Решающие деревья
Случайный лес

Методы анализа данных

Лекция 4

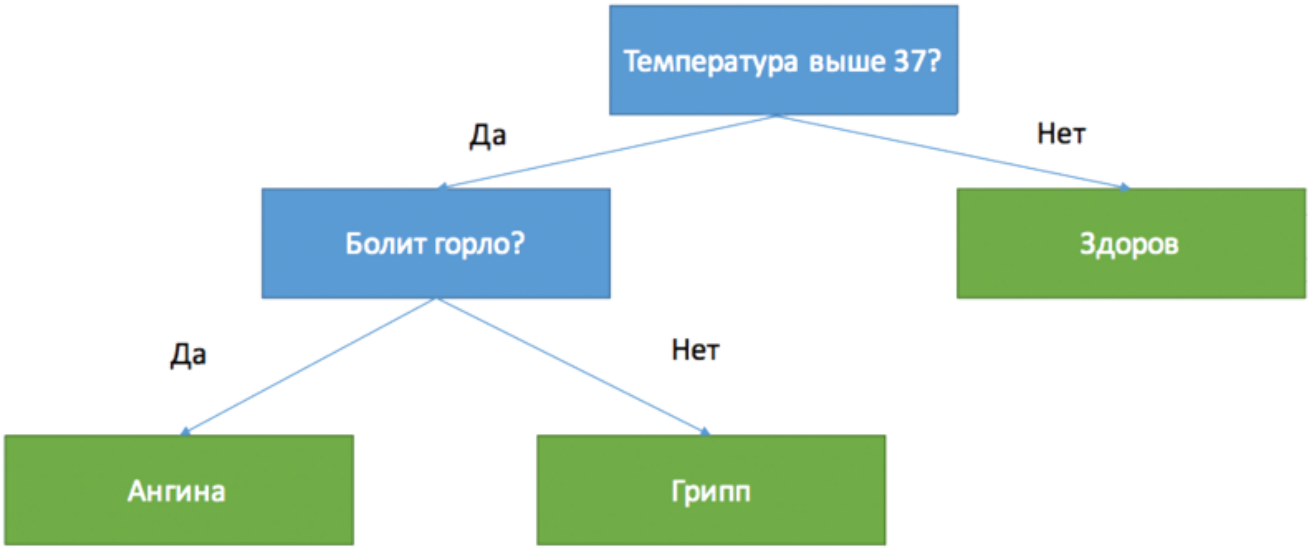
Москва, МФТИ, 2020

Линейные модели

- Линейные модели быстро учатся
- Линейные модели могут восстанавливать только простые зависимости
- Линейные модели можно использовать для восстановления нелинейных зависимостей
- Линейные модели не отражают особенности процесса принятия решений у людей

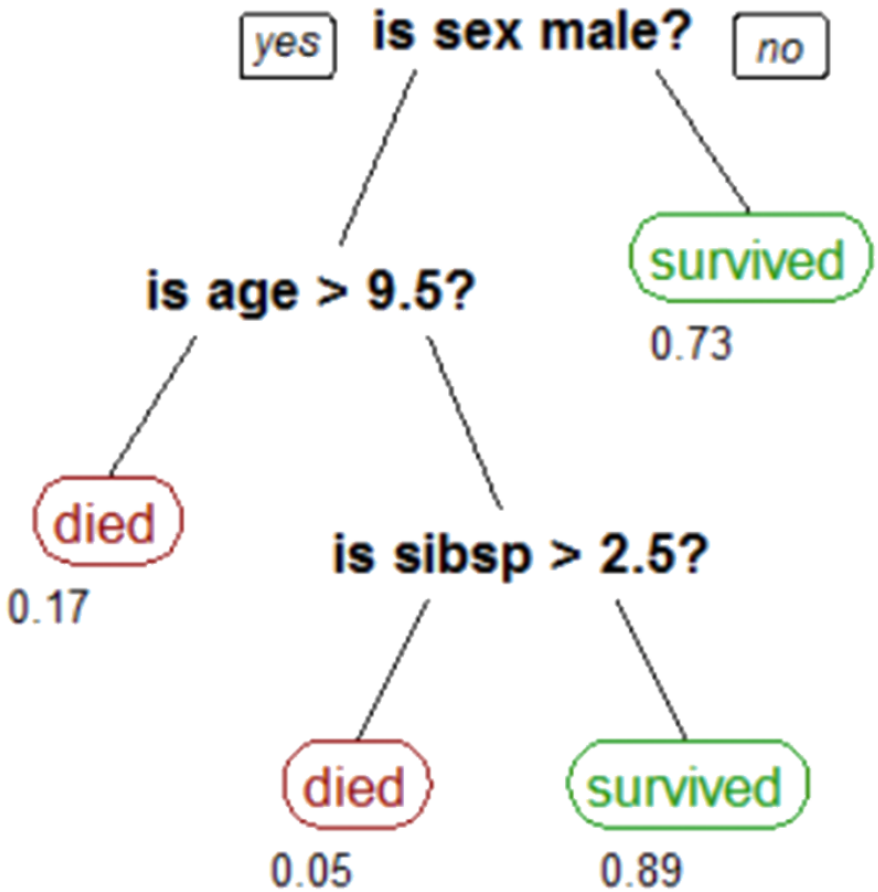
Решающие деревья (пример 1)

Медицинская диагностика



Решающие деревья (пример 2)

Выживет ли тот или иной пассажир Титаника



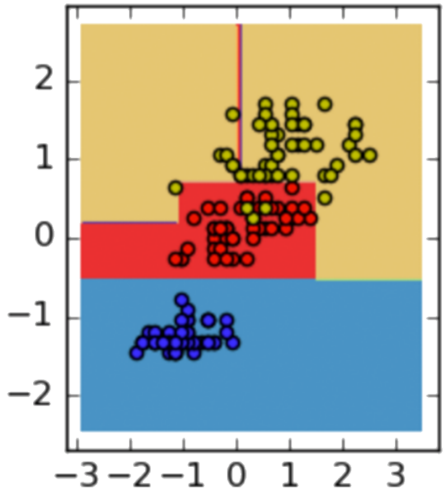
Решающие деревья

- Чаще всего решающие деревья представляют собой бинарные деревья, в каждой внутренней вершине записано условие, а в каждом листе дерева — прогноз.
- В качестве условия выступает проверка, лежит ли значение некоторого признака x^j левее, чем заданный порог t:

$$[x^j \leq t]$$

Решающие деревья в задаче классификации

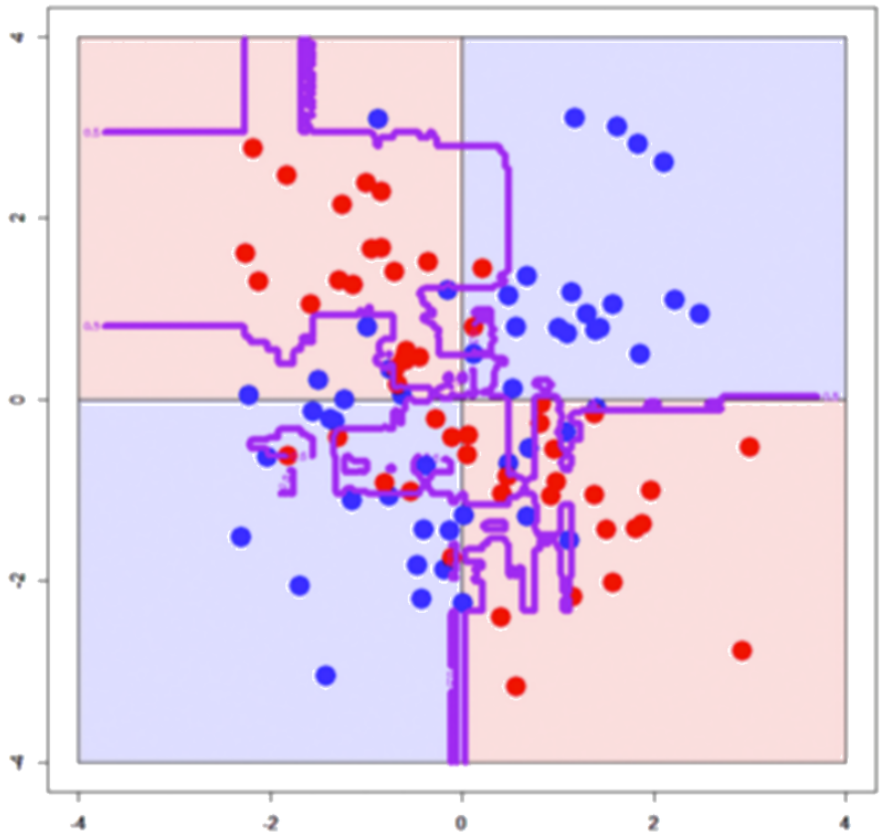
Решается задача классификации с двумя признаками и тремя классами



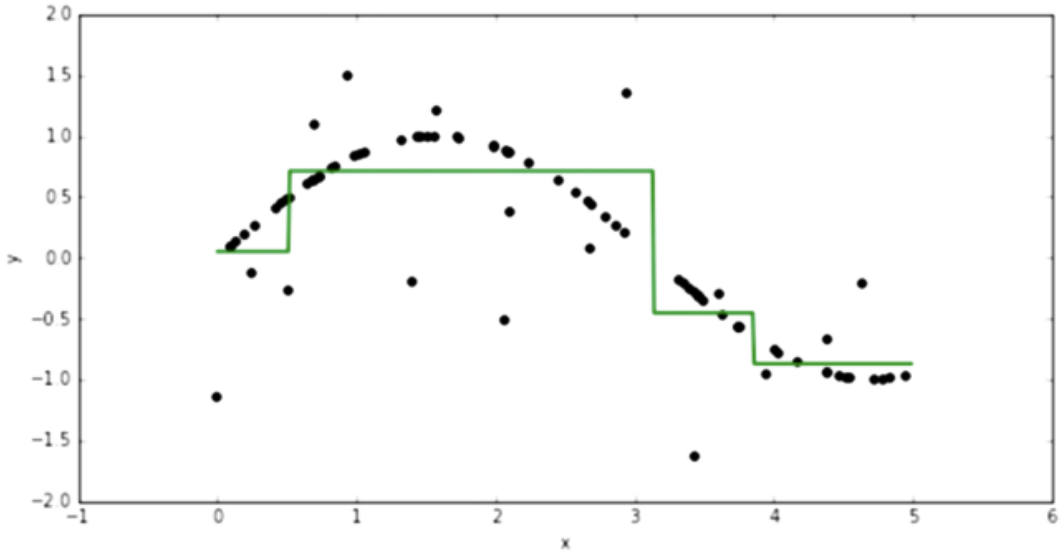
- разделяющая поверхность каждого класса кусочно-постоянная
- каждая сторона поверхности параллельна оси координат, так как каждое условие сравнивает значение равно одного признака с порогом

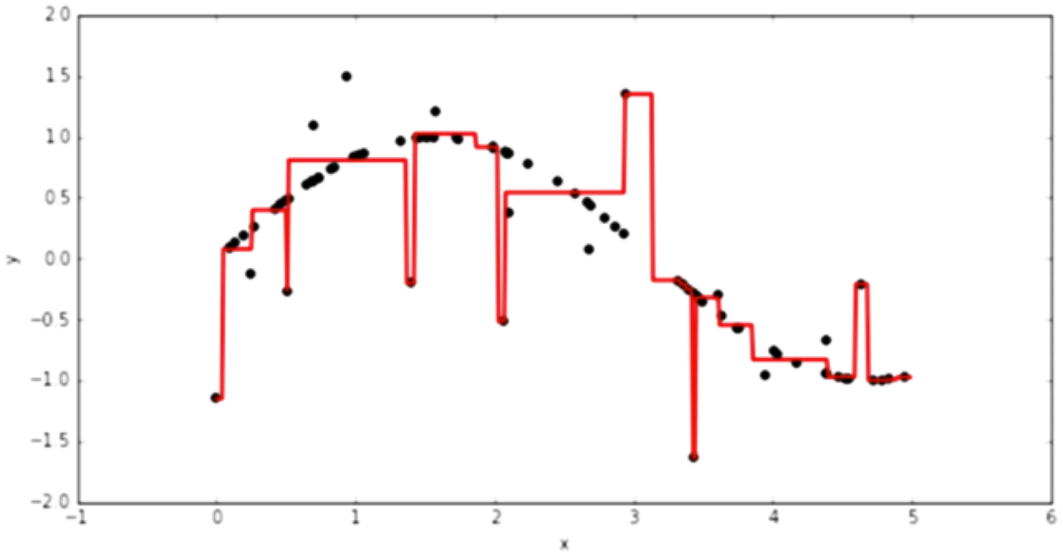
Решающие деревья в задаче классификации

Дерево может переобучиться: его можно сделать настолько глубоким, что каждый лист решающего дерева будет соответствовать одному объекту обучающей выборки

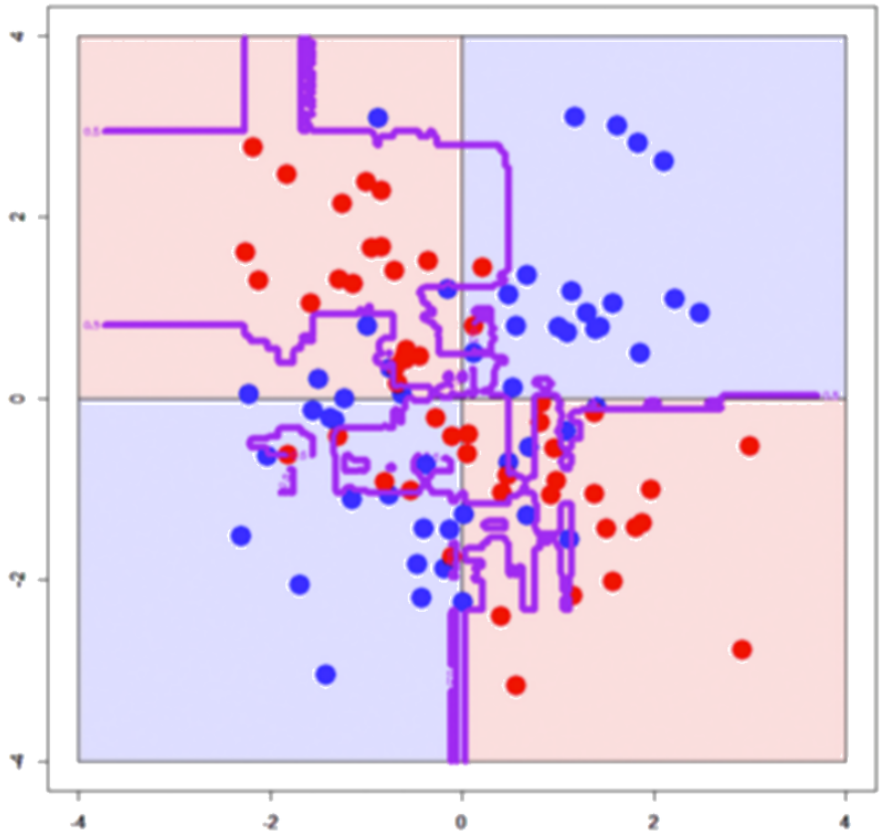


Решающие деревья в задаче регрессии





Обучение решающих деревьев



Жадный способ построения

- Выбирается корень, который разбивает выборку на две. Затем разбивается каждый из потомков этого корня и так далее.
- В каждой вершине строящегося дерева будет использоваться простейшее условие: значение одного из признаков будет сравниваться с некоторым порогом.

В вершину m попало множество X_m

$$Q(X_m, j, t) \to \min_{j,t}.$$

Параметры j и t можно подбирать перебором исходя из условия минимизации Q.
В результате исходная выборка разобьется на 2:

$$X_\ell = \left\{x \in X_m | [x^j \leq t]\right\}, \qquad X_r = \left\{x \in X_m | [x^j > t]\right\}$$

Жадный способ построения

- Если в вершину попал только один объект обучающей выборки или все объекты принадлежат одному классу (в задачах классификации), дальше разбивать не имеет смысла.
- Можно также останавливать разбиение, если глубина дерева достигла определенного значения

В задаче регрессии, если функционал — среднеквадратичная ошибка, оптимально давать средний ответ по этой подвыборке:

$$a_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i.$$

Жадный способ построения

В задаче классификации оптимально возвращать тот класс, который наиболее популярен среди объектов в X_m:

$$a_m = \text{argmax}_{y \in \mathbb{Y}} \sum_{i \in X_m} [y_i = y].$$

Если требуется указать вероятности классов, их можно указать как долю объектов разных классов в X_m:

$$a_{mk} = \frac{1}{|X_m|} \sum_{i \in X_m} [y_i = k].$$

Выбор критерия ошибки

- Критерий ошибки

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r)$$

- $H(X)$ называется критерием информативности: ее значение должно быть тем меньше, чем меньше разброс ответов в X
- В случае регрессии разброс ответов — это дисперсия

$$H(X) = \frac{1}{|X|} \sum_{i \in X} (y_i - \bar{y}(X))^2, \qquad \bar{y} = \frac{1}{|X|} \sum_{i \in X} y_i.$$

Критерий информативности Джини

- Пусть p_k — доля объектов класса k в выборке X:

$$p_k = \frac{1}{X} \sum_{i \in X} [y_i = k].$$

- Критерий информативности Джини формулируется в терминах p_k

$$H(X) = \sum_{k=1}^{K} p_k (1 - p_k).$$

Критерий информативности Джини

Рассмотрим вершину m и объекты X_m, попавшие в нее. Вершине m сопоставим алгоритм $a(x)$, который случайным образом выбирает класс K с вероятностью p_{mK}. Докажем, что матоживадание частоты ошибок алгоритма на объектах из X_m равно кртерию Джини.

Доказательсто:

$$\mathbb{E}\frac{1}{|X_m|}\sum_{x_i \in X_m}[y_i \neq a(x_i)] = \frac{1}{|X_m|}\sum_{x_i \in X_m}\mathbb{E}[y_i \neq a(x_i)]$$

$$= \frac{1}{|X_m|}\sum_{x_i \in X_m}(1 - p_{m,y_i}) = \sum_{k=1}^{K}\frac{\sum_{x_i \in X_m}[y_i = k]}{|X_m|}(1 - p_{mk})$$

$$= \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

Энтропийный критерий информативности

$$H(X) = -\sum_{k=1}^{K} p_k \ln p_k$$

Энтропийный критерий информативности

Докажем, что энтропия ограничена сверху и достигает максимума при равномерном распределении $p_1 = \ldots = p_k = \frac{1}{K}$

Доказательство:

$\forall$ вогнутой функции: $f\left(\sum_{i=1}^{n} a_i x_i\right) \geq \sum_{i=1}^{n} a_i f(x_i)$, если $\sum_{i=1}^{n} a_i = 1$

$$H(p) = -\sum_{k=1}^{K} p_k \log_2 p_k = \sum_{k=1}^{K} p_k \log_2 \frac{1}{p_k} \leq \log_2 \left(\sum_{k=1}^{K} p_k \frac{1}{p_k}\right) = \log_2 K$$

Энтропия равномерного распределения

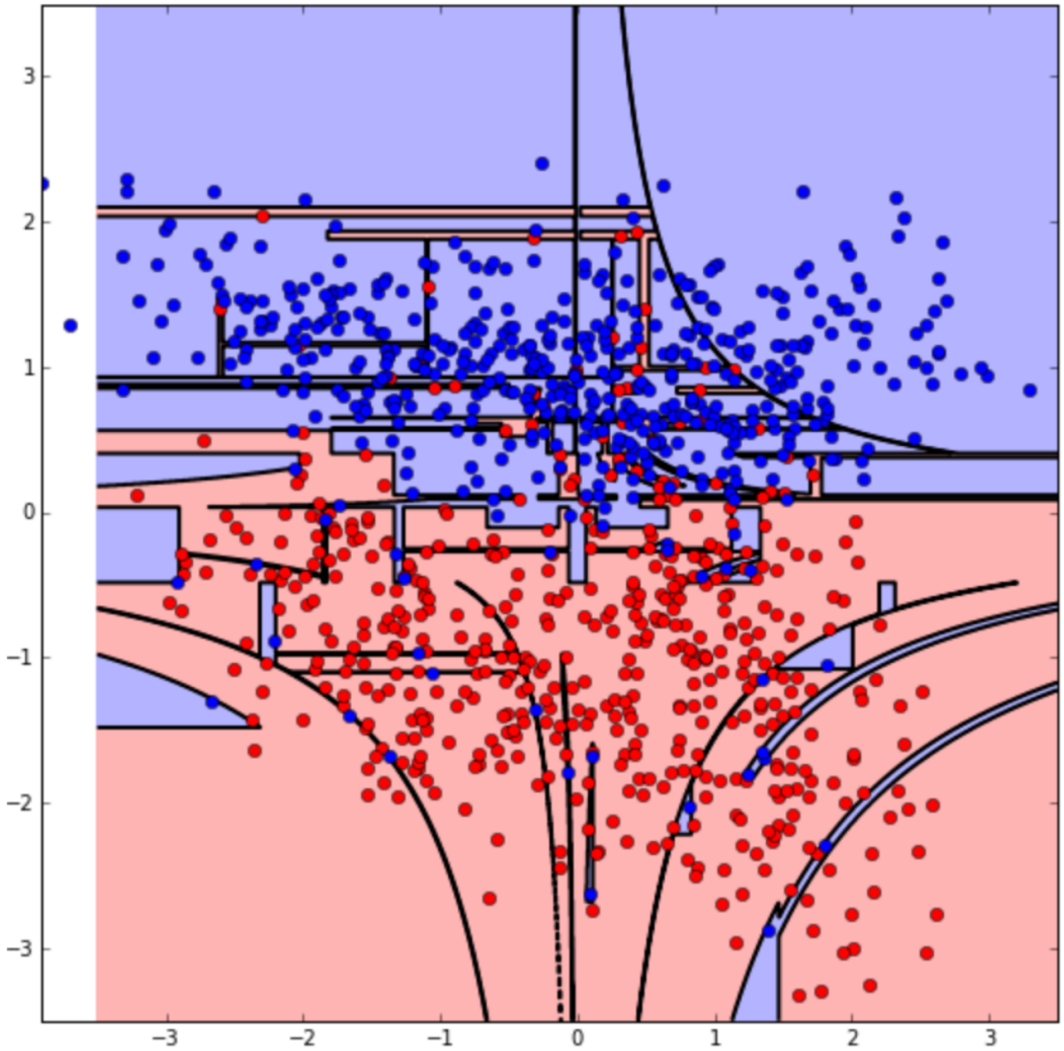
$$H(p)_{\text{равн}} = -\sum_{k=1}^{K} \frac{1}{K} \log_2 \frac{1}{K} = -K \frac{1}{K} \log_2 \frac{1}{K} = -\log_2 \frac{1}{K} = \log_2 K$$

Критерий останова и стрижка деревьев

- **Критерий останова:** устойчивый и полезный критерий проверяет, сколько объектов оказалось в вершине, и разбиение продолжается, если это число больше, чем некоторое выбранное n. Соответственно, если в вершину попало n объектов, она становится листовой. Параметр n нужно подбирать.
- **Стрижка деревьев:** строится решающее дерево максимальной сложности и глубины, до тех пор, пока в каждой вершине не окажется по 1 объекту обучающей выборки. После начинается «стрижка», удаление листьев в этом дереве. До тех пор, пока улучшается качество некоторой отложенной выборки. Стрижка — очень ресурсоёмкая процедура

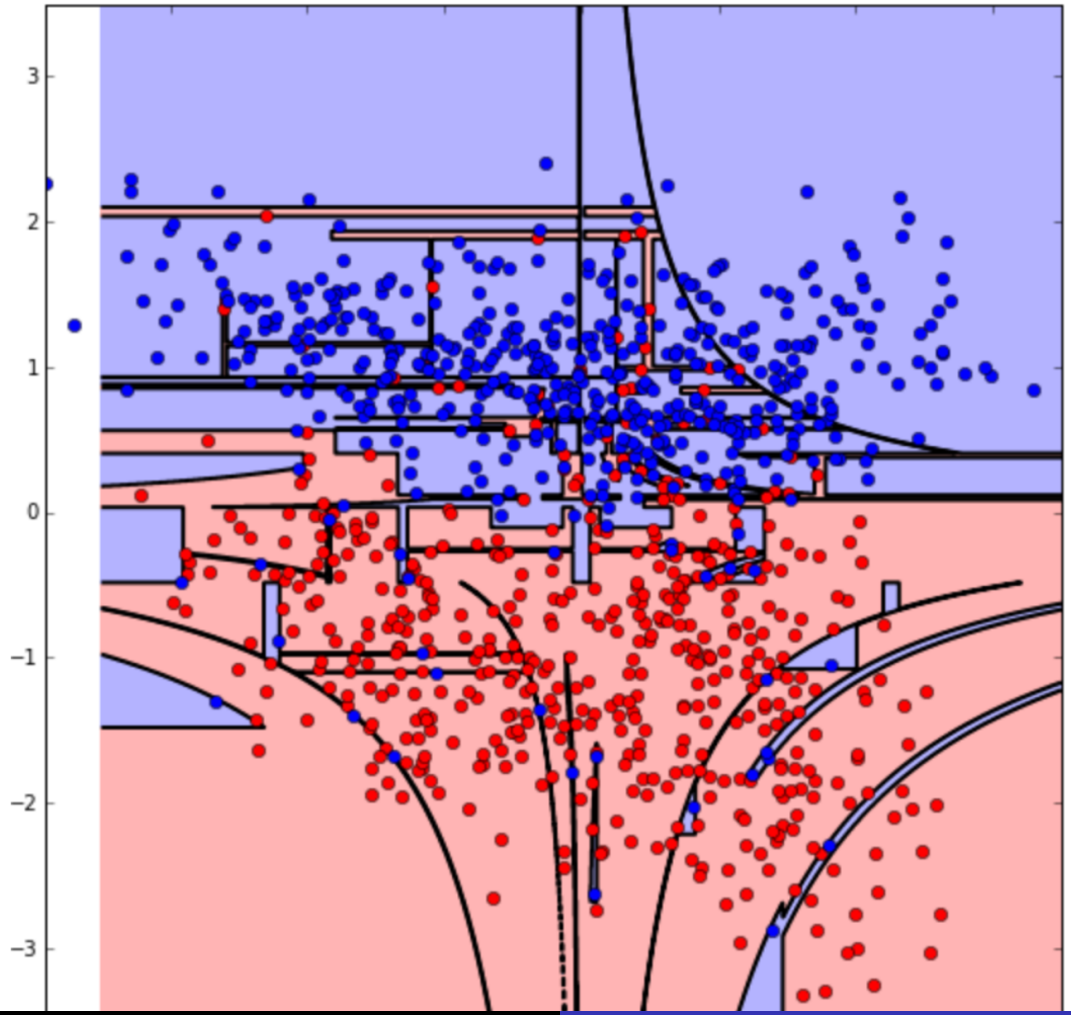
Переобучение решающих деревьев

Основным недостатком решающих деревьев является их высокая склонность к переобучению:



Неустойчивость к изменению обучающей выборки у решающих деревьев

Основным недостатком решающих деревьев является их высокая склонность к переобучению:



Недостатки решающих деревьев

- сильно переобучаются
- сильно меняются при небольшом изменении выборки

Композиция алгоритмов

- Композиция — это объединение N алгоритмов $b_1(x), ..., b_N(x)$ в один

Композиция алгоритмов для регрессии

$$a(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x).$$

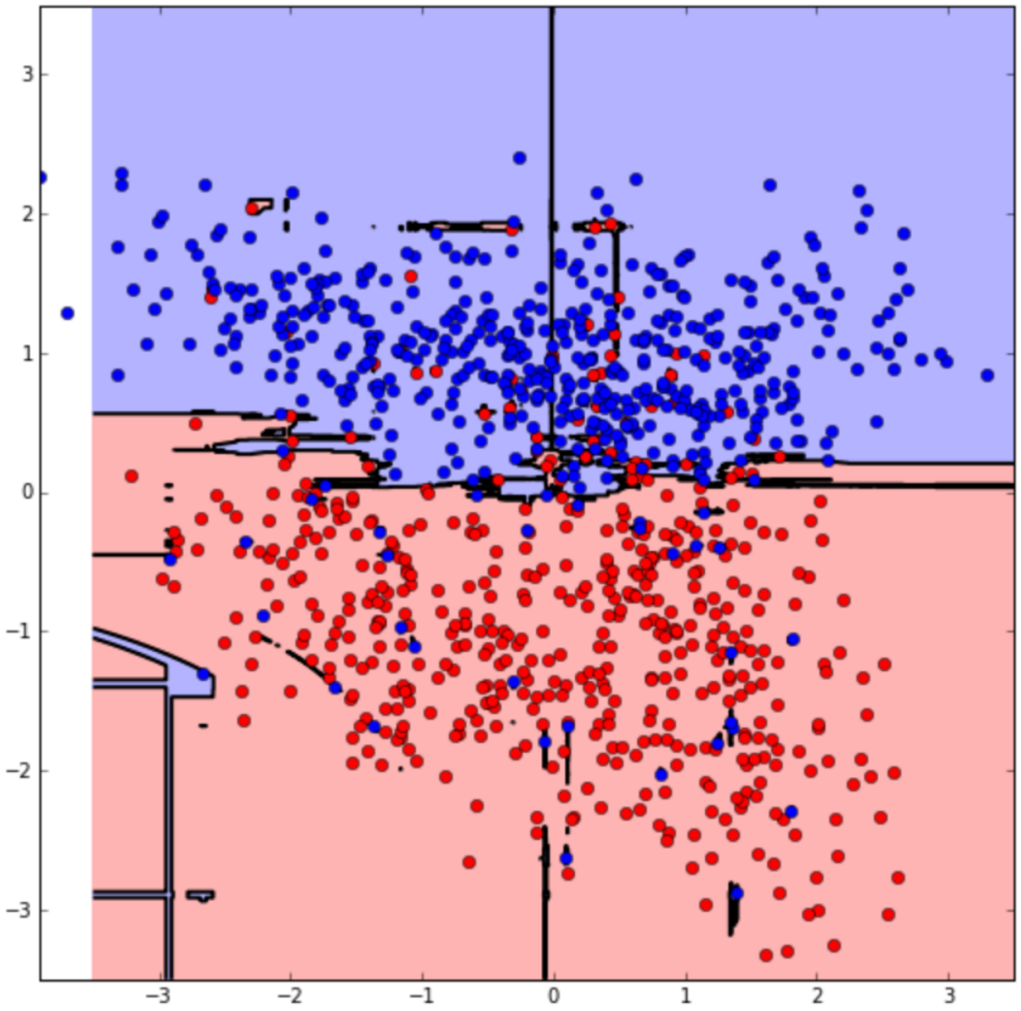
Композиция алгоритмов для классификации

$$a(x) = \text{sign}\, \frac{1}{N} \sum_{n=1}^{N} b_n(x)$$

Рандомизация

Бутстрап - метод построения подвыборок. Он заключается в том, что из обучающей выборки длины l выбирают с возвращением l объектов. При этом новая выборка также будет иметь размер l, но некоторые объекты в ней будут повторятся, а некоторые объекты из исходной выборки в нее не попадут.

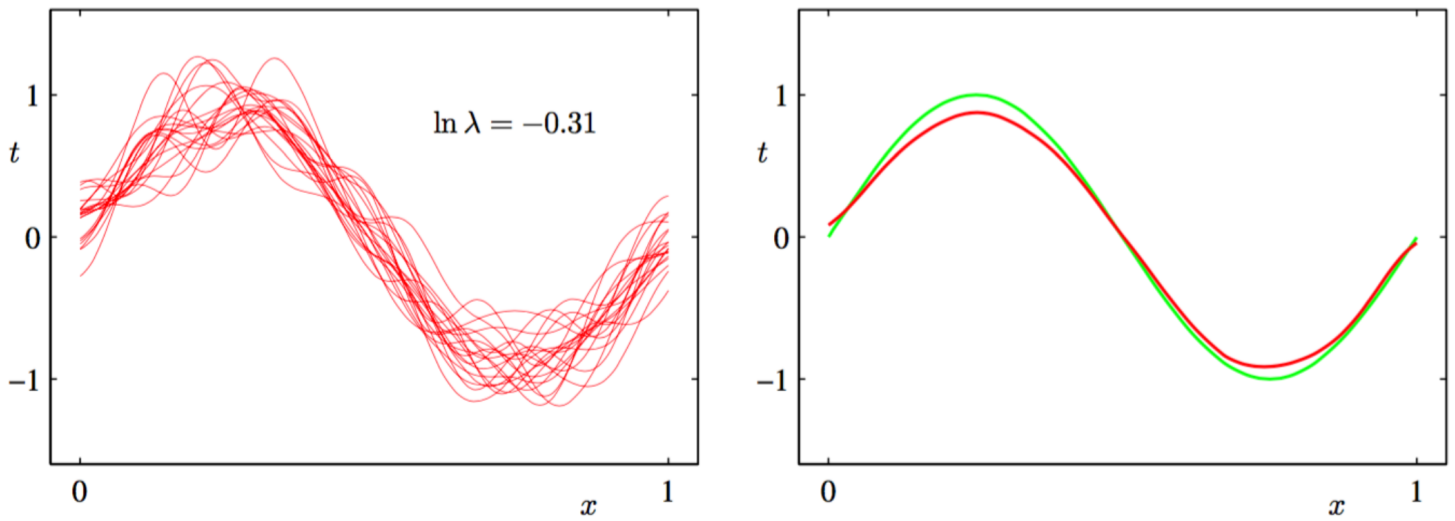
Композиция деревьев



Разложение ошибки на шум, смещение и разброс

- Шум — компонента ошибки алгоритма, которая будет проявляться даже на идеальной модели в этой задаче. Пусть выборка генерируется из некоторого вероятностного распределения.
- Смещение — отклонение, усредненного по различным обучающим выборкам, прогноза заданной модели от прогноза идеальной модели.
- Разброс — дисперсия ответов моделей, обученных по различным обучающим выборкам.

Визуализация смещения и разброса



Сравнение линейных моделей и решающих деревьев в отношении компонент разложения ошибки

Линейные модели:

- Высокое смещение
- Низкий разброс

Деревья:

- Низкое смещение
- Высокий разброс

Смещение и разброс композиции алгоритмов

- При вычислении композиции базовых алгоритмов (с одинаковым смещением) смещение композиции совпадает со смещением отдельного базового алгоритма. Следовательно, смещение композиции деревьев мало, и композиция способна восстанавливать сложные зависимости.
- Разброс композиции отличается от разброса базового алгоритма:

$$\begin{pmatrix}\textbf{разброс}\\ \textbf{композиции}\end{pmatrix} = \frac{1}{N}\begin{pmatrix}\textbf{разброс одного}\\ \textbf{базового алгоритма}\end{pmatrix} + \begin{pmatrix}\textbf{корелляция между}\\ \textbf{базовыми алгоритмами}\end{pmatrix}\cdot$$

Уменьшение корреляции между базовыми алгоритмами

- Беггинг: Обучение базовых алгоритмов происходит на случайных подвыборках обучающей выборки. Причем чем меньше размер случайной подвыборки, тем более независимыми получаются базовые ал- горитмы.
- Метод случайных подпространств: выбирается случайное подмножество признаков и очередной базовый алгоритм обучается только на этих признаках. Доля выбираемых признаков является гиперпараметром этого метода.

Промежуточные результаты

- Ошибка может быть разложена на смещение и разброс
- Смещение композиции близко к смещению одного базового алгоритма
- Разброс при построении композиции уменьшается, причем тем сильнее, чем менее коррелированы базовые алгоритмы

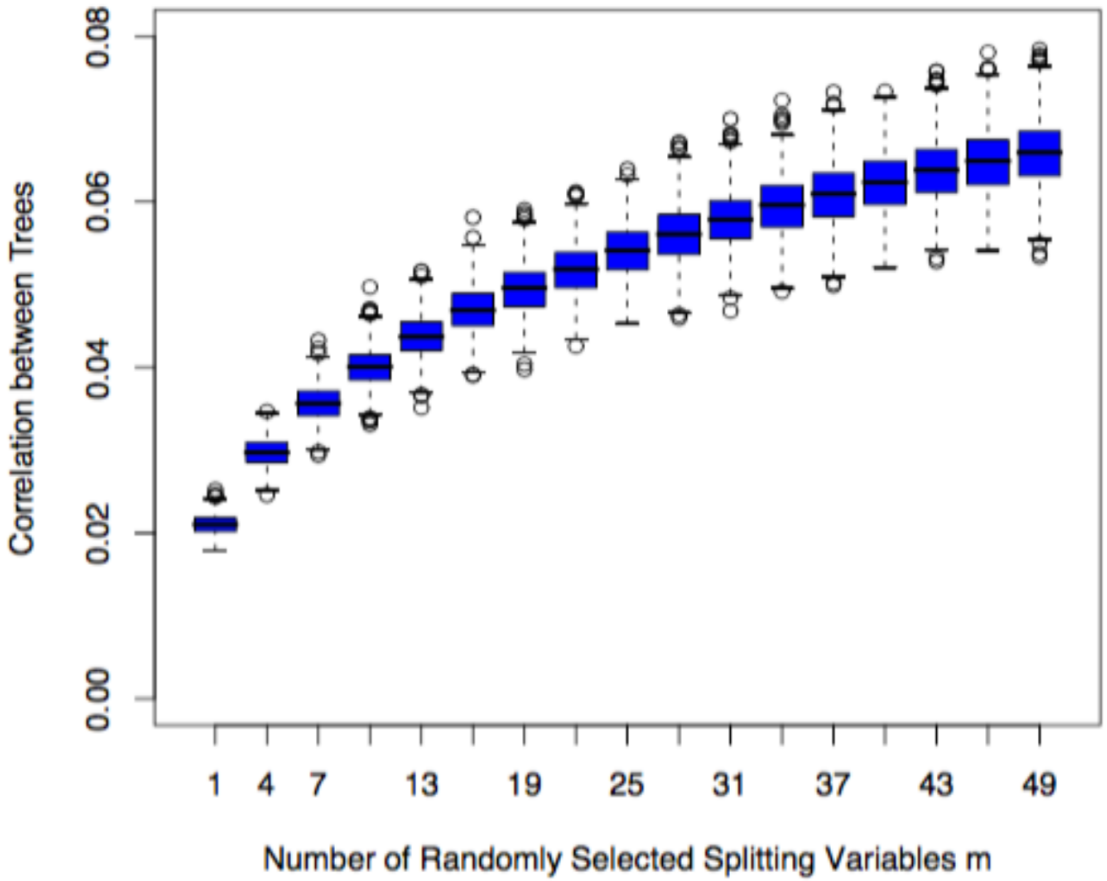
Рандомизация построения решающих деревьев

- В вершину m попало множество X_m

$$Q(X_m, j, t) \to \min_{j,t}.$$

- Параметры j и t можно подбирать перебором исходя из условия минимизации Q.
- Рандомизировать процесс построения можно, если в задаче поиска оптимальных параметров выбирать j из случайного подмножества признаков размера q

Общие рекомендации по рандомизации



- В задаче регрессии имеет смысл брать $q = d/3$
- В задаче классификации имеет смысл брать $q = \sqrt{d}$

Алгоритм построения случайного леса

1: Построить с помощью бутстрапа N случайных подвыборок $\tilde{X}_n, n = 1, \ldots, N$

2: Каждая подвыборка $\tilde{X}_n$ используется как обучающая выборка для построения решающего дерева $b_n(x)$. Причем:

- Дерево строится, пока в каждом листе окажется не более n_{min} объектов
- Процесс построения дерева рандомизирован: этап выбора оптимального признака происходит по случайному подмножеству размера q
- Случайное подмножество размера q выбирается заново каждый раз, когда необходимо разбить очередную вершину

3: Построенные деревья объединяются в композицию:

- В задачах регрессии $a(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$
- В задачах классификации $a(x) = sign \frac{1}{N} \sum_{n=1}^{N} b_n(x)$

Возможность распараллеливания

Так как каждое дерево обучается независимо от всех остальных базовых решающих деревьев, его можно обучать на отдельном ядре

Out of bag

- Формула для оценки качества случайного леса из N деревьев в рамках подхода out-of-bag имеет вид:

$$OOB = \sum_{i=1}^{\ell} L\left(y_i, \frac{1}{\sum_{n=1}^{N}[x_i \notin X_n]} \sum_{n=1}^{N}[x_i \notin X_n] b_n(x_i)\right).$$

Отбор признаков у решающих деревьев

- Критерий выбора признака и порога:

$$Q(X_m, j, t) = \frac{|X_l|}{|X_m|} H(X_l) + \frac{|X_r|}{|X_m|} H(X_r) \to \min_{j,t}$$

- Уменьшение критерия информативности:

$$H(X_m) - \frac{|X_l|}{|X_m|} H(X_l) - \frac{|X_r|}{|X_m|} H(X_r)$$

- R_j - сумма уменьшения по всем вершинам, в которых происходило разбиение по признаку j. Признак j тем важнее, чем больше это уменьшение.