ЛЕКЦИЯ 6. Интеллектуальный анализ данных: дерево решений

Деревья решений (decision trees) относятся к числу самых популярных и мощных инструментов Data Mining, позволяющих эффективно решать задачи классификации и регрессии. В отличие от методов, использующих статистический подход, таких как классификатор Байеса, линейная и логистическая регрессия, деревья решений основаны на машинном обучении и в большинстве случаев не требуют предположений о статистическом распределении значений признаков. В основе деревьев решений лежат решающие правила вида «если... то...», которые могут быть сформулированы на естественном языке. Поэтому деревья решений являются наиболее наглядными и легко интерпретируемыми моделями.

Для удобства приведем базовые понятия теории деревьев решений в табл. 1.1.

Таблица 1.1. Понятия, встречающиеся в теории деревьев решений

Название	Описание
Объект	Пример, шаблон, наблюдение, запись
Атрибут	Признак, независимая переменная, свойство, входное поле
Метка класса	Зависимая переменная, целевая переменная, выходное поле
Узел	Внутренний узел дерева
Лист	Конечный узел дерева, узел решения
Проверка	Условие в узле

Атрибутами в теории деревьев решений называются признаки, описывающие классифицируемые объекты. В основе работы деревьев решений лежит процесс рекурсивного разбиения исходного множества наблюдений или объектов на подмножества, ассоциированные с классами. Разбиение производится с помощью решающих правил, в которых осуществляется проверка значений атрибутов по заданному условию. Рекурсивными называются алгоритмы, которые работают в пошаговом режиме, при этом на каждом последующем шаге используются результаты, полученные на предыдущем шаге. Рассмотрим главную идею алгоритмов построения деревьев решений на примере, Пусть требуется предсказать возврат или невозврат кредита с помощью набора решающих правил на основе единственного атрибута *Возраст клиента*. Для этого будем использовать множество наблюдений, в каждом из которых указывается возраст, а также факт возврата/невозврата кредита, графически такое множество наблюдений представлено на рис. 1.1. Условно примем, что объект в форме круга указывает на дефолт, в форме прямоугольника на возврат по кредиту, а внутри каждого объекта указан возраст. Необходимо разбить множество объектов на

подмножества таким образом, чтобы в каждое из них попали объекты только одного класса.

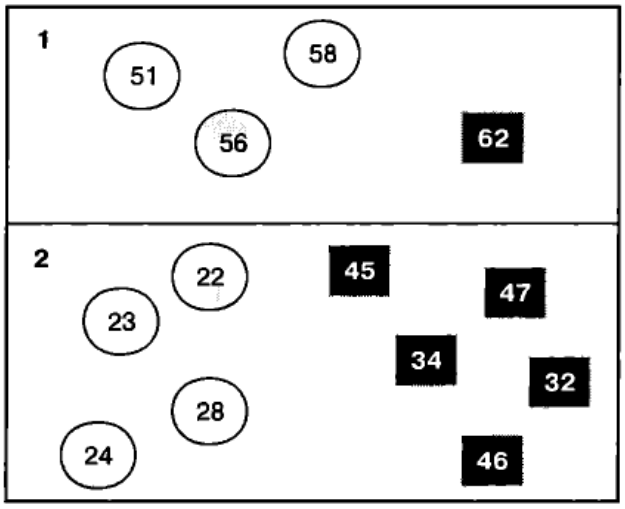


Рис. 1.1. Разделение на классы

Выберем некоторое значение возрастного порога, например равное 50, и разобьем исходное множество на два подмножества в соответствии с условием *Возраст* > *50*. В результате разбиения в одном подмножестве окажутся все записи, для которых значение атрибута Возраст больше 50, а во втором меньше 50. На рис. 1.1 данные подмножества обозначены номерами 1 и 2 соответственно. Легко увидеть, что выбор возрастного порога 50 не позволил получить подмножества, содержащие только объекты одного класса, поэтому для решения задачи применяется разбиение полученных подмножеств. Поскольку для этого имеется только один атрибут *Возраст*, будем использовать его и в дальнейшем, но в условиях выберем другой порог. Например, для подмножества 1 применим порог – 60, а для подмножества 2 – 30. Результаты повторного разбиения представлены на рис. 1.2.

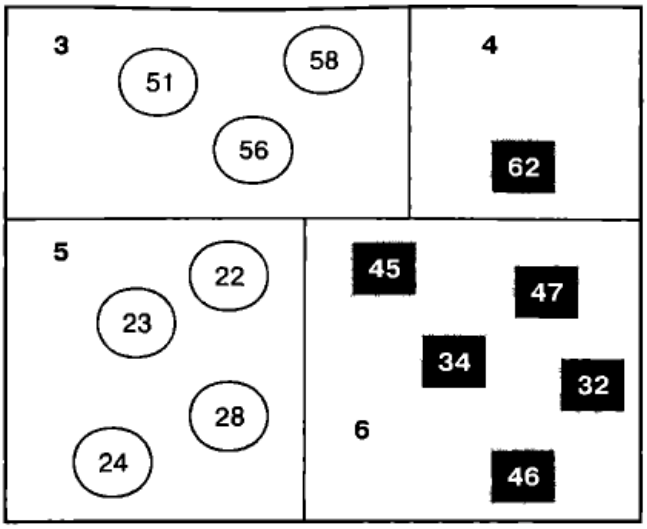


Рис. 1.2. Продолжение деления на классы

На рис. 1.2 можно увидеть, что задача решена: исходное множество удалось разбить на чистые подмножества, содержащие только наблюдения одного класса.

Дерево, реализующее данную процедуру, представлено на рис. 1.3.

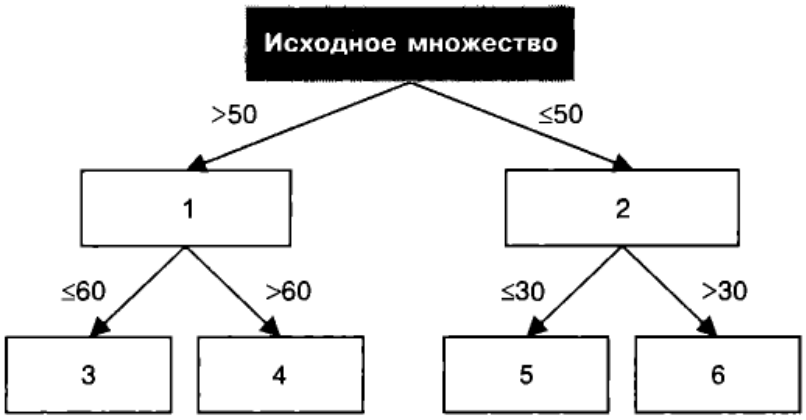


Рис. 1.3. Сформированное дерево решений

Сопоставление рис. 1.2 и 1.3 показывает, что подмножества 4 и 6 ассоциированы с классом добросовестных клиентов, а 3 и 5 с классом клиентов, не погасивших кредит. Применяя построенную модель к новым клиентам, мы можем предсказать риск, связанный с выдачей им кредита, на основе того, в какое из подмножеств модель поместит соответствующую запись.

Конечно, приведенный пример тривиален. В реальности задача классификации по одному атрибуту встречается очень редко, поскольку для эффективного разделения на классы требуется несколько атрибутов. Но даже этот пример позволяет увидеть причину привлекательности деревьев решений, Они не только классифицируют объекты и наблюдения, но и объясняют, почему объект был отнесен к данному классу. Так, если с помощью дерева решений было предсказано, что вероятность возврата кредита данным клиентам слишком мала и на этом основании в выдаче кредита было отказано, то можно не только принять решение, но и объяснить его причину. В нашем случае это возраст клиента, обладая высокой объясняющей способностью, деревья решений могут использоваться и как эффективные классификаторы, и как инструмент исследования предметной области.

Таким образом, мы получили систему правил вида «если…то…», которые позволяют принять решение относительно принадлежности объекта к определенному классу. Решающие правила образуют иерархическую древовидную структуру, дающую возможность выполнять классификацию объектов и наблюдений. Эта структура и называется *деревом решений*.

Деревья решений стали одним из наиболее популярных методов Data Mining, используемых при решении задач классификации. Это обусловлено следующими факторами:

- Деревья решений – это модели, основанные на обучении. Процесс обучения сравнительно прост в настройке и управлении.
- Процесс обучения деревьев решений быстр и эффективен.
- Деревья решений универсальны способны решать задачи как классификации, так и регрессии.
- Деревья решений обладают высокой объясняющей способностью и интерпретируемостью.

Для эффективного построения дерева решений должны выполняться следующие условия:

- *Описание атрибутов.* Анализируемые данные должны быть представлены в виде структурированного набора, в котором вся информация об объекте или наблюдении должна быть выражена совокупностью атрибутов.
- *Предварительное определение классов*. Категории, к которым относятся наблюдения (метки классов), должны быть заданы предварительно, то есть имеет место обучение с учителем.
- *Различимость классов.* Должна обеспечиваться принципиальная возможность установления факта принадлежности или непринадлежности примера к определенному классу. При этом количество примеров должно быть намного больше, чем количество классов.
- *Полнота данных.* Обучающее множество должно содержать достаточно большое количество различных примеров. Необходимая численность зависит от таких факторов, как количество признаков и классов, сложность классификационной модели и т.д.

Структура дерева решений

Как можно увидеть на рис. 1.3. структура деревьев решений проста и в целом аналогично древовидным иерархическим структурам, используемым в других областях Data Mining, например деревьям ассоциативных правил. В состав деревьев решений входят два вида объектов узлы (node) и листья (leaf). В узлах содержатся правила, с помощью которых производится проверка атрибутов и множество объектов в данном узле разбивается на подмножества. Листья – это конечные узлы дерева, в которых содержатся подмножества, ассоциированные с классами. Основным отличием листа от узла является то, что в листе не производится проверка, разбивающая ассоциированное с ним подмножество и, соответственно, нет ветвления. В принципе, листом может быть объявлен любой узел, если принято решение, что множество в узле достаточно однородно в плане классовой принадлежности объектов и дальнейшее разбиение не имеет смысла,

поскольку не приведет к значимому увеличению точности классификации, а только усложнит дерево.

В дереве, представленном на рис. 1.3. объекты с номерами 1 и 2 узлы, а с номерами 3, 4, 5 и 6 – листья. Обратим внимание на то, что в дереве имеется по два листа, ассоциированных с одним классом. В этом нет никакого противоречия, просто для классификации объектов в них использовались различные способы проверки. Для каждого листа в дереве имеется уникальный путь. Начальный узел дерева является входным: через него проходят все объекты, предъявляемые дереву. Обычно входной узел называют корневым узлом (root node). Следовательно, дерево растет сверху вниз. Узлы и листья, подчиненные узлу более высокого иерархического уровня, называются потомками, или дочерними узлами, а тот узел по отношению к ним предком, или родительским узлом. Обобщенная структура дерева проиллюстрирована на рис. 1.4.

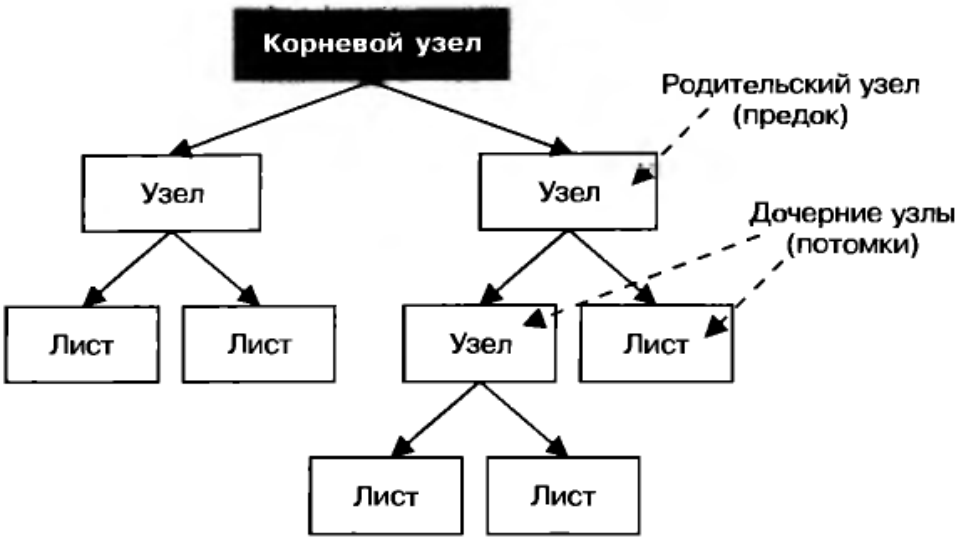


Рис. 1.4. Узлы и листья в дереве решений

Как и любая модель Data Mining, дерево решений строится на основе обучающего множества. Атрибуты могут быть как числовыми (непрерывными), так и категориальными (дискретными). Одно из полей обязательно должно содержать независимую переменную метку класса. Иными словами, для каждой записи в обучающем множестве должна быть задана метка класса, определяющая классовую принадлежность связанного с ней объекта.

В процессе построения дерева решений формируются решающие правила и для каждого из них создается узел. Для каждого узла нужно выбрать атрибут, по которому будет производиться проверка правила. Его принято называть атрибутом ветвления, или атрибутом разбиения (splitting attribute), и от того, насколько удачно он выбран, зависит классифицирующая сила правила. Метод, в соответствии с которым осуществляется выбор атрибута ветвления на каждом

шаге, называется алгоритмом построения дерева решений. Разработано достаточно много таких алгоритмов. Сформулируем общую цель, которая должна преследоваться при выборе атрибута ветвления: очередной выбранный атрибут должен обеспечивать наилучшее разбиение в узле. Наилучшим разбиением считается то, которое позволяет классифицировать наибольшее число примеров и создавать максимально чистые подмножества, в которых примесь объектов другого класса (то есть не ассоциированного с данным узлом или листом) минимальна.

Хотя между алгоритмами построения деревьев решений имеются существенные различия, все они основаны на одной и той же процедуре рекурсивном разбиении данных на все более малые труппы таким образом, чтобы каждое новое поколение узлов содержало больше примеров одного класса, чем родительский.

Выбор атрибута разбиения в узле.

Процесс создания дерева начинается с подготовки обучающего множества. В итоге будет построено дерево, которое назначает класс (или вероятность принадлежности к классу) для выходного поля новых записей на основе значений входных переменных.

Мерой оценки возможного разбиения является так называемая чистота (purity), под которой понимается отсутствие примесей. Существует несколько способов определения чистоты, но все они имеют один и тот же смысл. Низкая чистота означает, что в подмножестве представлены объекты, относящиеся к различным классам. Высокая чистота свидетельствует о том, что члены отдельного класса доминируют. Наилучшим разбиением можно назвать то, которое дает наибольшее увеличение чистоты дочерних узлов относительно родительского. Кроме того, хорошее разбиение должно создавать узлы примерно одинакового размера или как минимум не создавать узлы, содержащие всего несколько записей. Рассмотрим рис. 1.5.

В исходном множестве представлена смесь объектов различной формы (треугольники и круги) в пропорции 1:1 (10 кругов и 10 треугольников). Разбиение слева признано плохим, потому что оно не увеличивает чистоту результирующих узлов: пропорция объектов обоих классов в них сохраняется и также составляет 1:1 (5 кругов и 5 треугольников). Во втором случае плохоro разбиения (справа) с помощью условия В родительском узле удалось добиться доминирования класса в одном из дочерних узлов (круги), но к классу было отнесено только два объекта, Такое разбиение неудачно по двум причинам. Во-

первых, правило, с помощью которого было получено это разбиение, имеет очень низкую значимость, то есть относится к малому числу примеров.

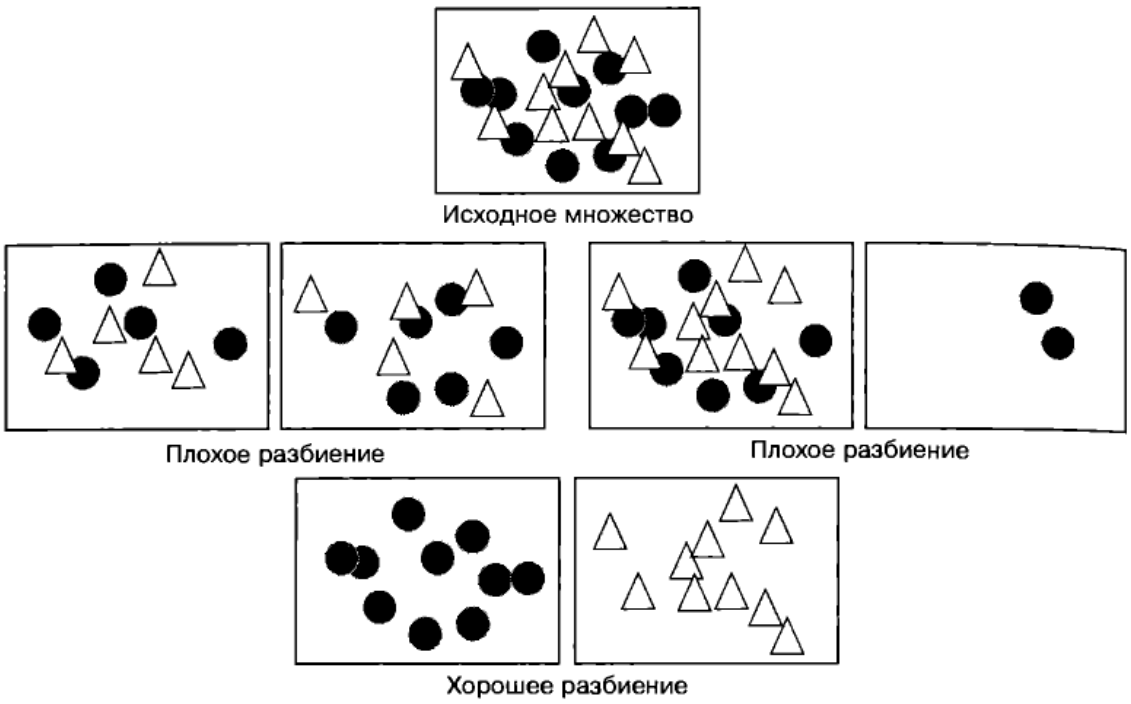


Рис. 1.5. Различные варианты разбиений

Во-вторых, чистота другого узла осталась низкой: соотношение объектов в нем 0,8:1 (8 кругов и 10 треугольников). И хотя в целом чистота относительно родительского узла улучшилась (масс треугольников стал слабо доминирующим), это не позволяет говорить о положительном результате, Наконец, случай хорошего разбиения обеспечивает абсолютную чистоту обоих дочерних узлов, поскольку в каждый из них распределены объекты только одного класса.

Алгоритмы построения деревьев решений являются «жадными». Они развиваются путем рассмотрения каждого входного атрибута по очереди и оценивают увеличение чистоты, которое обеспечило разбиение с помощью данного атрибута.

Принцип «разделяй и властвуй».

Процесс рекурсивного разбиения подмножеств в узлах дерева решений получил название «разделяй и властвуй!» (divide and conquer). В его основе лежит следующий принцип. Пусть задано множество T, в котором определены классы $\{C_1, C_2, \dots C_k\}$. Тогда существуют три возможных варианта разбиения,

1, Множество T содержит два примера или более, которые относятся к одному классу C_j. Деревом решений для множества T будет лист, идентифицирующий класс. Это тривиальный случай, который на практике не представляет интереса. Графически он может быть интерпретирован, как показано на рис. 1.6.

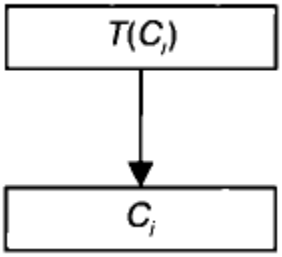


Рис. 1.6. Иллюстрация первого случая

2. Множество T не содержит примеров, то есть является пустым деревом решений, также будет лист, но класс, ассоциированный с данным листом, должен быть определен из другого множества, например родительского, Этот случай иллюстрируется на рис. 1.7.

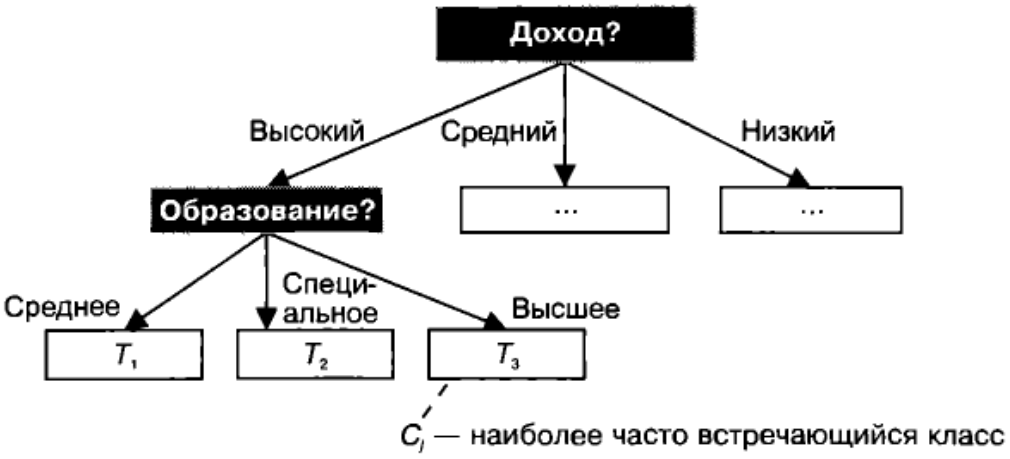


Рис. 9.7. Иллюстрация второго случая

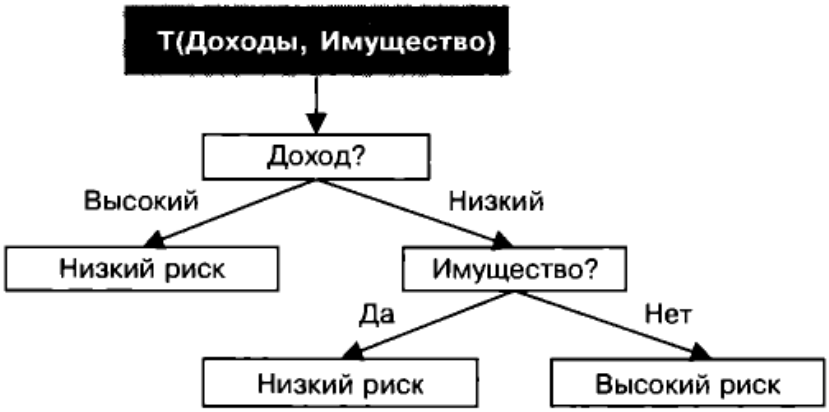
После разбиения по атрибуту *Доход* ветвление продолжилось по атрибуту *Образование*.

Как должен вести себя алгоритм, если в обучающем множестве нет ни одного наблюдения, где *Доход = Высокий* и *Образование = Среднее*? Алгоритм должен создать три дочерних узла, и в T_3 окажется пустое множество. Тогда сгенерируется узел, который будет ассоциирован с классом, наиболее часто встречающимся в родительском множестве.

3. Множество T содержит примеры, относящиеся к различным классам. Задача заключается в разделении T на подмножества, ассоциированные с классами.

Выбирается один из входных атрибутов, принимающий два отличных друг от друга значения $v_1, v_2 \dots v_n$, или более, после чего T разбивается на подмножества $\{T_1(v_1), T_2(v_2) \dots T_n(v_n)\}$, где каждое подмножество T: содержит все примеры из исходного множества, в которых выбранный атрибут принимает значение v_i. Данная процедура будет рекурсивно повторяться до тех пор, пока

подмножества не будут содержать примеры только одного класса. Этот случай проиллюстрирован на рис. 1.8.



Пусть требуется определить класс риска, связанный с выдачей кредита клиенту, на основе двух атрибутов *Доход* и *Имущество*. Атрибуты принимают по два значения *Высокий* и *Низкий*. На первом шаге алгоритм разобьет исходное множество на два подмножества, в одно из которых будут помещены клиенты с высокими доходами, а в другое с низкими. Предположим, что все клиенты с высокими доходами относятся к классу низкого кредитного риска, поэтому соответствующее подмножество будет объявлено листом и дальнейшее разбиение в нем остановится. Другое подмножество, в котором остались клиенты с низким доходом, подвергается проверке с помощью атрибута *Имущество*. Логика проста: если клиент не имеет высокого дохода, но имеет достаточно имущества, чтобы покрыть кредит при невозможности его выплатить, то кредитный риск может быть определен как низкий.

Данный случай является наиболее распространенным и практически важным в процессе построения деревьев решений.

Процесс построения дерева решений не является однозначно определенным. Для различных атрибутов и даже для различного порядка их применения могут быть сгенерированы различные деревья решений. В идеальном случае разбиения должны быть такими, чтобы результирующее дерево оказалось наиболее компактны.

Встает вопрос: почему бы тогда не исследовать все возможные деревья и не

выбрать самое компактное? К сожалению, во многих реальных задачах перебор всех возможных деревьев решений приводит к комбинаторному взрыву. Даже для небольшой базы данных, содержащей всего 5 атрибутов и 15 примеров, возможное число деревьев превышает 106 в зависимости от количества значений, принимаемых каждым атрибутом.

Эффективность разбиения оценивается по чистоте полученных дочерних узлов относительно целевой переменной. От ее типа и будет зависеть выбор предпочтительного критерия разбиения. Если выходная переменная является категориальной, то необходимо использовать такие критерии, как индекс Джини, прирост информации или тест хи-квадрат. Если выходная переменная является непрерывной, то для оценки эффективности разбиения используются метод уменьшения дисперсии или F-тест (рис. 1.9).

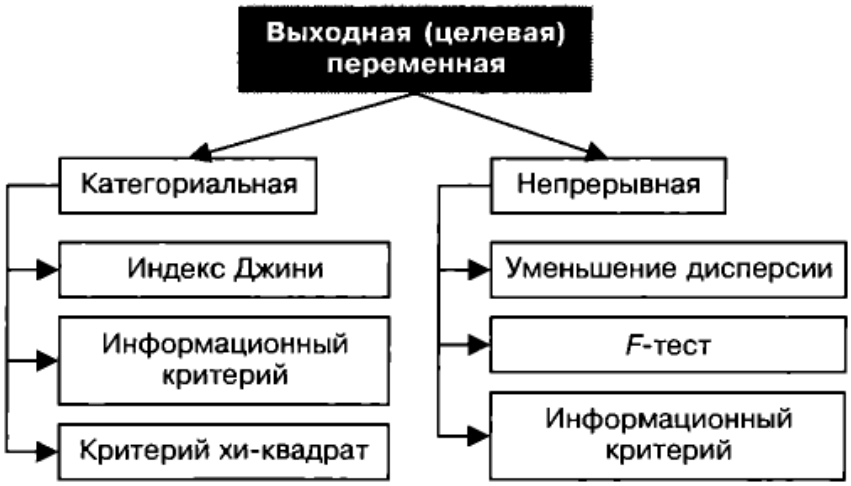


Рис. 1.9. Критерии разбиения

Рассмотрим несколько важных понятий.

Полное дерево решений.

Процесс роста дерева решений начинается с разбиения корневого узла на два потомка или более, каждый из которых рекурсивно подвергается дальнейшему разбиению. При этом каждый раз все входные атрибуты рассматриваются как потенциальные атрибуты разбиения, даже те, что уже использовались ранее, кроме атрибутов, все значения в которых одинаковы. Такие атрибуты исключаются из рассмотрения, поскольку с их помощью нельзя разделить при меры различных классов. Когда больше не удается обнаружить разбиения, значимо повышающие чистоту дочерних узлов, или когда число примеров в узле достигает некоторого заданного минимума, процесс разбиения для данной ветви заканчивается. Узел объявляется листом,

Когда найти в дереве какие-либо новые разбиения, повышающие его точность, не удается и разбиение прекращается по всем ветвям, это значит, что построено полное дерево.

Меры эффективности деревьев решений.

В целом эффективность деревьев решений определяется с помощью тестового множества набора примеров, которые не использовались при построении дерева. Дереву предъявляется набор тестовых примеров и

вычисляется, для какого процента примеров класс был определен правильно. Это позволяет оценить ошибку классификации, а также качество решения задачи классификации или регрессии отдельных ветвей в дереве.

Каждый узел или лист дерева обладает следующими характеристиками:

количество примеров, попавших в узел (лист);

доли примеров, относящихся к каждому из классов;

число классифицированных примеров (для листьев);

процент примеров, верно классифицированных данным узлом (листом).

Особый интерес представляет количество правильно классифицированных записей в данном узле или листе, Поэтому для оценки качества классификации вводятся два показателя *поддержка* (support) и *достоверность* (confidence).

Поддержка определяется как отношение числа правильно классифицированных примеров в данном узле или листе к общему числу попавших в него примеров, то есть:

$$S = \frac{N_{\text{кл}}}{N_{\text{общ}}}$$

Очевидно, что значение поддержки может изменяться от 0 до 1.

Достоверность определяется как отношение числа правильно классифицированных примеров к числу ошибочно классифицированных, то есть:

$$C = \frac{N_{\text{кл}}}{N_{\text{ош}}}$$

Значит, чем больше число правильно классифицированных примеров в узле, тем выше достоверность. Поддержка и достоверность могут использоваться в качестве параметров построения дерева решений. Например, можно задать, что разбиение должно производиться до тех пор, пока в узле не будет достигнут заданный порог поддержки.

Критерии выбора наилучших атрибутов ветвления

Существует множество различных подходов к оценке потенциальных разбиений. При этом методы, разработанные в рамках теории машинного обучения, в основном сосредотачиваются на повышении чистоты результирующих подмножеств, в то время как статистические методы фокусируются на статистической значимости различий между распределением значений выходной переменной в узлах. Альтернативные методы разбиения часто приводят к построению совершенно разных деревьев, которые, впрочем, функционируют примерно одинаково. Различные меры оценки чистоты ведут к выбору различных атрибутов разбиения, но, поскольку все меры основаны на одной и той же идее, полученные с их помощью модели будут похожи.

Для выбора атрибута ветвления в случае категориальной целевой переменной используются такие методы, как:

индекс Джини, или метод разнообразия выборки (Gini-index);

энтропия, или прирост информации (information gain);

отношение прироста информации (gain-ratio);

тест хи-квадрат (chi-square test).

Данные методы применимы и тогда, когда целевая переменная является непрерывной, но в этом случае необходимо предварительно выполнить ее квантование.

Индекс Джини.

Один из популярных критериев разбиения получил название индекса Джини в честь итальянского статистика и экономиста. Эта мера основана на исследовании разнообразия совокупности. Она определяет вероятность того что два объекта, случайным образом выбранные из одной совокупности, относятся к одному классу.

Очевидно, что для абсолютно чистой выборки данная вероятность равна 1. Мера Джини для узла представляет собой простую сумму квадратов долей классов в узле. В случае, представленном на рис.1.10, родительский узел содержит одинаковое количество светлых и темных кругов.

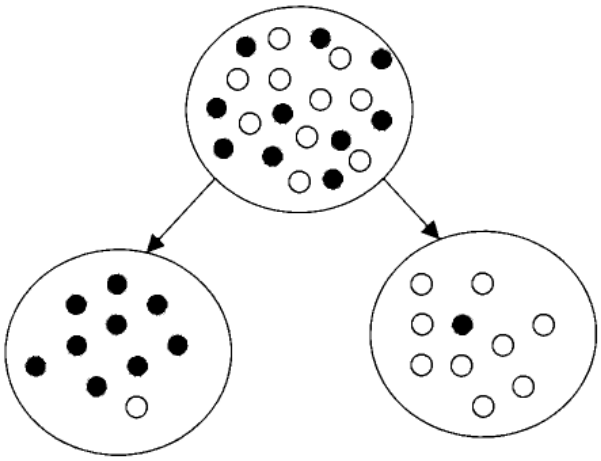


Рис. 1.10. Пример разбиения

Для узла, в котором содержится равное число объектов, относящихся к двум классам, можно записать: $0{,}5^2 + 0{,}5^2 = 0{,}5$, Такой результат вполне ожидаем, поскольку вероятность того, что случайно дважды будет выбран пример, относящийся к одному классу, равна 1/2. Индекс Джини для любого из двух результирующих узлов будет $0{,}1^2 + 0{,}9^2 = 0{,}82$. Идеально чистый узел имеет значение индекса Джини, равное 1. Узел, в котором содержится равное число объектов двух классов, будет иметь значение индекса 0,5. Таким образом,

предпочтение следует отдать тому атрибуту, который обеспечит максимальное значение индекса Джини.

Уменьшение энтропии, или прирост информации

Из теории информации известно, что чем больше состояний может принимать некоторая система, тем сложнее ее описать и тем больше информации для этого потребуется. Примером может служить кодирование цифровых изображений. Для представления каждой точки черно-белого изображения достаточно одного бита. Если он принимает значение 1, точка черная, а если 0 – белая. Когда нужно описать изображение, содержащее 256 оттенков, для задания каждой его точки потребуется $k = log_2 256 == 8$ бит и т. д.

Аналогичный подход можно применить к описанию подмножества в некотором узле дерева решений. Если лист совершенно чистый, то все попавшие в него примеры относятся к одному классу и его описание будет очень простым. Но если лист содержит смесь объектов различных классов, то его описание усложнится и для этого потребуется большее количество информации. В теории существует мера количества информации – энтропия, Она отражает степень неупорядоченности системы.

В контексте нашего рассмотрения энтропия – это мера разнообразия классов в узле. Проще говоря, будем считать, что это мера, определяющая количество вопросов «да/нет», на которые нужно ответить, чтобы определить состояние системы. Если существует 16 возможных состояний, это даст $log_2(16)$, или 4 бита, необходимых для описания всех состояний.

Целью разбиения узла в дереве решений является получение дочерних узлов с более однородным классовым составом, В результате разбиения должны образовываться узлы с меньшим разнообразием состояний выходной переменной. Следовательно, энтропия падает, а количество внутренней информации в узле растет. Уменьшение энтропии эквивалентно приросту информации.

Формально энтропия определённого узла T дерева решений определяется формулой

$$Info(T) = \sum_{i=1}^{k} p_j \, log_2 \, p_j \qquad (1)$$

и представляет собой сумму всех вероятностей появления примеров, относящихся к -классу, умноженную на логарифм этой вероятности. Поскольку вероятность меньше или равна 1, значение логарифма всегда будет отрицательным. На практике эта сумма обычно умножается на –1 для получения положительного числа.

Энтропия всего разбиения – это сумма энтропий всех узлов, умноженных на долю записей каждого узла в числе записей исходного множества.

Пусть в некотором узле дерева решений содержится множество T, которое состоит из N примеров. При этом неважно, является ли данный узел корневым или потомком, Torda по формуле (1) для него может быть рассчитана энтропия $I\text{п}fo(T)$. В результате разбиения S для данноuo узла были созданы k потомков $T_1, T_2, \dots T_k$, каждый из которых содержит число записей $N_1, N_2, \dots N_k$. Для потомков рассчитывается энтропия по формуле (1): $I\text{п}fo(T_1), I\text{п}fo(T_2), ., I\text{п}fo(T_k)$ Torda общая энтропия разбиения S:

$$Info(S) = \frac{N_1}{N} I\text{п}fo(T_1) + \frac{N_2}{N} I\text{п}fo(T_2) + \cdots + \frac{N_k}{N} I\text{п}fo(T_k) = \sum_{i=1}^{k} \frac{N_i}{N} I\text{п}fo(T_i) \quad (2)$$

Для бинарной целевой переменной, показанной на рис. 1.10, формула энтропии отдельного узла будет:

$$Info(T) = (-1)(P_{(\text{темные})} log_2 P_{(\text{темные})} + P_{(\text{светлые})} log_2 P_{(\text{светлые})})$$

Здесь $P_{(\text{темные})}$ вероятность того, что случайно выбранная из подмножества

запись имеет значение целевой переменной *Темные*. В нашем примере вероятности $P_{(\text{темные})}$ и $P_{(\text{светлые})}$ равны 0,5. Подставив значение 0,5 в предыдущую формулу, получим:

$$Info(T) = -1(0{,}5 log_2 0{,}5 + 0{,}5 log_2 0{,}5) = -1 log_2(0{,}5) = 1$$

Что собой представляет энтропия узлов, полученных в результате разбиения? Один из них содержит 1 темный и 9 светлых объектов, в то время как другой 1 светлый и 9 темных. Очевидно, что оба узла имеют одинаковую энтропию:

$$Info(T_1) = Info(T_2) == -1(0{,}1 log_2 0{,}1 + 0{,}9 log_2 0{,}9) = 0{,}33 + 0{,}14 = 0{,}47$$

Для вычисления общей энтропии разбиения воспользуемся формулой (2):

$$Info(S) = \frac{N_1}{N} I\text{п}fo(T_1) + \frac{N_2}{N} I\text{п}fo(T_2) = \frac{10}{20} 0{,}47 + \frac{10}{20} 0{,}47 = 0{,}47$$

В результате разбиения уменьшение полной энтропии, или прирост информации, составит $Gain(S) = 1 - 0.47 = 0{,}53$. Это показывает, что данное разбиение является эффективным.