

Решающие деревья

Виктор Владимирович Китов

МГУ им.Ломоносова, ф-т ВМиК, кафедра ММП.

I семестр 2015 г.

Содержание

1 Алгоритм MARS

Алгоритм MARS

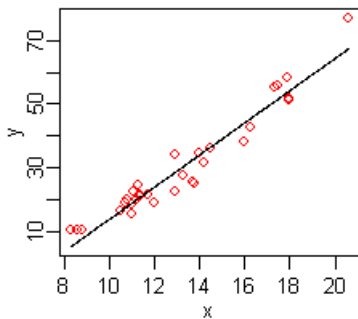
- MARS-multivariate adaptive regression splines
- MARS-торговая марка, эквивалентное «свободное» название - earth
- Существуют бесплатные реализации алгоритма в python, R, matlab.
- J.H. Friedman Multivariate adaptive regression splines. The Annals of Statistics, 19 (1991), pp. 1–141.

Мотивация

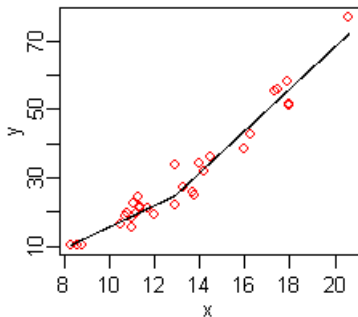
- Цель: в многомерном признаковом пространстве строить регрессию «не слишком сложной» непрерывной функцией
- Алгоритм CART:
 - отбирает признаки, применим в многомерном пространстве
 - аппроксимирует кусочно-постоянными функциями
 - ⇒ для непрерывной функции может потребоваться много листьев, описывающих «ступени»

Пример с одним признаком

$$y = a_0 + a_1x$$

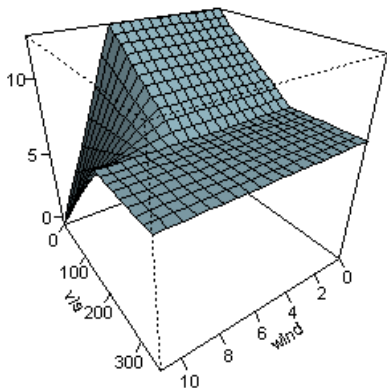


$$y = a_0 + a_1[x - b_1]_+ + a_2[b_1 - x]_+$$



Пример с многими признаками

$$y = a_0 + a_1[x_1 - b_1]_+ + a_2[x_1 - b_2]_+ + a_3[x_2 - b_3]_+ + a_4[x_3 - b_4]_+ + [x_4 - b_5]_+$$



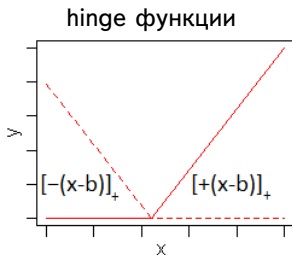
Модель

Прогнозирующая функция:

$$\hat{y} = \sum_{k=1}^K a_k R_k(x, \mathbf{b}^k, \mathbf{s}^k, \mathbf{i}^k)$$

где $\mathbf{b}_k, \mathbf{s}_k, \mathbf{i}_k \in \mathbb{R}^P$, $R_k = \prod_{p=1}^P [s_p^k (x_{i_p^k} - b_p^k)]_+$,
 $s_p^k \in \{-1, +1\} \forall p, k$

$$[u]_+ = \begin{cases} u, & u \geq 0 \\ 0, & u < 0 \end{cases}$$



Алгоритм построения

Алгоритм MARS состоит из 2х этапов:

- 1 наращивание модели
 - 2 упрощение модели
- Обозначим RSS - residual sum of squares

$$RSS = \sum_{n=1}^N (\hat{y}_n - y_n)^2$$

Построение модели-наращивание

НАРАЩИВАНИЕ МОДЕЛИ:

пока $K < K_{max}$ и изменение RSS выше порога:

для каждого $k = 1, 2, \dots, K$:

для каждого $i = 1, 2, \dots, D$:

для каждого $b \in \text{dom}\{x_i\}$

если k, i, b удовлетворяют ограничениям:

рассчитать уменьшение RSS

при добавлении к модели

$$R_{k+1} = R_k[+(x_i - b)]_+ \text{ и } R_{k+2} = R_k[-(x_i - b)]_+$$

добавить ту пару слагаемых, которая дает
наибольшее уменьшение RSS .

$$K \leftarrow K + 2$$

Возможные ограничения на модель:

- количество слагаемых $K < K_{max}$
- количество множителей $P < P_{max}$ (обычно $P_{max} = 1, 2$)
- разрешено произведение только некоторых типов признаков

Построение модели-упрощение

УПРОЩЕНИЕ МОДЕЛИ:

пока GCV уменьшается:

для каждого $k = 1, 2, \dots, K$:

рассчитать уменьшение GCV

исключить слагаемое R_k , дающее максимальное уменьшение GCV .

$K \leftarrow K - 1$

- Добавление слагаемых производилось парами, а исключение производится по одному.
- GCV (generalized cross validation) - критерий, аппроксимирующий LOO-CV:

$$GCV = \frac{RSS}{N(1 - C/N)^2}$$

- RSS - оценка точности, а $C = K + \alpha \frac{K-1}{N}$ - оценка сложности модели, α -параметр, обычно $\alpha \in [2, 5]$.

Комментарии

- Вместо множителей $[s(x_i - b)]_+$ могут использоваться $[s(x_i - b)]_+^q, q > 0$.
 - При $q \rightarrow 0$ $[u]_+^q \rightarrow \mathbb{I}[u \geq 0]$ и MARS сходится к кусочно-постоянному решению.
 - При $q > 1$: непрерывные производные.
- Для классификации:
 - кодируем класс вектором $\tilde{y} \in \mathbb{R}^C$, где C -число классов, а $\tilde{y}^j = \mathbb{I}[x \in \omega_j]$
 - для $i = 1, 2, \dots, C$: оцениваем $g_i(x)$ по $(x_1, \tilde{y}_1^i), (x_2, \tilde{y}_2^i), \dots, (x_N, \tilde{y}_N^i)$
 - классификация x : $\hat{y} = \arg \max_i g_i(x)$

Обсуждение алгоритма

- Преимущества MARS:
 - более гибкий подход, чем линейная модель
 - гибкость можно регулировать, ограничивая K_{max} и P_{max}
 - интерпретируемость
 - возможность одновременного учета непрерывных и дискретных признаков
 - встроенный отбор признаков
 - в отличие от решающих деревьев - аппроксимация кусочно-линейная, а не кусочно-постоянная
- Недостатки MARS:
 - субоптимальность решения, сходимость к лок. оптимуму
 - как следствие:
 - чувствительность к обучающей выборке
 - неустойчивость
 - меньше интерпретируемость, чем у деревьев, т.к. одновременно влияют все слагаемые суммы.