3. Решающие деревья

Цель занятия

В результате обучения на этой неделе вы:

- научитесь использовать решающие деревья в задачах машинного обучения
- познакомитесь с критериями информативности: энтропией и Джини
- узнаете, как использовать решающие деревья в задаче регрессии

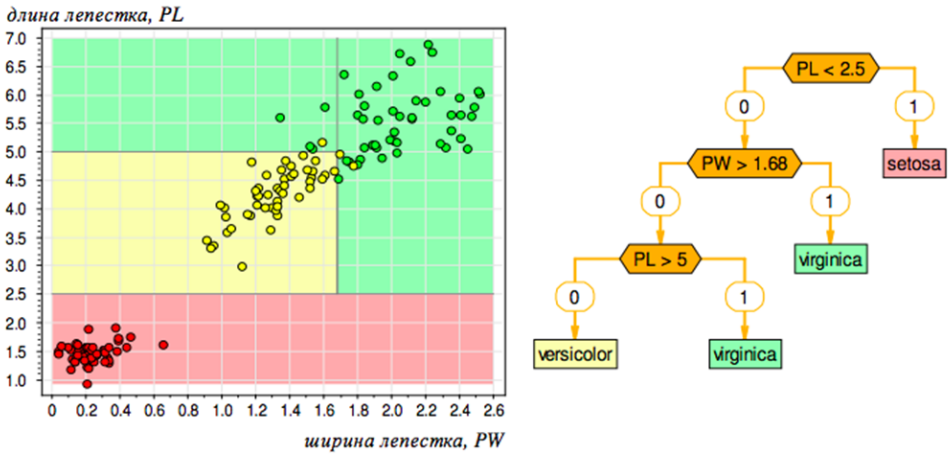
План занятия

Конспект занятия

1. Решающие деревья

Решающие деревья стоят несколько особняком от уже рассмотренных ранее линейных моделей или наивного Байеса, но при этом являются одними из основополагающих моделей в мире машинного обучения. Это наш первый шаг в сторону описания нелинейных зависимостей, не используя при этом ручные методы вроде подмены ядра в методе опорных векторов или ручной генерации признаков для линейных моделей.

Рассмотрим классический датасет ирисов:



setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

В данном датасете 4 признака. Мы визуализируем только 2 из них. И по двум из них построено решающее дерево.

Дерево – набор нескольких условий (предикатов), каждое из которых разделяет признаковое пространство на область, где предикат истинный, и область, где предикат ложный. Далее берем подвыборку, которая удовлетворила первому условию, и воспроизводим с ней ту же операцию, но с другим признаком и порогом (трешхолдом).

Дерево можно визуализировать в виде некоторого графа (см. рисунок выше) или отобразить схематично.

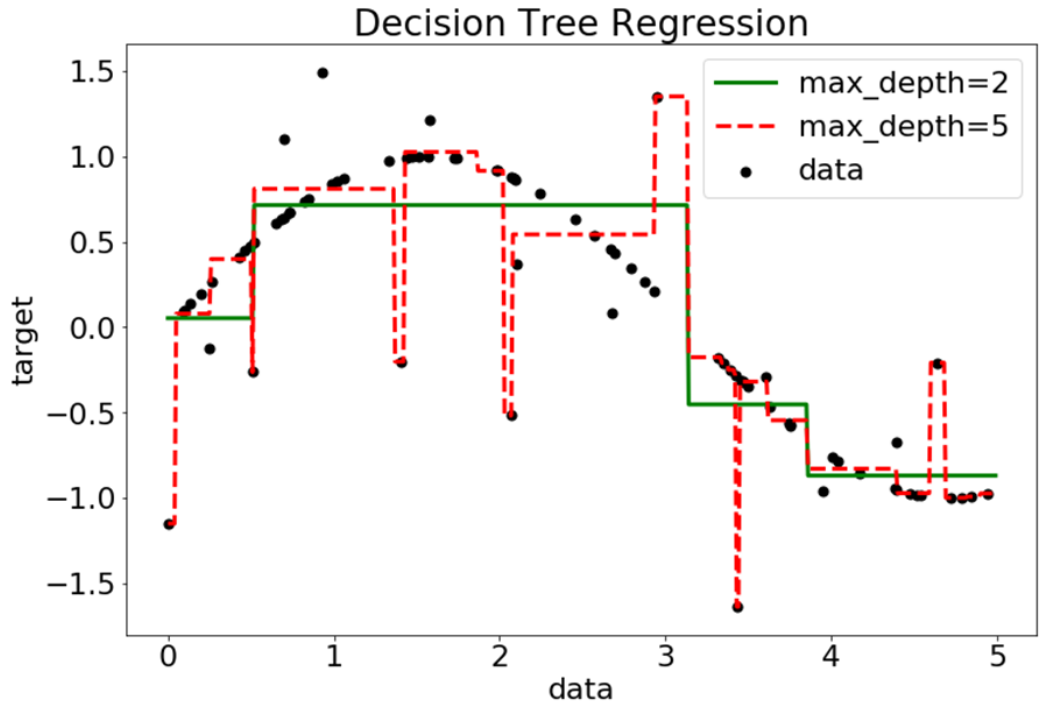
В целом дерево – это набор условий if. В качестве ответа получаем один из листьев. Обычно в каждой вершине делят на два дерева. Можно разделять на большее, но на практике так делают крайне редко.

Дерево описывает нелинейную разделяющую поверхность. Чем глубже дерево, тем более сложную поверхность дерево может описать. При этом все пороги строго параллельны одной из признаковых осей.

В случае классификации мы можем предсказывать метку класса или в более общем случае – вероятность метки класса.

Что делать в случае регрессии?

В случае регрессии ничего не меняется. Дерево выглядит примерно так:



Зеленое дерево решений – дерево глубины 2.

Красное дерево решений – дерево глубины 5.

Мы берем признаковое пространство. В данном случае для простоты визуализации взят одномерный объект – некоторые данные. Мы пытаемся описать данную зависимость с помощью решающего дерева. Оно также разбивает признаковое пространство на подобласти – в данном случае это полуинтервалы, в которых оно задает некоторую константу, которая описывает наш признак.

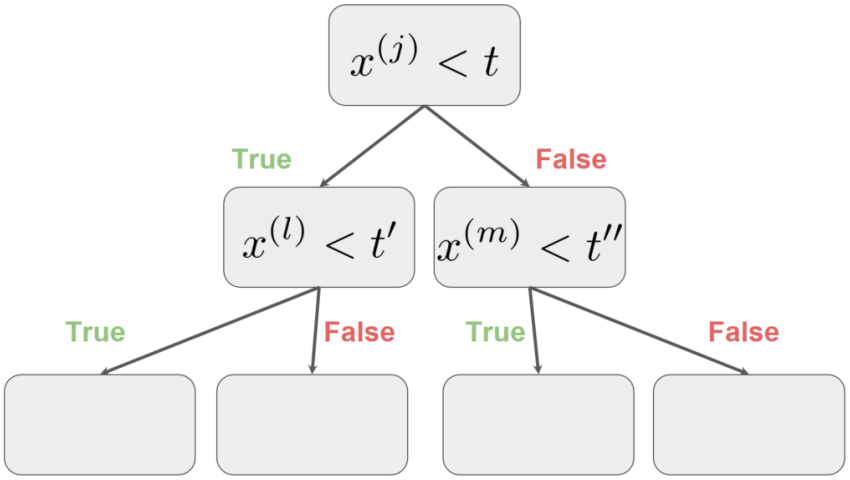
На рисунке видим, что красное дерево глубины 5 лучше описывает наши данные, включая также и все выбросы. Дерево в данном случае переобучилось. Оно обращает много внимания на выбросы. Это следствие того, как именно дерево строится.

Чем глубже дерево, тем более оно склонно к переобучению. Стоит вспомнить, что у нас есть зависимость между сложностью модели, её обобщающей способностью и тем, как модель может переобучаться. Если модель может запомнить слишком много информации из обучающей выборки, она, как нерадивый студент на экзамене, будет знать ответы на все вопросы, но при этом совершенно не понимать, что происходит. Поэтому делать очень глубокие деревья, то есть переобученные модели – не очень хорошая идея.

2. Процедура построения дерева решений

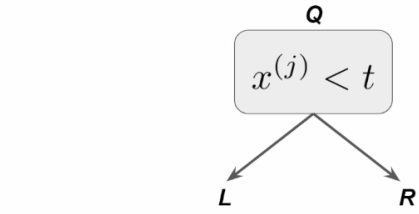
Процедура построения дерева решений не так проста, как кажется на первый взгляд. Дерево – это не дифференцируемая функция. У нее производная либо 0, либо не определена в точке разрыва. В данном случае градиентная оптимизация не подходит. Будем использовать **жадную оптимизацию** – каждый раз будем выбирать наиболее подходящее решение.

Представим, что у нас уже есть некоторая выборка X, и эта выборка попала в какую-то вершину t. Процедура разбиения выборки на две подвыборки абсолютно одинакова независимо от того, находимся мы в корне дерева или в предпоследней вершине перед листьями. Каждый раз повторяется одна и та же процедура:



На каждом шаге мы выбираем новый признак и величину порога. Совершаем рекурсию.

Как выбирать, по какому признаку разделять выборку, и как выбирать значение порога? Для этого предполагается использовать простую эвристику.



$$\frac{|L|}{|Q|}H(L) + \frac{|R|}{|Q|}H(R) \longrightarrow \min_{j,t}$$

Допустим, есть некоторая “мера нехорошести” – насколько данные не упорядочены внутри той подвыборки, которая у нас есть. Это функция H. Эту меру будем пытаться понизить с помощью разбиения.

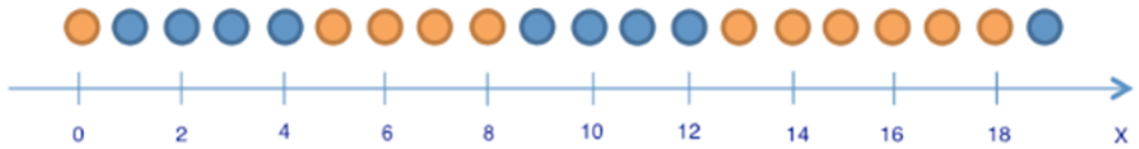
Пример. Есть выборка, в ней объекты класса “синие” и класса “желтые”. Мы хотим ее разбить на две подвыборки, в которых в одной только объекты класса “синие”, в правой – класса “желтые”. Это и можно использовать в качестве “меры нехорошести” – насколько у нас разнородны объекты в той или иной подвыборке.

3. Критерии информативности: Энтропия

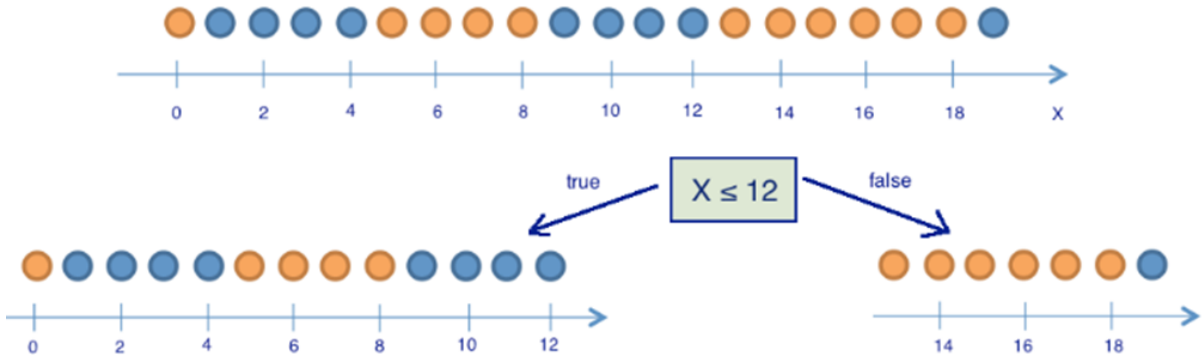
Введем формально “меру нехорошести” – понятие информативного критерия.

Пусть $H(R)$ будет измерять меру разнородности, гетерогенности наших данных, которые попали в ту или иную вершину.

Пример. Рассмотрим пример с желтыми и синими шарами. У нас один признак, все данные упорядочены вдоль одной оси:



Мы хотим разбить выборку на подвыборки так, чтобы в каждой подвыборке было больше одинаковых шариков:



Мы интуитивно нашли первый порог.

Для задачи бинарной классификации можно формально определить следующие критерии информативности:

1. Вероятность ошибиться, если для каждой выборки будем предсказывать метку класса, доминирующего в этой выборке. Это общий подход, не встречается на практике.

   Критерий классификации (Misclassification criteria):

   $$H(R) = 1 - max\{p_0, p_1\}$$

   Данный метод не подходит, если классов много.

2. Энтропийный критерий

   $$H(R) = -p_0 log p_0 - p_1 log p_1$$

3. Критерий Джини

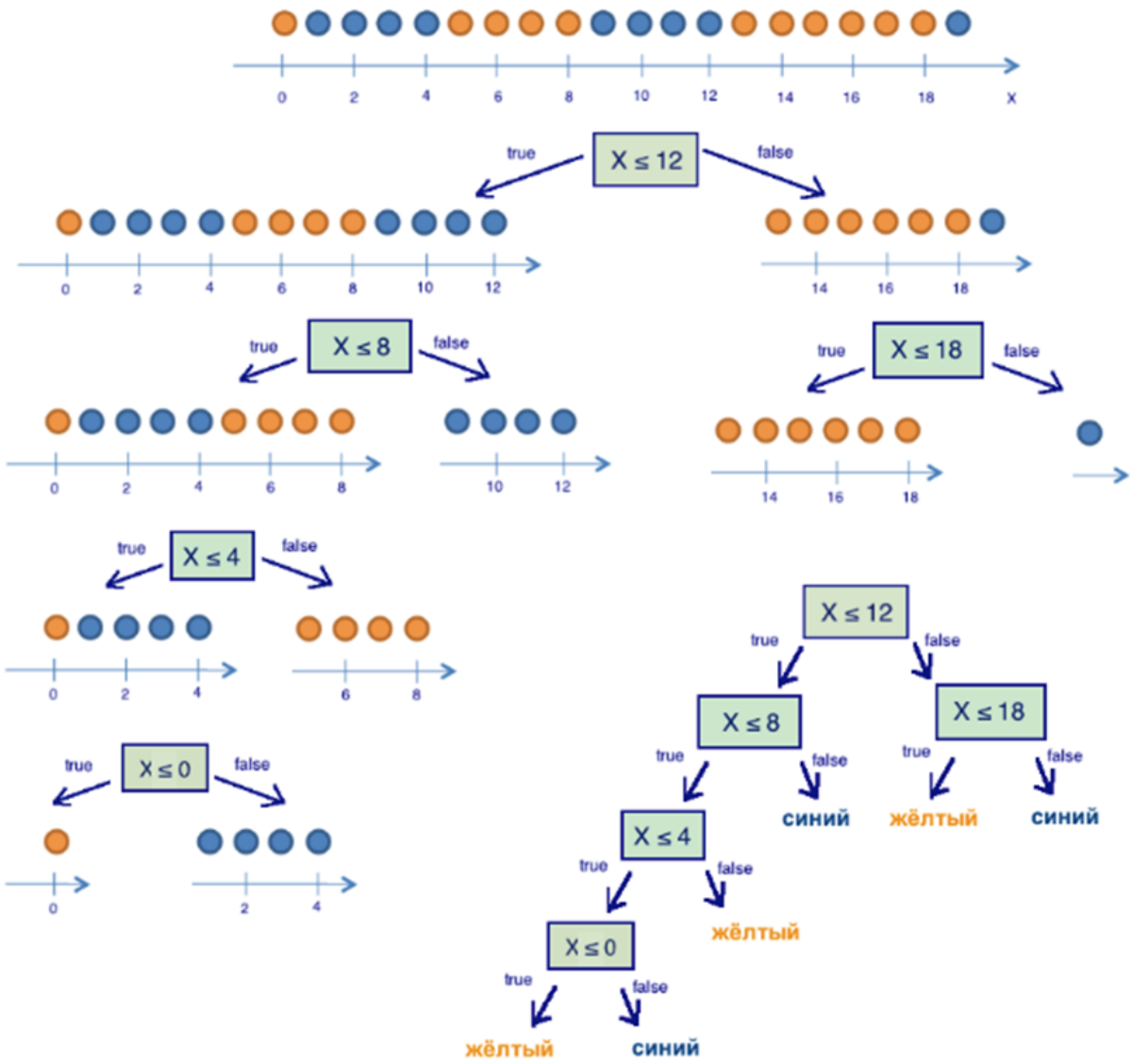
   $$H(R) = 1 - p_0^2 - p_1^2 = 2p_0p_1$$

Важно! Следует отличать критерий Джини (неупорядоченность Джини) от индекса Джини в экономической литературе. Это разные вещи.

В линейных моделях (логистической регрессии, SVM) при переходе от бинарной классификации к многоклассовой приходилось строить набор моделей “один против всех”, “один против одного”. Это не очень удобно. С решающими деревьями, также как и с методом наивного Байеса, и методом kNN, все работает “из коробки”. Мы можем переформулировать эти критерии для многоклассовой классификации:

1. $$H(R) = 1 - \max_k\{p_k\}$$
2. Энтропийный критерий $H(R) = -\sum_{k=0}^{K} p_k log p_k$
3. Критерий Джини $H(R) = 1 - \sum_k \left(p_k\right)^2$

Вернемся к нашему примеру с синими и желтыми шариками. Все дерево решений будет выглядеть так:



Посмотреть более подробно эту задачу можно здесь.

Формула для энтропии:

$$S = -M \sum_{k=0}^{K} p_k log p_k$$

K – количество классов, которые у нас есть, k – индекс класса, p_k – вероятность выбрать объект k-го класса в той или иной выборке.

Поскольку энтропию мы минимизируем, нам все равно, какое основание у логарифма в формуле выше. Обычно используют или натуральный, или десятичный логарифм. Получаем то же выражение для энтропии с точностью до константы M.

Нас интересует аргумент, при котором достигается минимум энтропии.

Энтропия показывает, насколько разнородна наша выборка. Будет минимальна, когда распределение вырожденное, то есть для всех классов, кроме одного, вероятность определения 0. В этом случае энтропия равна 0.

Величина $p_k \in [0,1]$ – неотрицательная, $logp_k$ – величина отрицательная. То есть энтропия в целом не отрицательна. Поэтому у энтропии в нуле достигается минимум, когда никакой случайности нет.

Посчитаем энтропию для каждого шага разделения синих и желтых шариков из нашего примера.

Как функция, энтропия также фиксирована сверху, если количество объектов K –конечное.

> **Домашнее задание**. Докажите, что для равномерного распределения энтропия будет максимальной (задача условной максимизации, может помочь метод множителей Лагранжа).

Рассмотрим энтропию для бинарного случая с точностью до константы:

$$S = -p_+ log_2 p_+ - p_- log_2 p_- = -p_+ log_2 p_+ - \left(1 - p_+\right) log_2 \left(1 - p_+\right).$$

p_+ – вероятность положительного класса,

p_- – вероятность отрицательного класса.

Получаем очень похожую формулу на логистическую функцию потерь. Логистическую функцию потерь часто еще называют кросс-энтропией.

4. Критерий Джини

Критерий также позволяет строить решающее дерево, оценивая, насколько “хороша” та или иная выборка, и как разбить её на две подвыборки.

Критерий Джини:

$$G = 1 - \sum_k \left(p_k\right)^2$$

Суть критерия Джини: какова вероятность выбрать объект класса k из какой-то выборки.

Теперь будем вытаскивать два объекта. Выбираем объекты с возвращением – взяли и положили на место. Для первого объекта вероятность быть из класса k – p_k, для второго – тоже p_k. Вероятность выбрать пару объектов класса k с возвращением –

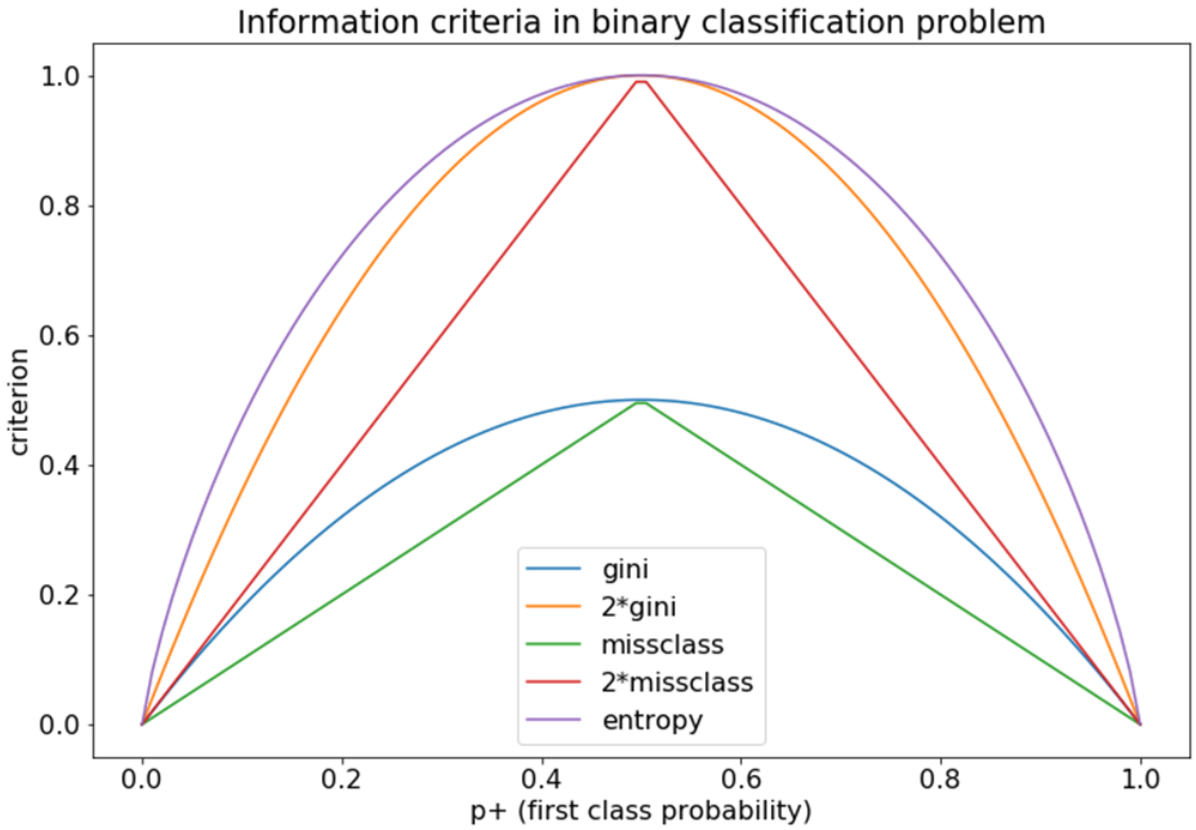
$(p_k)^2$. Критерий Джини – вероятность того, что мы выбрали два объекта с возвращением из нашей выборки, и они оказались разных классов. Это еще один способ измерить некоторую разнородность выборки. Часто используют при построении деревьев.

Для бинарного случая:

$$G = 1 - p_+^2 - p_-^2 = 1 - p_+^2 - (1 - p_+)^2 = 2p_+(1 - p_+).$$

$(p_+ + p_- = 1)$

Рассмотрим, как соотносятся друг с другом разные критерии информативности для задачи бинарной классификации:



Видим, что критерии Джини и энтропийный критерий очень похожи друг на друга. В основном именно эти критерии заложены в программах построения большинства решающих деревьев.

5. Критерии в задаче регрессии. Усечение деревьев

Задача регрессии

Рассмотрим, как можно использовать решающие деревья в задачах регрессии.

Дерево представляет собой кусочно-постоянную функцию, которая предсказывает некоторую константу для каждой подобласти, выделенной отдельным листом.

Для разделения областей нужен подходящий критерий информативности. Будем считать ошибку между каждым значением целевой переменной для данного объекта и предсказанной константой.

Мы можем взять среднюю квадратичную ошибку:

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - c)^2$$

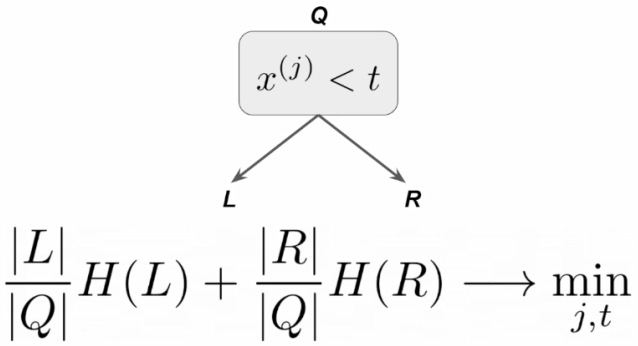
Константа c в этом случае:

$c *= \frac{1}{|R|} \sum_{y_i \in R} y_i$ – мат. ожидание, то есть выборочное среднее в данном случае.

$|R|$ – количество элементов в множестве R, то есть множестве, в которое попали все элементы нашей подвыборки.

Мы поменяли критерии информативности, при этом вся процедура построения решающего дерева осталась та же самая.

Рассмотрим еще раз схему решающего дерева:



$$\frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \longrightarrow \min_{j,t}$$

На вход приходит некоторая выборка Q, которая обладает некоторым множеством элементов. Мы разбиваем по j-му признаку, по порогу t на левую (L) и правую (R) подвыборки. Для подвыборок L и R мы можем посчитать H, которую мы выбираем в зависимости от задачи. В классификации это критерий Джини или энтропия, в регрессии – например MSE. Для регрессии можно взять среднюю абсолютную ошибку (MAE), в этом случае константой будет медиана.

На какие подвыборки поделить Q? Для любой подвыборки мы умеем считать критерий информативности. Используем жадную оптимизацию. Мы перебираем все возможные признаки j, и для каждого из них перебираем все возможные значения порога. Это конечно долго и не эффективно. Но, с другой стороны, нет необходимости считать все значения порога. Достаточно их ставить в осмысленных местах (вспомним пример про синие и желтые шары).

Также необходимо учитывать, сколько объектов было распределено в подвыборку L и в подвыборку R. Отнести один объект в подвыборку, а остальные – в другую, не очень хорошая идея. Если у нас миллион объектов, и на каждом шаге убирать по одному объекту, получится дерево глубины миллион, и оно точно переобучится.

Усечения деревьев

Усечение деревьев (Pruning) – попытка избавиться от избыточной его сложности.

Разделяют два типа усечения:

- Препрунинг (Pre-pruning) – внесение некоторых ограничений на структуру модели до построения дерева.
    - Ограничение глубины дерева
    - Ограничение числа листьев в дереве
    - Минимальный размер листа (количество элементов, которые в него попали)
- Постпрунинг (Post-pruning) – основной подход к усечению деревьев. Это попытка усечь дерево, которое уже было построено. Для этого запускаем еще одну процедуру жадной оптимизации, чтобы понять, какое из разбиений можно выкинуть, чтобы дерево ”ухудшилось” максимально слабо.

На практике решающие деревья вручную усекают очень редко. В одиночестве решающее дерево, как правило, практически нигде не используется. Поскольку в одиночку оно очень слабое: легко переобучается, неустойчиво. Лучше использовать ансамбли деревьев.

Поэтому прунинг на практике используется лишь, когда он уже “вшит” в процедуру построения дерева.

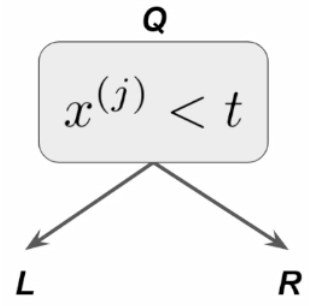
6. Специфические свойства деревьев

Специфические свойства деревьев полезно понимать, если мы хотим построить интегральную картину того, как деревья работают.

Пропуски в данных

Деревья очень полезны на практике, потому что они позволяют работать с **пропусками в данных**. Дерево может обрабатывать пропуски на уровне своей архитектуры.

Рассмотрим пример. Есть выборка Q, она пришла на вход дерева.



Что делать, если есть объект, который не обладает j-м признаком? Отправим его одновременно и в левое, и в правое поддерево. Смотрим и слева, и справа, какое получится предсказание, возвращаемся назад и строим взвешенную сумму левого и правого ответов:

$$\hat{y} = \frac{|L|}{|Q|}\hat{y}_L + \frac{|R|}{|Q|}\hat{y}_R$$

Взвешенная сумма берется для того, чтобы понять, насколько часто объекты попадали влево и вправо на этапе обучения.

При работе с пропущенными признаками конечно ухудшается предсказываемая величина, становится менее точной.

Решающее дерево, как линейная модель

Это свойство позволяет “породниться” решающим деревьям и линейным моделям. Рассмотрим предсказание от дерева:

$$\hat{y} = \sum_j w_j [x \in J_j]$$

Предсказываемая величина – взвешенная сумма индикаторных величин. Получаем линейную модель. Мы неявным образом построили процедуру построения признакового описания, где для каждого объекта вместо того, чтобы смотреть на его значение признаков, мы говорим, что он принадлежит определенному листу и описывается некоторой константой ω. То есть мы строим информативные признаки, но вместо SVM мы подбирали их не вручную.

Механизмы построения деревьев

- **ID-3**. Использует энтропийный критерий. Строится жадным образом до того момента, когда уже нельзя уменьшить минимум энтропии. Дерево переобучается.
- **C4.5**. Использует нормированный энтропийный критерий. Останавливается на основании ограничений, которые были получены извне – минимальный размер листа. Включал в себя постпрунинг.
- **C5.0**. Улучшение C4.5. Доработаны некоторые алгоритмы.
- **CART**. Использует критерий Джини, прунинг, который зависел от того, насколько сложным получилось дерево, суррогатные предикаты.

Дополнительные материалы для самостоятельного изучения

1. https://habr.com/ru/company/ods/blog/322534/