Разработка эффективной реализации методов, основанных на поиске оптимального баланса между дивергенцией и точностью аппроксимации

A Preprint

Аристархов Данила Дмитриевич
ВМК МГУ
aristarkhov.danila@yandex.ru

Сенько Олег Валентинович
ВМК МГУ
senkoov@mail.ru

Abstract

Рассмотрен новый метод построения ансамбля деревьев при решении задачи регрессии. При этом оптимизация производится исходя из одновременного достижения расходимости алгоритмов в пространстве прогнозов и хорошей аппроксимации данных отдельными алгоритмами ансамбля.

Ключевые слова регрессия · коллективные методы · бэггинг · градиентный бустинг

1 Введение

Методы решения задачи регрессии, основанные на вычислении более точного предсказания с помощью набора менее точных и более простых базовых алгоритмов, получили самое широкое распространение в современном машинном обучении. К числу таких методов может быть отнесен регрессионный случайный лес, а также методы, основанные на использовании адаптивного или градиентного бустинга. Важную роль при построении таких алгоритмов играет способ получения ансамбля так называемых слабых алгоритмов. Теоретический анализ показывает, что увеличение обобщающей способности может быть достигнуто за счет выбора ансамбля алгоритмов, обладающих не только высокой точностью, но и максимально расходящимися прогнозами [Докунин and Сенько, 2015]. В случайном лесе расходимость прогнозов достигается за счет обучения алгоритмов ансамбля на различных выборках, генерируемых из исходной выборки с использованием процедуры бутстрэпа [Brieman, 1996]. В методе градиентного бустинга [Hastie et al., 2009] ансамбль генерируется последовательно. При этом на каждой итерации в ансамбль добавляются деревья, аппроксимирующие первые производные функции потерь в точке, соответствующей текущему прогнозу алгоритма на данном шаге.

Построение ансамбля, имеющего как высокую точность предсказаний, так и сильное расхождение прогнозов отдельных алгоритмов, продемонстрировало улучшение качества модели [Журавлев et al., 2021]. Целью данной работы является продолжение исследования данного подхода.

2 Постановка задачи

Дана выборка $S = (X_1, y_1), \ldots, (X_m, y_m)$, где X_i — вектор признакового описания объекта, y_i — метка объекта. Рассматривается задача регресии: $X_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$. Требуется построить ансамбль базовых алгоритмов $A_1(X), \ldots, A_k(X)$, предсказывающих значения метки по вектору признаков.

3 Дисперсия ансамбля

Рассмотрим известное разложение среднеквадратичного риска модели $\mu(x)$ (bias-variance decomposition):

$$L(\mu) = N(\mu) + B(\mu) + V(\mu),$$

где

$$N(\mu) = \mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y|x]\big)^2\Big] \text{ — шум,}$$
$$B(\mu) = \mathbb{E}_{x}\Big[\big(\mathbb{E}_X\big[\mu(X)\big] - \mathbb{E}[y|x]\big)^2\Big] \text{ — смещение,}$$
$$V(\mu) = \mathbb{E}_{X}\Big[\big(\mu(X) - \mathbb{E}_X\big[\mu(X)\big]\big)^2\Big] \text{ — разброс.}$$

Проанализируем это разложение в случае ансамбля базовых алгоритмов $\mu(X) = \frac{1}{k}\sum_{i=1}^{k} A_i(X)$. Шум является свойством выборки и не зависит от алгоритма. Выражения для смещения принимает вид:

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

Таким образом, ансамблирование не изменяет смещенности относительно базового алгоритма.

Запишем выражения для разброса ансамбля:

$$\begin{aligned} V(\mu) &= \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(\frac{1}{k}\sum_{i=1}^{k} A_i(X)(x) - \mathbb{E}_X\Big[\frac{1}{k}\sum_{i=1}^{k} A_i(X)(x)\Big]\Big)^2\Big]\Big] \\ &= \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\frac{1}{k^2}\Big(\sum_{i=1}^{k}\Big[A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big]\Big)^2\Big]\Big] = \\ &= \mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\frac{1}{k^2}\sum_{i=1}^{k}\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)^2 + \\ &\quad + \frac{1}{k^2}\sum_{i\neq j}\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)\Big(A_j(X)(x) - \mathbb{E}_X\big[A_j(X)(x)\big]\Big)\Big]\Big] \\ &= \frac{1}{k^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\sum_{i=1}^{k}\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)^2\Big]\Big] + \\ &\quad + \frac{1}{k^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\sum_{i\neq j}\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)\Big(A_j(X)(x) - \mathbb{E}_X\big[A_j(X)(x)\big]\Big)\Big]\Big] \\ &= \frac{1}{k}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)^2\Big]\Big] + \\ &\quad + \frac{k(k-1)}{k^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_X\Big[\Big(A_i(X)(x) - \mathbb{E}_X\big[A_i(X)(x)\big]\Big)\Big(A_j(X)(x) - \mathbb{E}_X\big[A_j(X)(x)\big]\Big)\Big]\Big] \end{aligned}$$

В данном разложении первое слагаемое — это дисперсия одного базового алгоритма, деленная на длину композиции k. Второе — ковариация между двумя базовыми алгоритмами. При малой коррелляции алгоритмов происходит существенное снижение разброса ансамбля по сравнению с базовым алгоритмом.

4 Случайный лес

В качестве базового алгоритма в случайных лесах используются решающие деревья. Они имеют достаточную сложность, и, как следствие, низкую смещенность (при неограниченной глубине дерево

может идеально подстроиться под выборку), но переобучаются и имеют высокий разброс, который можно уменьшить с помощью ансамблирования. В случайных лесах коррелляция между деревьями понижается двумя способами:

- Бэггинг. Каждый алгоритм обучается на случайной подвыборке, сгенерированной из выборки с помощью бутстрэпа, т.е. выбираются m объектов с возвращениями. Таким образом, в одной выборке некоторые объекты встретятся несколько раз, а некоторые — ни разу.
- Рандомизация признаков. При построении очередного дерева в каждой вершине выбор наилучшего признака для разбиения происходит не из всех возможных признаков, а из случайно выбранной подвыборки.

Так удается добавить случайность в построение деревьев и уменьшить коррелированность их прогнозов.

5 Предлагаемый метод

Обозначим $L_k(X) = \frac{1}{k}\sum_{i=1}^{k} A_i(X)$, $Q_k(X) = \frac{1}{k}\sum_{i=1}^{k} A_i^2(X)$. Введем критерий Φ_E:

$$\Phi_E(A_1(X), \ldots, A_k(X)) = \frac{1}{mk}\sum_{i=1}^{k}\sum_{j=1}^{m}(y_j - A_i(X_j))^2,$$

являющийся среднеквадратичной ошибкой алгоритма, и Φ_V

$$\Phi_V(A_1(X), \ldots, A_k(X)) = \frac{1}{mk}\sum_{i=1}^{k}\sum_{j=1}^{m}(L_k(X_j) - A_i(X_j))^2,$$

представляющий собой дисперсию прогнозов вычисляемых алгоритмов.

При построении ансамбля предлагается предлагается явно минимизировать Φ_E и максимизировать Φ_V. Данная задача может быть сведена к минимизации Φ_G:

$$\Phi_G = (1-\mu)\Phi_E - \mu\Phi_V,$$

где $\mu \in [0, 1]$ является гиперпараметром, определяющим соотношение точности и разнородности прогнозов отдельных деревьев.

Поскольку каждое дерево в ансамбле строится отдельно от других, необходимо получить критерий для построения очередного дерева. Обозначим через D_E^k и D_V^k изменение функционалов Φ_E и Φ_V при включении в ансамбль дополнительного алгоритма A_{k+1}.

$$\begin{aligned}
D_E^k &= \Phi_E(A_1(X), ..., A_{k+1}(X)) - \Phi_E(A_1(X), \ldots, A_k(X)) \\
&= \frac{1}{k+1}\left(\Phi_E(A_1(X), \ldots, A_k(X)) \cdot k + \frac{1}{m}\sum_{i=1}^{m}(y_j - A_{k+1}(X_j))^2\right) - \Phi_E(A_1(X), \ldots, A_k(X)) \\
&= \frac{1}{m(k+1)}\sum_{j=1}^{m}(y_j - A_{k+1}(X_j))^2 - \frac{1}{k+1}\Phi_E(A_1(X), \ldots, A_k(X)) \\
&= \frac{1}{m(k+1)}\sum_{j=1}^{m}(y_j - A_{k+1}(X_j))^2 - C_E,
\end{aligned}$$

где C_E не зависит от $A_{k+1}(X)$.

Для расчета D_V^k воспользуемся известным соотношением для выборочной дисперсии:

$$\sum_{j=1}^{m}\sum_{i=1}^{k}(L_k(X_j) - A_i(X_j))^2 = \sum_{j=1}^{m}\left(\frac{1}{k}\sum_{i=1}^{k}A_i^2(X_j) - \left(\frac{1}{k}\sum_{i=1}^{k}A_i(X_j)\right)^2\right) = \sum_{j=1}^{m}(Q_k(X_j) - L_k^2(X_j))$$

Тогда выражение для D_V^K принимает вид:

$$D_V^K = \Phi_V(A_1(X), \ldots, A_{k+1}(X)) - \Phi_V(A_1(X), \ldots, A_k(X))$$

$$= \frac{1}{m}\sum_{j=1}^{m}(Q_{k+1}(X_j) - L_{k+1}^2(X_j)) - \frac{1}{m}\sum_{j=1}^{m}(Q_k(X_j) - L_k^2(X_j))$$
$$= \frac{1}{m}\sum_{j=1}^{m}(\frac{1}{k+1}\sum_{i=1}^{k+1}A_i^2(X_j) - \frac{1}{k}\sum_{i=1}^{k}A_i^2(X_j) - (kL_k(X_j) + A_{k+1}(X_j))^2 + L_k^2(X_j))$$
$$= \frac{1}{m(k+1)}\sum_{j=1}^{m}(A_{k+1}^2(X_j) + \sum_{i=1}^{k}A_i^2(X_j) - \frac{k+1}{k}\sum_{i=1}^{k}A_i^2(X_j)$$
$$- \frac{k^2}{k+1}L_k^2(X_j) - \frac{2k}{k+1}L_k(X_j)A_{k+1}(X_j) - \frac{1}{k+1}A_{k+1}^2(X_j) + (k+1)L_k^2(X_j))$$
$$= \frac{1}{m(k+1)}\sum_{j=1}^{m}(\frac{k}{k+1}A_{k+1}^2(X_j) - Q_k(X_j) - \frac{2k}{k+1}L_k(X_j)A_{k+1}(X_j) - \frac{k^2}{k+1}L_k^2(X_j) + (k+1)L_k^2(X_j))$$
$$= \frac{k}{m(k+1)^2}\sum_{j=1}^{m}(A_{k+1}^2(X_j) - 2L_k(X_j)A_{k+1}(X_j)) + C_V,$$

где C_V не зависит от $A_{k+1}(X)$.

Объединяя эти выражения, получаем функционал, который необходимо минимизировать при построении очередного дерева $A_{k+1}(X)$:

$$D_G^k = (1-\mu)D_E^k - \mu D_V^k =$$
$$= \frac{1-\mu}{m(k+1)}\sum_{j=1}^{m}(y_j - A_{k+1}(X_j))^2 - \frac{\mu k}{m(k+1)^2}\sum_{j=1}^{m}(A_{k+1}^2(X_j) - 2L_k(X_j)A_{k+1}(X_j)) + C_G, \qquad (1)$$

где C_G не зависит от $A_{k+1}(X)$.

Теперь рассмотрим вопрос оптимального значения в листе дерева $A_{k+1}(X)$. Пусть в лист попали объекты $(X_{n_1}, y_{n_1}), \ldots, (X_{n_p}, y_{n_p})$. В листе алгоритм предсказывает одно значение для всех объектов, попавших в этот лист: $A_{k+1}(X_{n_j}) \equiv \tilde{A},\ j = \overline{1,p}$. Найдем производную функционала (1) относительно прогноза $\tilde{A}$:

$$\frac{\partial D_G^k}{\partial \tilde{A}} = \frac{2(1-\mu)}{p(k+1)}\sum_{j=1}^{p}(\tilde{A} - y_{n_j}) - \frac{2\mu k}{p(k+1)^2}\sum_{j=1}^{p}(\tilde{A} - L_k(X_{n_j}))$$
$$= \frac{2}{p(k+1)}\sum_{j=1}^{p}\left((1-\mu\frac{2k+1}{k+1})\tilde{A} - (1-\mu)y_{n_j} + \frac{k\mu}{k+1}L_k(X_{n_j})\right)$$

Приравнивая производную к нулю, получаем оптимальный прогноз:

$$\tilde{A} = \sum_{j=1}^{p}\frac{(k+1)(1-\mu)y_{n_j} - \mu k L_k(X_{n_j})}{p(k+1-\mu(2k+1))} \qquad (2)$$

Заметим, что в формулах (1) и (2) используются $L_k(X)$, т.е., для построения $A_{k+1}(X)$ требуется хранить среднее предсказаний $A_1(X), \ldots, A_k(X)$. Также формулы (1) и (2) зависят от k. Следовательно, построение деревьев очередного дерева в ансамбле зависит от его номера и предсказаний всех предыдущих деревьев.

Подводя итог, построение ансамбля $A_1(X), \ldots, A_N(X)$ происходит следующим образом:

Algorithm 1 Предложенный алгоритм

1: Сгенерировать выборку $\tilde{X}_1$ с помощью бутстрэпа
2: Построить решающее дерево $A_1(x)$ по выборке $\tilde{X}_1$, используя только среднеквадратичную ошибку
3: Вычислить $L_1(X) = A_1(X)$ для всех $X_1, \ldots, X_m$
4: for $k = 2, \ldots, N$ do
5: Сгенерировать выборку $\tilde{X}_k$ с помощью бутстрэпа
6: Построить решающее дерево $A_k(x)$ по выборке $\tilde{X}_k$, используя $L_{k-1}(X)$:
- В каждой вершине ищется оптимальное разбиение относительно функционала (1)
- Для вычисления значений в листе используется выражение (2)
7: Вычислить $L_k(X) = \frac{1}{k}((k-1)L_{k-1}(X) + A_k(X))$ для всех $X_1, \ldots, X_m$
8: end for
9: Вернуть композицию $\mu_N(X) = \frac{1}{N}\sum_{k=1}^{N} A_k(X)$

Отметим, что в данном алгоритме при построении дерева используются и обычные способы увеличения дисперсии, описанные в разд. 4. Также используются параметры, ограничивающие глубину деревьев, такие как:

- Максимальная глубина
- Минимальное число объектов для разбиения вершины
- Минимальное число объектов в листе

6 Эксперименты

Метод был реализован на языке python с помощью библиотеки numpy. В качестве референсного метода для оценки эффективности использовался обычный случайный лес RandomForestRegressor, реализованный в библиотеке scikit-learn Pedregosa et al. [2011]. Для экспериментов были зафиксированы следующие параметры:

- Количество алгоритмов в ансамбле (n_estimators): 100
- Максимальная глубина дерева (max_depth): 7
- Минимальное количество объектов в листе (min_samples_leaf): 5
- Размер подвыборки признаков при разбиении (max_features): 1/3 (от числа всех признаков)

Алгоритм исследовался для нескольких значений параметра μ: 0.05, 0.1, 0.2, 0.3

Разработанные метод использовался на следующих данных:

1. Предсказание развития диабета по медицинским показателям пациента (Diabetes)
2. Предсказание стоимости жилья в районе по средним характеристикам домов (California housing)
3. Синтетический датасет, имеющий 1000 объектов и 10 признаков, 5 из которых являются информативными, а 5 — шумовыми (Synthetic)
4. Предсказание стоимости жилья в районе по характеристикам города (Boston)

Результаты экспериментов описаны на рис. 1. Для оценки точности использовался коэффициент детерминации r^2 на тестовой выборке. Для этого данные предварительно разбивались на обучающую и тестовую выборки в соотношении 8:2. В связи с тем, что генерация выборок с помощью бутстрэпа и построение деревьев происходит в значительной мере случайно, результаты решения одной и той же задачи изменяются от эксперимента к эксперименту. В связи с этим каждый эксперимент проводился 10 раз, а результаты приведены в виде «ящика с усами».

По результатам экспериментов метод показал прирост качества по сравнению с референсным методом. При этом качество росло с увеличением параметра μ, но до некоторого предела, после которого качество падало. На данных California housing и Synthetic метод продемонстрировал существенное улучшение, оптимальным значением параметра оказалось $\mu = 0.2$. На данных Diabetes прирост был меньше, лучшее качество было достигнуто при $\mu = 0.1$. Таким образом, для достижения наилучшей точности требуется подбор оптимального значения гиперпараметра μ.

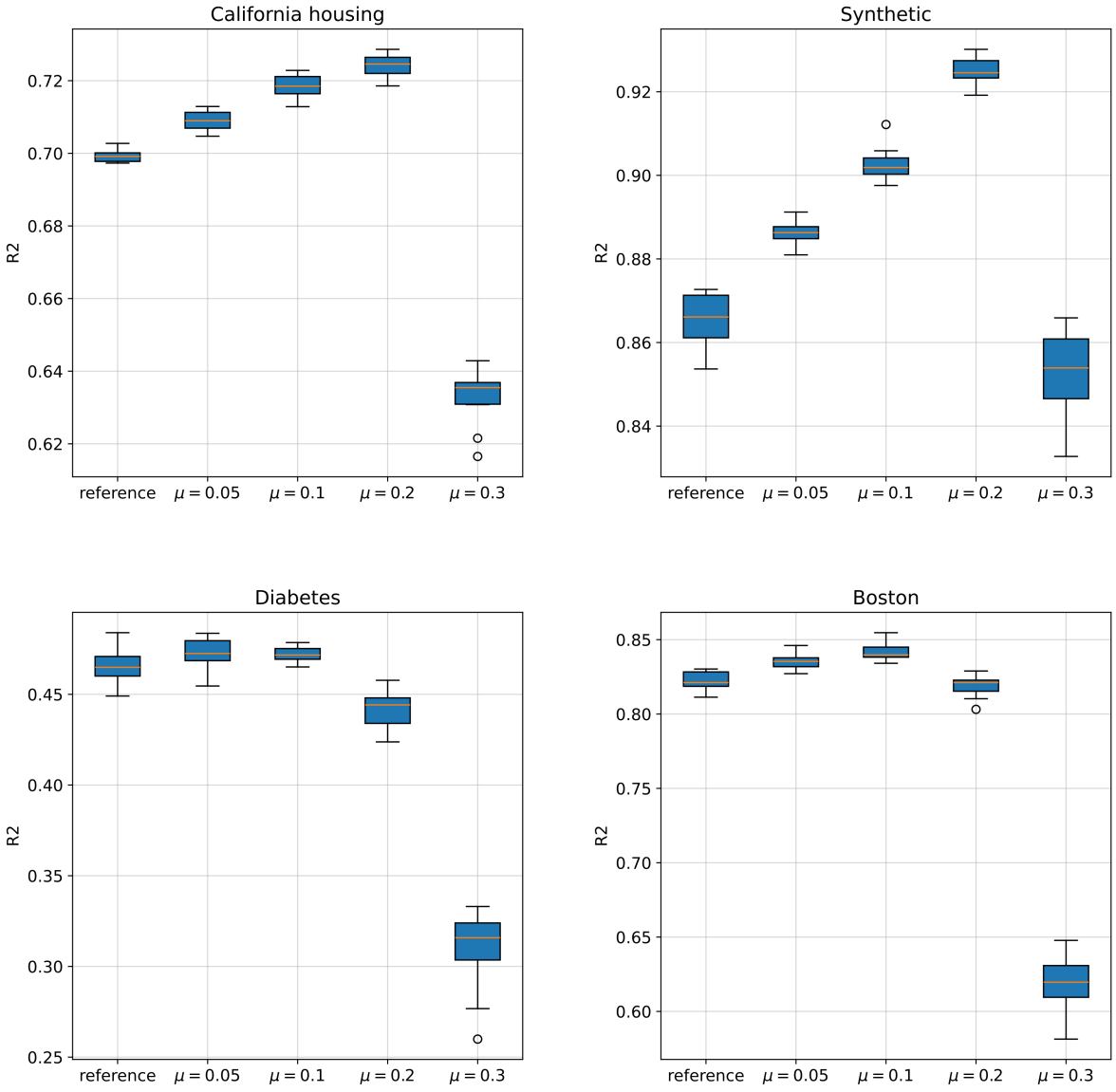

Рис. 1: Эксперименты

Литература

А. А. Докунин and О. В. Сенько. Регрессионная модель, основанная на выпуклых комбинациях, максимально коррелирующих с откликом. Ж. вычисл. матем и матем. физ., 53:530–544, 2015.

L. Brieman. Bagging predictors. Machine Learning., 24:123–140, 1996.

T. Hastie, R. Tibshirani, and J. H. Friedman. The Elements of Statistical Learning (2nd ed.). 2009.

Ю. И. Журавлев, О. В. Сенько, А. А. Докунин, Н. Н. Киселева, and И. А. Саенко. Двухуровневый метод регрессионного анализа, использующий ансамбли деревьев с оптимальной дивергенцией. Докл. РАН. Матем., информ., проц. упр., 499:63–66, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.