Машинное обучение, ФКН ВШЭ
Семинар №8

1 Разложение на смещение и разброс

На лекции была выведено следующая формула, показывающая, как можно представить ошибку алгоритма регрессии в виде суммы трех компонент:

$$L(\mu) = \mathbb{E}_{x,y}\big[\mathbb{E}_X\big[(y - \mu(X)(x))^2\big]\big] =$$

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

- $\mu(X)$ — алгоритм, обученный по выборке $X = \{(x_1, y_1), \dots (x_\ell, y_\ell)\}$;
- $\mu(X)(x)$ — ответ алгоритма, обученного по выборке X, на объекте x;
- $\mathbb{E}_X$ — мат. ожидание по всем возможным выборкам;
- $\mathbb{E}_X[\mu(X)(x)]$ — «средний» ответ алгоритма, обученного по всем возможным выборкам X, на объекте x.

С помощью этой формулы мы можем анализировать свойства алгоритма обучения модели μ, если зададим вероятностную модель порождения пар $p(x, y)$.

§1.1 Разложение для линейной модели

Чтобы лучше понять смысл трех компонент, входящих в разложение, рассмотрим модельный пример одномерной линейной регрессии.

Алгоритм обучения μ. В одномерной линейной регрессии зависимость целевого признака y от объекта x моделируется с помощью примитивной линейной функции $y = kx$. Оптимальный параметр k находится по выборке $X = \{(x_1, y_1), \dots (x_\ell, y_\ell)\}$ минимизацией $\text{MSE} = \sum_{i=1}^{\ell}(y_i - kx_i)^2$. В результате получается следующий алгоритм $\mu(X)$ (см. семинар по линейной регрессии):

$$\mu(X)(x) = k(X)\,x, \quad k(X) = \frac{\sum_i x_i y_i}{\sum_i x_i^2}.$$

Модель порождения данных. Будем считать, что объекты x генерируются из нормального распределения $x \sim p(x) = \mathcal{N}(0, \sigma_1^2)$. Правильный ответ на объекте x определяется зашумленной функцией $f(x)$: $y = f(x) + \varepsilon$, $\varepsilon \sim p(\varepsilon) = \mathcal{N}(0, \sigma_2^2)$. Иными словами, $y \sim p(y|x) = \mathcal{N}(f(x), \sigma_2^2)$.

Выборка X составляется из ℓ независимых пар (x_i, y_i).

Мы будем рассматривать два простых частных случая: $f(x) = ax$, когда модель зависимости отвечает искомой зависимости, и $f(x)$ — четная функция, т. е. $f(-x) = f(x)$.

Задача 1.1. *Найдите шумовую компоненту для одномерной линейной регрессии.*

Решение. Так как распределение $p(y|x)$ нормальное, для него легко вычислить мат. ожидание:

$$\mathbb{E}[y|x] = f(x).$$

Тогда

$$\mathbb{E}_{x,y}\big[(y - \mathbb{E}[y|x])^2\big] = \mathbb{E}_{x,\varepsilon}\big[(f(x) + \varepsilon - f(x))^2\big] = \mathbb{E}_\varepsilon \varepsilon^2 = \mathbb{D}\varepsilon + \big(\mathbb{E}\varepsilon\big)^2 = \sigma_2^2 + 0 = \sigma_2^2.$$

■

Задача 1.2. *Найдите смещение алгоритма одномерной линейной регрессии для $f(x) = ax$ и для произвольной четной $f(x)$.*

Решение. Для начала найдем «средний» по всем выборкам ответ алгоритма на объекте x:

$$\mathbb{E}_X[\mu(X)(x)] = \mathbb{E}_X[k(X)]\, x.$$

Итак, нам нужно найти «среднее» по всем выборкам значение коэффициента k:

$$\mathbb{E}_X[k(X)] = \int \frac{\sum_i x_i(f(x_i) + \varepsilon_i)}{\sum_i x_i^2} \prod_i \big(p(x_i)p(\varepsilon_i)\big)\, dx_1 \dots dx_\ell\, d\varepsilon_1 \dots d\varepsilon_\ell. \tag{1.1}$$

Здесь записан несобственный интеграл, в котором каждая переменная принимает значения от $-\infty$ до ∞. Значение этого интеграла определяется функцией $f(x)$ и не всегда вычисляется аналитически. В случае, когда истинная зависимость в данных линейная, мы получим:

$$\mathbb{E}_X[k(X)] = \int \frac{\sum_i x_i(a\, x_i + \varepsilon_i)}{\sum_i x_i^2} p(\bar{x})p(\bar{\varepsilon})d\bar{x}d\bar{\varepsilon} =$$

$$= a \int \frac{\sum_i x_i^2}{\sum_i x_i^2} p(\bar{x})p(\bar{\varepsilon})d\bar{x}d\bar{\varepsilon} + \int \frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2} p(\bar{x})p(\bar{\varepsilon})d\bar{x}d\bar{\varepsilon}.$$

Мы сократили обозначение для дифференциалов и для плотностей распределений. Первый интеграл равен a (интеграл по всему пространству от плотности распределения). Второй интеграл берется по симметричным относительно нуля интервалам

от нечетной по x_i и по ε_i функции, а значит, равен 0. Итак, «средний» коэффициент равен a.

Найдем смещение:

$$\mathbb{E}_x\big[(\mathbb{E}_X[\mu(X)(x)] - \mathbb{E}[y|x])^2\big] = \mathbb{E}_x\big[(a\,x - a\,x)^2\big] = 0.$$

Это интуитивно понятный результат: логично, что перебрав все возможные выборки длины ℓ и усреднив по ним значение k, мы обязательно найдем истинную величину коэффициента.

Теперь найдем «среднее» k для произвольной четной $f(x)$. По аналогии с предыдущим случаем:

$$\mathbb{E}_X[k(X)] = \int \frac{\sum_i (x_i f(x_i))}{\sum_i x_i^2} p(\bar{x})d\bar{x} + \int \frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2} p(\bar{x})p(\bar{\varepsilon})d\bar{x}d\bar{\varepsilon}.$$

Как мы уже выяснили, второй интеграл равен 0. Первый интеграл тоже равен 0, так как подынтегральное выражение — это нечетная по всем x_i функция. Итак, «среднее» значение коэффициента равно нулю. И это тоже понятный результат, потому что четную функцию логично приближать четной, а единственная четная линейная функция, проходящая через 0 — это $y = 0$.

Найдем смещение:

$$\mathbb{E}_x\big[(\mathbb{E}_X[\mu(X)(x)] - \mathbb{E}[y|x])^2\big] = \mathbb{E}_x\big[(0 - f(x))^2\big] = \mathbb{E}_x f^2(x).$$

Чем меньше четная функция $y = f(x)$ похожа на четную линейную $y = 0$, тем больше будет смещение. Таким образом, если мы пытаемся приблизить нелинейную функцию $f(x)$ в классе линейных, мы получаем большое смещение. Обратите внимание, что если бы $f(x)$ не была четной, мы бы не смогли просто аналитически вычислить интегралы.

■

Задача 1.3. *Найдите разброс алгоритма одномерной линейной регрессии для $f(x) = ax$ и для произвольной четной $f(x)$.*

Решение. Для $f(x) = ax$:

$$\mathbb{E}_x\big[\mathbb{E}_X\big[(\mu(X)(x) - \mathbb{E}_X[\mu(X)(x)])^2\big]\big] = \mathbb{E}_x\big[\mathbb{E}_X\big[(ax + \frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2}x - ax)^2\big]\big] =$$

$$= \left(\mathbb{E}_x x^2\right)\left(\mathbb{E}_X\left(\frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2}\right)^2\right) = \sigma_1^2\mathbb{E}_X\left(\frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2}\right)^2.$$

Мат. ожидание можно немного упростить, раскрыв квадрат суммы в числителе и внеся внутрь суммы мат. ожидание по $\bar{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_\ell)$:

$$\mathbb{E}_X\left(\frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2}\right)^2 = \mathbb{E}_{\bar{x}}\left[\frac{\sum_{i\neq j} x_i x_j \mathbb{E}_{\bar{\varepsilon}}[\varepsilon_i\varepsilon_j] + \sum_i x_i^2\mathbb{E}_{\bar{\varepsilon}}\varepsilon_i^2}{(\sum_i x_i^2)^2}\right].$$

Так как ε_i и ε_j независимы, $\mathbb{E}[\varepsilon_i\varepsilon_j] = 0$, а $\mathbb{E}_{\bar{\varepsilon}}\varepsilon_i^2 = \sigma_2^2$. Тогда

$$\mathbb{E}_X\left(\frac{\sum_i x_i\varepsilon_i}{\sum_i x_i^2}\right)^2 = \mathbb{E}_{\bar{x}}\left[\frac{\sum_i x_i^2\sigma_2^2}{(\sum_i x_i^2)^2}\right] = \sigma_2^2\mathbb{E}_{\bar{x}}\left[\frac{1}{\sum_i x_i^2}\right],$$

а разброс

$$\mathbb{E}_x\big[\mathbb{E}_X\big[(\mu(X)(x) - \mathbb{E}_X[\mu(X)(x)])^2\big]\big] = \sigma_1^2\sigma_2^2\mathbb{E}_{\bar{x}}\left[\frac{1}{\sum_i x_i^2}\right].$$

Последнее мат. ожидание также можно рассчитать, но мы не будем этого делать. Мы получили, что если шум в ответах небольшой, то и разброс модели будет небольшой.

Для четной $f(x)$:

$$\mathbb{E}_x\big[\mathbb{E}_X\big[(\mu(X)(x) - \mathbb{E}_X[\mu(X)(x)])^2\big]\big] = \mathbb{E}_x\big[\mathbb{E}_X\big[(0 - \frac{\sum_i x_i(f(x_i)+\varepsilon_i)}{\sum_i x_i^2}x)^2\big]\big] =$$

$$= \left(\mathbb{E}_x x^2\right)\left(\mathbb{E}_X\left(\frac{\sum_i x_i(f(x_i)+\varepsilon_i)}{\sum_i x_i^2}\right)^2\right) = \sigma_1^2\mathbb{E}_X\left[\frac{\sum_i x_i(f(x_i)+\varepsilon_i)}{\sum_i x_i^2}\right]^2.$$

■

§1.2 Разложение для решающих деревьев

Мы выяснили, что линейные модели имеют маленькое смещение, когда истинная зависимость в данных также линейна, и большое смещение, если это не так. С решающими деревьями ситуация противоположна. Мы не будем формально это обосновывать, но интуитивно понятно, что поскольку для любой выборки можно построить дерево, имеющую нулевую ошибку на обучении, то смещение решающего дерева будет небольшое для любой истинной зависимости $f(x)$. Разброс, наоборот, будет большой, потому что при малом изменении в выборке мы можем получить совершенно другое решающее дерево.

С другой стороны, можно ограничивать многообразие деревьев, установив ограничение на глубину или минимальное число объектов в листовых вершинах. Тогда смещение будет увеличиваться, а разброс уменьшаться. В граничном случае, когда $\mu(x) = C = \text{const}$, разброс, очевидно, будет равен 0.

§1.3 Приближенное вычисление интегралов

Разложение на смещение и разброс — это теоретическая конструкция, показывающая, из-за чего происходит переобучние и недообучение алгоритмов. Обычно при анализе сложности алгоритма оперируют терминами «смещение» и «разброс», качественно оценивая их величину (например, мы так сделали с деревьями). Вычислить компоненты аналитически для большинства алгоритмов не представляется возможным. Однако, если есть необходимость количественно оценить их, можно воспользоваться техниками приближенного вычисления интегралов с помощью семплирования.

Если нам нужно оценить математическое ожидание $\mathbb{E}_{x\sim p(x)}f(x) = \int f(x)p(x)dx$, то можно просемплировать выборку $\{x_1, \ldots, x_n\}$ из распределения $p(x)$ и приближенно вычислить интеграл:

$$\mathbb{E}_{x\sim p(x)}f(x) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

Из областей с большим значением плотности в выборку попадет больше точек, и они внесут больший вклад в значение интеграла. Несложно показать, что данная оценка является несмещенной.

Для вычисления математического ожидания по случайным выборкам нужно сгенерировать несколько выборок:

$$\mathbb{E}_X[\mu(X)(x)] \approx \frac{1}{n}\sum_{i=1}^{n}\mu(X_i)(x), \quad X_i = \{(x_{i,1}, y_{i,1}), \ldots, (x_{i,\ell}, y_{i,\ell})\},\ x_{i,j}, y_{i,j} \sim p(x,y)$$

В numpy для генерации выборок можно пользоваться модулем numpy.random.

2 Композиционные алгоритмы

§2.1 Беггинг

Существуют способы уменьшить разброс алгоритма. Наиболее известный из них — бэггинг. Беггинг заключается в генерации нескольких новых выборок $X_1, \ldots, X_m$ на основе имеющейся, обучении алгоритма на каждой из сгенерированных выборок и усреднении ответов всех алгоритмов на новом объекте.

Задача 2.1. *При бэггинге новую выборку $\widetilde{X}$ составляют, генерируя элементы из X с возвращением. При этом объекты в $\widetilde{X}$ могут повторяться. Будем считать, что число объектов в $\widetilde{X}$ и в X одинаковое и равно ℓ. Найдите вероятность того, что конкретный объект попадет в выборку.*

Решение.

Вероятность того, что объект попадет в выборку при одном вытаскивании — $\frac{1}{\ell}$, ℓ — число объектов выборки. Вероятность того, что не попадет — $1 - \frac{1}{\ell}$; не попадет ни при одном вытаскивании — $\left(1 - \frac{1}{\ell}\right)^{\ell}$. Наконец, искомая вероятность

$$1 - \left(1 - \frac{1}{\ell}\right)^{\ell} \to_{\ell\to\infty} 1 - \frac{1}{e}.$$

■

На лекции было показано, что усреднение с помощью бэггинга уменьшает разброс алгоритма в m раз, если базовые алгоритмы мало коррелированы.

Такой эффект наблюдается при усреднении предсказаний любых алгоритмов регрессии, необязательно полученных бэггингом, если эти алгоритмы выдают слабо коррелированные ответы.

§2.2 Простое голосование

Идею о том, что при выборе итогового предсказания для объекта x с помощью агрегации предсказаний нескольких базовых алгоритмов предсказание будет точнее, можно применить и к классификации. Наиболее простой способ это сделать — построить несколько классификаторов, предсказывающих класс для объекта x, и выбирать класс, который чаще всего предсказывали эти алгоритмы (простое голосование).

Задача 2.2. *Пусть у нас есть три бинарных классификатора, каждый из которых ошибается с вероятностью p. С какой вероятностью будет ошибаться классификатор, построенный с помощью простого голосования? При каких значениях p эта вероятность будет меньше p?*

Решение. Простое голосование выдаст правильный ответ, если не более чем один алгоритм ошибется. Вероятность того, что все три алгоритма ответят правильно, равна $(1-p)^3$, вероятность того, что ровно два из трех ответят правильно: $3p(1-p)^2$. Итоговая вероятность ошибки:

$$1-(1-p)^3-3p(1-p)^2 = 1-1+3p-3p^2+p^3-3p+6p^2-3p^3 = -2p^3+3p^2 = p^2(3-2p).$$

$$-2p^3 + 3p^2 < p;$$

$$p(-2p^2 + 3p - 1) < 0;$$

$$-2p(p-1)(p-\frac{1}{2}) < 0.$$

На участке $p \in [0,1]$ решением этого неравенства является полуинтервал $p \in (0, \frac{1}{2})$. Таким образом, если каждый из алгоритмов хотя бы чуть лучше, чем случайный, то композиция будет давать меньше ошибок, чем каждый алгоритм по отдельности.

■