# Урок 7

# Решающие деревья

## 7.1. Решающие деревья

Решающие деревья — это семейство алгоритмов, которое очень сильно отличается от линейных моделей, но в то же время играет важную роль в машинном обучении.

### 7.1.1. Линейные модели (обзор)

До этого момента изучались линейные модели. К особенностям линейных моделей относится следующее:

- Линейные модели быстро учатся. В случае со среднеквадратичной ошибкой для вектора весов даже есть аналитическое решение. Также легко применять для линейных моделей градиентный спуск.
- При этом линейные модели могут восстанавливать только простые зависимости из-за ограниченного количества параметров (степеней свободы).
- В то же время линейные модели можно использовать для восстановления нелинейных зависимостей за счет перехода к спрямляющему пространству, что является довольно сложной операцией.

Отдельно стоит отметить, что линейные модели не отражают особенности процесса принятия решений у людей. На самом деле, когда человек хочет понять ту или иную вещь, он будет задавать последовательность из простых вопросов, которые в итоге приведут его к какому-нибудь ответу.

### 7.1.2. Решающие деревья (пример 1)

Чтобы понять принцип работы решающих деревьев, полезно рассмотреть следующий сильно упрощенный пример.

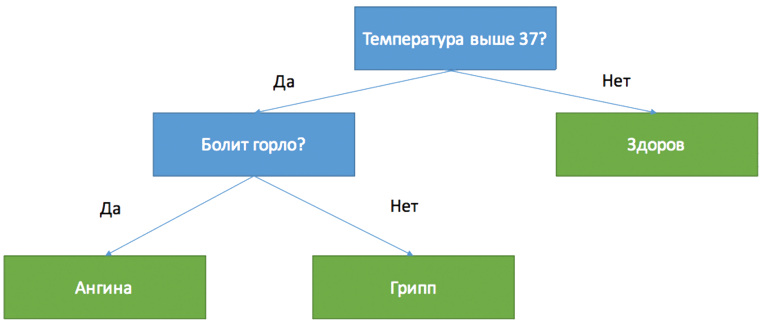


Рис. 7.1

Необходимо провести медицинскую диагностику. Врач, который проводит эту диагностику, знает только 2 заболевания — ангина и грипп. Поэтому сначала он спрашивает, какая температура у пациента. Если она меньше 37 градусов, он заключает, что пациент здоров, в ином случае — переходит к следующему вопросу, а

именно, спрашивает, болит ли у пациента горло. Если оно болит, врач ставит диагноз ангина, в ином случае — грипп.

*Создатели курса не рекомендуют серьезно относиться к предложенному методу диагностики заболеваний.*

### 7.1.3. Решающие деревья (пример 2)

Другой пример — для известной задачи определения того, выживет или не выживет тот или иной пассажир Титаника. Задача очень неплохо решается следующим решающим деревом:

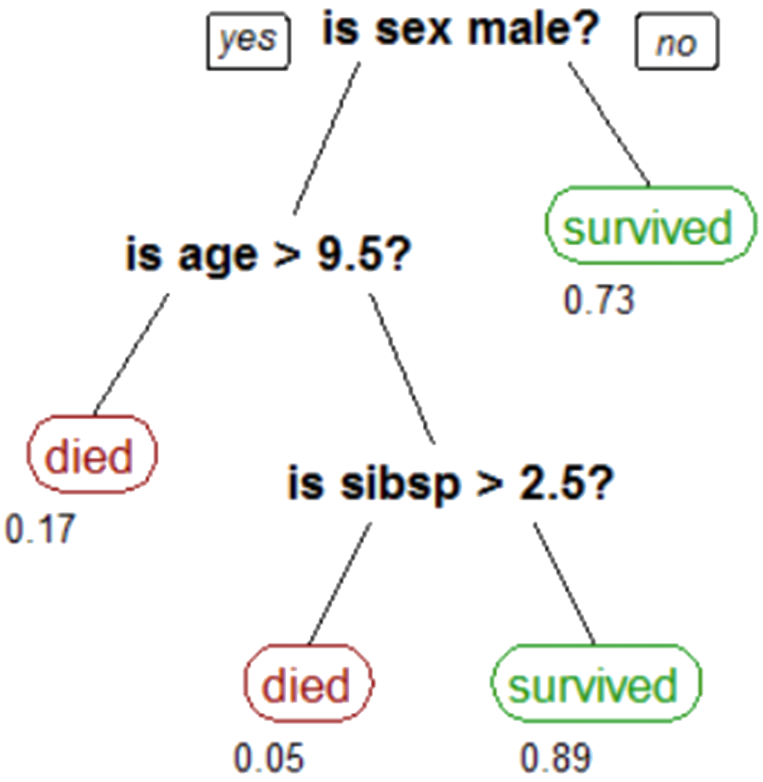


Рис. 7.2

В первую очередь спрашивается пол пассажира. Если это женщина, то решающее дерево сразу заявляет, что она выживает, и этот ответ верен в 73% случаев, и так далее.

### 7.1.4. Решающие деревья

Итак, были рассмотрены два примера решающих деревьев, которые представляли собой бинарные деревья, в каждой внутренней вершине записано условие, а в каждом листе дерева — прогноз. Строго говоря, не обязательно решающее дерево должно быть бинарным, но как правило используются именно бинарные.

Условия во внутренних вершинах выбираются крайне простыми. Наиболее частый вариант — проверить, лежит ли значение некоторого признака $x^j$ левее, чем заданный порог $t$:

$$[x^j \leq t].$$

Это очень простое условие, которое зависит всего от одного признака, но его достаточно, чтобы решать многие сложные задачи.

Прогноз в листе является вещественным числом, если решается задача регрессии. Если же решается задача классификации, то в качестве прогноза выступает или класс, или распределение вероятностей классов.

### 7.1.5. Решающие деревья в задаче классификации

Пусть решается задача классификации с двумя признаками и тремя классами.

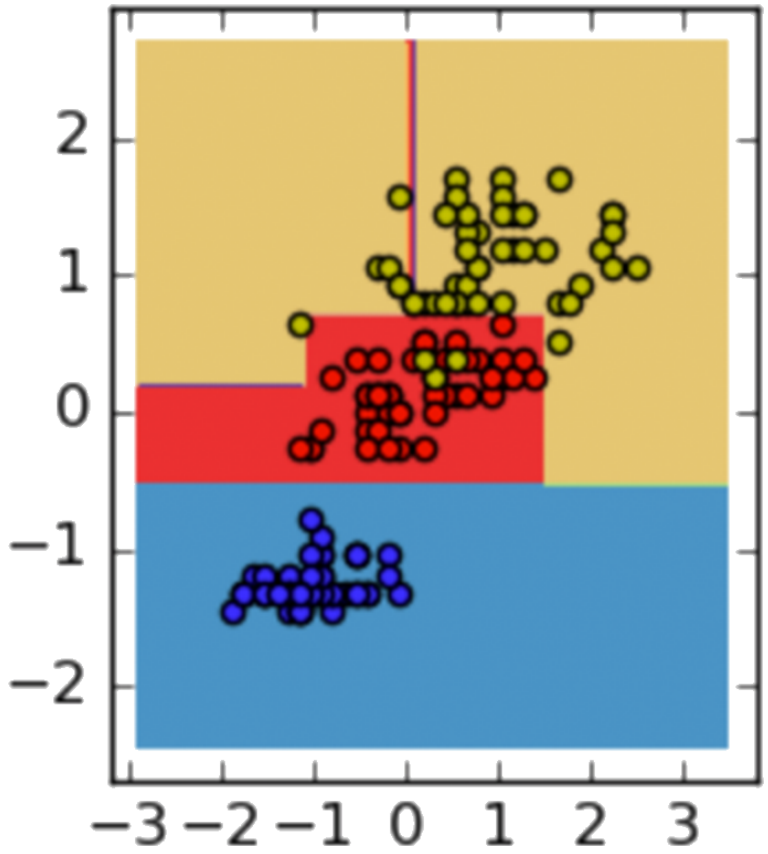


Рис. 7.3: Использование решающих деревьев в задачах классификации

Видно, что решающее дерево может очень неплохо отделить каждый класс от всех остальных. Видно, что разделяющая поверхность каждого класса кусочно-постоянная, и при этом каждая сторона поверхности параллельна оси координат, так как каждое условие сравнивает значение равно одного признака с порогом.

В то же время решающее дерево вполне может переобучиться: его можно сделать настолько глубоким, что каждый лист решающего дерева будет соответствовать ровно одному объекту обучающей выборки. В этом случае, если записать в каждом листе ответ соответствующего объекта, на обучающей выборке получается нулевая ошибка. Дерево получается явно переобученным. Вот пример такого дерева:

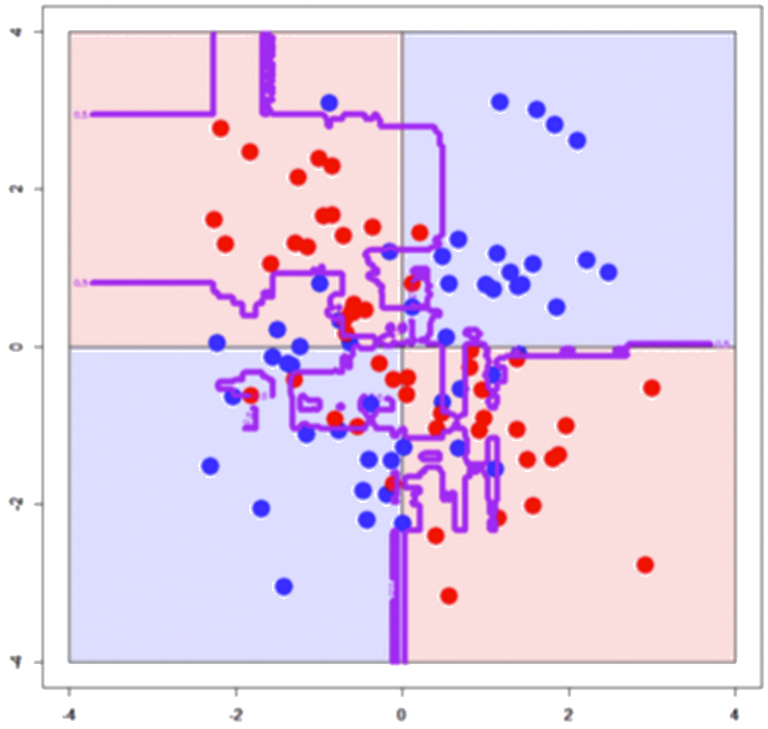

Рис. 7.4: Переобученное решающее дерево

Это дерево идеально отделило синий от красного класса, но разделяющая поверхность получилась безумно сложной — видно, что этот алгоритм переобучился и от него не будет никакой пользы на тестовой выборке.

### 7.1.6. Решающие деревья в задаче регрессии

Пусть решается задача регрессии с одним признаком, по которому нужно восстановить значение целевой переменной. Не очень глубокое дерево восстанавливает зависимость примерно так:

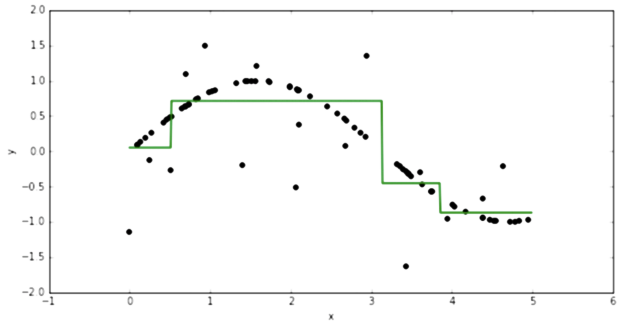

Рис. 7.5: Использование решающих деревьев в задачах регрессии

Восстановленная зависимость будет кусочно-постоянной, но в целом будет иметь неплохое качество. При увеличении глубины дерева получившаяся функция будет иметь следующий вид:

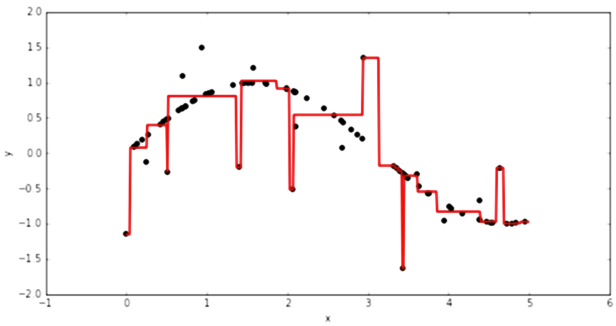

Рис. 7.6: Переобученное решающее дерево

Видно, что дерево подогналось под выбросы и его качество уже будет не таким хорошим. Дерево переобучилось из-за того, что его глубина слишком большая.

## 7.2. Обучение решающих деревьев

В данном разделе будет рассмотрен вопрос, как строить решающие деревья и как обучать их по конкретной выборке.

### 7.2.1. Переобучение деревьев

В предыдущем разделе было показано, что решающие деревья очень легко переобучаются. В том числе можно построить дерево, у которого каждый лист будет соответствовать одному объекту обучающей выборки.

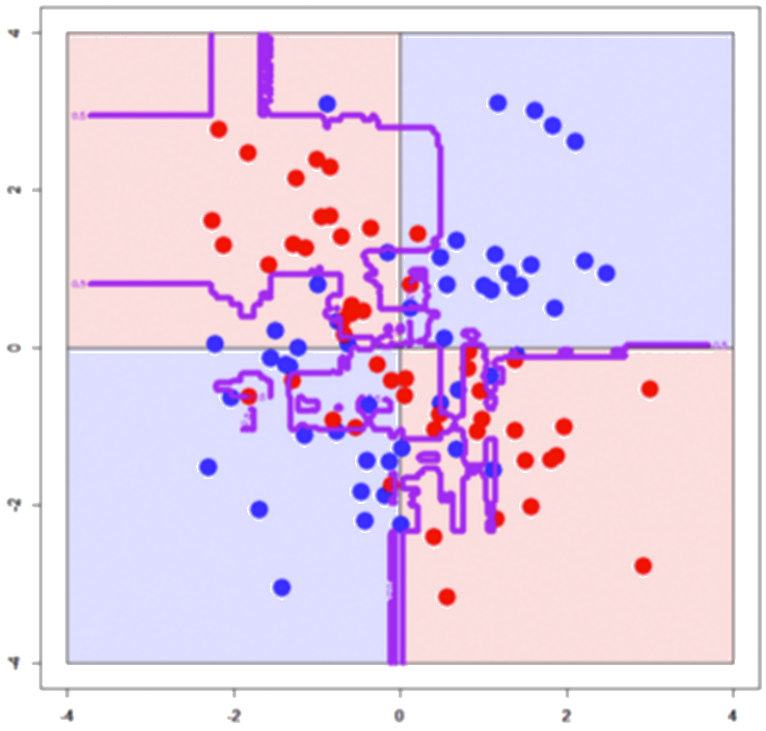

Рис. 7.7: Переобученное решающее дерево

Это дерево строит очень-очень сложную разделяющую поверхность и, очевидно, переобучено.

Поскольку всегда можно построить такое дерево, которое не ошибается на обучающей выборке и будет переобученным, имеет смысл искать минимальное (например, с минимальным числом листьев) дерево из имеющих нулевую ошибку. Но, к сожалению, задача отыскания такого дерева – NP-полная, то есть ее невозможно решить за разумное время.

### 7.2.2. Жадный способ построения

В машинном обучении применяется жадный способ построения решающего дерева от корня к листьям. Сначала выбирается корень, который разбивает выборку на две. Затем разбивается каждый из потомков этого корня и так далее. Дерево ветвится до тех пор, пока этого не будет достаточно.

Остается уточнить способ разбиения каждого потомка. Как было сказано ранее, в качестве условия в каждой вершине строящегося дерева будет использоваться простейшее условие: значение одного из признаков будет сравниваться с некоторым порогом.

Пусть в вершину $m$ попало множество $X_m$ объектов из обучающей выборки. Параметры в условии $[x^j \leq t]$ будут выбраны так, чтобы минимизировать данный критерий ошибки $Q(X_m, j, t)$, зависящий от этих параметров:

$$Q(X_m, j, t) \to \min_{j,t}.$$

Параметры $j$ и $t$ можно подбирать перебором. Действительно, признаков конечное число, а из всех возможных значений порога $t$ можно рассматривать только те, при которых получаются различные разбиения. Можно показать, что таких значений параметра $t$ столько, сколько различных значений признака $x^j$ на обучающей выборке.

После того, как параметры были выбраны, множество $X_m$ объектов из обучающей выборки разбивается на два множества

$$X_\ell = \left\{x \in X_m | [x^j \leq t]\right\}, \qquad X_r = \left\{x \in X_m | [x^j > t]\right\},$$

каждое из которых соответствует своей дочерней вершине.

Предложенную процедуру можно продолжить для каждой из дочерних вершин: в этом случае дерево будет все больше и больше углубляться. Такой процесс рано или поздно должен остановиться, и очередная дочерняя вершина будет объявлена листком, а не разделена пополам. Этот момент определяется критерием остановки. Существует много различных вариантов критерия остановки:

- Если в вершину попал только один объект обучающей выборки или все объекты принадлежат одному классу (в задачах классификации), дальше разбивать не имеет смысла.
- Можно также останавливать разбиение, если глубина дерева достигла определенного значения.

Возможные критерии останова будут обсуждаться позднее в этом уроке.

Если какая-то вершина не была поделена, а была объявлена листом, нужно определить прогноз, который будет содержаться в данном листе. В этот лист попала некоторая подвыборка $X_m$ исходной обучающей выборки и требуется выбрать такой прогноз, который будет оптимален для данной подвыборки.

В задаче регрессии, если функционал — среднеквадратичная ошибка, оптимально давать средний ответ по этой подвыборке:

$$a_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i.$$

В задаче классификации оптимально возвращать тот класс, который наиболее популярен среди объектов в $X_m$:

$$a_m = \text{argmax}_{y \in \mathbb{Y}} \sum_{i \in X_m} [y_i = y].$$

Если требуется указать вероятности классов, их можно указать как долю объектов разных классов в $X_m$:

$$a_{mk} = \frac{1}{|X_m|} \sum_{i \in X_m} [y_i = k].$$

## 7.3. Критерии информативности

В этом разделе речь пойдет о критериях информативности, с помощью которых можно выбирать оптимальное разбиение при построении решающего дерева.

### 7.3.1. Выбор критерия ошибки

Критерий ошибки записывается следующим образом:

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r)$$

и состоит из двух слагаемых, каждое из которых соответствует своему листу.

Функция $H(X)$ называется критерием информативности: ее значение должно быть тем меньше, чем меньше разброс ответов в $X$.

В случае регрессии разброс ответов — это дисперсия, поэтому критерий информативности в задачах регрессии записывается следующим образом:

$$H(X) = \frac{1}{|X|} \sum_{i \in X} \left(y_i - \bar{y}(X)\right)^2, \qquad \bar{y} = \frac{1}{|X|} \sum_{i \in X} y_i.$$

### 7.3.2. Критерий информативности Джини

Сформулировать критерий информативности для задачи классификации несколько сложнее. Пусть $p_k$ — доля объектов класса $k$ в выборке $X$:

$$p_k = \frac{1}{X} \sum_{i \in X} [y_i = k].$$

Критерий информативности Джини формулируется в терминах $p_k$:

$$H(X) = \sum_{k=1}^{K} p_k (1 - p_k).$$

Все слагаемые в сумме неотрицательные, поэтому критерий Джини также неотрицателен. Его оптимум достигается только в том случае, когда все объекты в $X$ относятся к одному классу.

Одна из интерпретаций критерий Джини — это вероятность ошибки случайного классификатора. Классификатор устроен таким образом, что вероятность выдать класс $k$ равна $p_k$.

### 7.3.3. Энтропийный критерий информативности

Еще один критерий информативности — энтропийный критерий:

$$H(X) = -\sum_{k=1}^{K} p_k \ln p_k.$$

В этом выражении полагается, что $0 \ln 0 = 0$.

Энтропийный критерий, как и критерий Джини, неотрицателен, а его оптимум также достигается только в том случае, когда все объекты в $X$ относятся к одному классу.

Энтропийный критерий имеет интересный физический смысл. Он заключается в том, что показывает, насколько распределение классов в $X$ отличается от вырожденного. Энтропия в случае вырожденного распределения равна 0: такое распределение характеризуется минимальной возможной степенью неожиданности. Напротив, равномерное распределение самое неожиданное, и ему соответствует максимальная энтропия.

## 7.4. Критерий останова и стрижка деревьев

В этом разделе речь пойдет о способах борьбы с переобучением деревьев, а именно о критериях останова и стрижке деревьев.

### 7.4.1. Критерий останова

Критерий останова используется, чтобы принять решение: разбивать вершину дальше или сделать листовой.

Худший случай решающего дерева — такое, в котором каждый лист соответствует своему объекту обучающей выборки. В этом случае дерево будет максимально переобученным и не будет обобщать информацию, полученную из обучающей выборки. Грамотно подобранный критерия останова позволяет бороться с переобучением.

Самый простой критерий останова проверяет, все ли объекты в вершине относятся к одному классу. Однако такой критерий останова может быть использован только в случае простых выборок, так как для сложных он остановится только тогда, когда в каждом листе останется примерно по одному объекту.

Гораздо более устойчивый и полезный критерий проверяет, сколько объектов оказалось в вершине, и разбиение продолжается, если это число больше, чем некоторое выбранное $n$. Соответственно, если в вершину попало $\leq n$ объектов, она становится листовой. Параметр $n$ нужно подбирать.

Случай $n = 1$ является худшим случаем, описанным выше. При этом выбирать $n$ нужно так, чтобы по $n$ объектам, которые попали в вершину, можно было устойчиво построить прогноз. Существует рекомендация, что $n$ нужно брать равным 5.

Еще один критерий, гораздо более грубый, заключается в ограничении на глубину дерева. Этот критерий хорошо себя зарекомендовал при построении композиций, когда много решающих деревьев объединяют в один сложный алгоритм. Об этом пойдет речь позже.

### 7.4.2. Стрижка деревьев

Существует и другой подход к борьбе с переобучением деревьев — стрижка. Он заключается в том, что сначала строится решающее дерево максимальной сложности и глубины, до тех пор, пока в каждой вершине не окажется по 1 объекту обучающей выборки.

После этого начинается «стрижка», то есть удаление листьев в этом дереве по определенному критерию. Например, можно стричь до тех пор, пока улучшается качество некоторой отложенной выборки.

Существует мнение, и это подкреплено многими экспериментами, что стрижка работает гораздо лучше, чем простые критерии, о которых говорилось раньше. Но стрижка — очень ресурсоёмкая процедура, так как, например, может потребоваться вычисление качества дерева на некоторой валидационной выборке на каждом шаге.

На самом деле, сами по себе деревья на сегодняшний день почти не используются, они бывают нужны только для построения композиции и объединения большого числа деревьев в один алгоритм. В случае с композициями такие сложные подходы к борьбе с переобучением уже не нужны, так как достаточно простых критериев останова, ограничения на глубину дерева или на число объектов в листе.

## 7.5. Решающие деревья и категориальные признаки

В этом разделе будет рассказано, как использовать категориальные признаки в решающих деревьях.

До этого момента использовалось следующее условие в вершине каждого дерева:

$$[x^j \leq t].$$

Очевидно, что такое условие можно записывать только для вещественных или бинарных признаков.

### 7.5.1. N-арные деревья

Подход, который позволяет включить категориальные признаки в деревья, состоит в том, чтобы строить $n$-арные деревья, то есть такие деревья, что из каждой вершины могут выходить до $n$ ребер. Пусть необходимо разбить некоторую вершину по некоторому признаку. Если этот признак — вещественный или бинарный, то все еще можно использовать простое условие с порогом $t$, поэтому интерес представляет именно случай категориального признака.

Если $x^j$ — категориальный признак, который может принимать значения

$$\{c_1, ..., c_n\},$$

можно разбить вершину на $n$ вершин таким образом, что в $i$-ую дочернюю вершину идут объекты с $x^j = c_i$. Критерий ошибки такого разбиения строится по аналогии со случаем бинарного дерева. Поскольку $X_n$ разбивается на $n$ частей, а не на две, в выражении будет $n$ слагаемых:

$$Q(X_m, j) = \sum_{i=1}^{n} \frac{|X_i|}{|X_m|} H(X_i) \to \min_j .$$

Таким образом, если вершину $m$ нужно разбить, рассматриваются все возможные вещественные, бинарные и категориальные признаки. Для вещественных и бинарных признаков считается $Q(X_m, j, t)$, а для $n$-арных — так, как написано выше. Разбиение вершины будет происходить по тому признаку, для которого значение критерия ошибки будет минимальным.

Важное замечание состоит в том, что при делении вершины по категориальному признаку получается больше дочерних вершин, на которых, очень вероятно, будет достигаться более высокое качество и более низкое значение критерия информативности, поэтому, скорее всего, при таком подходе предпочтение почти всегда будет отдаваться разбиению по категориальным признакам с большим числом возможных значений. В результате получается много листьев в дереве, что почти гарантированно может привести к переобучению.

Однако это не всегда так. Если выборки настолько большие, что даже после разбиения по категориальному признаку в каждом поддереве будет оставаться много объектов, то такое дерево будет неплохо работать, поскольку оно будет восстанавливать сложные зависимости, и при этом не переобучаться при использовании должного критерия останова.

### 7.5.2. Бинарные деревья с разбиением множества значений

Другой подход позволяет не переходить к $n$-арным деревьям и продолжать работать с бинарными деревьями. Пусть также необходимо сделать разбиение вершины $m$, а категориальный признак $x^j$ может принимать значения $C = \{c_1, ..., c_n\}$.

Для этого сначала необходимо разбить множество значений категориального признака на два не пересекающихся подмножества:

$$C = C_1 \cup C_2, \qquad C_1 \cap C_2 = \varnothing.$$

После того, как такое разбиение построено, условие в данной вершине будет выглядеть просто:

$$[x^j \in C_1]$$

Это условие проверяет, в какое из подмножеств попадает значение признака в данный момент на данном объекте. Главным вопросом остается то, как именно нужно разбивать множество $C$.

Полное количество возможных разбиений множества на два подмножества — $2^n$. К счастью, есть хитрость, которая позволяет избежать полного перебора и, более того, работать с категориальным признаком как с вещественным. Для этого возможные значения категориального признака сортируются специальным образом $c_{(1)}, ..., c_{(n)}$ и заменяются на натуральные числа 1,...,$n$. После этого с данными признаком следует уже работать как с вещественным, а значение порога $t$ будет определять разделение множества $C$ на два подмножества.

Сортировать значения категориального признака в случае задачи бинарной классификации нужно по следующему принципу:

$$\frac{\sum_{i \in X_m} [x_i^j = c_{(1)}][y_i = +1]}{\sum_{i \in X_m} [x_i^j = c_{(1)}]} \le \ ... \ \le \frac{\sum_{i \in X_m} [x_i^j = c_{(n)}][y_i = +1]}{\sum_{i \in X_m} [x_i^j = c_{(n)}]}$$

Фактически, значения категориального признака сортируются по возрастанию доли объектов $+1$ класса среди объектов выборки $X_n$ с соответствующим значением этого признака.

Для задачи регрессии сортировка происходит похожим образом, но вычисляется не доля объектов положительного класса, а средний ответ по всем объектам, у которых значение категориального признака равно $c$:

$$\frac{\sum_{i \in X_m} [x_i^j = c_{(1)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(1)}]} \le \ ... \ \le \frac{\sum_{i \in X_m} [x_i^j = c_{(n)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(n)}]}.$$

Главная особенность такого подхода состоит в том, что полученный результат полностью эквивалентен результату, который можно было бы получить в результате полного перебора. Это условие работает для критерия Джини, MSE и энтропийного критерия.