Глава 11

Ансамбли моделей

Одна голова хорошо, а две лучше – согласно этой известной пословице, два человека, работающие вместе, зачастую добиваются лучших результатов. Если заменить «голова» на «признак», то эта пословица будет в полной мере применима и к машинному обучению, в чем мы имели возможность убедиться в предыдущих главах. Но можно улучшить результаты еще сильнее, если комбинировать не просто признаки, а целые модели – это мы и продемонстрируем в данной главе. Комбинации моделей часто называют *ансамблями моделей*. Это один из самых мощных методов машинного обучения, нередко превосходящий другие по качеству и эффективности. Правда, за это приходится расплачиваться увеличением сложности алгоритмов и моделей.

Тема комбинирования моделей имеет богатую и разнообразную историю, которой мы можем посвятить лишь немного времени в этой короткой главе. Своими корнями она уходит в вычислительную теорию обучения, с одной стороны, и в статистику – с другой. В статистике хорошо известно интуитивное соображение, согласно которому усреднение результатов измерений может дать более устойчивую и надежную оценку, поскольку сокращается влияние случайных флуктуаций в отдельном измерении. Поэтому если бы удалось построить ансамбль немного различающихся моделей по одним и тем же обучающим данным, то мы смогли бы уменьшить влияние случайных флуктуаций в отдельных моделях. Ключевой вопрос – как обеспечить необходимое разнообразие моделей? Как мы увидим, зачастую этого можно добиться путем обучения моделей на случайно выбранных подмножествах данных и даже путем их конструирования из случайно выбранных подмножеств имеющихся признаков.

Из теории вычислительного обучения заимствован следующий ход рассуждений. Как мы видели в разделе 4.4, обучаемость языков гипотез исследовалась в контексте модели обучения, которая и определяет, что понимать под обучаемостью. PAC-обучаемость предполагает, что гипотеза почти правильна в большинстве случаев. Альтернативная модель, известная под названием *слабая обучаемость*, требует только, чтобы обученная гипотеза была лишь немного лучше случайной. И хотя кажется очевидным, что PAC-обучаемость сильнее, чем слабая обучаемость, оказывается, что на самом деле обе модели обучения эквивалентны: язык гипотез PAC-обучаем тогда и только тогда, когда он слабо обучаем. Это было доказано конструктивно – с помощью итеративного алгоритма, который повторяет конструирование гипотезы, имеющей целью исправление ошибок предыдущей гипотезы, и тем самым «усиливает» ее. Окончательная

модель комбинирует гипотезы, обученные на каждой итерации, и потому определяет ансамбль.

По существу, у всех методов построения ансамблей в машинном обучении есть две общие черты:

- они конструируют несколько различающихся прогностических моделей из адаптированных вариантов обучающих данных (чаще всего с помощью изменения весов или повторной выборки);
- они каким-то образом комбинируют предсказания этих моделей, часто с помощью простого усреднения или голосования (возможно, взвешенного).

Однако необходимо подчеркнуть, что, несмотря на наличие этих общих черт, разнообразие применяемых методов очень велико, и не следует удивляться тому, что некоторые методы на практике сильно различаются, пусть даже и обладают поверхностным сходством. Например, очень многое зависит от того, учитываются ли при адаптации обучающих данных для следующей итерации предсказания предыдущих моделей. Из всего разнообразия мы рассмотрим два наиболее известных метода построения ансамблей: баггинг (раздел 11.1) и усиление (раздел 11.2). Далее в разделе 11.3 мы кратко обсудим эти и родственные им методы построения ансамблей и завершим главу обычным подведением итогов и рекомендацией литературы для дальнейшего чтения.

11.1 Баггинг и случайные леса

Баггинг (сокращение от «bootstrap aggregating») – простой, но весьма эффективный метод создания различающихся моделей на основе различных случайных выборок из исходного набора данных. Выборки производятся равномерно с заменой и называются *усиливающими выборками* (bootstrap samples). Поскольку выборка производится с заменой, то в общем случае усиливающая выборка содержит дубликаты, и потому некоторые исходные примеры будут отсутствовать, даже если размер усиливающей выборки такой же, как размер исходного набора. Чтобы понять, насколько различными могут быть усиливающие выборки, заметим, что вероятность того, что конкретный пример не входит в усиливающую выборку размера n, равна $(1 - 1/n)^n$; при $n = 5$ она равна примерно $1/3$ и стремится к $1/e = 0.368$, когда $n \to \infty$. Это означает, что в каждой усиливающей выборке отсутствует примерно треть исходных данных.

В алгоритме 11.1 описан базовый алгоритм баггинга, который возвращает ансамбль в виде множества моделей. Для комбинирования предсказаний моделей мы можем воспользоваться голосованием – выигрывает класс, предсказанный большинством моделей, – или усреднением, которое подходит лучше, если базовые классификаторы возвращают оценки или вероятности. На рис. 11.1 приведена иллюстрация. Я обучил пять базовых линейных классификаторов на усиливающих выборках из набора, содержащего по 20 положительных и отри-

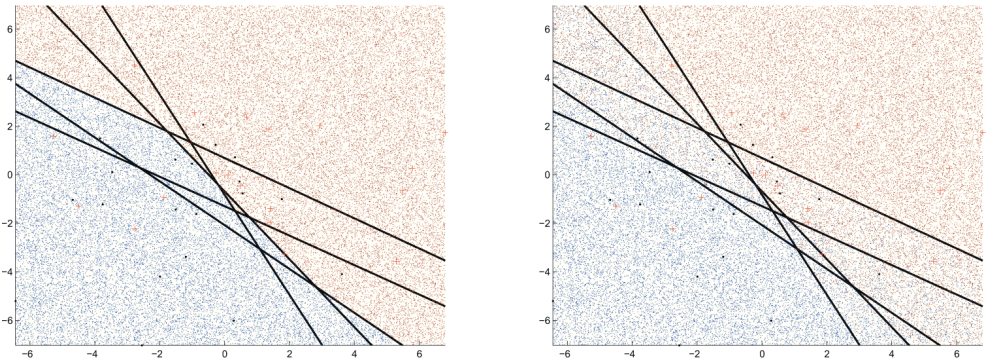

Рис. 11.1. (Слева) Ансамбль из пяти базовых линейных классификаторов, построенный из усиливающих выборок с помощью баггинга. Решающим правилом является большинство голосов, оно порождает кусочно-линейную решающую границу. **(Справа)** Если преобразовать голоса в вероятности, то ансамбль превращается в группирующую модель: каждый сегмент пространства объектов получает немного отличающуюся вероятность

цательных примеров. Отчетливо видно, насколько различны классификаторы, этому способствует также то, что набор данных очень мал. Рисунок демонстрирует различие между комбинированием предсказаний путем голосования (слева) и создания вероятностного классификатора (справа). В случае голосования баггинг порождает кусочно-линейную решающую границу – вещь, невозможную для одного линейного классификатора. Если преобразовать голоса моделей в оценки вероятностей, то мы увидим, что различные решающие границы разбивают пространство объектов на сегменты, каждый из которых потенциально может получить различные оценки.

Алгоритм 11.1. Bagging($D, T, \mathscr{A}$) – обучить ансамбль моделей на усиливающих выборках

Вход: набор данных D; размер ансамбля T; алгоритм обучения $\mathscr{A}$.
Выход: ансамбль моделей, предсказания которых необходимо скомбинировать путем голосования или усреднения.

1 **for** $t = 1$ до T **do**
2 | построить усиливающую выборку D_t из D, выбрав $|D|$ примеров с заменой;
3 | прогнать $\mathscr{A}$ на D_t и получить в результате модель M_t;
4 **end**
5 **return** $\{M_t \,|\, 1 \le t \le T\}$

Баггинг особенно полезен в сочетании с древовидными моделями, которые очень чувствительны к небольшому изменению обучающих данных. В применении к древовидным моделям баггинг нередко сочетается еще с одной идеей: строить каждое дерево по разным случайно выбранным подмножествам признаков; этот процесс называется *выборкой подпространства* (subspace sampling). В результате разнообразие ансамбля еще повышается, и в качестве дополнитель-

ного бонуса мы получаем уменьшение времени обучения каждого дерева. Получающийся ансамбль называется *случайным лесом*, а соответствующий алгоритм описан в алгоритме 11.2.

Алгоритм 11.2. RandomForest(D, T, d) – обучить ансамбль древовидных моделей на усиливающих выборках и случайных подпространствах

Вход: набор данных D; размер ансамбля T; размерность подпространства d.
Выход: ансамбль древовидных моделей, предсказания которых необходимо скомбинировать путем голосования или усреднения.

```
for t = 1 до T do
    построить усиливающую выборку D_t из D, выбрав |D| примеров с заменой;
    случайным образом выбрать d признаков и соответственно уменьшить размерность D_t;
    обучить древовидную модель M_t по D_t без редукции;
end
return {M_t | 1 ≤ t ≤ T}
```

Поскольку решающее дерево – это группирующая модель, а его листья образуют разбиение пространства объектов, то таковым является и случайный лес: соответствующее ему разбиение пространства объектов представляет собой пересечение разбиений, образуемых входящими в ансамбль деревьями. Хотя разбиение, образуемое случайным лесом, мельче разбиений, образуемых большинством деревьев, его, в принципе, можно отобразить обратно на одиночную древовидную модель (потому что пересечение соответствует комбинированию ветвей двух разных деревьев). В этом состоит отличие от баггинга линейных классификаторов, при котором решающая граница ансамбля не может быть получена в результате обучения одного базового классификатора. Таким образом, можно сказать, что алгоритм построения случайного леса в случае древовидных моделей – это реализация альтернативного алгоритма обучения.

11.2 Усиление

Усиление, или бустинг (boosting), – техника построения ансамблей, на первый взгляд, похожая на баггинг, однако для внесения разнообразия в обучающие наборы в ней используются более изощренные методы. Основная идея проста и соблазнительна. Допустим, что мы обучили линейный классификатор на некотором наборе данных и обнаружили, что частота ошибок обучения равна ϵ. Мы хотим добавить в ансамбль еще один классификатор, который ошибается реже, чем первый. Добиться этого можно, например, продублировав неправильно классифицированные объекты: если наша модель представляет собой базовый линейный классификатор, то это приведет к смещению средних значений классов в сторону дубликатов. Еще лучше было бы сопоставить неправильно классифицированным объектам больший вес и изменить классификатор, так чтобы он

принимал веса во внимание (например, базовый линейный классификатор умеет вычислять средние значения классов как средневзвешенные).

Но насколько следует изменить вес? Идея в том, чтобы половину общего веса отдать неправильно классифицированным примерам, а другую половину – всем остальным. Поскольку мы начинали с одинаковых весов, которые в сумме дают 1, то текущий вес, приходящийся на долю неправильно классифицированных примеров, в точности равен частоте ошибок ϵ, поэтому мы умножаем их веса на $1/2\epsilon$. В предположении, что $\epsilon < 0.5$, это и будет желаемым увеличением веса. Веса правильно классифицированных примеров умножаются на $1/2(1-\epsilon)$, так чтобы сумма подправленных весов осталась равной 1. В следующем раунде мы делаем то же самое, только при вычислении частоты ошибок учитываем неодинаковые веса.

Пример 11.1 (изменение весов в методе усиления). Предположим, что качество линейного классификатора такое, как в таблице сопряженности слева. Частота ошибок равна $\epsilon = (9 + 16)/100 = 0.25$. Веса неправильно классифицированных примеров умножаются на $1/2\epsilon = 2$, а правильно классифицированных – на $1/2(1-\epsilon) = 2/3$.

	Предсказано ⊕	*Предсказано* ⊖	
Фактически ⊕	**24**	**16**	40
Фактически ⊖	**9**	**51**	60
	33	67	100

	⊕	⊖	
⊕	**16**	**32**	48
⊖	**18**	**34**	52
	24	66	100

С учетом измененных весов получаем таблицу сопряженности справа, в которой взвешенная частота ошибки равна 0.5.

В алгоритме усиления необходим еще один компонент – коэффициент доверия α для каждой модели в ансамбле; им мы воспользуемся для формирования ансамблевого предсказания, равного взвешенному среднему отдельных моделей. Понятно, что нам хотелось бы, что α увеличивался при уменьшении ϵ: обычно его вычисляют по формуле

$$\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t} = \ln\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}, \qquad (11.1)$$

обоснование которой мы приведем чуть ниже. Базовый алгоритм усиления приведен в алгоритме 11.3. На рис. 11.2 слева показано, как усиленный ансамбль из пяти базовых линейных классификаторов может достичь нулевой ошибки обучения. Очевидно, что получившаяся решающая граница гораздо сложнее той, что может построить один базовый линейный классификатор. Напротив, ансамбль из пяти базовых линейных классификаторов, построенный методом баггинга, дал пять очень похожих решающих границ, и объясняется это тем, что усиленные выборки были очень похожи.

Алгоритм 11.3. Boosting(D, T, $\mathscr{A}$) – обучить ансамбль бинарных классификаторов по обучающим наборам с пересчитываемыми весами

Вход: набор данных D; размер ансамбля T; алгоритм обучения $\mathscr{A}$.
Выход: взвешенный ансамбль моделей.

$w_{1i} \leftarrow 1/|D|$ для всех $x_i \in D$; // начинаем с одинаковых весов
for $t = 1$ до T **do**
  прогнать $\mathscr{A}$ на D с весами w_{ti}, получив в результате модель M_t;
  вычислить взвешенную ошибку ϵ_t;
  if $\epsilon_t \geq 1/2$ **then**
    положить $T \leftarrow t-1$ и выйти из цикла
  end
  $\alpha_t \leftarrow \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$; // доверие для этой модели
  $w_{(t+1)i} \leftarrow w_{ti}/2\epsilon_t$ для неправильно классифицированных $x_i \in D$; // увеличить вес
  $w_{(t+1)j} \leftarrow w_{tj}/2(1-\epsilon_t)$ для правильно классифицированных $x_j \in D$; // уменьшить вес
end
return $M(x) = \Sigma_{t=1}^{T}\alpha_t M_t(x)$

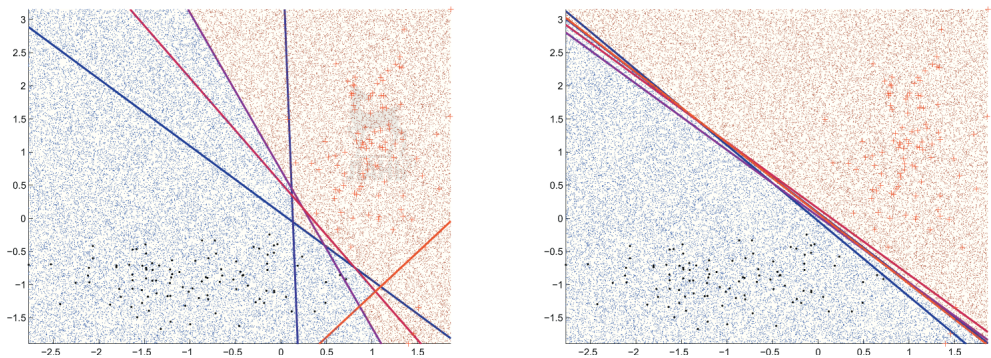

Рис. 11.2. (Слева) Ансамбль из пяти усиленных базовых линейных классификаторов с мажоритарным голосованием. Линейные классификаторы обучались в порядке от **синего** до **красного**, ни один из них в отдельности не достигает нулевой ошибки обучения, а ансамбль достигает. **(Справа)** Применение баггинга приводит к гораздо более однородному ансамблю, доказывая, что усиленные выборки различаются слабо

Перейду к вопросу о том, чем объясняется выбор коэффициента α_t в уравнении (11.1). Во-первых, я покажу, что модификации веса для неправильно и правильно классифицированных примеров можно записать в виде взаимно обратных членов δ_t и $1/\delta_t$, нормированных на некоторую величину Z_t:

$$\frac{1}{2\epsilon_t} = \frac{\delta_t}{Z_t}; \quad \frac{1}{2(1-\epsilon_t)} = \frac{1/\delta_t}{Z_t}.$$

Второе выражение дает $\delta_t = 2(1 - \epsilon_t)/Z_t$; подставляя это в первое выражение, получаем:

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}; \quad \delta_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = \exp(\alpha_t). \tag{11.2}$$

Таким образом, модификация веса для неправильно классифицированных примеров составляет $\exp(\alpha_t)/Z_t$, а для правильно классифицированных $\exp(-\alpha_t)/Z_t$. Пользуясь тем фактом, что $y_i M_t(x_i) = +1$ для объектов, правильно классифицированных моделью Mt, и -1 в противном случае, мы можем записать модификацию веса в виде:

$$w_{(t+1)i} = w_{ti}\frac{\exp(-\alpha_t y_i M_t(x_i))}{Z_t},$$

и именно это выражение часто встречается в литературе.

Теперь вернемся на шаг назад и сделаем вид, что мы еще не решили, какими должны быть значения α_t в каждом раунде. Так как модификации весов мультипликативны, то имеем

$$w_{(T+1)i} = w_{1i}\prod_{t=1}^{T}\frac{\exp(-\alpha_t y_i M_t(x_i))}{Z_t} = \frac{1}{|D|}\frac{\exp(-y_i M(x_i))}{\prod_{t=1}^{T} Z_t},$$

где $M(x_i) = \Sigma_{t=1}^{T}\alpha_t M_t(x_i)$ – модель, представленная усиленным ансамблем. Сумма весов по всему пространству объектов всегда равна 1, поэтому

$$\prod_{t=1}^{T} Z_t = \frac{1}{|D|}\sum_{i=1}^{|D|}\exp(-y_i M(x_i)).$$

Отметим, что $\exp(-y_i M(x_i))$ всегда положительно и не меньше 1, если $-y_i M(x_i)$ положительно, как бывает, когда объект x_i неправильно классифицирован ансамблем (то есть $\text{sign}(M(x_i)) \neq y_i$). Таким образом, правая часть этого выражения не меньше ошибки обучения усиленного ансамбля, и $\prod_{t=1}^{T} Z_t$ – верхняя граница ошибки обучения. Следовательно, в качестве простой эвристики можно взять жадную минимизацию величины

$$Z_t = \sum_{i=1}^{|D|} w_{ti}\exp(-\alpha_t y_i M_t(x_i)) \tag{11.3}$$

в каждом раунде усиления. Сумма весов объектов, неправильно классифицированных моделью M_t, равна ϵ_t, и потому

$$Z_t = \epsilon_t \exp(\alpha_t) + (1 - \epsilon_t)\exp(-\alpha_t).$$

Приравнивая к нулю производную α_t и решая получившееся уравнение относительно α_t, находим α_t в виде, приведенном в формуле (11.1), и Z_t – в виде (11.2).

Отметим, что из уравнения (11.3) следует, что функция потерь, минимизированная алгоритмом усиления, представляет собой ☞ *экспоненциальную функцию потерь* $\exp(-y\hat{s}(x))$, с которой мы уже встречались на рис. 2.6 на стр. 77. Отметим далее, что минимизация Z_t в соответствии с (11.2) означает также минимизацию $2\sqrt{\epsilon_t(1-\epsilon_t)}$. Вероятно, вы узнаете в этом выражении ☞ *меру нечистоты Джини*, которую мы изучали в главе 5. Там мы видели, что этот критерий разделения нечувствителен к изменениям в распределении по классам (см. рис. 5.7 на стр. 159 Здесь он возникает в основном из-за способа модификации весов в алгоритме усиления.

Обучение усиленных правил

Интересный вариант усиления возникает, когда наши базовые модели являются частичными классификаторами, которые иногда воздерживаются от предсказания. Предположим, к примеру, что наши базовые классификаторы – это конъюнктивные правила, заголовок которых фиксирован и всегда предсказывает положительный класс. Тогда отдельное правило либо предсказывает положительный класс для тех объектов, которые покрывает, либо воздерживается от предсказания. Мы можем воспользоваться усилением, чтобы обучить ансамбль таких правил, который возвращает взвешенный результат голосования среди своих членов.

В уравнения усиления необходимо внести показанные ниже небольшие коррективы. Отметим, что ϵ_t – взвешенная ошибка t-го базового классификатора. Поскольку наши правила всегда предсказывают положительный класс для покрытых объектов, эти ошибки относятся только к покрытым отрицательным объектам, которые мы будем обозначать $\epsilon_t^{\ominus}$. Аналогично взвешенную сумму покрытых положительных объектов обозначим $\epsilon_t^{\oplus}$, она будет играть ту же роль, что $1-\epsilon_t$. Однако при наличии воздерживающихся правил есть еще третий компонент, обозначаемый ϵ_t^0, – взвешенная сумма объектов, не покрытых правилом ($\epsilon_t^0 + \epsilon_t^{\ominus} + \epsilon_t^{\oplus} = 1$). Тогда имеем:

$$Z_t = \epsilon_t^0 + \epsilon_t^{\ominus}\exp(\alpha_t) + \epsilon_t^{\oplus}\exp(-\alpha_t).$$

Это выражение достигает максимума при следующем значении α_t:

$$\alpha_t = \frac{1}{2}\ln\frac{\epsilon_t^{\oplus}}{\epsilon_t^{\ominus}} = \ln\sqrt{\frac{\epsilon_t^{\oplus}}{\epsilon_t^{\ominus}}}, \tag{11.4}$$

откуда получаем

$$Z_t = \epsilon_t^0 = 2\sqrt{\epsilon_t^{\oplus}\epsilon_t^{\ominus}} = 1-\epsilon_t^{\oplus}-\epsilon_t^{\ominus}+2\sqrt{\epsilon_t^{\oplus}\epsilon_t^{\ominus}} = 1-\left(\sqrt{\epsilon_t^{\oplus}}-\sqrt{\epsilon_t^{\ominus}}\right)^2.$$

Это означает, что в каждом раунде усиления мы конструируем правило, которое максимизирует величину $\left|\sqrt{\epsilon_t^{\oplus}}-\sqrt{\epsilon_t^{\ominus}}\right|$, и задаем для него коэффициент доверия α_t,

показанный в формуле (11.4). Чтобы получить от ансамбля предсказание для некоторого объекта, мы складываем коэффициенты доверия всех покрывающих его правил. Отметим, что эти коэффициенты доверия отрицательны, если $\epsilon_t^{\oplus} < \epsilon_t^{\ominus}$, откуда следует, что правило коррелирует с отрицательным классом; само по себе это не проблема, но тем не менее корреляцию можно устранить, изменив целевую функцию для отдельных правил на $\sqrt{\epsilon_t^{\oplus}} - \sqrt{\epsilon_t^{\ominus}}$.

Модификации весов после каждой итерации усиления такие же, как и раньше, с тем отличием, что веса примеров, не покрытых правилом, не изменяются. Таким образом, обучение усиленных правил аналогично алгоритму построения ☞ *взвешенного покрытия* (алгоритм 6.5 на стр. 194) для выявления подгрупп. Различие в том, что там мы хотели поощрить перекрытие правил безотносительно к классу и потому уменьшали веса всех покрытых примеров, тогда как здесь мы уменьшаем веса покрытых положительных примеров и увеличиваем веса покрытых отрицательных примеров.

11.3 Карта ансамблевого ландшафта

Теперь, после того как мы ближе познакомились с часто используемыми методами построения ансамблей, поговорим о том, чем объяснить различия в их качестве, а затем уже обратимся к некоторым из многочисленных ансамблевых методов, описанных в литературе.

Смещение, дисперсия и зазоры

Методы построения ансамблей – хороший способ лучше разобраться в ☞ *дилемме смещения-дисперсии*, которую мы рассматривали в разделе 3.2 в контексте регрессии. Вообще говоря, есть три причины неверной классификации тестового примера моделью. Во-первых, это просто неизбежно, если в данном пространстве признаков объекты разных классов описываются одинаковыми признаками. В вероятностном контексте это случается, когда условные распределения $P(X|Y)$ перекрываются, так что у некоторого объекта вероятности принадлежности к нескольким классам отличны от нуля. В такой ситуации лучшее, на что можно надеяться, – это аппроксимация целевого концепта.

Вторая причина ошибок классификации – недостаточная выразительность модели для представления целевого концепта. Например, если данные не являются линейно разделимыми, то даже самый лучший линейный классификатор будет делать ошибки. Это смещение классификатора, между ним и выразительностью существует обратная зависимость. И хотя не существует общепринятого способа измерения выразительности или смещения классификатора[1], интуитивно понят-

[1] Хотя квадратичная потеря хорошо раскладывается на квадратичное смещение и дисперсию, как видно из уравнения (3.2) на стр. 107, функции потерь, применяемые в классификации, например функция потерь типа 0–1, допускают различные разложения.

но, что, скажем, у гиперболической решающей границы смещение меньше, чем у линейной. Ясно также, что у древовидных моделей смещение наименьшее из возможных, поскольку их листовые узлы можно сделать настолько мелкими, что они будут покрывать только по одному объекту.

Может показаться, что модели с низким смещением в общем случае предпочтительнее. Однако в практике применения машинного обучения действует эвристическое правило: *у моделей с низким смещением обычно высокая дисперсия, и наоборот*. Дисперсия – это третий источник ошибок классификации. Модель имеет высокую дисперсию, если ее решающая граница сильно зависит от обучающих данных. Например, в случае классификатора по ближайшему соседу сегменты пространства объектов определяются одной точкой обучающего набора, поэтому если я сдвину точку в сегменте, примыкающем к решающей границе, то сдвинется и сама граница. У древовидных моделей дисперсия высока по другой причине: если изменить обучающие данные настолько сильно, что в корне дерева изменится выбранный для разделения признак, то, скорее всего, и все дерево станет другим. Примером модели с низкой дисперсией может служить базовый линейный классификатор, поскольку он производит усреднение по всем точкам класса.

Взгляните теперь на рис. 11.1. Ансамбль базовых линейных классификаторов, построенный методом баггинга, обучил кусочно-линейную решающую границу, которая по выразительности превосходит любой отдельно взятый линейный классификатор. Это говорит о том, что баггинг, как и всякий ансамблевый метод, способен уменьшить смещение базовой модели, которая изначально характеризуется высоким смещением, например линейного классификатора. Однако если сравнить это с результатами метода усиления, показанными на рис. 11.2, то мы увидим, что уменьшение смещения, получающееся в результате баггинга, гораздо меньше, чем в результате усиления. Фактически *баггинг – прежде всего техника уменьшения дисперсии, тогда как усиление – преимущественно метод уменьшения смещения*. Это объясняет, почему баггинг часто применяется в сочетании с моделями, имеющими высокую дисперсию, например древовидными (☞*случайные леса* в алгоритме 11.2), а усиление – с моделями, имеющими высокое смещение, например линейными классификаторами или одномерными решающими деревьями (которые называются также *решающими пнями*).

По-другому усиление можно интерпретировать в терминах зазоров. Интуитивно представляется, что зазор – это расстояние со знаком до решающей границы, причем знак показывает, по правильную ли сторону от границы мы оказались. Экспериментально было отмечено, что усиление дает хороший эффект в плане увеличения зазоров примеров, даже если они уже расположены по правильную сторону от решающей границы. Улучшение качества на тестовом наборе в результате усиления может продолжаться даже после того, как ошибка обучения свелась к нулю. Если учесть, что усиление первоначально возникло в контексте PAC-обучения, не рассчитанном на увеличение зазоров, то этот результат покажется удивительным.

Другие ансамблевые методы

Существует много других методов построения ансамблей, помимо баггинга и усиления. Основные отличия связаны с тем, как комбинируются предсказания базовых моделей. Отметим, что этот вопрос и сам по себе можно было бы определить как проблему обучения: рассматривая предсказания некоторых базовых классификаторов как признаки, обучить *метамодель*, которая будет комбинировать их наилучшим способом. Например, в методе усиления мы могли бы обучить веса α_t, а не выводить их из частот ошибок каждой отдельной базовой модели. Обучение линейной метамодели называется *укладкой* (stacking). Существует несколько вариаций на эту тему, например в качестве метамоделей применялись решающие деревья. Можно также комбинировать различные базовые модели в гетерогенный ансамбль, при этом разнообразие достигается за счет того, что базовые модели обучаются различными алгоритмами, поэтому можно использовать один и тот же обучающий набор. Для некоторых базовых моделей можно задавать различные параметры, например ансамбль может включать несколько машин опорных векторов с разными значениями параметра сложности, который определяет, в какой мере терпимы ошибки зазора.

Итак, в общем случае ансамбль моделей состоит из множества базовых моделей и метамодели, которая обучена решать, как именно следует комбинировать предсказания базовых моделей. Обучение метамодели неявно подразумевает оценку качества каждой базовой модели, например если метамодель линейна, как в случае укладки, то вес, близкий к нулю, означает, что соответствующий базовый классификатор не дает большого вклада в ансамбль. Можно даже допустить, что базовый классификатор получает отрицательный вес, тогда в контексте остальных базовых моделей его предсказания следует инвертировать. Можно было бы пойти еще дальше и попытаться *предсказать* ожидаемое качество базовой модели еще до ее обучения! Сформулировав это как проблему обучения на метауровне, мы вступаем в область метаобучения.

Метаобучение

Прежде всего метаобучение подразумевает обучение ряда моделей на большой совокупности наборов данных. Цель состоит в том, чтобы сконструировать модель, которая поможет отвечать на такие вопросы:

- в каких случаях решающее дерево, скорее всего, окажется лучше метода опорных векторов?
- когда от линейного классификатора следует ожидать плохого качества?
- можно ли использовать данные для рекомендования конкретных параметров?

Ключевой вопрос метаобучения – как спроектировать признаки, на основе которых строится метамодель? Эти признаки должны сочетать в себе характеристики набора данных и релевантных аспектов обученной модели. Характеристики набора данных отнюдь не должны ограничиваться простым перечислением

количества и вида признаков и количества объектов, потому что маловероятно, что на основе только лишь этой информации можно будет предсказать что-то содержательное о качестве модели. Например, мы можем попытаться оценить уровень зашумленности данных, измерив размер обученного решающего дерева до и после редукции. Обучение на наборе данных простых моделей типа решающих пней и последующее измерение их качества также дает полезную информацию.

В замечании 1.1 на стр. 31 мы упоминали теорему о бесплатных завтраках, которая утверждает, что никакой алгоритм обучения не может быть лучше всех остальных алгоритмов обучения на множестве всех возможных проблем обучения. Отсюда следует, что попытка метаобучения на всех возможных проблемах обучения тщетна, – иначе мы могли бы построить одну гибридную модель, которая использует метамодель, чтобы сообщить, у какой базовой модели качество будет превосходить случайное на конкретном наборе данных. А это означает, что мы можем лишь надеяться, что метаобучение окажется полезным на проблемах обучения с неравномерным распределением.

11.4 Ансамбли моделей: итоги и литература для дальнейшего чтения

В этой короткой главе мы обсудили некоторые фундаментальные идеи методов построения ансамблей. У всех таких методов есть общая черта: они строят несколько базовых моделей на основе модифицированных обучающих данных, а затем применяют тот или иной способ комбинирования предсказаний или оценок отдельных базовых моделей для получения предсказания всего ансамбля. Мы рассмотрели два широко распространенных ансамблевых метода: баггинг и усиление. Хорошее введение в ансамбли моделей имеется в работе Brown (2010). Стандартный справочник по комбинированию классификаторов – работа Kuncheva (2004), а более современный обзор имеется в работе Zhou (2012).

- В разделе 11.1 обсудили баггинг и случайные леса. По методу баггинга на выборках из обучающих данных обучаются различающиеся модели, он впервые был описан в работе Breiman (1996a). Случайные леса, обычно приписываемые работе Breiman (2001), объединяют обученные по методу баггинга решающие деревья со случайными подпространствами; аналогичные идеи были развиты в работах Ho (1995) и Amit, Geman (1997). Эти методы особенно полезны для уменьшения дисперсии моделей с низким смещением, например решающих деревьев.
- В разделе 11.2 обсуждался метод усиления. Его основная идея – обучить различающиеся модели путем увеличения веса примеров, которые раньше были классифицированы неправильно. Это позволяет уменьшить смещение устойчивых методов обучения, каковыми являются, например, линейные классификаторы и решающие пни. Доступный обзор данной тематики имеется в работе Schapire (2003). В работах Kearns, Valiant (1989, 1994)

был поставлен вопрос, верно ли, что слабый алгоритм обучения, который работает лишь немного лучше случайного угадывания, можно усилить до алгоритма обучения с произвольно высокой точностью. В работе Schapire (1990) дано теоретическое определение усиления, позволившее доказать эквивалентность слабой и сильной обучаемости. Алгоритм AdaBoost, лежащий в основе алгоритма 11.3, впервые был обнародован в работе Freund, Schapire (1997). В работе Schapire, Singer (1999) приведены обобщения AdaBoost на случаи нескольких классов и нескольких меток. Ранжирующий вариант AdaBoost был предложен в работе Freund et al. (2003). Подход к обучению усиленных правил, допускающий классификаторы, которые могут воздерживаться от предсказания, основан на алгоритме Slipper (Cohen, Singer, 1999), усиленном варианте алгоритма Ripper (Cohen, 1995).

☞ В разделе 11.3 мы обсудили баггинг и усиление с точки зрения смещения и дисперсии. В работе Schapire, Freund, Bartlett, Lee (1998) приведен детальный теоретический и экспериментальный анализ усиления в терминах улучшения маргинального распределения. Я упомянул также некоторые другие ансамблевые методы, предназначенные для обучения метамодели, комбинирующей результаты базовых моделей. В методе укладки применяется линейная метамодель, он был введен в работе Wolpert (1992) для классификации и распространен на регрессию в работе Breiman (1996b). Решающие метадеревья впервые описаны в работе Todorovski, Dzeroski (2003).

☞ Мы также обсудили метаобучение – технику, позволяющую предсказывать качество алгоритмов обучения. Эта направление зародилось в раннем эмпирическом исследовании, документированном в работе Michie et al. (1994). Из недавних работ следует отметить Brazdil et al. (2009, 2010). Редуцированные и нередуцированные решающие деревья использовались для получения характеристик наборов данных в работе Peng et al. (2002). Идея обучения простых моделей для получения дополнительных характеристик данных известна под названием межевания (landmarking) (Pfahringer et al. 2000).