Введение в искусственный интеллект. Машинное обучение

Лекция 7. Решающие деревья. Случайный лес

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

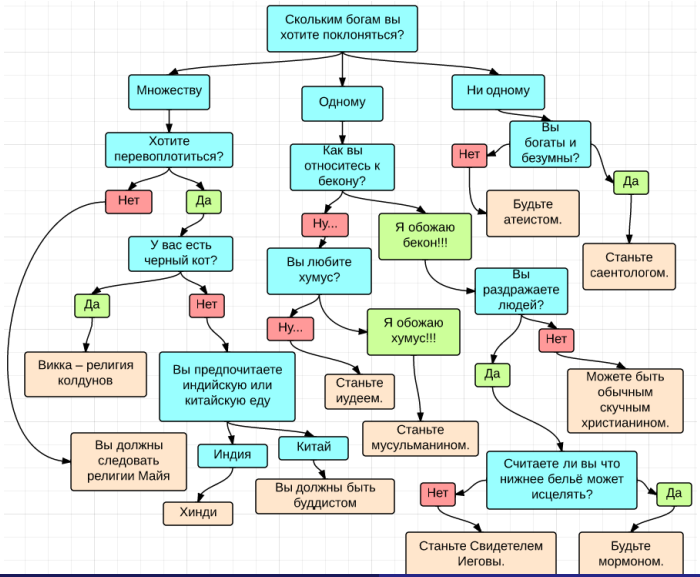
кафедра Математической Теории Интеллектуальных Систем

17 ноября 2020г.

План лекции

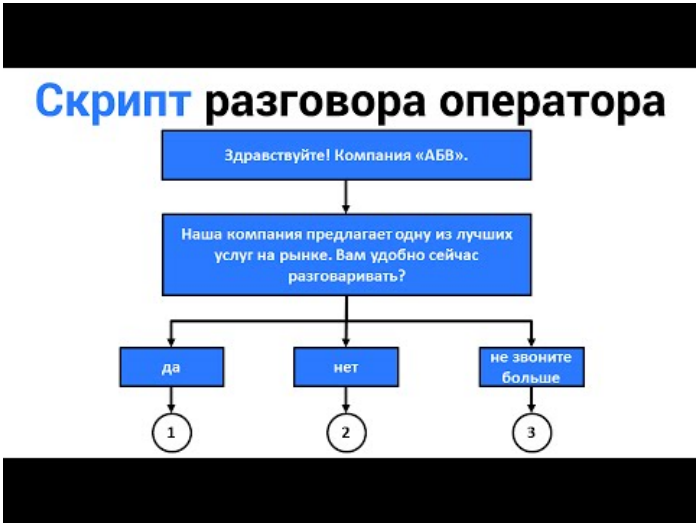
1. Случайные деревья
   - ID3 (Iterative Dichotomiser 3)
   - CART (Classification And Regression Tree)
2. Случайный лес

Примеры деревьев из жизни



Примеры деревьев из жизни

Примеры деревьев из жизни



Примеры деревьев из жизни

- Инструкции любого call-центра

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза
- Многие алгоритмы имеют древовидную структуру

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза
- Многие алгоритмы имеют древовидную структуру

Достоинства

Главные достоинства дерева — интерпретируемость и простота

Определение решающего дерева

Определение

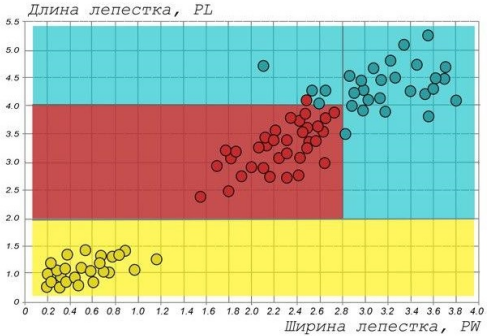
Решающее дерево — алгоритм машинного обучения, задающийся деревом со следующими свойствами:

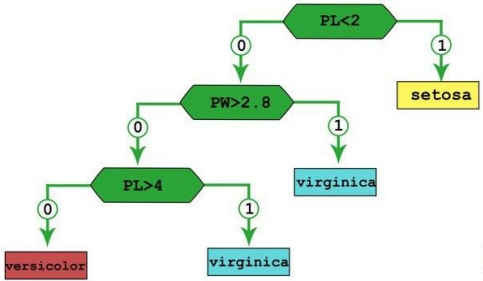
1. Выделена вершина — корень дерева
2. Листовой вершине (у которой нет потомков) соотвествует некоторый ответ алгоритма $y \in Y$
3. Каждой внутренней вершине соотвествует предикат. Каждому ребру из внутренней вершины соотвествует некоторое значение предиката

Пример решающего дерева для задачи классификации

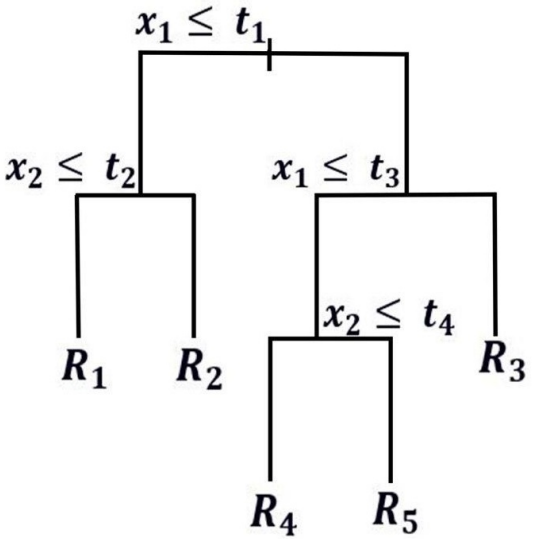
Ирисы Фишера

Задача Фишера о классификации цветков ириса на 3 класса





Пример решающего дерева для задачи регрессии



Часто используемые виды предикатов

- Пороговое условие

Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий

Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий

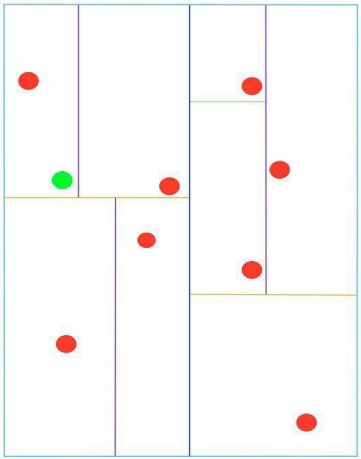
Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий
- Полуплоскость — линейная пороговая функция

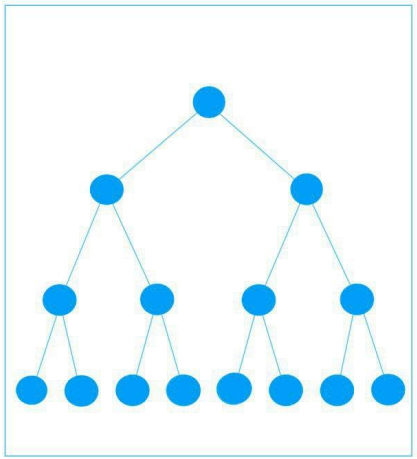
Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий
- Полуплоскость — линейная пороговая функция
- Шар — пороговая функция близости

K-мерное дерево для kNN



Kd-Tree in 2D



Multiple Randomized Kd-Trees

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

Проблема

Построение дерева наименьшего размера, которое не ошибается — NP - сложная задача

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

Проблема

Построение дерева наименьшего размера, которое не ошибается — NP - сложная задача

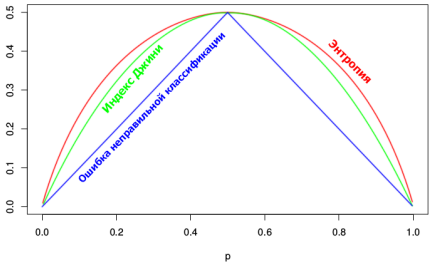
Решение

Итеративные жадные алгоритмы

Критерии информативности для задачи двухклассовой классификации

Пусть p — частота встречаемости одного из классов

Ошибка классификации	$1 - max(p, 1-p)$
Индекс Джинни	$2p(1-p)$
Кросс-энтропийный критерий	-p log p - (1-p) log(1-p)

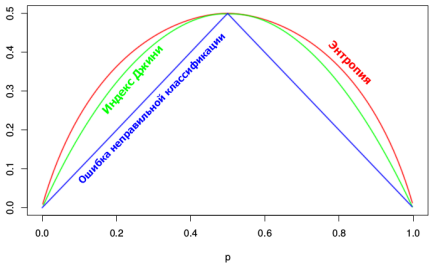


- Критерии довольно похоже
- Последние два критерия — дифференцируемые

Критерии информативности для задачи двухклассовой классификации

Пусть p — частота встречаемости одного из классов

Ошибка классификации	$1 - max(p, 1-p)$
Индекс Джинни	$2p(1-p)$
Кросс-энтропийный критерий	-p log p - (1-p) log(1-p)



- Критерии довольно похоже
- Последние два критерия — дифференцируемые
- Есть по 400 наблюдений из двух классов. Вопрос: какое разделение лучше (300, 100) и (100, 300) против (200, 400) и (200, 0)?

Критерии информативности для задачи многоклассовой классификации

Определения

Пусть R_m — некотороя часть пространста, сожержащая N_m объектов.

Пусть $\hat{p}_{mk} = \frac{1}{N_m} \sum\limits_{x_i \in R_m} [y_i = k]$.

$k(m) = \arg\max\limits_k \hat{p}_{mk}$ — мажоритарный класс

Ошибка классификации	$\frac{1}{N_m} \sum\limits_{i: x_i \in R_m} [y_i \neq k(m)] = 1 - \hat{p}_{mk(m)}$
Индекс Джинни	$\sum\limits_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum\limits_k \hat{p}_{mk}(1 - \hat{p}_{mk})$
Кросс-энтропийный критерий	$- \sum\limits_k \hat{p}_{mk} log \hat{p}_{mk}$

Ветвление

Дано

U — множество объектов, $\mathcal{B}$ - множество предикатов, $I(U)$ — критерий информативности.

Ветвление

Тогда для предиката β множество U разбивается на U_0 и U_1.
Определим предикат с максимально информативностью:

$$\beta = \arg\max_{\beta \in \mathcal{B}} I(\beta, U),$$

где $I(\beta, U) = \frac{|U_0|}{|U|} I(U_0) + \frac{|U_1|}{|U|} I(U_1)$.

ID3 [1]

Вход:
U — обучающая выборка;
$\mathscr{B}$ — множество элементарных предикатов;
Выход:
возвращает корневую вершину дерева, построенного по выборке U;

ПРОЦЕДУРА LearnID3(U);
если все объекты из U лежат в одном классе $c \in Y$ **то**
  создать новый лист v;
  $c_v := c$;
  вернуть (v);
найти предикат с максимальной информативностью:
$\beta := \arg\max_{\beta \in \mathscr{B}} I(\beta, U)$;
разбить выборку на две части $U = U_0 \cup U_1$ по предикату β:
$U_0 := \{x \in U : \beta(x) = 0\}$;
$U_1 := \{x \in U : \beta(x) = 1\}$;
если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**
  создать новый лист v;
  $c_v :=$ класс, в котором находится большинство объектов из U;
иначе
  создать новую внутреннюю вершину v;
  $\beta_v := \beta$;
  $L_v :=$ LearnID3(U_0); (построить левое поддерево)
  $R_v :=$ LearnID3(U_1); (построить правое поддерево)
вернуть (v);

[1] http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf

Преимущества и недостатки алгоритма ID3

Преимущества

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Недостатки

- Жадность. Локальный выбор оптимального предиката не является глобальным

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Недостатки

- Жадность. Локальный выбор оптимального предиката не является глобальным
- Большая вариация алгоритма, при небольших изменениях в данных структура дерева полностью меняется

Преимущества и недостатки алгоритма ID3

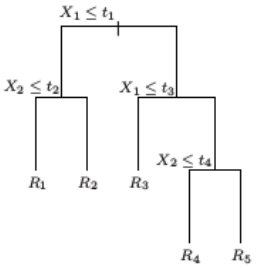
Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

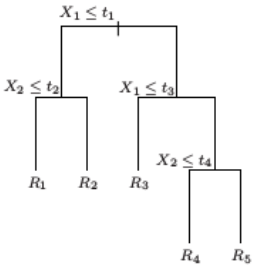
Недостатки

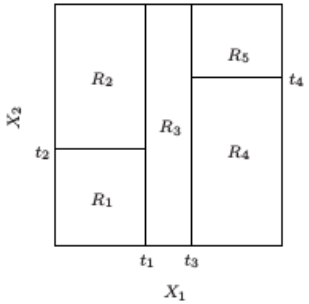
- Жадность. Локальный выбор оптимального предиката не является глобальным
- Большая вариация алгоритма, при небольших изменениях в данных структура дерева полностью меняется
- Алгоритм склонен к переобучения, так как усложняет структуру дерева

Алгоритм построения дерева для регрессии

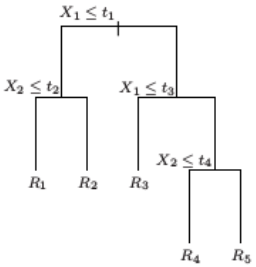


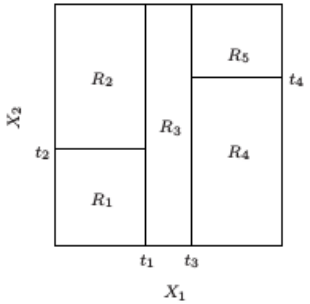
Алгоритм построения дерева для регрессии

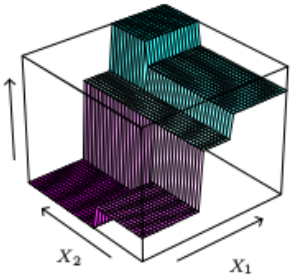




Алгоритм построения дерева для регрессии

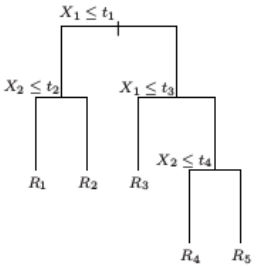


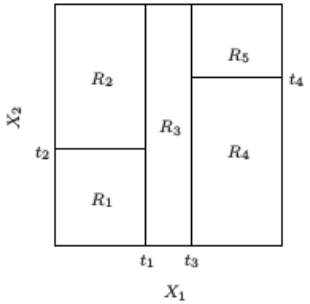


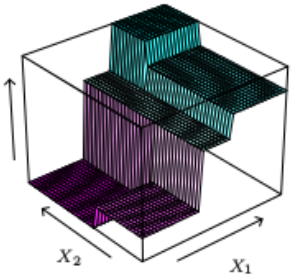


$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

Алгоритм построения дерева для регрессии







$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

$$c_m = ave(y_i | x_i \in R_m)$$

Алгоритм построения дерева для регрессии

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$
$$c_m = ave(y_i | x_i \in R_m)$$

Ветвление

Пусть $R_1(j,s) = \{X | X_j \leq s\}$ и $R_2(j,s) = \{X | X_j > s\}$
Тогда на каждом шаге будем решать задачу минимизации

$$\min_{j,s}(\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2)$$

Регуляризация деревьев

Проблема

Деревья обученные жадным алгоритмом склонны к переобучению

Регуляризация деревьев

Проблема

Деревья обученные жадным алгоритмом склонны к переобучению

Решение

Добавление различных условий на сложность дерева, редукция решающих деревьев

CART

Определения

Пусть T — некоторое поддерево дерева T_0. Обозначим:

- $|T|$ — количество листов в T
- $N_m = \#\{x_i \in R_m\}$
- $c_m = \frac{1}{N_m} \sum\limits_{x_i \in R_m} y_i$
- $Q_m(T) = \frac{1}{N_m} \sum\limits_{x_i \in R_m} (y_i - c_m)^2$

Стратегия усечения дерева

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)
- Минимальное число наблюдений для ветвления дерева (min_samples_split)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)
- Минимальное число наблюдений для ветвления дерева (min_samples_split)
- Минимальное число наблюдения в листе (min_samples_leaf)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)
- Минимальное число наблюдений для ветвления дерева (min_samples_split)
- Минимальное число наблюдения в листе (min_samples_leaf)
- Максимальное количество признаков, используемых для деления (max_features)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)
- Минимальное число наблюдений для ветвления дерева (min_samples_split)
- Минимальное число наблюдения в листе (min_samples_leaf)
- Максимальное количество признаков, используемых для деления (max_features)
- Максимальное число листьев (max_leaf_nodes)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (max_depth)
- Минимальное число наблюдений для ветвления дерева (min_samples_split)
- Минимальное число наблюдения в листе (min_samples_leaf)
- Максимальное количество признаков, используемых для деления (max_features)
- Максимальное число листьев (max_leaf_nodes)
- Минимальное изменение критерия информативности (min_impurity_decrease)

Random forest

Основная идея

Решащие деревья не устойчивы к небольшим изменениям данных. Поэтому при обучении на различных подвыборках решающие деревья будут ошибаться на разных объектах.

Random forest

Основная идея

Решащие деревья не устойчивы к небольшим изменениям данных. Поэтому при обучении на различных подвыборках решающие деревья будут ошибаться на разных объектах.

Определение

- Обучение каждого решающего дерева происходит на сгенерированной случайной подвыборки с повторениями размера обучающей выборки
- Обучение происходит на случайной подвыборке признаков (их количество — входной параметр алгоритма, для классификации берут $\sqrt{n}$, для регрессии $\frac{n}{3}$)
- Дерево строится до полного исчерпания подвыборки и не подвергается процедуре прунинга
- Решающее правило $a(x) = \frac{1}{T}\sum\limits_{t=1}^{T} a_t(x)$

Отбор признаков

Важность признаков

- При построении случайного леса считается количество вершин, в которых используется признак с весом (количество наблюдений в вершине)

Отбор признаков

Важность признаков

- При построении случайного леса считается количество вершин, в которых используется признак с весом (количество наблюдений в вершине)
- Признак тем важнее, чем большее разница между метрикой качества на обычных данных и на данных, где этот признак случайно перемешан

Преимущества и недостатки случайного леса

Преимущества

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Недостатки

- Большой размер и сложность получающихся моделей

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Недостатки

- Большой размер и сложность получающихся моделей
- Нулевая интерпретируемость — черный ящик

Заключение

- Решающие деревья — хорошо интерпритируемый алгоритм машинного обучения

Заключение

- Решающие деревья — хорошо интерпритируемый алгоритм машинного обучения
- На его основе строятся самые сильные модели машинного обучения (например, случайный лес)

Источники

На основе материалов сайта http://www.machinelearning.ru.