Лекция 27
AutoML

Е. А. Соколов
ФКН ВШЭ

1 Введение

Допустим вы пришли работать и вам поставили задачу сделать классификатор обращений пользователей в службу поддержки. Выделим основные этапы решения данной задачи:

1. Сбор данных;
2. Предварительная обработка данных;
3. Выбор целевой метрики;
4. Изготовление признаков;
5. Выбор семейства моделей, подбор гиперпараметров.

Сразу заметим, что выбор целевой метрики мы никак не сможем автоматизировать, потому что он требует понимания сути бизнес-требований. Сбор данных тоже не факт что может быть полностью автоматизирован. Этап предобработки данных состоит из какого-то стандартного набора действий (заполнить пропуски, выкинуть выбросы), которые можно выполнять автоматически. Для изготовления новых признаков можно перебирать комбинации исходных признаков с применением к ним каких-то преобразований (посчитать производную, взять логарифм). Выбор семейства моделей и подбор гиперпараметров тоже поддается автоматизации.

2 Подбор гиперпараметров

Пусть есть набор гиперпараметров $\theta \in \mathbb{R}^n$, определяющий модель, подготовку данных, и так далее. Пусть также есть функция $f(\theta)$, которая говорит, какая ошибка получится, если взять модель и обучить её с гиперпараметрами θ. Тогда мы ищем решение задачи

$$f(\theta) \to \min_{\theta}.$$

Рассмотрим различные способы её решения.

Grid Search: перебрать по сетке тысячу гиперпараметров не представляется возможным в принципе — очень долго.

Random Search: может быть быстрее Grid Search и легче управлять вычислительными ресурсами.

Градиентный спуск: не подходит, потому что

- $f(\theta)$ очень долго считать: нужно пройти весь пайплайн, обучить модель, и ещё качество на отложенной выборке посчитать;
- $f(\theta)$ не дифференцируема, а некоторые из гиперпараметров так вообще могут быть категориальными.

Дискретная оптимизация: алгоритм имитации отжига (нормально работает при $n \approx 100$), генетические алгоритмы и любой другой алгоритм дискретной оптимизации. Проблема одна – они не работают, поскольку у нас много параметров, по которым мы оптимизируем.

§2.1 Оптимизация через суррогатные функции

Предположим, что мы знаем $f(\theta_1), \ldots, f(\theta_k)$. Построим функцию $g\colon \Theta \to \mathbb{R}$ такую, что

- Для любого $i \in \{1, \ldots, k\}$ $g(\theta_i) \approx f(\theta_i)$. Будем надеяться, что тогда $g(\theta) \approx f(\theta)$ для любого $\theta \in \Theta$;
- Функция g является хорошей, то есть непрерывной, сколько нужно раз дифференцируемой, и так далее.

Функцию g будем называть *суррогатной функцией*. Тогда

$$\theta_{k+1} \coloneqq \arg\min_{\theta \in \Theta} g(\theta).$$

После этого мы добавляем $f(\theta_{k+1})$ в исходную выборку, строим новую суррогатную функцию, и так далее.

§2.2 Гауссовские процессы

Опр. 2.1 (GPGO). *Будем говорить, что на $\mathbb{R}^n$ задан гауссовский процесс, если каждому $x \in \mathbb{R}^n$ сопоставлена случайная величина $\xi(x) \sim \mathcal{N}(\mu(x), \sigma^2)$ такая, что*

$$\operatorname{cov}(\xi(x_1), \xi(x_2)) = K(x_1, x_2).$$

На самом деле это не вполне корректное определение, и мы хотим потребовать

$$(\xi(x_1), \ldots, \xi(x_n)) \sim \mathcal{N}\left(\begin{bmatrix}\mu(x_1)\\ \vdots \\ \mu(x_n)\end{bmatrix}, \Sigma\right),$$

где Σ вычисляется через K.

Интуиция Пусть у нас есть пространство $\mathbb{R}^2$, в нем в каждой точке задана случайная величина. Допустим, мы знаем, что $\xi(x_1) = 10$. Возьмем какую-то точку x_2 рядом с x_1, для которой мы еще не знаем $\xi(x_2)$. Но, поскольку мы знаем, что это все гауссовский процесс, то мы гарантируем, что точка x_2 будет сильно скореллирована с x_1, поэтому мы можем с помощью ядра K оценить значение $\xi(x_2)$. Чем дальше от x_1 мы возьмем x_2, тем больше будет дисперсия наших представлений об этой точке, и тем меньше мы будем знать информации о ней.

Пример ядра K В выкладках выше можно взять K равным ядру Матерна, то есть

$$K_{\text{Matern}}(x_1, x_2) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} r(x_1, x_2)\right)^{\nu} K_\nu \left(\sqrt{2\nu} r(x_1, x_2)\right),$$

где σ_f^2 – гиперпараметр, $r(x_1, x_2) = \sqrt{(x_1 - x_2)^\top \Lambda (x_1 - x_2)}$ – расстояние Махаланобиса с гиперпараметром Λ, $K_\nu(\cdot)$ – функция Бесселя 2-го рода, ν – гиперпараметр.

Это все нам нужно, чтобы мы могли записать распределение в какой-то точке при условии значений в других точках нашего пространства. Допустим у нас есть выборка гиперпараметров $\theta_1, \ldots, \theta_k$, которые мы уже попробовали. Также у нас есть набор значений $f(\theta_1), \ldots, f(\theta_k)$ функции потерь для этого набора гиперпараметров. Тогда мы можем посчитать $p(f(\theta) \mid \theta_1, \ldots \theta_k, f(\theta_1), \ldots, f(\theta_k))$ и таким образом оценить неопределенность на наших значениях гиперпараметров.

Здесь нужно ответить на важный вопрос: какую следующую точку брать? Нам нужно выбрать функцию, которая будет отвечать на вопрос, какую следующую точку брать. Например,

$$\lambda(f(\theta)) = \min(f(\theta), \eta),$$

где η – лучшее значение f, найденное на данный момент. То есть мы хотим как можно сильнее уменьшить значение ошибки относительно наилучшего значения ошибки, найденного на этот момент. Далее будем решать следующую задачу оптимизации:

$$\mathbb{E}\lambda(f(\theta)) \to \min_\theta .$$

Примечание Утверждается, что гауссовские процессы работают хорошо, если k не очень большое. Чем больше у нас k, тем сложнее посчитать $p(f(\theta) \mid \theta_1, \ldots, \theta_k, f(\theta_1), \ldots, f(\theta_k))$, поскольку нам нужно обращать большую матрицу ковариаций, и тем вычислительно сложнее становится оптимизация.

Ещё одно примечание Если мы хотим попробовать больше разных гиперпараметров, то можно увеличить η. По умолчанию η берут как лучшее значение f.

§2.3 Tree-structured Parzen Estimator

Допустим мы уже попробовали какое-то количество гиперпараметров $\theta_1, \ldots, \theta_k$ и посчитали $y_1 = f(\theta_1), \ldots, y_k = f(\theta_k)$. Построим три распределения:

$$p(y < y_\star), \quad p(\theta \mid y < y_\star), \quad p(\theta \mid y \geqslant y_\star),$$

где:

- Первое распределение задаёт априорную вероятность того, что мы сможем найти такой набор гиперпараметров, который сильнее улучшит значение f (например, 0.15);
- Второе распределение задаёт распределение на гиперпараметрах при условии, что мы смогли улучшить ошибку;
- Третье распределение задаёт распределение на гиперпараметрах при условии, что мы не смогли улучшить ошибку.

Для приближения второго и третьего распределений мы просто берем набор гиперпараметров θ, который удовлетворяет условию, и строим непараметрическую оценку плотности.

По формуле Байеса можно найти $p(y \mid \theta)$, подставить в

$$\mathbb{E}\lambda(f(\theta)) \to \min_{\theta}$$

и найти нужный параметр θ_{k+1}.

Примечание Метод называется «Tree-structured», потому что обычно используется для случаев, когда одни гиперпараметры зависят от других. Например, при подборе количества нейронов в слое нейросети, искомое число может зависеть от числа нейронов в предыдущем слое.