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 3](#_Toc483786974)

[Обознаечния и сокращения 4](#_Toc483786974)

[1 Алгоритмы построения деревьев](#_Toc483786975) 5

[1.1 ID3 5](#_Toc483786976)

[1.2 C4.5 9](#_Toc483786977)

[1.3 CART 9](#_Toc483786977)

[1.4 Random Forest 9](#_Toc483786977)

[1.5 CHAID 9](#_Toc483786977)

[2 Обучение дерева решений](#_Toc483786975) 5

[Выводы и заключения 66](#_Toc483786999)

[Список использованной литературы 67](#_Toc483786999)

**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.

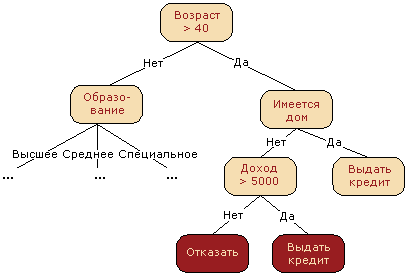


Рисунок 1 Пример дерева решений

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).

Cвойства 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).

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

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

Дикие Пеньки Забывчивые Нечёткие

**ОБОЗНАЕЧНИЯ И СОКРАЩЕНИЯ**

 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.

1 АЛГОРИТМЫ ПОСТРОЕНИЯ ДЕРЕВЬЕВ

1.1 ID3

Полное название - Iterative Dichotomiser 3. Разработан Джоном Р. Квинланом. Впоследствии Квинлан 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:
   1. A — признак, который лучше всего классифицирует примеры (с максимальной информационной выгодой).
   2. Создать корень дерева решения; признаком в корне будет являться {\displaystyle A}A.
   3. Для каждого возможного значения {\displaystyle A}A({\displaystyle v\_{i}}Vi):
      1. Добавить новую ветвь дерева ниже корня с узлом со значением {\displaystyle A=v\_{i}}A = Vi
      2. Выделить подмножество {\displaystyle Examples(v\_{i})}Ex(Vi) примеров, у которых {\displaystyle A=v\_{i}}A=Vi
      3. Если подмножество примеров пусто, то ниже этой новой ветви добавить узел с меткой, которая больше других встречается в значениях целевого признака в примерах.
      4. Иначе, ниже этой новой ветви добавить поддерево, вызывая рекурсивно ID3({\displaystyle Examples(v\_{i})}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 алгоритма**

* Алгоритм удовлетворяет всем данным

**Недостатки алгоритма**

* 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

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:
   1. Обрежем ветку с корнем в этой вершине
   2. Если обрезанное дерево будет лучше справляться с тестами, так и оставим обрезанную ветку, иначе вернём, как было

1.2 C4.5

Successor of ID3

Только классификация

Энтропия

Любые факторы

Ветвление: для числовых используем границу разбиения, а остальные по старому

Остановка: все значения лежат в одном классе или невозможно разделить примеры или нет изменения критерия информативности

Стрижка по CV

C4.5 Ross Quinlan Строим классификатор. Алгоритм: T: T(l) = − c∈C p(c) log p(c) X: дискретные и непрерывные фичи, веса точек, не все точки размечены E: пока в листе не окажется либо ∅, либо все значения будут в одном классе L: создаем листья по количеству возможных значений фактора (в случае непрерывной фичи используем разницу не менее α), если множество пустое, то используем значение, которое чаще всего встречается на предыдущем уровне P: По CV сворачиваем ветви, которые не работают

1.3 CART

CART I Classiﬁcation and Regression Tree (L. Breiman, J. Friedman, et.al.) Строим классификатор или регрессию. Дерево бинарное. Алгоритм: T: T(l) X: все возможные разбиения по каждому фактору E: пока можно найти разбиение lt+1 : T(lt) > T(lt+1) L: создаем 2 листа с средневыборочным значением в каждом P: нет

И классификация и регрессия

Дерево бинарное

Gini impurity или дисперсия

Ветвление: граница по всем возможным значениям

Остановка: пока есть изменение критерия информативности

Стрижка по CV

Тонкости На практике мы используем варианты CART, поэтому дальше только о нем. Одинаковые по форме деревья решений могут нести существенно разную информацию Уверенность в значении в листе не всегда можно просто вычислить Не всегда оптимально назначать средневыборочное как значение в листе

1.4 MARS

1.5 CHAID

2 ОБУЧЕНИЕ ДЕРЕВА РЕШЕНИЙ

2 ДОСТОИНСТВА И НЕДОСТАТКИ

Достоинства метода:

* 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 метода

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.

**ВЫВОДЫ И ЗАКЛЮЧЕНИЯ**

Для того, чтобы правильно спроектировать систему необходимо грамотно проработать предметную область, также разработать основной процесс предметной области. Это позволит подробно разобраться и понять, что должно выполнять приложение.

Важным этапом является проектирование такого интерфейса системы, чтобы подготовленному пользователю было привычно пользоваться приложением.

Система может быть улучшена за счет добавления алгоритмов докига. Такой функционал уместен при профессиональной и коммерческой разработке препаратов. Такой функционал может быть реализован в виде мобильного приложения, синхронизирующийся с основной шиной данных.

Согласно проведенному технико-экономическому обоснованию разрабатываемая система поддержки работы департамента взыскания является экономически эффективной. По расчетам выявлено, что все дополнительные капитальные затраты на освоение, сопровождение и адаптацию нового ПО окупятся в течение второго года эксплуатации. Положительный экономический эффект достигнут за счет экономии затрат на заработную плату.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

[1] БИОМОЛЕКУЛА [Электронный ресурс]. – Электронные данные. – Режим доступа: https://biomolecula.ru/ – Дата доступа: 05.04.18.

[10] И. Куралёнок, Н. Поваров – Деревья решений.

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

**4.2 Дерево решений**