

Машинное обучение (Machine Learning)

Деревья решений (Decision trees)

Уткин Л.В.

Санкт-Петербургский политехнический университет Петра Великого

- 1 Определения и основные понятия и элементы деревьев решений
- 2 Алгоритм конструирования деревьев на примере алгоритма CART для классификации
- 3 Алгоритм конструирования деревьев для регрессии
- 4 Процедуры расщепления, остановки, сокращения дерева или отсечения ветвей
- 5 Наиболее известные алгоритмы
- 6 Достоинства и недостатки деревьев решений

Презентация является компиляцией и заимствованием материалов из замечательных курсов и презентаций по машинному обучению:

К.В. Воронцова, А.Г. Дьяконова, Н.Ю. Золотых, С.И. Николенко, Andrew Moore, Lior Rokach, Rong Jin, Luis F. Teixeira, Alexander Statnikov и других.

Общие определения деревьев решений

Деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение

Деревья решений – это логический алгоритм классификации, основанный на поиске конъюнктивных закономерностей.

Деревья решений

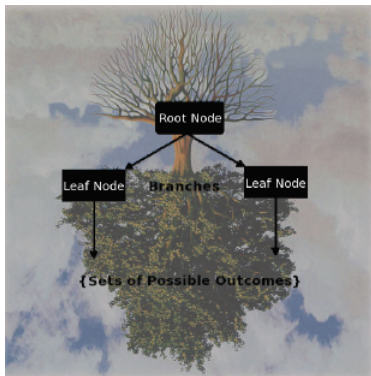


http://statweb.stanford.edu/~lpekelis/talks/13_datafest_cart_talk.pdf

Деревья решений (определение)

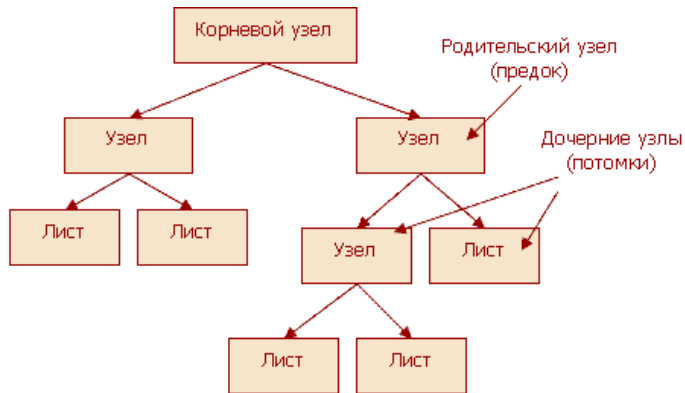
- Дерево решений - это классификатор в форме древовидной структуры, где каждая вершина либо:
 - Вершина решения (узел, промежуточная вершина) - характеризует некоторый тест для одного признака
 - Корень дерева
 - Лист (конечная вершина, терминальная вершина) - показывает значение класса обучающих примеров
- Дерево решений пытается классифицировать пример посредством серии вопросов
- Дерево называется бинарным, если из любой его внутренней вершины выходит ровно два ребра.

Деревья решений



http://statweb.stanford.edu/~lpekelis/talks/13_datafest_cart_talk.pdf

Элементы дерева



Определение бинарных деревьев

Бинарное решающее дерево – это алгоритм классификации, задающийся бинарным деревом, в котором каждой внутренней вершине $v \in V$ приписан предикат $\beta_v : X \rightarrow \{0, 1\}$, каждой терминальной вершине $v \in V$ приписано имя класса $c_v \in Y$. При классификации объекта $x \in X$ он проходит по дереву путь от корня до некоторого листа.

Применение деревьев решений

- **Описание данных:** ДР позволяют хранить информацию о данных в компактной форме, вместо них мы можем хранить ДР, которое содержит точное описание объектов.
- **Классификация:** ДР отлично справляются с задачами классификации, т.е. отнесения объектов к одному из заранее известных классов. Целевая переменная имеет дискретные значения.
- **Регрессия:** Если целевая переменная имеет непрерывные значения, ДР позволяют установить зависимость целевой переменной от входных переменных

Пример обучающей выборки (выдача кредита)

	возраст	наличие дома	доход	образование	кредит
x_1	32	нет	2000	среднее	нет
x_2	54	да	12000	высшее	да
x_3	73	нет	800	специальное	нет
...			
x_{50}	18	да	200	среднее	да

Пример дерева классификации (Выдавать ли кредит?)



Этапы конструирования деревьев

- 1 “Построение” или “создание” дерева (tree building):
выбор критерия расщепления и остановки обучения
- 2 “Сокращение” дерева (tree pruning): сокращения дерева и отсечение некоторых его ветвей

Критерий расщепления

- Расщепление должно разбивать исходное множество данных таким образом, чтобы объекты подмножеств, получаемых в результате этого разбиения, являлись представителями одного класса или же были максимально приближены к такому разбиению.
- Количество объектов из других классов, так называемых “примесей”, в каждом классе должно стремиться к минимуму.

Общий жадный алгоритм построения ДР

Жадный алгоритм – алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Алгоритм:

- 1 На каждой итерации для входного подмножества обучающего множества строится такое разбиение пространства гиперплоскостью (ортогональной одной из осей координат), которое минимизировало бы среднюю меру **неоднородности** двух полученных подмножеств.
- 2 Данная процедура выполняется **рекурсивно** для каждого полученного подмножества до тех пор, пока не будут достигнуты критерии остановки.

Алгоритм CART (Classification and Regression Tree) разработан в 1974-1984 годах L.Breiman (Berkeley), J.Friedman (Stanford), C.Stone (Berkeley) и R.Olshen (Stanford).

Алгоритм CART предназначен для построения *бинарного* дерева решений.

Особенности алгоритма CART:

- *функция оценки качества разбиения;*
- механизм отсечения дерева;
- построение деревьев регрессии.

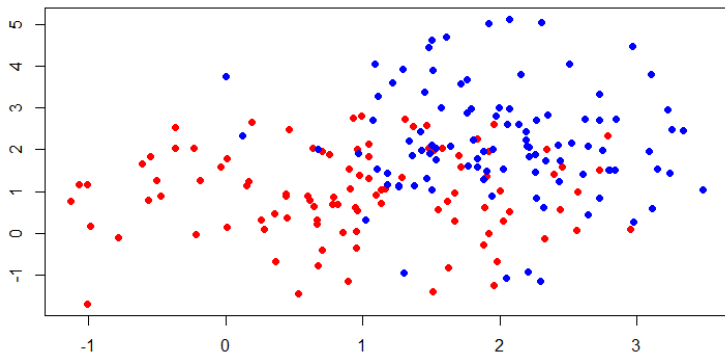
Процедура расщепления в алгоритме CART (1)

- 1 Выбирается k -ый признак f_k с множеством значений $X^{(k)}$.
- 2 Определяется такое значение $x_0^{(k)} \in X^{(k)}$ для всех признаков f_k , $k = 1, \dots, m$, чтобы мера неоднородности $\text{Gini}_{\text{split}}(T)$ была минимальной, т.е.

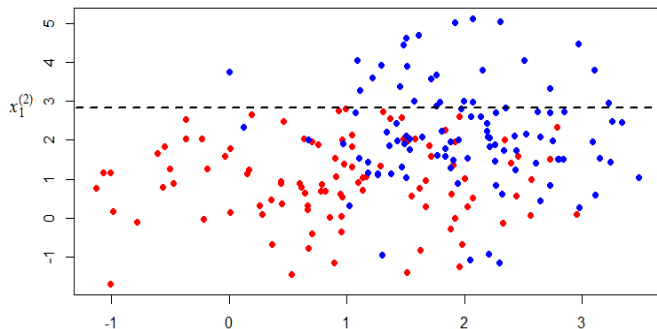
$$x_0^{(k)} = \arg \min_{f_k: X^{(k)} \in X^{(k)}} \text{Gini}_{\text{split}}(T, x^{(k)})$$

- 3 Данная процедура выполняется рекурсивно для каждого полученного подмножества до тех пор, пока не будут достигнуты критерии остановки.

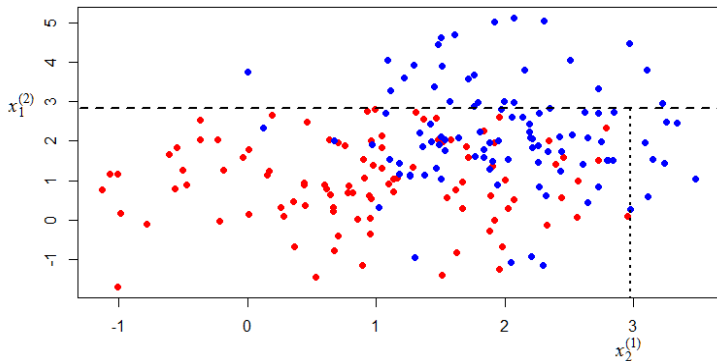
Процедура расщепления в алгоритме CART (2)



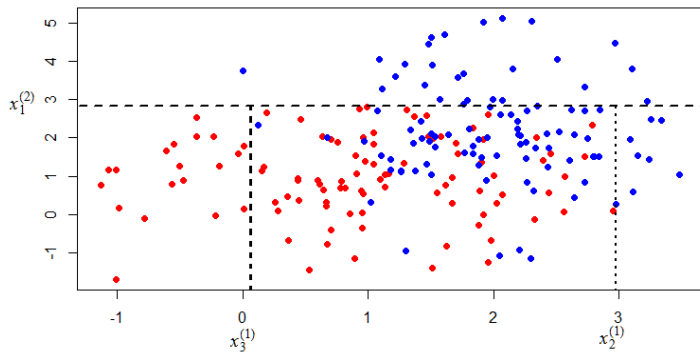
Процедура расщепления в алгоритме CART (3)



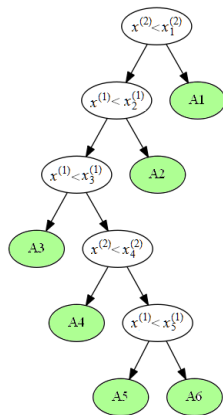
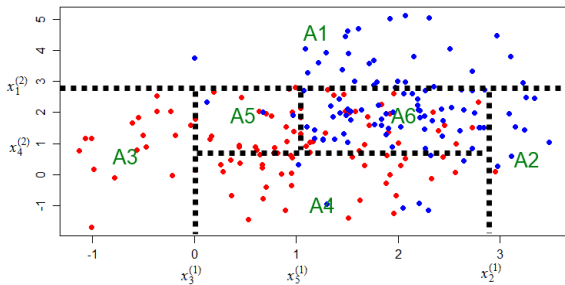
Процедура расщепления в алгоритме CART (4)



Процедура расщепления в алгоритме CART (5)



Процедура расщепления в алгоритме CART (6)



Критерий расщепления и алгоритм CART

Критерии расщепления или меры неоднородности множества относительно его меток:

- мера энтропии (cross-entropy): $-\sum_{i=1}^C p_i \log(p_i)$
- индекс Gini: $\sum_{i=1}^C p_i(1 - p_i)$

p_i - частота или вероятность точек i -го класса в блоке;
Если набор данных T разбивается на две части T_1 и T_2 с числом примеров в каждой N_1 и N_2 соответственно, тогда показатель качества разбиения будет равен:

$$\text{Gini}_{\text{split}}(T) = \frac{N_1}{N} \cdot \text{Gini}(T_1) + \frac{N_2}{N} \cdot \text{Gini}(T_2)$$

Чем меньше критерий расщепления, тем лучше расщепление.

Процедура расщепления в алгоритме CART (7)

- 1 Выбирается k -ый признак f_k с множеством значений $X^{(k)}$.
- 2 Определяется такое значение $x_0^{(k)} \in X^{(k)}$ для всех признаков f_k , $k = 1, \dots, m$, чтобы мера неоднородности $Gini_{\text{split}}(T)$ была минимальной, т.е.

$$x_0^{(k)} = \arg \min_{f_k, x^{(k)} \in X^{(k)}} Gini_{\text{split}}(T, x^{(k)})$$

- 3 Данная процедура выполняется рекурсивно для каждого полученного подмножества до тех пор, пока не будут достигнуты критерии остановки.

Остановка - такой момент в процессе построения дерева, когда следует прекратить дальнейшие ветвления

- достигнута максимальная глубина узла;
- вероятность доминирующего класса в разбиении превышает некоторый порог (например, 0.95);
- количество элементов в подмножестве меньше некоторого порога.

Сокращение дерева или отсечение ветвей

Сокращение - это компромисс между получением дерева "подходящего размера" и получением наиболее точной оценки классификации.

- Осуществляется путем **отсечения** (pruning) некоторых ветвей.
- Отсечение (прореживание) важно не только для упрощения деревьев, но и для избежания переобучения.

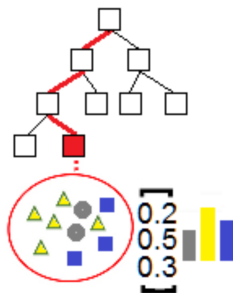
Основные характеристики алгоритма CART

- бинарное расщепление с критерием расщепления - индексом Gini,
- специальный механизм отсечения (minimalcost-complexity tree pruning),
- V-fold cross-validation,
- принцип “вырастить дерево, а затем сократить”,
- высокая скорость построения.

Другие известные алгоритмы

- **Алгоритм C4.5** строит дерево решений с неограниченным количеством ветвей у узла, может работать только с дискретным зависимым атрибутом, может решать только задачи классификации
- **Алгоритм ID3**. В основе лежит понятие информационной энтропии. Использует рекурсивное разбиение подмножеств в узлах дерева по одному из выбранных атрибутов.
- **Алгоритм MARS** (Multivariate adaptive regression splines).
- **Алгоритм CHAID** (CHi-squared Automatic Interaction Detection).

Вероятности классов

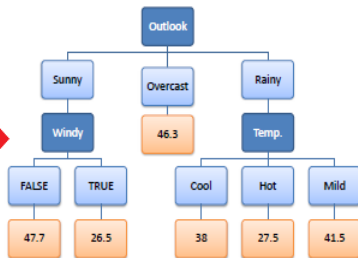


Деревья решений для регрессии (1)

- Основной алгоритм - ID3, предложенный J.R. Quinlan
- Идея - замена информационного критерия расщепления, например, энтропии, критерием понижения СКО (Standard Deviation Reduction).
- СКО используется для определения однородности
- Идеальный случай - нулевое СКО

Деревья решений для регрессии (2)

Predictors				Target
Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	46
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Деревья решений для регрессии (3)

СКО для одного признака:

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$S = \sqrt{\frac{\sum (x - \mu)^2}{n}}$$



Standard Deviation

$$S = 9.32$$

Деревья решений для регрессии (4)

СКО для двух признаков:

$$S(T, X) = \sum_{c \in X} P(c) S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

$S(\text{Hours}, \text{Outlook})$

$$= P(\text{Sunny})S(\text{Sunny}) + P(\text{Overcast})S(\text{Overcast}) + P(\text{Rainy})S(\text{Rainy})$$

$$= (4/14) \cdot 3.49 + (5/14) \cdot 7.78 + (5/14) \cdot 10.87 = 7.66$$

Понижение СКО (1)

Поиск признака, который дает наибольшее понижение СКО

Шаг 1: Вычисляется СКО зависимой переменной

$$СКО(\text{Hours Played}) = 9.32$$

Шаг 2: Обучающая выборка разбивается по всем признакам. СКО для каждой ветви расщепления вычисляется. Результирующее СКО вычитается из СКО перед расщеплением. Это и есть понижение СКО.

Понижение СКО (2)

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
		SDR=1.66

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
		SDR=0.17

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
		SDR=0.28


		Hours Played (StDev)
Windy	False	7.87
	True	10.59
		SDR=0.29

$$SDR(T, X) = S(T) - S(T, X)$$

$$\begin{aligned}SDR(\text{Hours}, \text{Outlook}) &= S(\text{Hours}) - S(\text{Hours}, \text{Outlook}) \\ &= 9.32 - 7.66 = 1.66\end{aligned}$$

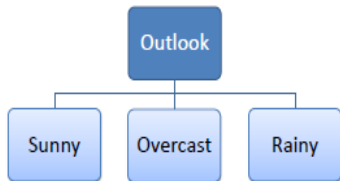
Понижение СКО (3)

Шаг 3: Признак с наибольшим снижением СКО выбирается для вершины решения

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

Понижение СКО (4)

Шаг 4 а: Обучающее множество разделяется на основе значений выбранного признака



Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44

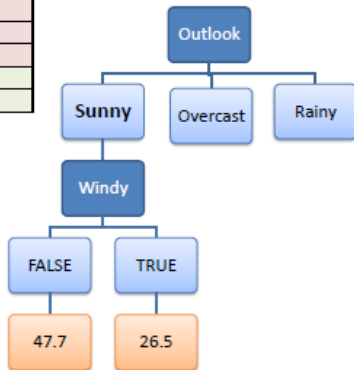
Шаг 4 b: Множество ветвей с СКО больше 0
расщепляется далее (на практике используется критерий
остановки расщепления, например, когда СКО для ветви
становится менее, чем 5% от СКО всего обучающего
множества или когда небольшое количество примеров
остается в одной ветви).

Понижение СКО (6)

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Mild	Normal	FALSE	46
Cool	Normal	TRUE	23
Mild	High	TRUE	30

★		Hours Played (StDev)
Windy	False	3.09
	True	3.50
SDR= 7.62		

$$SDR = 10.87 - ((3/5) * 3.09 + (2/5) * 3.5)$$



Шаг 5: Процесс повторяется рекурсивно. Когда число примеров на конечных вершинах более одного, вычисляется среднее как окончательное значение зависимой переменной.

Преимущества:

- интерпретируемость, допускаются разнотипные данные, возможность обхода пропусков;

Недостатки:

- переобучение, неустойчивость к шуму, составу выборки, критерию;

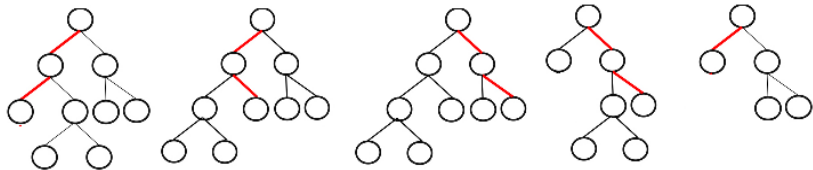
Способы устранения этих недостатков:

- редукция, композиции (леса) деревьев

Случайный лес



Случайный лес



- <https://cran.r-project.org/web/views/MachineLearning.html>
- Пакет **rpart**, функция **rpart**
- Пакет **C50**, функция **C5.0.default**
- Пакет **data.tree**, функция **data.tree**

?