Лекция. Логические методы машинного обучения. Деревья решений

5 ноября 2025 г.

ВОПРОСЫ:

- Определения, области практического применения
- Этапы построения дерева решений
- Критерии остановки алгоритма
- Достоинства и недостатки метода

1 Определения, области практического применения

Деревья решений Основополагающие идеи послужившие толчком к появлению и развитию деревьев решений были заложены в 1950-х гг. в области исследований моделирования человеческого поведения с помощью компьютерных систем: К. Ховеленд «Компьютерное моделирование мышления», Е. Ханта и др. «Эксперименты по индукции».

Деревья решений воспроизводят логические схемы для классификации объекта с помощью ответов в иерархически организованную систему вопросов. Вопрос, задаваемый на последующем иерархическом уровне, зависит от ответа, полученного на предыдущем уровне.

Метод представления решающих правил в иерархической структуре, состоящей из элементов двух типов: узлов (node) и листьев (leaf). В узлах находятся решающие правила и производится проверка соответствия примеров этому правилу по какому-либо атрибуту обучающего множества. В результате проверки, множество примеров, попавших в узел, разбивается на два подмножества, в одно из которых попадают примеры, удовлетворяющие правилу, а в другое - не удовлетворяющие

Структура дерева решений представляет собой связный ориентированный граф – ориентированное дерево Корневая вершина (узел), связанная с выходящим рёбрами. Внутренние вершины, связанны с входящим ребром и нескольким выходящим ребрами. Ребра - это возможные варианты решений. Листья (концевые вершины, узлы) восходят только к одному входящему ребру. Каждый уровень рассматривается как решение. Узел обеспечивает проверку решений. Онсовные термины, используемые при построении дерева решений приведены на рис.1.

В результате проверки, множество примеров, попавших в узел, разбивается на два подмножества, в одно из которых попадают примеры, удовлетворяющие правилу, а в другое — не удовлетворяющие. Затем к каждому подмножеству вновь применяется правило и процедура рекурсивно повторяется пока не будет достигнуто некоторое условие остановки алгоритма. В результате в последнем узле проверка и разбиение не производится и он объявляется листом. Лист определяет решение для каждого попавшего в него примера. Для дерева классификации — это класс, ассоциируемый с узлом, а для дерева регрессии — соответствующий листу модальный интервал целевой переменной.

Название	Описание
Объект	Пример, шаблон, наблюдение
Атрибут	Признак, независимая переменная, свойство
Целевая переменная	Зависимая переменная, метка класса
Узел	Внутренний узел дерева, узел проверки
Корневой узел	Начальный узел дерева решений
Лист	Конечный узел дерева, узел решения, терминальный узел
Решающее правило	Условие в узле, проверка

Рис. 1:

Таким образом, в отличие от узла, в листе содержится не правило, а подмножество объектов, удовлетворяющих всем правилам ветви, которая заканчивается данным листом.

Очевидно, чтобы попасть в лист, пример должен удовлетворять всем правилам, лежащим на пути к этому листу. Поскольку путь в дереве к каждому листу единственный, то и каждый пример может попасть только в один лист, что обеспечивает единственность решения.

Область применения метода: ботаника, зоология, минералогия, медицина и др. машинное обучение.

- Классификация — отнесение объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения
- Регрессия (численное предсказание) — предсказание числового значения независимой переменной для заданного входного вектора
- Описание объектов — набор правил в дереве решений позволяет компактно описывать объекты

Дерево для классификации, когда предсказываемый результат является классом, к которому принадлежат данные;//

Дерево для регрессии, когда предсказываемый результат можно рассматривать как вещественное число//

Любое решающее дерево может быть преобразовано в набор продукционных правил: каждому пути от корня дерева до терминальной вершины соответствует одно продукционное правило. Его посылкой является конъюнкция условий «признак – значение», соответствующих пройденным вершинам и ребрам дерева, а заключением – имя или номер класса, соответствующего терминальной вершине.

Деревья решений применяют при разработке правил продукций в экспертных системах. Правило продукций формулируется в виде описания посылки условий и заключения в виде действия, которое необходимо выполнить.

Правило 1: Если «условие 1», то «заключение 1»(F1)
Правило 2: Если «условие 2», то «заключение 2» (F2)
Правило 3: Если «условие 2», то «заключение 2» (F3)

(F1, F2, F3- коэффициенты истинности правил продукций).

Дерево принятия решений (также называют деревом классификации или регрессионным деревом).

Правила автоматически генерируются в процессе обучения на обучающем множестве и, поскольку они формулируются практически на естественном языке, деревья решений как аналитические модели более вербализуемы и интерпретируемы, чем нейронные сети.

По аналогии с соответствующим методом логического вывода их называют индуктивными правилами, а сам процесс обучения —

индукцией деревьев решений.

2 Этапы построения дерева решений

В ходе построения дерева решений нужно решить несколько основных проблем, с каждой из которых связан соответствующий шаг процесса обучения: 1. Выбор атрибута, по которому будет производиться разбиение в данном узле (атрибута разбиения).
2. Выбор критерия остановки обучения.
3. Выбор метода отсечения ветвей (упрощения).
4. Оценка точности построенного дерева.

При формировании правила для разбиения в очередном узле дерева необходимо выбрать атрибут, по которому это будет сделано. Общее правило для этого можно сформулировать следующим образом: выбранный атрибут должен разбить множество наблюдений в узле так, чтобы результирующие подмножества содержали примеры с одинаковыми метками класса, или были максимально приближены к этому, т.е. количество объектов из других классов («примесей») в каждом из этих множеств было как можно меньше. Для этого были выбраны различные критерии, наиболее популярными из которых стали теоретико-информационный и статистический.

Теоретико-информационный критерий

Критерий основан на понятиях теории информации, а именно — информационной энтропии.

будет тот, который обеспечит максимальное снижение энтропии результирующего подмножества относительно родительского. На практике, однако, говорят не об энтропии, а о величине, обратной ей, которая называется информацией. Тогда лучшим атрибутом разбиения будет тот, который обеспечит максимальный прирост информации результирующего узла относительно исходного:

$$H = -\sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right)$$

где n — число классов в исходном подмножестве, N_i число примеров i-го класса, N — общее число примеров в подмножестве.

Энтропия может рассматриваться как мера неоднородности подмножества по представленным в нём классам. Когда классы представлены в равных долях и неопределённость классификации наибольшая, энтропия также максимальна. Если все примеры в узле относятся к одному классу, т.е. $N = N_i$

логарифм от единицы обращает энтропию в ноль

лучшим атрибутом разбиения A_j

будет тот, который обеспечит максимальное снижение энтропии результирующего подмножества относительно родительского. На практике, однако, говорят не об энтропии, а о величине, обратной ей, которая называется информацией. Тогда лучшим атрибутом разбиения будет тот, который обеспечит максимальный прирост информации результирующего узла относительно исходного:

$$\text{Gain}(A) = \text{Info}(S) - \text{Info}(S_A)$$

где Info(S) - информация, связанная с подмножеством S до разбиения, $Info(S_A)$ — информация, связанная с подмножеством, полученными при разбиении по атрибуту A. задача выбора атрибута разбиения в узле заключается в максимизации величины Gain, называемой приростом информации (от англ. gain — прирост, увеличение). Поэтому сам теоретико-информационный подход известен как критерий прироста информации. Он впервые был применён в алгоритме ID3, а затем в C4.5 и других алгоритмах.

Статистический поход В основе статистического подхода лежит использование индекса Джини (назван в честь итальянского статистика и экономиста Коррадо Джини). Статистический смысл данного показателя в том, что он показывает — насколько часто случайно выбранный пример обучающего множества будет распознан неправильно, при условии, что целевые значения в этом множестве были взяты из определённого статистического распределения.

Таким образом индекс Джини фактически показывает расстояние между двумя распределениями — распределением целевых значений, и распределением предсказаний модели. Очевидно, что чем меньше данное расстояние, тем лучше работает модель.

Индекс Джини может быть рассчитан по формуле:

$$\text{Gini}(Q) = 1 - \sum_{i=1}^{n} p_i^2,$$

где Q — результирующее множество, n — число классов в нём, p_i вероятность i-го класса (выраженная как относительная частота примеров соответствующего класса). Очевидно, что данный показатель меняется от 0 до 1. При этом он равен 0, если все примеры Q относятся к одному классу, и равен 1, когда классы представлены в равных пропорциях и равновероятны. Тогда лучшим будет то разбиение, для которого значение индекса Джини будут минимальным.

3 Критерий остановки алгоритма

Теоретически, алгоритм обучения дерева решений будет работать до тех пор, пока в результате не будут получены абсолютно «чистые» подмножества, в каждом из которых будут примеры одного класса. Правда, возможно при этом будет построено дерево, в котором для каждого примера будет создан отдельный лист. Очевидно, что такое дерево окажется бесполезным, поскольку оно будет переобученным — каждому примеру будет соответствовать свой уникальный путь в дереве, а следовательно, и набор правил, актуальный только для данного примера.

Переобучение в случае дерева решений ведёт к тем же последствиям, что и для нейронной сети — точное распознавание примеров, участвующих в обучении и полная несостоятельность на новых данных. Кроме этого, переобученные деревья имеют очень сложную структуру, и поэтому их сложно интерпретировать.

Очевидным решением проблемы является принудительная остановка построения дерева, пока оно не стало переобученным. Для этого разработаны следующие подходы.

1. Ранняя остановка — алгоритм будет остановлен, как только будет достигнуто заданное значение некоторого критерия, например процентной доли правильно распознанных примеров. Главным недостатком является то, что ранняя остановка всегда делается в ущерб точности дерева

2. Ограничение глубины дерева — задание максимального числа разбиений в ветвях, по достижении которого обучение останавливается. Данный метод также ведёт к снижению точности дерева.

3. Задание минимально допустимого числа примеров в узле — запретить алгоритму создавать узлы с числом примеров меньше заданного (например, 5). Это позволит избежать создания тривиальных разбиений и, соответственно, малозначимых правил.

Все перечисленные подходы являются эвристическими, т.е. не гарантируют лучшего результата или вообще работают только в каких-то частных случаях. Поэтому к их использованию следует подходить с осторожностью. Каких-либо обоснованных рекомендаций по тому, какой метод лучше работает, в настоящее время тоже не существует. Поэтому аналитикам приходится использовать метод проб и ошибок.

Глубина дерева решений — это количество уровней (или слоев) от корневого узла до самого дальнего листового узла. Она определяет максимальное количество шагов или правил, которые модель использует для принятия решения, и является важным параметром для контроля сложности модели и предотвращения переобучения.

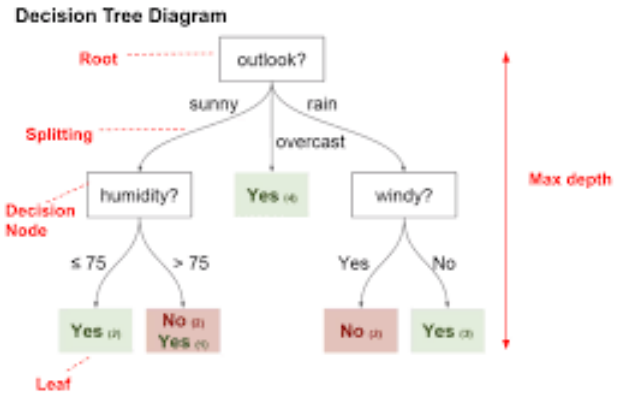

Рис. 2: Глубина дерева решений

Регулирование глубины дерева — это техника, которая позволяет уменьшать размер дерева решений, удаляя участки дерева, которые имеют маленький вес.

Один из вопросов, который возникает в алгоритме дерева решений — это оптимальный размер конечного дерева. Так, небольшое дерево может не охватить ту или иную важную информацию о выборочном пространстве. Тем не менее, трудно сказать, когда алгоритм должен остановиться, потому что невозможно спрогнозировать, добавление какого узла позволит значительно уменьшить ошибку. Эта проблема известна как «эффект горизонта». Тем не менее, общая стратегия ограничения дерева сохраняется, то есть удаление узлов реализуется в случае, если они не дают дополнительной информации.

Регулирование глубины дерева должно уменьшить размер обучающей модели дерева без умень-

шения точности её прогноза или с помощью перекрестной проверки. Есть много методов регулирования глубины дерева, которые отличаются измерением оптимизации производительности.

Некоторые методы позволяют построить более одного дерева решений (ансамбли деревьев решений):

1. Бэггинг над деревьями решений, наиболее ранний подход. Строит несколько деревьев решений, неоднократно интерполируя данные с заменой (бутстреп), и в качестве консенсусного ответа выдаёт результат голосования деревьев (их средний прогноз); 2. Классификатор «Случайный лес» основан на бэггинге, однако в дополнение к нему случайным образом выбирает подмножество признаков в каждом узле, с целью сделать деревья более независимыми; 3. Бустинг над деревьями может быть использован для задач как регрессии, так и классификации. Одна из реализаций бустинга над деревьями, алгоритм XGBoost, неоднократно использовался победителями соревнований по анализу данных. 3. «Вращение леса» — деревья, в которых каждое дерево решений анализируют первым применением метода главных компонент (PCA) на случайные подмножества входных функций.

4 Достоинства и недостатки метода

К достоинствам метода относят следующие.

- Метод прост в понимании и интерпретации результатов;
- не требует специальной подготовки данных, как например: нормализации данных, добавления фиктивных переменных, а также удаления пропущенных данных;
- способен работать как с категориальными, так и с интервальными переменными. (Прочие методы работают лишь с теми данными, где присутствует лишь один тип переменных. Например, метод отношений может быть применён только на номинальных переменных, а метод нейронных сетей только на переменных, измеренных по интервальной шкале.);
- использует модель «белого ящика», то есть если определённая ситуация наблюдается в модели, то её можно объяснить при помощи булевой логики. Примером «черного ящика» может быть искусственная нейронная сеть, так как полученные результаты сложно объяснить;
- позволяет оценить модель при помощи статистических тестов, что позволяет оценить надежность модели.
- метод хорошо работает даже в том случае, если были нарушены первоначальные предположения, включенные в модель;
- позволяет работать с большим объёмом информации без специальных подготовительных процедур. Данный метод не требует специального оборудования для работы с большими базами данных.

Недостатки метода:

- проблема получения оптимального дерева решений является NP-полной задачей, с точки зрения некоторых аспектов оптимальности даже для простых задач. Таким образом, практическое применение алгоритма деревьев решений основано на эвристических алгоритмах, таких как алгоритм «жадности», где единственно оптимальное решение выбирается локально в каждом узле. Такие алгоритмы не могут обеспечить оптимальность всего дерева в целом.

- в процессе построения дерева решений могут создаваться слишком сложные конструкции, которые недостаточно полно представляют данные. Данную проблему называют переобучением. Для решения данной проблемы используют метод «регулирования глубины дерева».
- существуют понятия, которые сложно понять из модели, так как модель описывает их сложным путём. Данное явление может быть вызвано проблемами XOR, чётности или мультиплексарности.