

Семинары по композиционным методам

Евгений Соколов
sokolov.evg@gmail.com

9 марта 2014 г.

1 Композиционные методы машинного обучения

§1.1 Бутстрэппинг

Рассмотрим простой пример построения композиции алгоритмов. Пусть дана конечная выборка X^ℓ и вещественные ответы на ней Y^ℓ . Будем решать задачу линейной регрессии. Сгенерируем подвыборку с помощью *бутстрэппинга*. Равномерно возьмем из выборки ℓ объектов с возвращением. Из-за возвращения среди них окажутся повторы; оставив по одной копии каждого объекта, мы получим подвыборку меньшего размера X_1 . Повторив процедуру n раз, сгенерируем n подвыборок X_1, \dots, X_n . Обучим по каждой из них линейную функцию регрессии, получив алгоритмы $a_1(x), \dots, a_n(x)$.

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

$$\varepsilon_i(x) = a_i(x) - y(x), \quad i = 1, \dots, n,$$

и записать матожидание среднеквадратичной ошибки

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

Средняя ошибка построенных функций регрессии имеет вид

$$E_1 = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_x \varepsilon_i^2(x).$$

Предположим, что ошибки несмещены и некоррелированы:

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

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

$$a(x) = \frac{1}{n} \sum_{i=1}^n a_i(x).$$

Найдем ее среднеквадратичную ошибку:

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

Таким образом, усреднение ответов позволило уменьшить ошибку 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. Ниже мы рассмотрим более сложные методы объединения алгоритмов в композицию, которые позволяют добиться высокого качества в реальных задачах.

§1.2 Адаптивный бустинг

Алгоритм AdaBoost (Adaptive Boosting) — одна из первых реализаций идеи о том, что путем объединения алгоритмов можно улучшить их качество.

Рассмотрим задачу классификации на два класса: $\mathbb{Y} = \{-1, +1\}$. Нашей задачей является поиск классификатора, минимизирующего число ошибок на обучении:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} [y_i a(x_i) < 0] \rightarrow \min_a. \quad (1.1)$$

Данный функционал является дискретным, что затрудняет его оптимизацию. Чтобы обойти эту проблему, перейдем к его верхней оценке. Заметим, что индикатор можно оценить сверху экспонентой:

$$[z < 0] \leq \exp(-z).$$

Применив эту оценку, получим новую задачу:

$$\tilde{Q}(a, X^\ell) = \sum_{i=1}^{\ell} \exp(-y_i a(x_i)) \rightarrow \min_a. \quad (1.2)$$

Пусть дано некоторое семейство базовых классификаторов \mathcal{A} , каждый алгоритм из которого является отображением из пространства объектов в $\{-1, +1\}$. В AdaBoost требуется, чтобы в данном семействе для любого набора весов w_1, \dots, w_ℓ нашелся алгоритм, взвешенная ошибка которого будет меньше $1/2$:

$$\exists a \in \mathcal{A} : \sum_{i=1}^{\ell} w_i [a(x_i) \neq y_i] < \frac{1}{2}. \quad (1.3)$$

Это свойство называется *слабой обучаемостью*.

Предлагается строить итоговый классификатор в виде суммы классификаторов из данного семейства с положительными коэффициентами:

$$a(x) = \text{sign} \sum_{n=1}^N \gamma_n a_n(x), \quad \gamma_n > 0.$$

Сумма строится последовательно. Каждый новый классификатор и вес выбираются так, чтобы минимизировать функционал (1.2). После добавления классификатор и его вес уже не меняются, т.е. процедура является жадной.

Пусть мы уже построили сумму из $(N - 1)$ -го классификатора.

Задача 1.1. Выведите формулы для $a_N(x)$ и γ_N .

Решение. Мы хотим получить классификатор вида

$$a(x) = \sum_{n=1}^N \gamma_n a_n(x).$$

Все слагаемые с 1-го по $(N - 1)$ -й фиксированы, и требуется найти лишь последний базовый классификатор и вес при нем. Запишем функционал:

$$\begin{aligned} \tilde{Q}(a, X^\ell) &= \sum_{i=1}^{\ell} \exp \left(-y_i \sum_{n=1}^N \gamma_n a_n(x_i) \right) = \\ &= \sum_{i=1}^{\ell} \underbrace{\exp \left(-y_i \sum_{n=1}^{N-1} \gamma_n a_n(x_i) \right)}_{=w_i} \exp(-y_i \gamma_N a_N(x_i)) = \\ &= \sum_{i=1}^{\ell} w_i \exp(-y_i \gamma_N a_N(x_i)). \end{aligned}$$

Выделим отдельно слагаемые, отвечающие за верно классифицированные объекты:

$$\begin{aligned} \tilde{Q}(a, X^\ell) &= \sum_{i=1}^{\ell} [y_i a_N(x_i) = 1] w_i \exp(-y_i \gamma_N a_N(x_i)) + \\ &\quad \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] w_i \exp(-y_i \gamma_N a_N(x_i)) = \\ &= e^{-\gamma_N} \sum_{i=1}^{\ell} [y_i a_N(x_i) = 1] w_i + e^{\gamma_N} \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] w_i = \\ &= (e^{\gamma_N} - e^{-\gamma_N}) \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] w_i + e^{-\gamma_N} \sum_{i=1}^{\ell} w_i. \end{aligned}$$

Все дальнейшие рассуждения получатся более стройными и интерпретируемыми, если мы отнормируем веса:

$$\tilde{Q}(a, X^\ell) = \left((e^{\gamma_N} - e^{-\gamma_N}) \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] \tilde{w}_i + e^{-\gamma_N} \underbrace{\sum_{i=1}^{\ell} \tilde{w}_i}_{=1} \right) \sum_{i=1}^{\ell} w_i,$$

где $\tilde{w}_i = w_i / \sum_i w_i$. Множитель $\sum_i w_i$ из функционала можно вычеркнуть, поскольку он не зависит от $a_N(x)$ и γ_N . Получаем задачу

$$\tilde{Q}(a, X^\ell) = (e^{\gamma_N} - e^{-\gamma_N})\varepsilon_N + e^{-\gamma_N} \rightarrow \min, \quad (1.4)$$

где через ε_N мы обозначили взвешенную ошибку

$$\varepsilon_N = \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] \tilde{w}_i.$$

Выясним, как искать оптимальный базовый классификатор $a_N(x)$. Заметим, что в функционале (1.4) от $a_N(x)$ зависит лишь ε_N . Стоящие при ε_N коэффициент $e^{\gamma_N} - e^{-\gamma_N}$ будет положительным, поскольку мы строим композицию с положительными коэффициентами ($\gamma_N > 0$); следовательно, его можно отбросить. Таким образом, выбирается классификатор, минимизирующий взвешенную ошибку:

$$a_N(x) = \arg \min \sum_{i=1}^{\ell} [y_i a_N(x_i) = -1] \tilde{w}_i.$$

Продифференцируем по коэффициенту γ_N функционал и приравняем к нулю:

$$\frac{\partial \tilde{Q}}{\partial \gamma_N} = (e^{\gamma_N} + e^{-\gamma_N})\varepsilon_N - e^{-\gamma_N} = 0.$$

Преобразуя, получаем

$$\gamma_N = \frac{1}{2} \log \frac{1 - \varepsilon_N}{\varepsilon_N}. \quad (1.5)$$

■

Заметим, что вес w_i будет большим, если отступ классификатора $a(x)$ на объекте x_i будет большим и отрицательным. Таким образом, AdaBoost пытается уменьшать ошибку на тех объектах, которые хуже всего получилось классифицировать на предыдущей итерации.

Задача 1.2. Обозначим через $\tilde{w}^{(n)}$ веса на n -й итерации. Покажите, что взвешенная ошибка базового классификатора a_N относительно весов со следующего шага $\tilde{w}^{(N+1)}$ равна $1/2$.

Решение.

$$\begin{aligned}
& \sum_{i=1}^{\ell} \tilde{w}_i^{(N+1)} [a_N(x_i) \neq y_i] = \\
&= \sum_{i=1}^{\ell} \frac{w_i^N \exp(-y_i \gamma_N a_N(x_i))}{\sum_{j=1}^N w_j^N \exp(-y_j \gamma_N a_N(x_j))} [a_N(x_i) \neq y_i] = \\
&= \frac{\sum_{i=1}^{\ell} w_i^N \exp(-y_i \gamma_N a_N(x_i)) [a_N(x_i) \neq y_i]}{\sum_{j=1}^N w_j^N \exp(-y_j \gamma_N a_N(x_j))} = \\
&= \frac{e^{\gamma_N} \sum_{i=1}^{\ell} w_i^N [a_N(x_i) \neq y_i]}{e^{\gamma_N} \sum_{j=1}^N w_j^N [a_N(x_j) \neq y_j] + e^{-\gamma_N} \sum_{j=1}^N w_j^N [a_N(x_j) = y_j]} = \\
&= \left\{ \gamma_N = \frac{1}{2} \log \left(\frac{1 - \varepsilon_N}{\varepsilon_N} \right) \right\} = \\
&= \frac{\sqrt{\frac{1 - \varepsilon_N}{\varepsilon_N}} \sum_{i=1}^{\ell} w_i^N [a_N(x_i) \neq y_i]}{\sqrt{\frac{1 - \varepsilon_N}{\varepsilon_N}} \sum_{j=1}^N w_j^N [a_N(x_j) \neq y_j] + \sqrt{\frac{\varepsilon_N}{1 - \varepsilon_N}} \sum_{j=1}^N w_j^N [a_N(x_j) = y_j]} = \\
&= \frac{\sqrt{\frac{1 - \varepsilon_N}{\varepsilon_N}} \varepsilon_N}{\sqrt{\frac{1 - \varepsilon_N}{\varepsilon_N}} \varepsilon_N + \sqrt{\frac{\varepsilon_N}{1 - \varepsilon_N}} (1 - \varepsilon_N)} = \\
&= \frac{\sqrt{\varepsilon_N (1 - \varepsilon_N)}}{\sqrt{\varepsilon_N (1 - \varepsilon_N)} + \sqrt{\varepsilon_N (1 - \varepsilon_N)}} = \\
&= \frac{1}{2}.
\end{aligned}$$

■

Данный результат означает, что распределение на новой итерации подбирается так, что классификатору с предыдущего шага сложнее всего справиться с ним.

Скорость сходимости. Выясним, с какой скоростью уменьшается ошибка композиции на обучающей выборке. Выше мы представили функционал $\tilde{Q}(a, X^\ell)$ в виде

$$\tilde{Q}(a, X^\ell) = \left((e^{\gamma_N} - e^{-\gamma_N}) \varepsilon_N + e^{-\gamma_N} \right) \sum_{i=1}^{\ell} w_i.$$

Заметим, что

$$\sum_{i=1}^{\ell} w_i = \sum_{i=1}^{\ell} \exp \left(-y_i \sum_{n=1}^{N-1} \gamma_n a_n(x_i) \right),$$

то есть сумма весов представляет собой ошибку композиции, состоящей из $(N - 1)$ -го алгоритма; обозначим ее через \tilde{Q}_{N-1} . Получаем, что ошибка на текущей итерации выражается через ошибку на предыдущей:

$$\tilde{Q}_N = \left((e^{\gamma_N} - e^{-\gamma_N}) \varepsilon_N + e^{-\gamma_N} \right) \tilde{Q}_{N-1}.$$

Подставим оптимальный коэффициент (1.5) в данное уравнение:

$$\begin{aligned}\tilde{Q}_N &= \left(\left(\sqrt{\frac{1-\varepsilon_N}{\varepsilon_N}} - \sqrt{\frac{\varepsilon_N}{1-\varepsilon_N}} \right) \varepsilon_N + \sqrt{\frac{\varepsilon_N}{1-\varepsilon_N}} \right) \tilde{Q}_{N-1} = \\ &= \left(\sqrt{\varepsilon_N(1-\varepsilon_N)} + (1-\varepsilon_N) \sqrt{\frac{\varepsilon_N}{1-\varepsilon_N}} \right) \tilde{Q}_{N-1} = \\ &= 2\sqrt{\varepsilon_N(1-\varepsilon_N)} \tilde{Q}_{N-1}.\end{aligned}$$

Ранее мы потребовали, чтобы семейство классификаторов \mathcal{A} обладало свойством слабой обучаемости; из него следует, что мы найдем такой классификатор a_N , что его взвешенная ошибка будет меньше $1/2$: $\varepsilon_N < 1/2$. Предположим, что на каждой итерации мы будем находить классификатор, ошибка которого отделена от $1/2$ константой α :

$$\varepsilon_n \leq \frac{1}{2} - \alpha, \quad \alpha > 0.$$

Учитывая это условие, оценим ошибку композиции на N -й итерации:

$$\begin{aligned}\tilde{Q}_N &= 2\sqrt{\varepsilon_N(1-\varepsilon_N)} \tilde{Q}_{N-1} \leq 2\sqrt{\left(\frac{1}{2} - \alpha\right) \left(\frac{1}{2} + \alpha\right)} \tilde{Q}_{N-1} = 2\sqrt{\frac{1}{4} - \alpha^2} \tilde{Q}_{N-1} = \\ &= \left(\sqrt{1 - 4\alpha^2}\right) \tilde{Q}_{N-1} \leq (1 - 4\alpha^2)^{N/2} \tilde{Q}_1.\end{aligned}$$

Таким образом, ошибка убывает со скоростью геометрической прогрессии с коэффициентом $\sqrt{1 - 4\alpha^2}$.