Алгоритмы создания дерева принятия решений.

Автор: Сидоров А.В.

Научный руководитель: Миронова Ю.Н.

Теория принятия решений — область исследования, вовлекающая понятия и
методыматематики, 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.

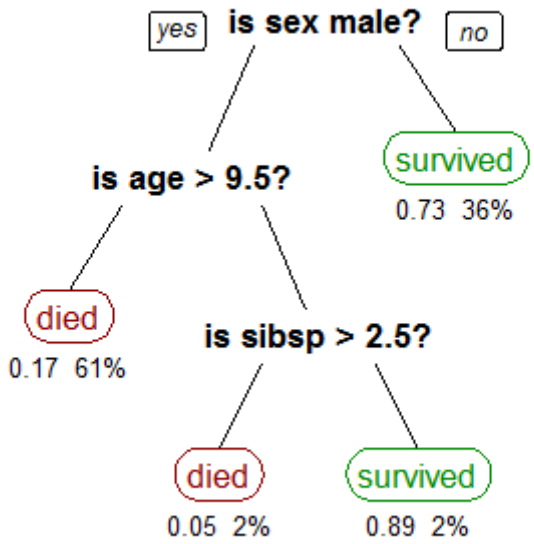
Принятие решения — это процесс рационального или иррационального выбора альтернатив, имеющий целью достижение осознаваемого результата. Различают *нормативную теорию*, которая описывает рациональный процесс принятия решения
и *дескриптивную теорию*, описывающую практику принятия решений.

Категории ТПР:

- Дерево принятия решений
- C4.5
- CART (алгоритм)

Дерево принятия решений:

Дерево принятия решений (также могут называться деревьями классификации или регрессионными деревьями) — используется в области статистики и анализа данных для прогнозных моделей. Структура дерева представляет собой следующее: «листья» и «ветки». На ребрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.



Каждый лист представляет собой значение целевой переменной, измененной в ходе движения от корня по листу. Каждый внутренний узел соответствует одной из входных переменных. Дерево может быть также «изучено» разделением исходных наборов переменных на подмножества, основанные на тестировании значений атрибутов. Это процесс, который повторяется на каждом из полученных подмножеств. Рекурсия завершается тогда, когда подмножество в узле имеет те же значения целевой переменной, таким образом, оно не добавляет ценности для предсказаний. Процесс, идущий «сверху вниз», индукция деревьев решений (TDIDT), является примером поглощающего «жадного» алгоритма, и на сегодняшний день является наиболее распространенной стратегией деревьев решений для данных, но это не единственная возможная стратегия. В интеллектуальном анализе данных, деревья решений могут быть использованы в качестве математических и вычислительных методов, чтобы помочь описать, классифицировать и обобщить набор данных, которые могут быть записаны следующим образом:

$$(x, Y) = (x_1, x_2, x_3 \dots x_k, Y)$$

Зависимая переменная Y является целевой переменной, которую необходимо проанализировать, классифицировать и обобщить. Вектор x состоит из входных переменных x_1, x_2, x_3 и т. д., которые используются для выполнения этой задачи.

Типология деревьев

Деревья решений, используемые в Data Mining, бывают двух основных типов:

- Анализ дерева классификации, когда предсказываемый результат является классом, к которому принадлежат данные;
- Регрессионный анализ дерева, когда предсказанный результат можно рассматривать как вещественное число (например, цена на дом, или продолжительность пребывания пациента в больнице).

Упомянутые выше термины впервые были введены Брейманом и др. Перечисленные типы имеют некоторые сходства, а также некоторые различия, такие, как процедура используется для определения, где разбивать. Некоторые методы позволяют построить более одного дерева решений:

1. Деревья решений «мешок», наиболее раннее дерево решений, строит несколько деревьев решений, неоднократно интерполируя данные с заменой, и деревья голосований для прогноза консенсуса;
2. Случайный классификатор «лесной» использует ряд деревьев решений, с целью улучшения ставки классификации;
3. «Повышенные» деревья могут быть использованы для регрессионного типа и классификации типа проблем.
4. «Вращение леса» — деревья, в которых каждое дерево решений анализируется первым применением метода главных компонент (PCA) на случайные подмножества входных функций.

Алгоритмы построения дерева

Общая схема построения дерева принятия решений по тестовым примерам выглядит следующим образом:

- Выбираем очередной атрибут Q, помещаем его в корень.
- Для всех его значений i:
  - Оставляем из тестовых примеров только те, у которых значение атрибута Q равно i
  - Рекурсивно строим дерево в этом потомке

Основной вопрос: как выбирать очередной атрибут?

Есть различные способы выбирать очередной атрибут:

- Алгоритм ID3, где выбор атрибута происходит на основании прироста информации (англ. *Gain*), либо на основании индекса Гини.
- Алгоритм C4.5 (улучшенная версия ID3), где выбор атрибута происходит на основании нормализованного прироста информации
- Алгоритм CART и его модификации — IndCART, DB-CART.

- Автоматический детектор взаимодействия Хи-квадрат (CHAID). Выполняет многоуровневое разделение при расчете классификации деревьев;
- MARS: расширяет деревья решений для улучшения обработки цифровых данных.

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 .

Достоинства метода

Среди прочих методов Data Mining, метод дерева принятия решений имеет несколько достоинств:

- 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 применен только на номинальных переменных, а метод нейронных сетей только на переменных, измеренных по интервальной шкале.
- Использует модель «белого ящика». Если определенная 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 дает возможность оценить надежность модели.
- Является надежным методом. Метод хорошо работает даже в том случае, если были нарушены первоначальные предположения, включенные в модель.
- Позволяет работать с большим объемом 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 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.
- Те, кто изучает метод дерева принятия решений, могут создавать слишком сложные конструкции, которые недостаточно полно представляют данные. Данная проблема называется . Для того, чтобы избежать данной проблемы, необходимо использовать Метод «регулирования глубины дерева».
- Существуют концепты, которые сложно понять из модели, так как модель описывает их сложным путем. Данное явление может быть вызвано проблемами XOR, четности 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).
- Для данных, которые включают категориальные переменные с большим набором уровней (закрытий), больший информационный вес присваивается тем атрибутам, которые имеют большее количество уровней.

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[12].

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.

Пример задачи

Предположим, что нас интересует, выиграет ли наша любимая футбольная команда следующий матч. Мы знаем, что это зависит от ряда параметров; перечислять их все — задача безнадежная, поэтому ограничимся основными:

- выше ли находится соперник по турнирной таблице;
- дома ли играется матч;
- пропускает ли матч кто-либо из лидеров команды;
- идет ли дождь.

У нас есть некоторая статистика на этот счет:

Соперник	Играем	Лидеры	Дождь	Победа
Выше	Дома	На месте	Да	Нет
Выше	Дома	На месте	Нет	Да
Выше	Дома	Пропускают	Нет	Нет
Ниже	Дома	Пропускают	Нет	Да
Ниже	В гостях	Пропускают	Нет	Нет
Ниже	Дома	Пропускают	Да	Да
Выше	В гостях	На месте	Да	Нет

Ниже	В гостях	На месте	Нет	???

C4.5

C4.5 — алгоритм для построения деревьев решений, разработанный Джоном Квинланом. C4.5 является усовершенствованной версией алгоритма ID3 того же автора. В частности, в новую версию были добавлены отсечение ветвей, возможность работы с числовыми атрибутами, а также возможность построения дерева из неполной обучающей выборки, в которой отсутствуют значения некоторых атрибутов**.** Требования к данным

Для того, чтобы с помощью C4.5 построить решающее дерево и применять его, данные должны удовлетворять нескольким условиям.

Информация об объектах, которые необходимо классифицировать, должна быть представлена в виде конечного набора признаков (*атрибутов*), каждый из которых имеет дискретное или числовое значение. Такой набор атрибутов назовём *примером*. Для всех примеров количество атрибутов и их состав должны быть постоянными.

Множество классов, на которые будут разбиваться примеры, должно иметь конечное число элементов, а каждый пример должен однозначно относиться к конкретному классу. Для случаев с нечёткой логикой, когда примеры принадлежат к классу с некоторой вероятностью, C4.5 неприменим.

В обучающей выборке количество примеров должно быть значительно больше количества классов, к тому же каждый пример должен быть заранее ассоциирован со своим классом. По этой причине C4.5 является вариантом машинного обучения с учителем.

Построение дерева

Пусть имеется T — обучающая выборка примеров, а C — множество классов, состоящее из k элементов. Для каждого примера из T известна его принадлежность к какому-либо из классов $C_1 \dots C_k$.

Построение дерева решений алгоритмом C4.5 принципиально не отличается от его построения в ID3. На первом шаге имеется корень и ассоциированное с ним множество T, которое необходимо разбить на подмножества. Для этого необходимо выбрать один из атрибутов в качестве проверки. Выбранный атрибут A имеет n значений, что даёт разбиение на n подмножеств. Далее создаются n потомков корня, каждому из которых поставлено в соответствие своё подмножество,

полученное при разбиении T. Процедура выбора атрибута и разбиения по нему рекурсивноприменяется ко всем n потомкам и останавливается в двух случаях:

- после очередного ветвления в вершине оказываются примеры из одного класса (тогда она становится *листом*, а класс, которому принадлежат её примеры, будет решением листа),
- вершина оказалась ассоциированной с пустым множеством (тогда она становится листом, а в качестве решения выбирается наиболее часто встречающийся класс у непосредственного предка этой вершины).

Алгоритм **CART**

Алгоритм **CART** (Classification and Regression Tree), 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).

На сегодняшний день существует большое число алгоритмов, реализующих деревья
решений: CART, C4.5, CHAID, CN2, NewId, ITrule и другие.

Основной смысл алгоритма

Алгоритм CART предназначен для построения бинарного дерева решений. Бинарные деревья также называют двоичными, значит, что каждый узел дерева при разбиении имеет только двух потомков. 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.[3]

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.

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.

Список литературных источников:

http://ru.wikipedia.org/wiki/Дерево_принятия_решений

http://ru.wikipedia.org/wiki/C4.5

http://ru.wikipedia.org/wiki/CART_(алгоритм)