(\theta) + \epsilon(\theta)$$

В этом случае мы не можем посчитать никакие производные, то есть градиента у нас нет. Что делать?

Стохастическая аппроксимация

Не будем считать настоящий градиент, а вычислим его приближение $\hat{g}(\hat{\theta}_k)$. Тогда, как и в **SGD**:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}(\hat{\theta}_k),$$

Как приближать?

Конечноразностная аппроксимация

Конечноразностная аппроксимация (FDSA) приближает $g(\hat{\theta})$ следующим образом:

$$\hat{g}_k(\hat{\theta}_k) = \begin{bmatrix} \frac{y(\hat{\theta}_k - c_k\xi_1) - y(\hat{\theta}_k + c_k\xi_1)}{2c_k} \\ \vdots \\ \frac{y(\hat{\theta}_k - c_k\xi_p) - y(\hat{\theta}_k + c_k\xi_p)}{2c_k} \end{bmatrix},$$

Где ξ_i - вектор, у которого на i - м месте стоит 1, а на остальных - нули. c_k - масштаб разности.

Конечноразностная аппроксимация

- При специальных ограничениях регулярности на функцию потерь, алгоритм сходится к θ^* по вероятности. При дополнительных ограничениях на параметры a_k и c_k он сходится почти наверное.

Конечноразностная аппроксимация

- При специальных ограничениях регулярности на функцию потерь, алгоритм сходится к θ^* по вероятности. При дополнительных ограничениях на параметры a_k и c_k он сходится почти наверное.

- Приближённый градиент теперь есть, но "дискретная производная"всё ещё вычисляется для каждого признака. **В итоге приходится считать функцию потерь $2p$ раз.**

Одновременные возмущения

Одновременные возмущения (SPSA) - метод, позволяющий сократить число вычислений $L(\theta)$ до **двух** на каждый шаг алгоритма. Все составляющие вектора $\hat{\theta}_k$ меняются одновременно с помощью вектора возмущений:

$$\Delta_k = \left[\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}\right].$$

Δ_{ki} - i.i.d случайные величины с нулевым средним.

Одновременные возмущения

Теперь $\hat{g}(\hat{\theta})$ вычисляется как

$$\hat{g}_k(\hat{\theta}_k) = \begin{bmatrix} \frac{y(\hat{\theta}_k - c_k\Delta_k) - y(\hat{\theta}_k + c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \vdots \\ \frac{y(\hat{\theta}_k - c_k\Delta_k) - y(\hat{\theta}_k + c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix}$$

$$= \frac{y(\hat{\theta}_k - c_k\Delta_k) - y(\hat{\theta}_k + c_k\Delta_k)}{2c_k} \left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right]^T$$

Одновременные возмущения

- Алгоритм сходится к θ^* п.н . Нужны определённые ограничения на моменты Δ_{ki}. Например, в качестве Δ_{ki} подходит симметричное распределение Бернулии на $\{+1, -1\}$, а $\mathcal{N}(0, \sigma)$ и $\mathcal{U}(-a, a)$ - **нет**.

Одновременные возмущения

- Алгоритм сходится к θ^* п.н . Нужны определённые ограничения на моменты Δ_{ki}. Например, в качестве Δ_{ki} подходит симметричное распределение Бернулии на $\{+1, -1\}$, а $\mathcal{N}(0, \sigma)$ и $\mathcal{U}(-a, a)$ - **нет**.

- Что с числом шагов? Множество экспериментов показывают, что SPSA достигает **такой же точности** что и FDSA при данном числе шагов, не смотря на то, что использует в $\mathbf{1/p}$ раз меньше вычислений функции потерь.

Генетические Алгоритмы

- **Генетический алгоритм (GA)** - метод поиска решения задачи оптимизации, в котором на каждом шаге мы имеем дело с **популяцией** предполагаемых решений.

Генетические Алгоритмы

- **Генетический алгоритм (GA)** - метод поиска решения задачи оптимизации, в котором на каждом шаге мы имеем дело с **популяцией** предполагаемых решений.

- На каждом шаге (в каждом новом поколении), популяция меняется, приближаясь к оптимуму. Элементы популяции (отдельные θ) называются ***хромосомами***.

Кодирование

- Каждая хромосома должна иметь удобное представление. Чаще всего в качестве кода используются обычные бинарные (0, 1) строки, но есть и способы, использующие вещественные числа.

Кодирование

- Каждая хромосома должна иметь удобное представление. Чаще всего в качестве кода используются обычные бинарные (0, 1) строки, но есть и способы, использующие вещественные числа.

- Функция потерь обычно трансформируется в функцию пригодности:

$$F(\theta) = -L(\theta) + C$$

так, чтобы $F(\theta) > 0$ на всём Θ.

Селекция

- Значение F вычисляется для каждой хромосомы. Из текущего поколения выбирается подмножество хромосом, которые выступят в качестве родителей следующего поколения.

Селекция

- Значение F вычисляется для каждой хромосомы. Из текущего поколения выбирается подмножество хромосом, которые выступят в качестве родителей следующего поколения.

- Здесь вступает в силу принцип **"выживают наиболее приспособленные"**: родители выбираются в соответствие с их F-значенями.

Селекция

- Значение F вычисляется для каждой хромосомы. Из текущего поколения выбирается подмножество хромосом, которые выступят в качестве родителей следующего поколения.

- Здесь вступает в силу принцип **"выживают наиболее приспособленные"**: родители выбираются в соответствие с их F-значенями.

- Важно оставить некоторую вероятность выбора неоптимальных хромосом для сохранения разнообразия.

Элитизм

- **Элитизм**:. Небольшая часть лучших хромосом напрямую переносится в следующее поколение без каких-либо изменений.

Элитизм

- **Элитизм**:. Небольшая часть лучших хромосом напрямую переносится в следующее поколение без каких-либо изменений.

- Это необходимо для того, чтобы формально гарантировать сходимость алгоритма.

Рулетка

Есть два самых популярных способа выбрать пары родителей следующего поколения.

Рулетка

Есть два самых популярных способа выбрать пары родителей следующего поколения.

- **Рулетка.** Каждой хромосоме сопоставляется вероятность выпадения на "рулетке"в соответствие с её F-значением. Родители выбираются путём последовательного вращения рулетки столько раз, каков размер поколения.

Турнир

- **Турнир** - другой популярный способ выбора родителей. Две пары хромосом выбираются с возвращением из текущего поколения. В каждой паре выигрывает хромосома с наибольшим F-значением. Победители становятся родителями следующего поколения.

Турнир

- **Турнир** - другой популярный способ выбора родителей. Две пары хромосом выбираются с возвращением из текущего поколения. В каждой паре выигрывает хромосома с наибольшим F-значением. Победители становятся родителями следующего поколения.

- Процесс продолжается до получения нужного количества родителей. Часто турнир показывает себя лучше чем рулетка.

Скрещивание

Скрещивание создаёт двух потомков из пары родителей. С некоторой вероятностью потомки будут представлять смесь хромосом родителей (иначе это будут их копии). В случае бинарного кодирования скрещивание может выглядеть так:

Родители		**Потомки**
1101|101101	$\Rightarrow$	1101|011010
1001|011010		1001|101101

Случайно выбирается место разделения, после которого биты хромосом меняются местами.

Мутации

- Исходная популяция может не быть достаточно вариативной, поэтому после скрещивания к потомкам применяются **мутации**.

Мутации

- Исходная популяция может не быть достаточно вариативной, поэтому после скрещивания к потомкам применяются **мутации**.

- В ходе мутации хромосомы потомков случайно изменяются. В случае бинарного кодирования каждый бит меняется на противоположный с какой-то маленькой вероятностью P_c.

Остановка

Когда останавливать GA?

Остановка

Когда останавливать GA?

- Простого способа нет.

Остановка

Когда останавливать GA?

- Простого способа нет.
- Можно остановиться, когда лучшая хромосома достаточно хороша.

Остановка

Когда останавливать GA?

- Простого способа нет.
- Можно остановиться, когда лучшая хромосома достаточно хороша.
- Маленькая разница между лучшей и худшей.

Ссылки

[1]. Handbook of Computational Statistics, (J. Gentle, W. Härdle, and Y. Mori, eds.), Springer Heidelberg 2004.