

# Альтернативные методы построения ансамблей [М.193]

Применение ансамблей моделей к решению различных задач анализа открывает широкие возможности для повышения эффективности моделей Data Mining. Поэтому в последние несколько лет в данной области проводятся активные исследования, в результате чего появилось большое количество методов.

Не все методы работают одинаково хорошо в разных ситуациях, а возможность выбора позволяет аналитику добиться наилучших результатов. Например, бустинг и бэггинг изначально возникли как методы повышения эффективности классификационных моделей, особенно деревьев решений, и не могли напрямую использоваться для регрессионных моделей. Поэтому для решения задач регрессии были разработаны варианты бустинга, получившие название аддитивной регрессии, в частности алгоритм **LogitBoost**. Распространены и такие методы комбинирования моделей, как деревья выбора и стэкинг. Рассмотрим основы их работы.

## Аддитивная регрессия

Появление бустинга вызывало большой интерес в Data Mining, поскольку он позволяет обеспечить очень высокую эффективность моделей. Одним из вариантов бустинга стало построение так называемых *аддитивных регрессионных моделей*. Понятие «аддитивная регрессия» относится к любым методам предсказания, основанным на комбинировании (суммировании) вкладов от нескольких регрессионных моделей.

В основе идеи аддитивной регрессии лежит *прямое ступенчатое аддитивное моделирование* (forward stagewise additive modeling), алгоритм которого содержит следующие шаги.

- 1 Сначала строится обычная регрессионная модель, например регрессионное дерево решений.
- 2 Вычисляются ошибки, полученные на обучающем множестве, как разность между желаемым и наблюдаемым значениями. Эти ошибки называются *остатками* (residuals).
- 3 Производится минимизация остатков с помощью второй модели (другого дерева решений), которая пытается предсказать наблюдаемые остатки. Для этого исходные целевые значения заменяются соответствующими остатками перед обучением второй модели.
- 4 Поскольку и вторая модель не является идеальной, она также даст остатки на некоторых примерах. Поэтому процесс продолжится: строится третья модель, которая обучается предсказывать остатки от остатков, и т. д.

Если каждая модель минимизирует квадратичную ошибку предсказания, как делает обычная линейная регрессионная модель, то алгоритм прямого ступенчатого моделирования минимизирует ошибку ансамбля в целом. Фактически при использовании обычной линейной регрессии как базовой модели для аддитивной регрессии сумма линейных регрессионных моделей также будет линейной регрессионной моделью, и регрессионный алгоритм сам минимизирует квадратичную ошибку.

Для удобства создание ступенчатой аддитивной регрессии обычно начинается с построения модели нулевого уровня, которая просто предсказывает среднее значение выхода на обучающем множестве, а каждая следующая модель подстраивает остатки.

Аддитивная регрессия подвержена переобучению, поскольку каждая модель, добавленная в ансамбль, делает его настройку все более точной. Чтобы определить момент остановки, используется ошибка на тестовом множестве, и выбирается то число итераций, которое ее минимизирует.

## Аддитивная логистическая регрессия

Аддитивная регрессия может применяться для решения задач бинарной классификации. Но поскольку для этого больше подходит логистическая регрессия, аддитивная модель должна быть соответствующим образом модифицирована. С помощью логит-преобразования производится оценка вероятности каждого класса, а затем задача регрессии решается с использованием ансамбля моделей, например регрессионных деревьев, так же, как и в случае аддитивной регрессии. На каждом шаге добавляется модель, которая максимизирует вероятность одного из классов.

Предположим, что  $f_j$  —  $j$ -я регрессионная модель в ансамбле, а  $f_j(a)$  — ее предсказание для примера  $a$ . Рассмотрим задачу бинарной классификации, используя аддитивную модель  $\sum_j f_j(a)$  для получения оценки вероятности первого класса:

$$P(1 | a) = \frac{1}{1 + \exp\left(-\sum_j f_j(a)\right)}.$$

Алгоритм бустинга, который использует в качестве базовой модели логистическую регрессию, получил название LogitBoost. Процесс генерации моделей этим алгоритмом можно представить в виде блок-схемы (рисунок 1). Здесь  $y_i = 1$  для примеров, относящихся к классу 1, и  $y_i = 0$  для примеров, относящихся к классу 2. На каждой итерации алгоритм подстраивает регрессионную модель  $f_j$  по взвешенной версии исходного множества данных.

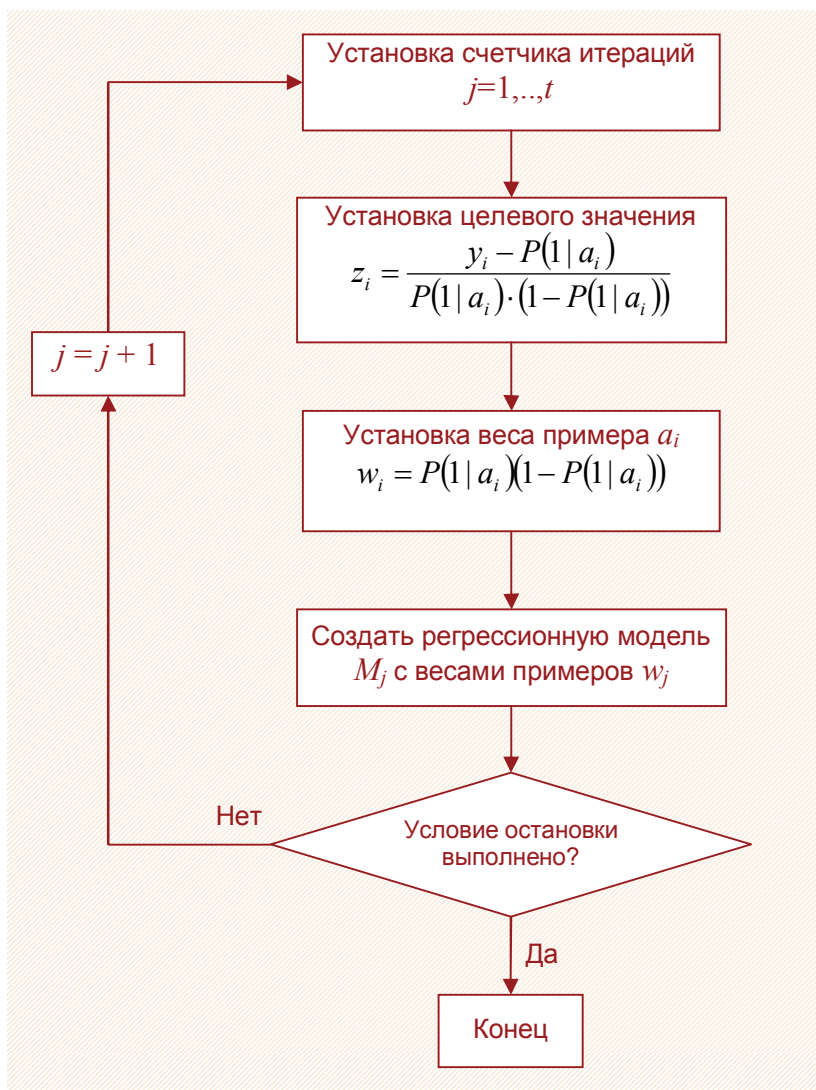


Рисунок 1 – Блок-схема алгоритма LogitBoost

Изначально алгоритм **LogitBoost** предназначен для решения задачи бинарной классификации, но при необходимости может быть обобщен на случай с несколькими классами.

## Деревья выбора

К недостаткам ансамблей моделей как сложных составных структур относится трудность интерпретации результатов. Решить эту проблему можно, сделав представление ансамбля более компактным. Одним из вариантов является использование структур, называемых **деревьями выбора** (option tree). Они отличаются от обычных деревьев решений тем, что содержат два типа узлов: узлы решений и узлы выбора. Простой пример такого дерева с одним узлом выбора представлен на рисунке 2. В узел решений, как и в обычном дереве решений, входит только одна ветвь, а в узел выбора — две. Таким образом, в узел выбора поступают решения из двух предыдущих узлов. Тогда выход узла выбора — это комбинация поступивших в него решений. Выбор может быть сделан обычным голосованием.

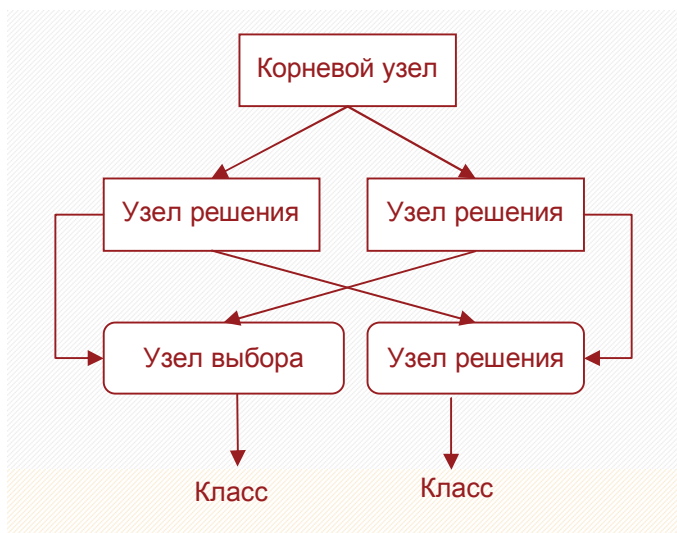


Рисунок 2 – Дерево выбора

Когда узел выбора содержит два входа и более, результат будет зависеть от того, являются ли они категориальными или числовыми. Если входы категориальные, то используется голосование, а если числовые — полученные различными способами вероятностные оценки, взвешенное среднее или байесовский подход.

Деревья выбора можно создавать с помощью модификации обычных деревьев решений путем встраивания в них узлов выбора.

Другая возможность построить дерево выбора — пошагово добавлять в него узлы. При этом используется алгоритм бустинга. Полученное дерево называется **знакопеременным деревом**, узлы решений — **узлами-разветвителями**, а узлы выбора — **узлами предсказания**. Узлы предсказания являются листьями, если из них больше не выходят узлы-разветвители. Обычные знакопеременные деревья применяются при решении задач бинарной классификации, и с каждым узлом предсказания ассоциировано положительное или отрицательное числовое значение. Чтобы предсказать класс определенного примера, его пропускают через все возможные ветви, после чего значения, полученные на выходах всех узлов предсказания, суммируются. Класс выбирается в зависимости от того, положительной или отрицательной оказалась сумма.

Схема знакопеременного дерева представлена на рисунке 3, где положительное значение соответствует классу Играть = Да, а отрицательное — классу Играть = Нет. Пример {Облачность = Солнечно, Температура = Жарко, Влажность = Нормальная, Ветер = Нет}, проходя по соответствующим листьям, получает значения  $-0,255$ ,  $0,213$ ,  $0,430$  и  $-0,331$ . После их сложения сумма положительная, значит, результат Играть = Да.



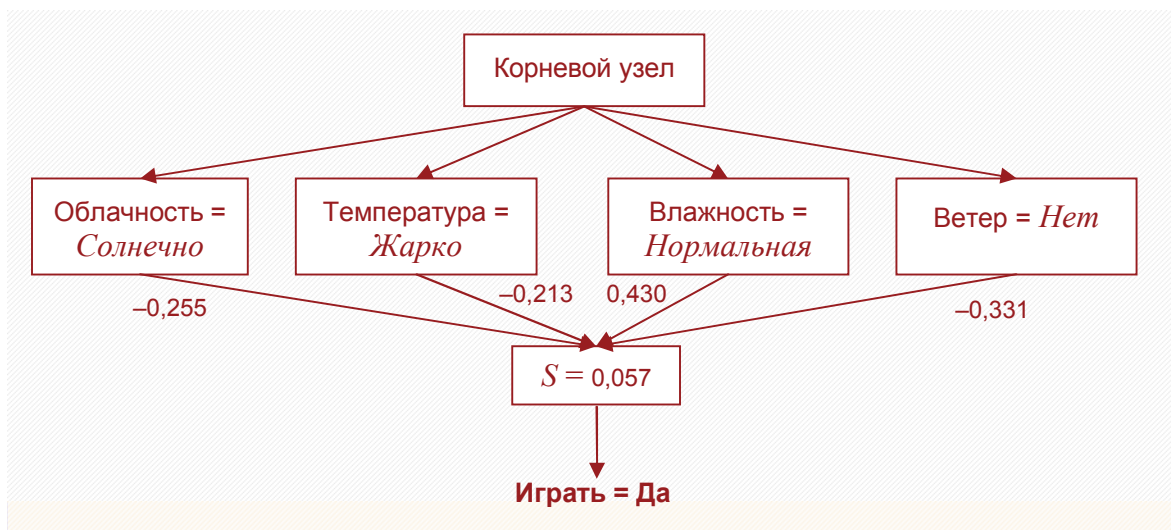


Рисунок 3 – Знакопеременное дерево

Дерево на рисунке 3 содержит четыре узла-разветвителя и один узел предсказания.

Знакопеременное дерево растет с использованием алгоритма бустинга, например **LogitBoost**. Предположим, что базовая модель создает отдельное правило на каждой итерации бустинга. После этого знакопеременное дерево может быть сгенерировано путем простого добавления каждого правила в дерево.

## Стэкинг и метаобучение

**Поуровневое обобщение** (stacked generalization), или просто **стэкинг** (stacking), — один из способов создания составных моделей. Хотя метод был разработан несколько лет назад, он все еще менее известен, чем бэггинг и бустинг. Отчасти это связано со сложностью теоретического анализа, а отчасти с тем, что общая концепция использования данного метода пока отсутствует — основная идея может применяться в самых разнообразных вариантах.

В отличие от бэггинга и бустинга стэкинг обычно не используется для комбинирования моделей одного и того же типа, например набора деревьев решений. Вместо этого он применяется к моделям, построенным с помощью различных алгоритмов обучения. Предположим, мы имеем три модели — дерево решений, байесовский классификатор и нейронную сеть, на основе которых нужно построить классификатор для определенного набора данных. Обычная процедура заключается в оценке ошибки каждого алгоритма и выборе лучшего из них, который и будет использоваться для классификации новых данных. Но действительно ли этот способ лучший? Если доступно три модели, то нельзя ли их все применять для классификации, комбинируя выходы?

Одной из техник комбинирования выходов является голосование (та же методика, что и в бэггинге). Но оно имеет смысл только тогда, когда обучающие схемы работают сравнительно хорошо. Если два из трех классификаторов работают плохо, то ожидать хороших результатов от всего ансамбля не приходится. Вместо этого стэкинг вводит концепцию **метаобучения**, которая заменяет процедуру голосования. Проблема голосования заключается в том, что неясно, какому классификатору следует доверять. Стэкинг пытается обучить каждый классификатор, используя алгоритм метаобучения, который позволяет обнаружить лучшую комбинацию выходов базовых моделей (рисунок 4).

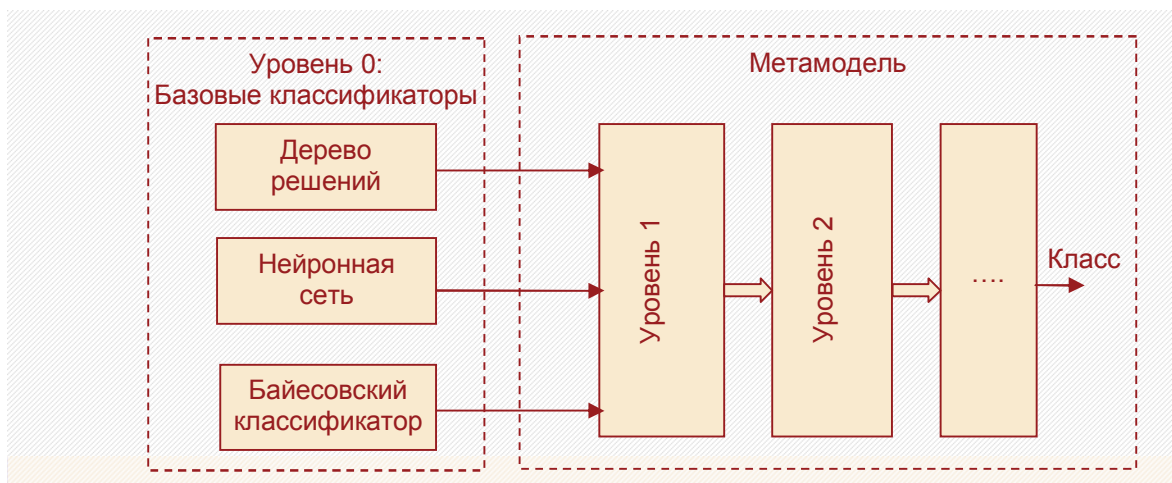


Рисунок 4 – Стэкинг

Базовые модели образуют уровень 0. На вход метамоделли, которая также называется моделью уровня 1, подаются результаты с выходов базовых моделей. Пример уровня 1 имеет столько атрибутов, сколько существует моделей уровня 0, а сами значения атрибутов — выходы моделей нулевого уровня. Затем на основе результатов, полученных моделью уровня 1, строится модель уровня 2 и т. д., пока не будет выполнено какое-либо из условий остановки обучения.