Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

РЕФЕРАТ
на тему

ДЕРЕВЬЯ РЕШЕНИЙ

Магистрант: Н.Р. Ровдо

МИНСК 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Деревья решений используются в повседневной жизни в самых разных областях человеческой деятельности, порой и очень далеких от машинного обучения. Деревом решений можно назвать наглядную инструкцию, что делать в какой ситуации. В терминах машинного обучения можно сказать, что это элементарный классификатор, который определяет форму по нескольким признакам. Дерево решений как алгоритм машинного обучения – по сути то же самое: объединение логических правил вида "Значение признака A меньше X И Значение признака B меньше Yследовательно Класс 1" в структуру данных "Дерево". Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку.

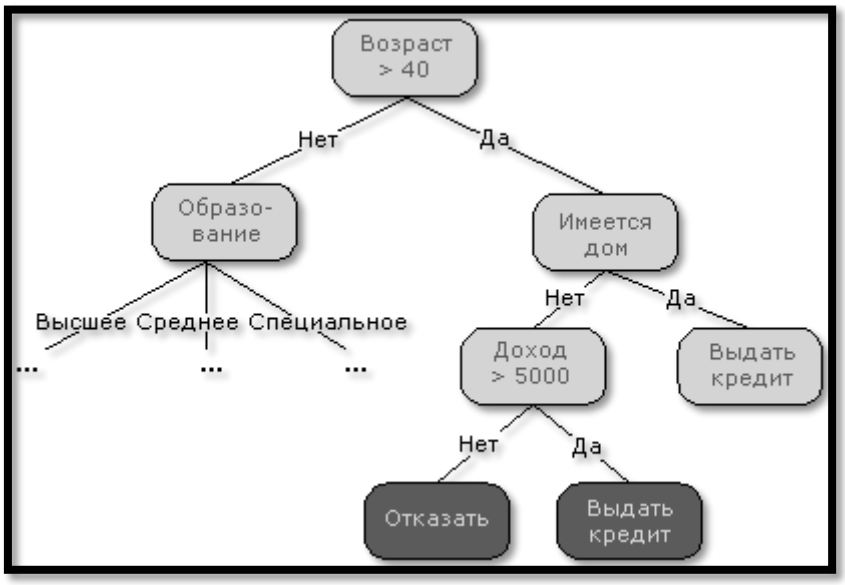

Рисунок 1 Пример дерева решений

Листьями дерева принятия решений являются классы. Чтобы классифицировать объект при помощи дерева принятия решений – нужно последовательно спускаться по дереву (выбирая направление основываясь на значениях предикатов применяемых к классифицируемому объекту). Путь от корня дерева до листьев можно трактовать как объяснение того, почему тот или иной объект отнесён к какому-либо классу.

Также, не накладывается ограничений на значения атрибутов объекта – они могут иметь как категориальную, так и числовую или логическую природу. Нужно только определить предикаты, которые умеют правильно обрабатывать значения атрибутов (например, вряд ли есть смысл использовать

предикаты «больше» или «меньше» для атрибутов с логическими значениями).

Использовать дерево решений очень просто спускаемся по дереву, выбирая нудные атрибуты, и в конце получаем ответ.

Свойства алгоритма деревьев решений:

1) Легко интерпретировать

2) Стремление к однородности групп, получившихся в результате деления

3) Деление всего множества

4) Жадность

Пример алгоритма построения дерева решений:

s0 = вычисляем энтропию исходного множества
Если s0 == 0 значит:
Все объекты исходного набора, принадлежат к одному классу
Сохраняем этот класс в качестве листа дерева
Если s0 != 0 значит:
Ищем предикат, который разбивает исходное множество таким образом чтобы уменьшилось среднее значение энтропии
Найденный предикат является частью дерева принятия решений, сохраняем его
Разбиваем исходное множество на подмножества, согласно предикату
Повторяем данную процедуру рекурсивно для каждого подмножества

В основе популярных алгоритмов построения дерева решений, таких как ID3 и C4.5, лежит принцип жадной максимизации прироста информации – на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим. Дальше процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине (если дерево не подгоняется идеально под обучающую выборку во избежание переобучения).

В разных алгоритмах применяются разные эвристики для "ранней остановки" или "отсечения", чтобы избежать построения переобученного дерева.

Свойства деревьев решений:

- Кусочно-постоянная природа
- Изменчивость при изменениях
- Линейная комбинация, произведение, степень деревьев — дерево
- Дерево деревьев — дерево решений.
- Любое дерево решений можно представить как бинарное

Виды деревьев

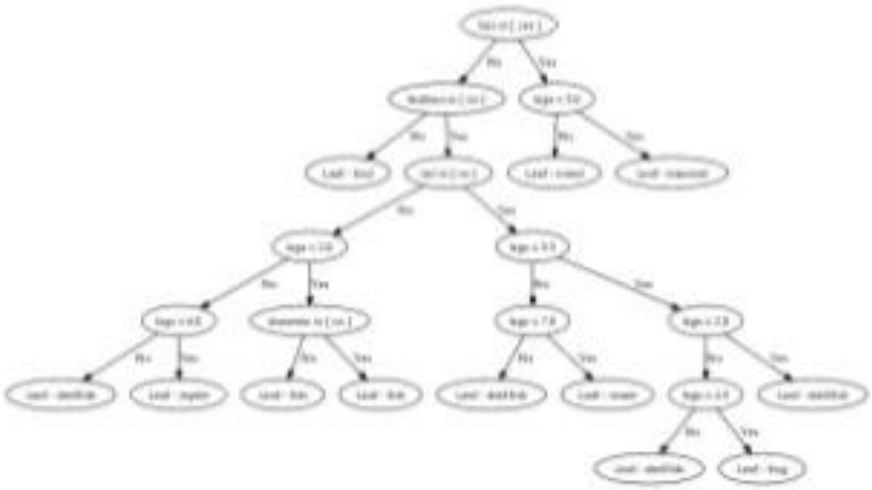

Рисунок 2 Дикие деревья решений

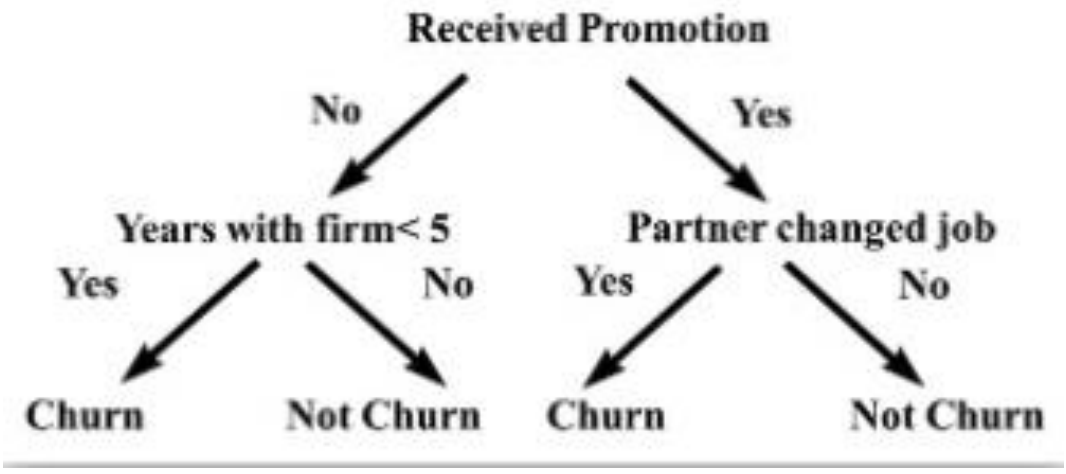

Рисунок 3 Деревья-пеньки

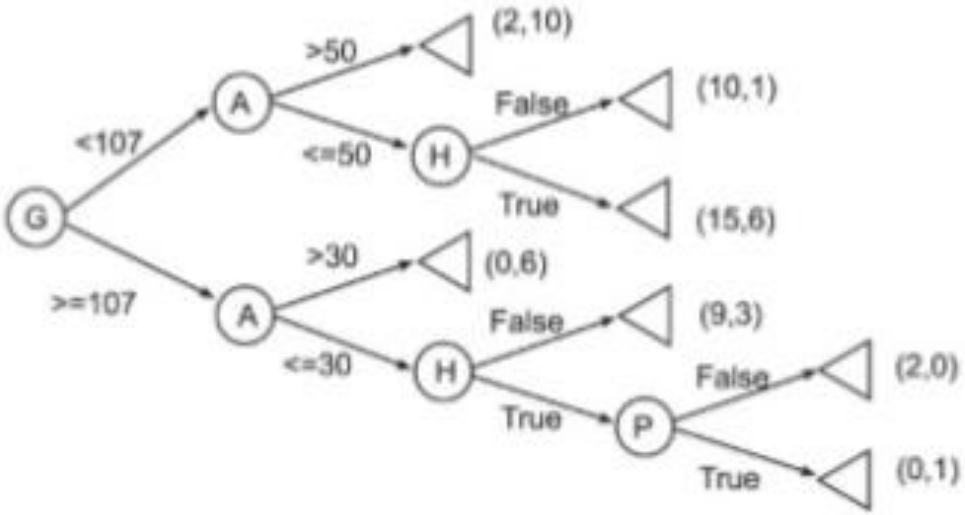

Рисунок 4 Забывчивые деревья

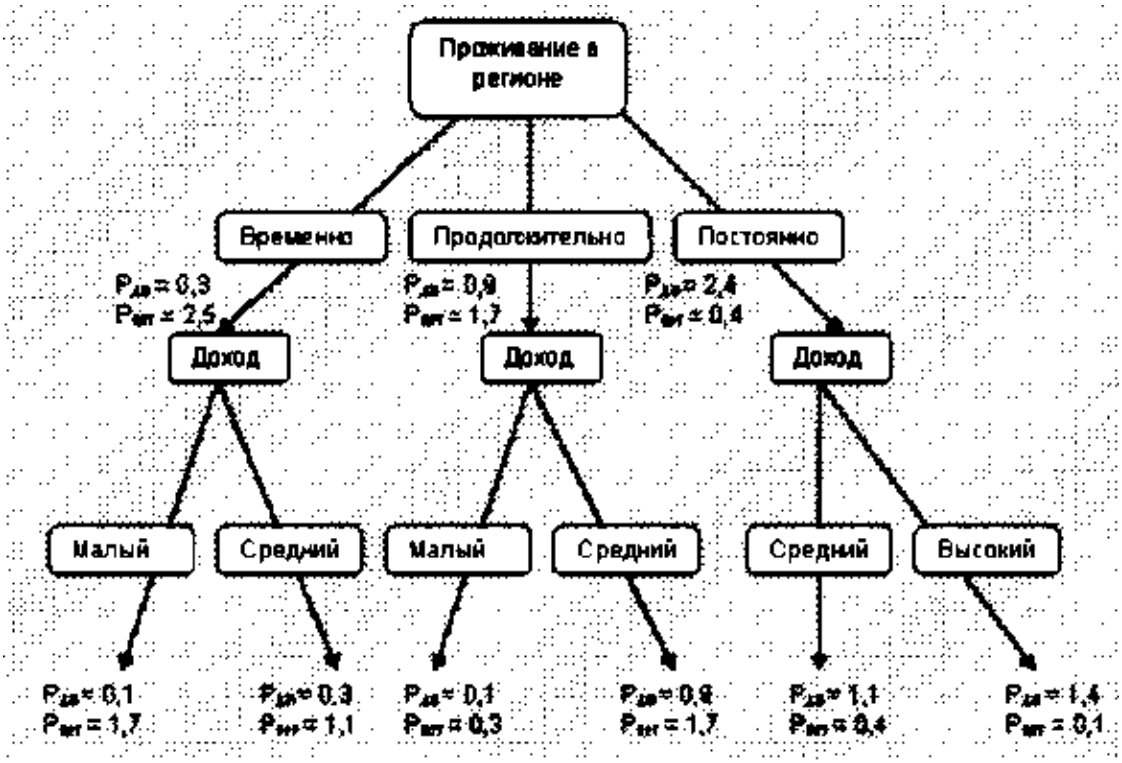

Рисунок 5 Нечёткие деревья

ОБОЗНАЕЧНИЯ И СОКРАЩЕНИЯ

Граф — абстрактный математический объект, представляющий собой множество вершин графа и набор *рёбер*, то есть соединений между парами вершин. Например, за множество вершин можно взять множество аэропортов, обслуживаемых некоторой авиакомпанией, а за множество рёбер взять

регулярные рейсы этой авиакомпании между городами. Граф называется связным, если для любых вершин есть путь друг к другу. Граф называется деревом, если он связный и не содержит нетривиальных циклов.

Дерево — это иерархическая структура данных, в которой каждый узел имеет значение,оно же является в данном случае и ключом, и ссылки на левого и правого потомка. Узел, находящийся на самом верхнем уровне (не являющийся чьим-либо потомком) называется корнем. Узлы, не имеющие потомков, оба потомка которых равны нулю называются листьями.

Двоичное дерево — это древовидная структура данных, в которой каждый узел имеет не более двух дочерних элементов, которые называются левым дочерним элементом и правым дочерним элементом. Двоичное дерево также можно интерпретировать как неориентированный, а не направленный граф, и в этом случае двоичное дерево является упорядоченным корневым деревом. Некоторые авторы используют корневое двоичное дерево вместо двоичного дерева, чтобы подчеркнуть тот факт, что дерево укоренено, но, как определено выше, двоичное дерево всегда укоренено. Двоичное дерево является частным случаем упорядоченного N-арного дерева, где n равно 2.

Теория принятия решений — область исследования, вовлекающая понятия и методы математики, статистики, экономики, менеджмента и психологи и с целью изучения закономерностей выбора людьми путей решения разного рода задач, а также способов поиска наиболее выгодных из возможных решений.

Принятие решения — это процесс рационального или иррационального выбора альтернатив, имеющий целью достижение осознаваемого результата. Различают нормативную теорию, которая описывает рациональный процесс принятия решения и дескриптивную теорию, описывающую практику принятия решений.

1 АЛГОРИТМЫ ПОСТРОЕНИЯ ДЕРЕВЬЕВ

1.1 ID3

Полное название – Третий Итерационный Дихотомизатор. Разработан Джоном Квинланом, который впоследствии создал усовершенствованную версию — алгоритм C4.5.

Описание Алгоритма

ID3(Таблица примеров, Целевой признак, Признаки)

1. Если все примеры положительны, то возвратить узел с меткой «+».
2. Если все примеры отрицательны, то возвратить узел с меткой «-».
3. Если множество признаков пустое, то возвратить узел с меткой, которая больше других встречается в значениях целевого признака в примерах.
4. Иначе:
    A. A — признак, который лучше всего классифицирует примеры (с максимальной информационной выгодой).
    B. Создать корень дерева решения; признаком в корне будет являться A.
    C. Для каждого возможного значения A(Vi):
        i. Добавить новую ветвь дерева ниже корня с узлом со значением A = Vi
        ii. Выделить подмножество Ex(Vi) примеров, у которых A=Vi
        iii. Если подмножество примеров пусто, то ниже этой новой ветви добавить узел с меткой, которая больше других встречается в значениях целевого признака в примерах.
        iv. Иначе, ниже этой новой ветви добавить поддерево, вызывая рекурсивно ID3(Ex(Vi), Целевой признак, Признаки)
5. Возвратить корень.

Свойства алгоритма

- Только классификация
- Энтропия
- Только категориальные или бинарные факторы
- Ветвление: по всем значениям фактора
- Остановка: все значения лежат в одном классе или невозможно разделить примеры

Достоинства и недостатки метода

Достоинства

- Алгоритм удовлетворяет всем данным

Недостатки

- Проблема критерия прироста информации
- Часть данных могут быть шумом или содержать ошибки и из-за этого дерево сильно растёт и хуже работает

Пример

Пусть команда «Зенит» дома выигрывает в 90% случаев, и ни отчего это больше не зависит.
И среди исходных данных имеется одно домашнее поражение
ID3 учтёт все «причины» и будет в дальнейшем предсказывать, что «Зенит» проиграет в аналогичных ситуациях.
Но на самом деле он будет выигрывать с вероятностью 90%

Решение недостатков: Обрезание

Алгоритм обрезания:

1. Обычно это делают так: ветку заменяют на значение, которое принимает большинство тестовых примеров в этой ветке.
2. Построим дерево по части исходных данных
3. Тестировать будем на оставшейся части
4. Для каждой вершины:
    a. Обрежем ветку с корнем в этой вершине
    b. Если обрезанное дерево будет лучше справляться с тестами, так и оставим обрезанную ветку, иначе вернём, как было

1.2 C4.5

C4.5 является расширением более раннего алгоритма ID3 Росса Квинлана. Деревья решений, сгенерированные C4. 5, могут использоваться для классификации, и по этой причине C4.5 часто называют статистическим классификатором. В 2011 году авторы программного обеспечения машинного обучения Weka описали алгоритм C4.5 как "знаковую программу дерева решений, которая, вероятно, является рабочей лошадкой машинного обучения, наиболее широко используемой на практике на сегодняшний день".

Алгоритм

C4.5 строит деревья решений из набора обучающих данных так же, как ID3, используя концепцию информационной энтропии. Обучающие данные представляют собой набор уже классифицированных данных S = s1, s2, s3…

Каждый элемент Si-тый состоит из p-мерного вектора (X1i, X2i, X3i…Xpi), где Xj-тый представляет значения атрибутов или признаков классифицированных данных, а также класс, в который попадает Si-тый вектор.

В каждом узле дерева алгоритм выбирает атрибут данных, который наиболее эффективно разбивает его набор выборок на подмножества, обогащенные в том или ином классе. Критерием расщепления является нормированный прирост информации (разница в энтропии). Для принятия решения выбирается атрибут с наибольшим нормированным коэффициентом усиления информации. Затем алгоритм рекурсивно проделывает работу на секционированных поддеревьях.

C4.5(T)

Input: training data set *T*; attributes *S*.

Output: decision tree *Tree*.

1: **if** *T* is NULL **then**
2: **return** failure
3: **end if**
4: **if** *S* is NULL **then**
5: **return** *Tree* as a single node with most frequent class label in *T*
6: **end if**
7: **if** $|S| = 1$ **then**
8: **return** Tree as a single node *S*
9: **end if**
10: set *Tree* = {}
11: **for** a ∈ *S* **do**
12: set $Info(a,T)=0$, and $SplitInfo(a,T)= 0$
13: compute *Entropy*(a)
14: for $v \in values(a,T)$ do
15: set $T_{a,v}$ as the subset of T with attribute $a = v$
16: $Info(a,T) += \frac{|Ta,v|}{|Ta|} Entroph(a_v)$
17: $SplitInfo(a,T) += \frac{|Ta,v|}{|Ta|} \log \frac{|Ta,v|}{|Ta|}$
18: **end for**
19: $Gain(a,T) = Entropy(a) - Info(a,T)$
20: $GainRatio(a,T) = \frac{Gain(a,T)}{SplitInfo(a,T)}$
21: **end for**
22: set $a_{best} = \underset{a}{argmax} \{GainRatio(a,T)\}$
23: attach a_{best} into *Tree*
24: **for** $v \in values(a_{best}, T)$ **do**
25: call $C4.5(Ta,v)$
26: **end for**
27: **return** *Tree*

Рисунок 6 Псевдокод алгоритма C4.5

Этот алгоритм имеет несколько базовых случаев.

1. Все образцы в списке принадлежат к одному классу. Когда это происходит, он просто создает конечный узел для дерева решений, говорящего, чтобы выбрать этот класс.
2. Ни одна из функций не обеспечивает получение какой-либо информации. В этом случае алгоритм создает узел решения выше по дереву, используя ожидаемое значение класса.

3. Обнаружен экземпляр ранее невидимого класса. Опять же, C4. 5 создает узел решения выше по дереву, используя ожидаемое значение.

В псевдокоде общий алгоритм построения деревьев решений выглядит следующим образом:

1. Проверить вышеуказанные базовые случаи.
2. Для каждого атрибута a найти нормализованный коэффициент усиления информации от разбиения на a.
3. Пусть переменная a_best - атрибут с наибольшим нормализованным коэффициентом усиления информации.
4. Создать узел решения, который разделяется на a_best.
5. Повторить в подсписках, полученных путем разбиения на a_best,и затем добавитт эти узлы в качестве дочерних узлов узла.

Достоинства и недостатки метода

Достоинства

Обработка как непрерывных, так и дискретных атрибутов - для обработки непрерывных атрибутов C4.5 создает порог и затем разбивает список на те, значение атрибута которых выше порога, и те, которые меньше или равны ему.

- Обработка обучающих данных с пропущенными значениями атрибутов-C4. 5 позволяет помечать значения атрибутов за пропажу. Отсутствующие значения атрибутов просто не используются в расчетах коэффициента усиления и энтропии.
- Обработка атрибутов с различными затратами.
- После создания дерево производится обрезка деревьев – алгоритм возвращается через дерево после его создания и пытается удалить ветви, которые не помогают, заменяя их листовыми узлами.

Недостатки

Ветвистые деревья, в случае если примеры представлены очень большим количеством атрибутов. Этот недостаток можно устранить, применив методику отсечения ветвей.

1.3 CART

Алгоритм CART (Дерево классификации и регрессии), как видно из названия, решает задачи классификации и регрессии построением дерева решений. Он разработан в 1974—1984 годах четырьмя профессорами статистики: Лео Брейманом (Беркли), Джеромом Фридманом (Jerome H. Friedman,Стэнфорд), Чарлзом Стоуном (Charles Stone, Беркли) и Ричардом Олшеном (Richard A. Olshen, Стэнфорд).

Алгоритм

Алгоритм CART предназначен для построения бинарного дерева решений. Для алгоритма CART «поведение» объектов выделенной группы означает долю модального значения выходного признака. Выделенные группы — те, для которых эта доля достаточно высока. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров на две части — часть, в которой выполняется правило (потомок — right) и часть, в которой правило не выполняется (потомок — left).

Преимуществом алгоритма CART является определенная гарантия того, что, если искомые детерминации существуют в исследуемой совокупности, то они будут выявлены. Кроме того, CART позволяет не «замыкаться» на единственном значении выходного признака, а искать все такие его значения, для которых можно найти соответствующее объясняющее выражение.

Метод CART применяется для номинальных (обычно двухуровневых) и порядковых предикторных переменных. В этом методе перебираются все возможные варианты ветвления для каждого узла, и выбирается та предикторная переменная, при которой оценочная функция дает наилучший показатель.

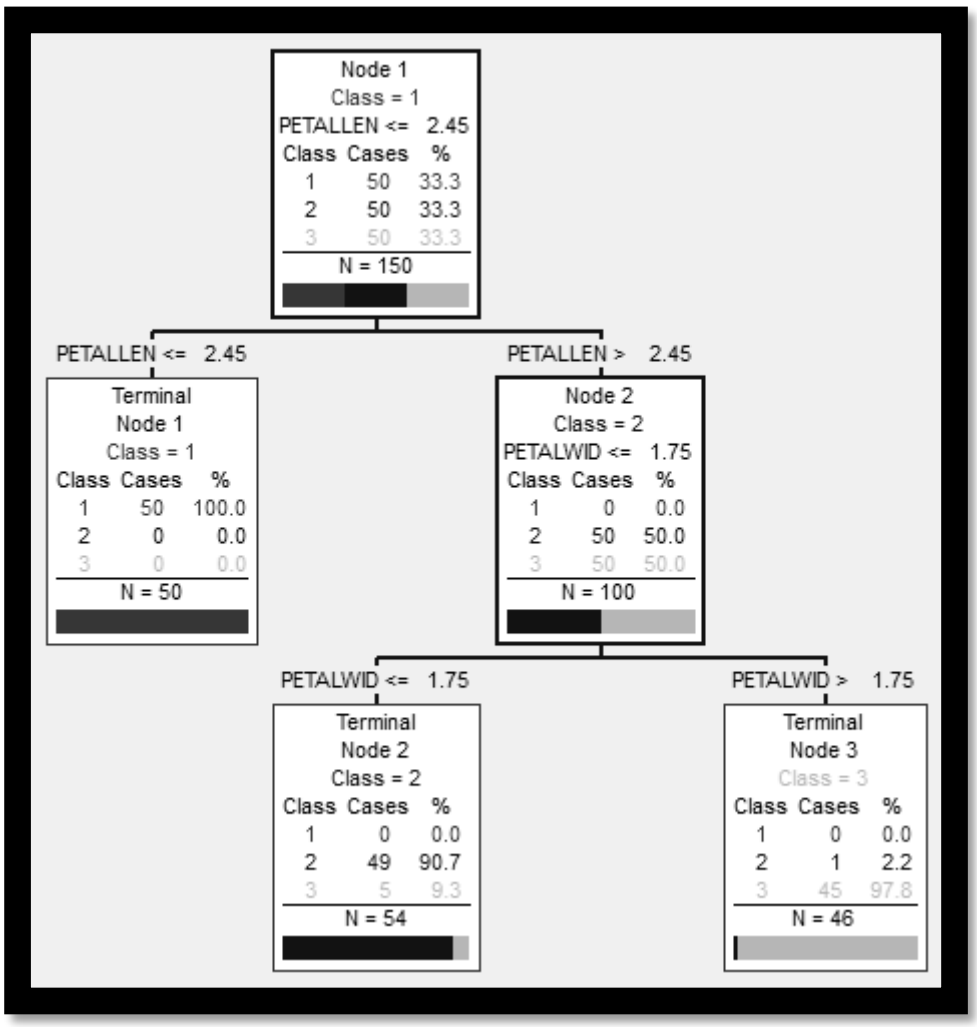

Рисунок 7 Работа алгоритма CART на классификации ирисов

Достоинства и недостатки метода

Достоинства

1. Данный метод является непараметрическим, это значит, что для его применения нет необходимости рассчитывать различные параметры вероятностного распределения.

2. Для применения алгоритма CART нет необходимости заранее выбирать переменные, который будут участвовать в анализе: переменные отбираются непосредственно во время проведения анализа на основании значения индекса Gini.

3. CART легко борется с выбросами: механизм «разбиения» (от англ. splitting), заложенный в алгоритме просто помещает «выбросы» в отдельный узел, что позволяет очистить имеющиеся данные от шумов.

4. Для применения этого алгоритма не надо принимать в расчет никаких предположений или допущений перед проведением анализа.

5. Большим преимуществом является скорость работы алгоритма.

Недостатки

1. Деревья решений, предложенные алгоритмом, не являются стабильными: результат, полученный на одной выборке, бывает не воспроизводим не другой (дерево может увеличиваться, уменьшаться, включать другие предикторы и т.д..)

2. В случае, когда необходимо построить дерево с более сложной структурой, лучше использовать другие алгоритмы, так как CART может не идентифицировать правильную структуру данных.

1.4 Random Forest

Случайный лес является композицией (ансамблем) множества решающих деревьев, что позволяет снизить проблему переобучения и повысить точность в сравнении с одним деревом. Прогноз получается в результате агрегирования ответов множества деревьев. Тренировка деревьев происходит независимо друг от друга (на разных подмножествах), что не просто решает проблему построения одинаковых деревьев на одном и том же наборе данных, но и делает этот алгоритм весьма удобным для применения в системах распределённых вычислений.

Суть алгоритма, что на каждой итерации делается случайная выборка переменных, после чего, на этой новой выборке запускают построение дерева принятия решений. При этом производится выборка случайных двух третей наблюдений для обучения, а оставшаяся треть используется для оценки результата. Такую операцию проделывают сотни или тысячи раз. Результирующая модель будет результатом “голосования” набора полученных при моделировании деревьев

Достоинства и недостатки метода

Достоинства

1. Высокое качество результата, особенно для данных с большим количеством переменных и малым количеством наблюдений.
2. Возможность распараллелить
3. Не требуется тестовая выборка

Недостатки

1. Каждое из деревьев огромное, в результате модель получается огромная
2. Долгое построение модели, для достижения хороших результатов.
3. Сложная интерпретация модели (Сотни или тысячи больших деревьев сложны для интерпретации)

1.5 CHAID

CHAID (Автоматическое обнаружение взаимодействия хи-квадрат) — это метод построения дерева решений, основанный на скорректированном тестировании значимости. Методика была разработана в Южной Африке и опубликована в 1980 году Гордоном В. Кассом, который защитил кандидатскую диссертацию на эту тему. CHAID может использоваться для прогнозирования, а также для классификации и для обнаружения корреляции между переменными. CHAID основан на формальном расширении процедур помощи AID (автоматическое обнаружение взаимодействия) и THAID (автоматическое обнаружение взаимодействия THeta) 1960-х и 1970-х годов, которые, в свою очередь, были продолжениями более ранних исследований, в том числе выполненных в Великобритании в 1950-х годах.

На практике CHAID часто используется в контексте прямого маркетинга для выбора групп потребителей и прогнозирования того, как их ответы на некоторые переменные влияют на другие переменные, хотя другие ранние приложения были в области медицинских и психиатрических исследований.

Как и другие деревья принятия решений, преимущества CHAID заключаются в том, что его результат очень нагляден и легко интерпретируется. Поскольку по умолчанию он использует многоходовые разбиения, для эффективной работы ему требуются довольно большие размеры выборки, поскольку при малых размерах выборки группы респондентов могут быстро стать слишком малыми для надежного анализа.

Одним из важных преимуществ CHAID перед альтернативами, такими как множественная регрессия, является то, что он непараметрический.

При анализе методом CHAID существует две опции для проверки модели:

- Разбиение – разбивка данных на две части: одна для построения модели, а bвторая – для проверки (применяется при большой выборке).
- N-кратная перекрестная проверка – применяется в случае небольшой выборки (менее 1000 наблюдений). Весь набор данных разбивается на n-выборок (обычно 10) приблизительно равного объема. Затем строятся n-деревьев с последовательным исключением каждой из выборок.

Алгоритм

1. Поиск самого сильного фактора, который наибольшим образом объясняет различия.
2. Перебор всех заданных предикторов, поиск комбинаций значений и нахождение лучшего результата (который максимизирует различия). Выделение групп по найденному лучшему результату.
3. Повторение процесса (пунктов 1 и 2) с целью нахождения оптимального решения для второго уровня и т.д. для всех возможных уровней.
4. Представление результатов в виде дерева решений.

Достоинства и недостатки метода

Достоинства

1. Метод работает с переменными всех типов, даже с номинальными (в отличие от других методов сегментации, в первую очередь, кластерного анализа)
2. Широкая сфера применимости деревьев классификации делает их весьма привлекательным инструментом анализа данных
3. Как метод разведочного анализа или как последнее средство, когда отказывают все традиционные методы, деревья классификации, по мнению многих исследователей, не знают себе равных

2 ДОСТОИНСТВА И НЕДОСТАТКИ ДЕРЕВЬЕВ РЕШЕНИЙ

Достоинства

- Прост в понимании и интерпретации. Люди способны интерпретировать результаты модели дерева принятия решений после краткого объяснения
- Не требует подготовки данных. Прочие техники требуют 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». Если определённая ситуация наблюдается в модели, то её можно объяснить при помощи булевой логики. Примером «черного ящика» может быть искусственная нейронная сеть, так как результаты данной модели поддаются объяснению с трудом.
- Позволяет оценить модель при помощи статистических тестов. Это даёт возможность оценить надёжность модели.
- Является надёжным методом. Метод хорошо работает даже в том случае, если были нарушены первоначальные предположения, включённые 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

1. Проблема получения оптимального дерева решений является NP-полной 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.

2. В процессе построения дерева решений могут создаваться слишком сложные конструкции, которые недостаточно полно представляют данные. Данная проблема называется переобучением. Для того, чтобы её избежать, необходимо использовать метод «регулирования глубины дерева».
3. Существуют концепты, которые сложно понять из модели, так как модель описывает их сложным путём. Данное явление может быть вызвано проблемами XOR, чётности или мультиплексарности. В этом случае мы имеем дело с непомерно большими деревьями. Существует несколько подходов решения данной проблемы, например, попытка изменить репрезентацию концепта в модели (составление новых суждений), или использование алгоритмов, которые более полно описывают и репрезентируют концепт (например, метод статистических отношений, индуктивная логика программирования).
4. Для данных, которые включают категориальные переменные с большим набором уровней (закрытий), больший информационный вес присваивается тем атрибутам, которые имеют большее количество уровней.

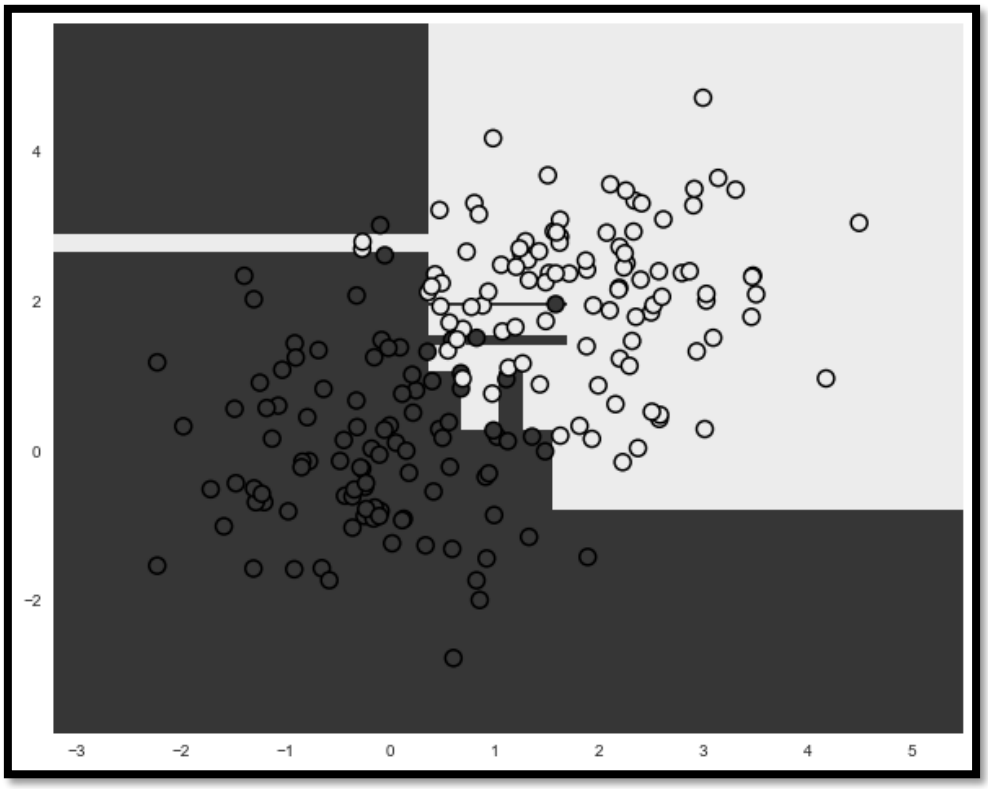

Рисунок 8 Пример разделяющей границы, построенной переобученным деревом.

3 РЕГУЛИРОВАНИЕ ГЛУБИНЫ ДЕРЕВА

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.

Необходимо отметить, что регулирование глубины дерева должно уменьшить размер обучающей модели дерева без уменьшения точности ее прогноза или с помощью перекрестной проверки. Есть много методов регулирования глубины дерева, которые отличаются измерением оптимизации производительности.

Методы регулирования

Сокращение дерева может осуществляться сверху вниз или снизу вверх. Сверху вниз — обрезка начинается с корня, снизу вверх — сокращается число листьев дерева. Один из простейших методов регулирования — уменьшение ошибки ограничения дерева. Начиная с листьев, каждый узел заменяется на самый популярный класс. Если изменение не влияет на точность предсказания, то оно сохраняется.

ВЫВОДЫ И ЗАКЛЮЧЕНИЯ

Главным достоинством деревьев решений является, получаемая в результате, структура предикатов, которая позволяет интерпретировать результаты классификации (хотя в силу своей «жадности», описанный алгоритм, не всегда позволяет обеспечить оптимальность дерева в целом).

На практике в результате работы этих алгоритмов часто получаются слишком детализированные деревья, которые при их дальнейшем применении дают много ошибок. Это связано с явлением переобучения. Для сокращения деревьев используется отсечение ветвей.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

MEDIUM [Электронный ресурс]. – Электронные данные. – Режим доступа: https://medium.com/greyatom/decision-trees/– Дата доступа: 05.04.18.

И. Куралёнок, Н. Поваров – Деревья решений.

Паклин Н.Б., Орешков В.И. Глава 9. // Аналитика деревьев: от данных к знаниям(+CD): Учебное пособие. 2-е изд.. — СПб: Питер, 2013. — С. 444-459. — ISBN 978-5-459-00717-6.

Левитин А. В. Глава 10. Ограничения мощи алгоритмов: Деревья принятия решения // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 409–417. — 576 с. — ISBN 978-5-8459-0987-9