Лекция 8
Бэггинг, случайные леса и разложение ошибки на смещение и разброс

Е. А. Соколов
ФКН ВШЭ

6 ноября 2020 г.

При обсуждении решающих деревьев мы упомянули, что они могут восстанавливать очень сложные закономерности, но при этом неустойчивы к малейшим изменениям в данных. Из-за этого сами по себе деревья не очень хороши, но при этом, как оказывается, при объединении в *композицию* они показывают очень хорошие результаты. Одним из подходов к построению композиций является *бэггинг*, который независимо строит несколько моделей и усредняет их ответы. На данной лекции мы изучим инструмент, который поможет нам в анализе бэггинга — декомпозицию ошибки на компоненты смещения и разброса (bias-variance decomposition) — а затем перейдем к самим методам. Также существует другой подход к построению композиций, называемый *бустингом*, который строит модели последовательно, и каждая следующая модель исправляет ошибки предыдущей. О таких методах речь пойдёт уже на следующих лекциях.

1 Бутстрап

Рассмотрим простой пример построения композиции алгоритмов. Пусть дана конечная выборка $X = (x_i, y_i)$ с вещественными ответами. Будем решать задачу линейной регрессии. Сгенерируем подвыборку с помощью *бутстрапа*. Равномерно возьмем из выборки ℓ объектов с возвращением. Отметим, что из-за возвращения среди них окажутся повторы. Обозначим новую выборку через X_1. Повторив процедуру N раз, сгенерируем N подвыборок $X_1, \ldots, X_N$. Обучим по каждой из них линейную модель регрессии, получив *базовые алгоритмы* $b_1(x), \ldots, b_N(x)$.

Предположим, что существует истинная функция ответа для всех объектов $y(x)$, а также задано распределение на объектах $p(x)$. В этом случае мы можем записать ошибку каждой функции регрессии

$$\varepsilon_j(x) = b_j(x) - y(x), \qquad j = 1, \ldots, N,$$

и записать матожидание среднеквадратичной ошибки

$$\mathbb{E}_x(b_j(x) - y(x))^2 = \mathbb{E}_x\varepsilon_j^2(x).$$

Средняя ошибка построенных функций регрессии имеет вид

$$E_1 = \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_x\varepsilon_j^2(x).$$

Предположим, что ошибки несмещены и некоррелированы:

$$\mathbb{E}_x\varepsilon_j(x) = 0;$$
$$\mathbb{E}_x\varepsilon_i(x)\varepsilon_j(x) = 0, \quad i \neq j.$$

Построим теперь новую функцию регрессии, которая будет усреднять ответы построенных нами функций:

$$a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x).$$

Найдем ее среднеквадратичную ошибку:

$$\begin{aligned}
E_N &= \mathbb{E}_x\left(\frac{1}{N}\sum_{j=1}^{n} b_j(x) - y(x)\right)^2 = \\
&= \mathbb{E}_x\left(\frac{1}{N}\sum_{j=1}^{N}\varepsilon_j(x)\right)^2 = \\
&= \frac{1}{N^2}\mathbb{E}_x\left(\sum_{j=1}^{N}\varepsilon_j^2(x) + \underbrace{\sum_{i\neq j}\varepsilon_i(x)\varepsilon_j(x)}_{=0}\right) = \\
&= \frac{1}{N}E_1.
\end{aligned}$$

Таким образом, усреднение ответов позволило уменьшить средний квадрат ошибки в N раз!

Следует отметить, что рассмотренный нами пример не очень применим на практике, поскольку мы сделали предположение о некоррелированности ошибок, что редко выполняется. Если это предположение неверно, то уменьшение ошибки оказывается не таким значительным. Позже мы рассмотрим более сложные методы объединения алгоритмов в композицию, которые позволяют добиться высокого качества в реальных задачах.

2 Bias-Variance decomposition

Допустим, у нас есть некоторая выборка, на которой линейные методы работают лучше решающих деревьев с точки зрения ошибки на контроле. Почему это так? Чем можно объяснить превосходство определенного метода обучения? Оказывается, ошибка любой модели складывается из трех факторов: сложности самой выборки, сходства модели с истинной зависимостью ответов от объектов в выборке, и богатства

семейства, из которого выбирается конкретная модель. Между этими факторами существует некоторый баланс, и уменьшение одного из них приводит к увеличению другого. Такое разложение ошибки носит название разложения на смещение и разброс, и его формальным выводом мы сейчас займемся.

Пусть задана выборка $X = (x_i, y_i)_{i=1}^{\ell}$ с вещественными ответами $y_i \in \mathbb{R}$ (рассматриваем задачу регрессии). Будем считать, что на пространстве всех объектов и ответов $\mathbb{X} \times \mathbb{Y}$ существует распределение $p(x, y)$, из которого сгенерирована выборка X и ответы на ней.

Рассмотрим квадратичную функцию потерь

$$L(y, a) = \big(y - a(x)\big)^2$$

и соответствующий ей *среднеквадратичный риск*

$$R(a) = \mathbb{E}_{x,y}\Big[\big(y - a(x)\big)^2\Big] = \int_{\mathbb{X}}\int_{\mathbb{Y}} p(x, y)\big(y - a(x)\big)^2 dxdy.$$

Данный функционал усредняет ошибку модели в каждой точке пространства x и для каждого возможного ответа y, причём вклад пары (x, y), по сути, пропорционален вероятности получить её в выборке $p(x, y)$. Разумеется, на практике мы не можем вычислить данный функционал, поскольку распределение $p(x, y)$ неизвестно. Тем не менее, в теории он позволяет измерить качество модели на всех возможных объектах, а не только на обучающей выборке.

§2.1 Минимум среднеквадратичного риска

Покажем, что минимум среднеквадратичного риска достигается на функции, возвращающей условное матожидание ответа при фиксированном объекте:

$$a_*(x) = \mathbb{E}[y \,|\, x] = \int_{\mathbb{Y}} yp(y \,|\, x)dy = \arg\min_a R(a).$$

Преобразуем функцию потерь:

$$\begin{aligned} L(y, a(x)) &= (y - a(x))^2 = (y - \mathbb{E}(y \,|\, x) + \mathbb{E}(y \,|\, x) - a(x))^2 = \\ &= (y - \mathbb{E}(y \,|\, x))^2 + 2\big(y - \mathbb{E}(y \,|\, x)\big)\big(\mathbb{E}(y \,|\, x) - a(x)\big) + (\mathbb{E}(y \,|\, x) - a(x))^2. \end{aligned}$$

Подставляя ее в функционал среднеквадратичного риска, получаем:

$$\begin{aligned} R(a) &= \mathbb{E}_{x,y} L(y, a(x)) = \\ &= \mathbb{E}_{x,y}(y - \mathbb{E}(y \,|\, x))^2 + \mathbb{E}_{x,y}(\mathbb{E}(y \,|\, x) - a(x))^2 + \\ &+ 2\mathbb{E}_{x,y}\big(y - \mathbb{E}(y \,|\, x)\big)\big(\mathbb{E}(y \,|\, x) - a(x)\big). \end{aligned}$$

Разберемся сначала с последним слагаемым. Перейдём от матожидания $\mathbb{E}_{x,y}[f(x, y)]$ к цепочке матожиданий

$$\mathbb{E}_x\mathbb{E}_y[f(x, y) \,|\, x] = \int_{\mathbb{X}} \left(\int_{\mathbb{Y}} f(x, y)p(y \,|\, x)dy\right) p(x)dx$$

и заметим, что величина $\big(\mathbb{E}(y\,|\,x)-a(x)\big)$ не зависит от y, и поэтому ее можно вынести за матожидание по y:

$$\begin{aligned}
&\mathbb{E}_x\mathbb{E}_y\Big[\big(y-\mathbb{E}(y\,|\,x)\big)\big(\mathbb{E}(y\,|\,x)-a(x)\big)\,|\,x\Big] = \\
&= \mathbb{E}_x\Big(\big(\mathbb{E}(y\,|\,x)-a(x)\big)\mathbb{E}_y\Big[\big(y-\mathbb{E}(y\,|\,x)\big)\,|\,x\Big]\Big) = \\
&= \mathbb{E}_x\Big(\big(\mathbb{E}(y\,|\,x)-a(x)\big)\big(\mathbb{E}(y\,|\,x)-\mathbb{E}(y\,|\,x)\big)\Big) = \\
&= 0
\end{aligned}$$

Получаем, что функционал среднеквадратичного риска имеет вид

$$R(a) = \mathbb{E}_{x,y}(y-\mathbb{E}(y\,|\,x))^2 + \mathbb{E}_{x,y}(\mathbb{E}(y\,|\,x)-a(x))^2.$$

От алгоритма $a(x)$ зависит только второе слагаемое, и оно достигает своего минимума, если $a(x)=\mathbb{E}(y\,|\,x)$. Таким образом, оптимальная модель регрессии для квадратичной функции потерь имеет вид

$$a_*(x) = \mathbb{E}(y\,|\,x) = \int_{\mathbb{Y}} yp(y\,|\,x)dy.$$

Иными словами, мы должны провести «взвешенное голосование» по всем возможным ответам, причем вес ответа равен его апостериорной вероятности.

§2.2 Ошибка метода обучения

Для того, чтобы построить идеальную функцию регрессии, необходимо знать распределение на объектах и ответах $p(x,y)$, что, как правило, невозможно. На практике вместо этого выбирается некоторый *метод обучения* $\mu : (\mathbb{X}\times\mathbb{Y})^\ell \to \mathcal{A}$, который произвольной обучающей выборке ставит в соответствие некоторый алгоритм из семейства $\mathcal{A}$. В качестве меры качества метода обучения можно взять усредненный по всем выборкам среднеквадратичный риск алгоритма, выбранного методом μ по выборке:

$$\begin{aligned}
L(\mu) &= \mathbb{E}_X\Big[\mathbb{E}_{x,y}\Big[\big(y-\mu(X)(x)\big)^2\Big]\Big] = \qquad (2.1)\\
&= \int_{(\mathbb{X}\times\mathbb{Y})^\ell}\int_{\mathbb{X}\times\mathbb{Y}}\big(y-\mu(X)(x)\big)^2 p(x,y)\prod_{i=1}^{\ell}p(x_i,y_i)dxdydx_1dy_1\ldots dx_\ell dy_\ell.
\end{aligned}$$

Здесь матожидание $\mathbb{E}_X[\cdot]$ берется по всем возможным выборкам $\{(x_1,y_1),\ldots,(x_\ell,y_\ell)\}$ из распределения $\prod_{i=1}^{\ell}p(x_i,y_i)$.

Обратим внимание, что результатом применения метода обучения $\mu(X)$ к выборке X является модель, поэтому правильно писать $\mu(X)(x)$. Но это довольно громоздкая запись, поэтому будем везде дальше писать просто $\mu(X)$, но не будем забывать, что это функция, зависящая от объекта x.

Выше мы показали, что среднеквадратичный риск на фиксированной выборке X можно расписать как

$$\mathbb{E}_{x,y}\Big[\big(y-\mu(X)\big)^2\Big] = \mathbb{E}_{x,y}\Big[\big(y-\mathbb{E}[y\,|\,x]\big)^2\Big] + \mathbb{E}_{x,y}\Big[\big(\mathbb{E}[y\,|\,x]-\mu(X)\big)^2\Big].$$

Подставим это представление в (2.1):

$$L(\mu) = \mathbb{E}_X\Big[\underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y\,|\,x]\big)^2\Big]}_{\text{не зависит от } X} + \mathbb{E}_{x,y}\Big[\big(\mathbb{E}[y\,|\,x] - \mu(X)\big)^2\Big]\Big] =$$
$$= \mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y\,|\,x]\big)^2\Big] + \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\big(\mathbb{E}[y\,|\,x] - \mu(X)\big)^2\Big]\Big]. \tag{2.2}$$

Преобразуем второе слагаемое:

$$\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\big(\mathbb{E}[y\,|\,x] - \mu(X)\big)^2\Big]\Big] =$$
$$= \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big] + \mathbb{E}_X\big[\mu(X)\big] - \mu(X)\big)^2\Big]\Big] =$$
$$= \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\underbrace{\big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big]\big)^2}_{\text{не зависит от } X}\Big]\Big] + \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\big(\mathbb{E}_X\big[\mu(X)\big] - \mu(X)\big)^2\Big]\Big] +$$
$$+ 2\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big]\big)\big(\mathbb{E}_X\big[\mu(X)\big] - \mu(X)\big)\Big]\Big]. \tag{2.3}$$

Покажем, что последнее слагаемое обращается в нуль:

$$\mathbb{E}_X\Big[\big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big]\big)\big(\mathbb{E}_X\big[\mu(X)\big] - \mu(X)\big)\Big] =$$
$$= \big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big]\big)\mathbb{E}_X\Big[\mathbb{E}_X\big[\mu(X)\big] - \mu(X)\Big] =$$
$$= \big(\mathbb{E}[y\,|\,x] - \mathbb{E}_X\big[\mu(X)\big]\big)\Big[\mathbb{E}_X\big[\mu(X)\big] - \mathbb{E}_X\big[\mu(X)\big]\Big] =$$
$$= 0.$$

Учитывая это, подставим (2.3) в (2.2):

$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y\,|\,x]\big)^2\Big]}_{\text{шум}} +$$
$$+ \underbrace{\mathbb{E}_x\Big[\big(\mathbb{E}_X\big[\mu(X)\big] - \mathbb{E}[y\,|\,x]\big)^2\Big]}_{\text{смещение}} + \underbrace{\mathbb{E}_x\Big[\mathbb{E}_X\Big[\big(\mu(X) - \mathbb{E}_X\big[\mu(X)\big]\big)^2\Big]\Big]}_{\text{разброс}}. \tag{2.4}$$

Рассмотрим подробнее компоненты полученного разложения ошибки. Первая компонента характеризует *шум* в данных и равна ошибке идеального алгоритма. Невозможно построить алгоритм, имеющий меньшую среднеквадратичную ошибку. Вторая компонента характеризует *смещение (bias)* метода обучения, то есть отклонение среднего ответа обученного алгоритма от ответа идеального алгоритма. Третья компонента характеризует *дисперсию (variance)*, то есть разброс ответов обученных алгоритмов относительно среднего ответа.

Смещение показывает, насколько хорошо с помощью данных метода обучения и семейства алгоритмов можно приблизить оптимальный алгоритм. Как правило, смещение маленькое у сложных семейств (например, у деревьев) и большое у простых семейств (например, линейных классификаторов). Дисперсия показывает, насколько сильно может изменяться ответ обученного алгоритма в зависимости от выборки — иными словами, она характеризует чувствительность метода обучения к изменениям

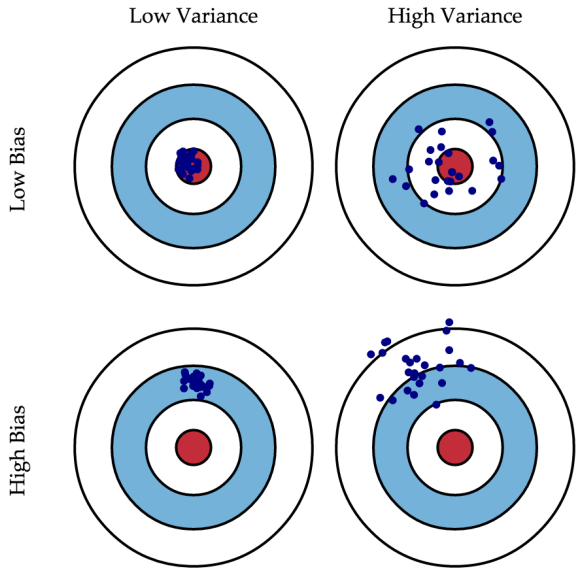

Рис. 1. Иллюстрация сдвига и разброса для различных моделей.

в выборке. Как правило, простые семейства имеют маленькую дисперсию, а сложные семейства — большую дисперсию.

На рис. 1 изображены модели с различными сдвигом и разбросом. Модели изображены синими точками, одна точка соответствует модели, обученной по одной из возможных обучающих выборок. Каждый круг характеризует качество модели — чем ближе точка к центру, тем меньше ошибок на контрольной выборке достигает данный алгоритм. Видно, что большой сдвиг соответствует тому, что в среднем точки не попадают в центр, то есть в среднем они не соответствуют лучшей модели. Большой разброс означает, что модель может попасть по качеству куда угодно — как в центр, так и в область с большой ошибкой.

Разложение для произвольной функции потерь. Разложение ошибки на три компоненты, которое мы только что вывели, верно только для квадратичной функции потерь. Существуют более общие формы этого разложения [2], которые состоят из трёх компонент с аналогичным смыслом, поэтому можно утверждать, что для большинства распространённых функций потерь ошибка метода обучения складывается из шума, смещения и разброса; значит, и дальнейшие рассуждения про изменение этих компонент в композициях также можно обобщить на другие функции потерь (например, на индикатор ошибки классификации).

3 Бэггинг

Пусть имеется некоторый метод обучения $\mu(X)$. Построим на его основе метод $\tilde{\mu}(X)$, который генерирует случайную подвыборку $\tilde{X}$ с помощью бутстрапа и подает ее на вход метода μ: $\tilde{\mu}(X) = \mu(\tilde{X})$. Напомним, что бутстрап представляет собой сэмплирование ℓ объектов из выборки с возвращением, в результате чего некоторые объекты выбираются несколько раз, а некоторые — ни разу. Помещение

нескольких копий одного объекта в бутстрапированную выборку соответствует выставлению веса при данном объекте — соответствующее ему слагаемое несколько раз войдет в функционал, и поэтому штраф за ошибку на нем будет больше.

В *бэггинге (bagging, bootstrap aggregation)* предлагается обучить некоторое число алгоритмов $b_n(x)$ с помощью метода $\tilde{\mu}$, и построить итоговую композицию как среднее данных базовых алгоритмов:

$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x) = \frac{1}{N}\sum_{n=1}^{N} \tilde{\mu}(X)(x).$$

Заметим, что в методе обучения для бэггинга появляется ещё один источник случайности — взятие подвыборки. Чтобы функционал качества $L(\mu)$ был детерминированным, мы будем далее считать, что матожидание $\mathbb{E}_X[\cdot]$ берётся не только по всем обучающим выборкам X, но ещё и по всем возможным подвыборкам $\tilde{X}$, получаемым с помощью бутстрапа. Это вполне логичное обобщение, поскольку данное матожидание вводится в функционал именно для учёта случайностей, связанных с процедурой обучения модели.

Найдём смещение из разложения (2.4) для бэггинга:

$$\begin{aligned}
&\mathbb{E}_{x,y}\Big[\Big(\mathbb{E}_X\Big[\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X)(x)\Big] - \mathbb{E}[y\,|\,x]\Big)^2\Big] = \\
&\quad = \mathbb{E}_{x,y}\Big[\Big(\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_X[\tilde{\mu}(X)(x)] - \mathbb{E}[y\,|\,x]\Big)^2\Big] = \\
&\quad = \mathbb{E}_{x,y}\Big[\big(\mathbb{E}_X\big[\tilde{\mu}(X)(x)\big] - \mathbb{E}[y\,|\,x]\big)^2\Big].
\end{aligned}$$

Мы получили, что смещение композиции, полученной с помощью бэггинга, совпадает со смещением одного базового алгоритма. Таким образом, бэггинг не ухудшает смещенность модели.

Теперь перейдём к разбросу. Запишем выражение для дисперсии композиции, обученной с помощью бэггинга:

$$\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X)(x) - \mathbb{E}_X\Big[\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X)(x)\Big]\Big)^2\Big]\Big].$$

Рассмотрим выражение, стоящее под матожиданиями:

$$\begin{aligned}
&\Big(\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X)(x) - \mathbb{E}_X\Big[\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X)(x)\Big]\Big)^2 = \\
&\quad = \frac{1}{N^2}\Big(\sum_{n=1}^{N}\Big[\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big]\Big)^2 = \\
&\quad = \frac{1}{N^2}\sum_{n=1}^{N}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)^2 + \\
&\qquad + \frac{1}{N^2}\sum_{n_1\neq n_2}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)
\end{aligned}$$

Алгоритм 3.1. Random Forest

1: **для** $n = 1, \ldots, N$
2: Сгенерировать выборку $\tilde{X}_n$ с помощью бутстрэпа
3: Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}_n$:

- дерево строится, пока в каждом листе не окажется не более $n_{\min}$ объектов
- при каждом разбиении сначала выбирается m случайных признаков из p, и оптимальное разделение ищется только среди них

4: Вернуть композицию $a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$

Возьмем теперь матожидания от этого выражения, учитывая, что все базовые алгоритмы одинаково распределены относительно X:

$$\begin{aligned}
&\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\frac{1}{N^2}\sum_{n=1}^{N}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)^2 + \\
&\quad + \frac{1}{N^2}\sum_{n_1\neq n_2}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\Big]\Big] = \\
&= \frac{1}{N^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\sum_{n=1}^{N}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)^2\Big]\Big] + \\
&\quad + \frac{1}{N^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\sum_{n_1\neq n_2}\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\times \\
&\qquad\qquad \times\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\Big]\Big] = \\
&= \frac{1}{N}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)^2\Big]\Big] + \\
&\quad + \frac{N(N-1)}{N^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\times \\
&\qquad\qquad \times\Big(\tilde{\mu}(X)(x) - \mathbb{E}_X\big[\tilde{\mu}(X)(x)\big]\Big)\Big]\Big]
\end{aligned}$$

Первое слагаемое — это дисперсия одного базового алгоритма, деленная на длину композиции N. Второе — ковариация между двумя базовыми алгоритмами. Мы видим, что если базовые алгоритмы некоррелированы, то дисперсия композиции в N раз меньше дисперсии отдельных алгоритмов. Если же корреляция имеет место, то уменьшение дисперсии может быть гораздо менее существенным.

§3.1 Случайные леса

Как мы выяснили, бэггинг позволяет объединить несмещенные, но чувствительные к обучающей выборке алгоритмы в несмещенную композицию с низкой дисперсией. Хорошим семейством базовых алгоритмов здесь являются решающие деревья — они достаточно сложны и могут достигать нулевой ошибки на любой выборке (следовательно, имеют низкое смещение), но в то же время легко переобучаются.

Метод *случайных лесов* [3] основан на бэггинге над решающими деревьями, см. алгоритм 3.1. Выше мы отметили, что бэггинг сильнее уменьшает дисперсию базовых алгоритмов, если они слабо коррелированы. В случайных лесах корреляция между деревьями понижается путем рандомизации по двум направлениям: по объектам и по признакам. Во-первых, каждое дерево обучается по бутстрапированной подвыборке. Во-вторых, в каждой вершине разбиение ищется по подмножеству признаков. Вспомним, что при построении дерева последовательно происходит разделение вершин до тех пор, пока не будет достигнуто идеальное качество на обучении. Каждая вершина разбивает выборку по одному из признаков относительно некоторого порога. В случайных лесах признак, по которому производится разбиение, выбирается не из всех возможных признаков, а лишь из их случайного подмножества размера m.

Рекомендуется в задачах классификации брать $m = \lfloor \sqrt{d} \rfloor$, а в задачах регрессии — $m = \lfloor d/3 \rfloor$, где d — число признаков. Также рекомендуется в задачах классификации строить каждое дерево до тех пор, пока в каждом листе не окажется по одному объекту, а в задачах регрессии — пока в каждом листе не окажется по пять объектов.

Случайные леса — один из самых сильных методов построения композиций. На практике он может работать немного хуже градиентного бустинга, но при этом он гораздо более прост в реализации.

3.1.1 Out-of-Bag

Каждое дерево в случайном лесе обучается по подмножеству объектов. Это значит, что те объекты, которые не вошли в бутстрапированную выборку X_n дерева b_n, по сути являются контрольными для данного дерева. Значит, мы можем для каждого объекта x_i найти деревья, которые были обучены без него, и вычислить по их ответам out-of-bag-ошибку:

$$\text{OOB} = \sum_{i=1}^{\ell} L\left(y_i, \frac{1}{\sum_{n=1}^{N}[x_i \notin X_n]} \sum_{n=1}^{N} [x_i \notin X_n] b_n(x_i)\right),$$

где $L(y, z)$ — функция потерь. Можно показать, что по мере увеличения числа деревьев N данная оценка стремится к leave-one-out-оценке, но при этом существенно проще для вычисления.

§3.2 Связь с метрическими методами

Случайные леса, по сути, осуществляют предсказание для объекта на основе меток похожих объектов из обучения. Схожесть объектов при этом тем выше, чем чаще эти объекты оказываются в одном и том же листе дерева. Покажем это формально.

Рассмотрим задачу регрессии с квадратичной функцией потерь. Пусть $T_n(x)$ — номер листа n-го дерева из случайного леса, в который попадает объект x. Ответ дерева на объекте x равен среднему ответу по всем обучающим объектам, которые попали в лист $T_n(x)$. Это можно записать как

$$b_n(x) = \sum_{i=1}^{\ell} w_n(x, x_i) y_i,$$

где

$$w_n(x, x_i) = \frac{[T_n(x) = T_n(x_i)]}{\sum_{j=1}^{\ell}[T_n(x) = T_n(x_j)]}.$$

Тогда ответ композиции равен

$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{\ell} w_n(x, x_i)y_i = \sum_{i=1}^{\ell}\left(\frac{1}{N}\sum_{n=1}^{N} w_n(x, x_i)\right) y_i.$$

Видно, что ответ случайного леса представляет собой сумму ответов всех объектов обучения с некоторыми весами, причём данные веса измеряют сходство объектов x и x_i на основе того, сколько раз они оказались в одном и том же листе. Таким образом, случайный лес позволяет ввести некоторую функцию расстояния на объектах. Как мы узнаем позже, на этом принципе основан целый класс *метрических* методов, наиболее популярным представителем которых является метод k ближайших соседей.

Отметим, что номер листа $T_n(x)$, в который попал объект, сам по себе является ценным признаком. Достаточно неплохо работает подход, в котором по выборке обучается композиция из небольшого числа деревьев с помощью случайного леса или градиентного бустинга, а затем к ней добавляются категориальные признаки $T_1(x), T_2(x), \ldots, T_N(x)$. Новые признаки являются результатом нелинейного разбиения пространства и несут в себе информацию о сходстве объектов.

Список литературы

[1] *Hastie, T., Tibshirani, R., Friedman, J.* (2001). The Elements of Statistical Learning. // Springer, New York.

[2] *Domingos, Pedro* (2000). A Unified Bias-Variance Decomposition and its Applications. // In Proc. 17th International Conf. on Machine Learning.

[3] *Breiman, Leo* (2001). Random Forests. // Machine Learning, 45(1), 5–32.