Разбор летучки

Лекция 12

Методы восстановления регрессии

Екатерина Тузова

Регрессия

X– объекты в $\mathbb{R}^n$; Y — ответы в $\mathbb{R}$
$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка
$y_i = y(x_i)$, $y : X \to Y$ – неизвестная зависимость

$a(x) = f(x, w)$ – модель зависимости,
$w \in \mathbb{R}^p$ – вектор параметров модели.

Метод наименьших квадратов (МНК):

$$Q(w, X^l) = \sum_{i=1}^{l} \alpha_i (f(x_i, w) - y_i)^2 \to \min_w$$

K ближайших соседей

K ближайших соседей

$\rho(u, x_1) \le \rho(u, x_2) \le \cdots \le \rho(u, x_l)$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Идея 1: Посмотрим на ближайшие объекты и отнесем u к доминирующему классу.

K ближайших соседей

$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$

x_i – i-й сосед объекта u
y_i – класс i-го соседа u

Идея 1: Посмотрим на ближайшие объекты и отнесем u к доминирующему классу.

Идея 2: Более близкие объекты важнее для классификации.

К ближайших соседей

$$a(u, X^l) = \arg\max_{y \in Y} \sum_{y_i = y} w(i, u)$$

$$w(i, u) = [i \leq k]$$

Как использовать для задачи регрессии?

K neighbors regressor

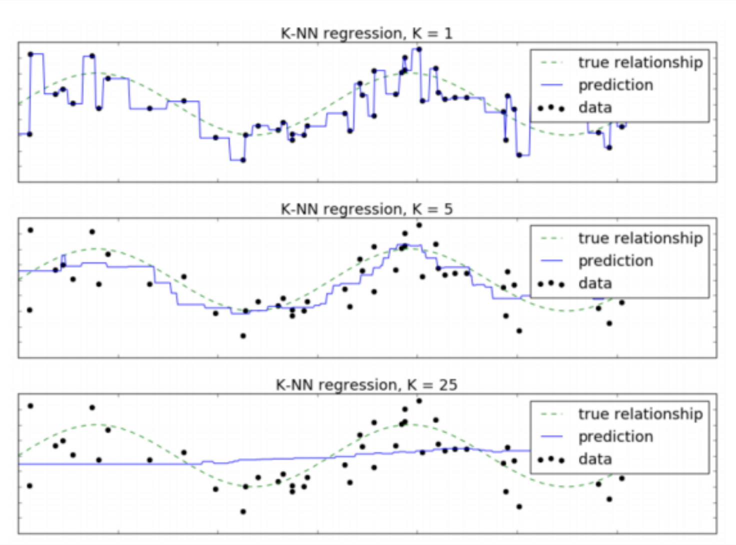
Идея: Усреднить характеристики k ближайших соседей.

K neighbors regressor

Идея: Усреднить характеристики k ближайших соседей.

$$a(u, X^l) = \frac{1}{k}\sum_{i=1}^{k} w(i, u) * y_i$$

Пример

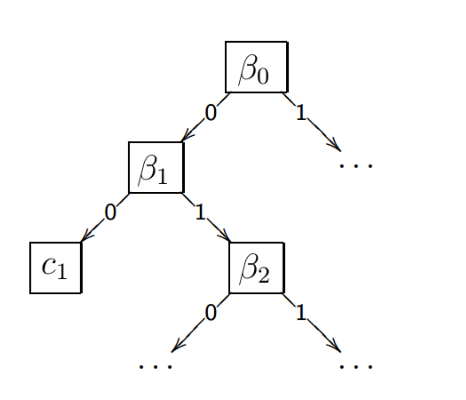

K-NN regression, K = 1
true relationship
prediction
data
K-NN regression, K = 5
true relationship
prediction
data
K-NN regression, K = 25
true relationship
prediction
data

Деревья принятия решений

Бинарное решающее дерево

Бинарное решающее дерево – алгоритм классификации $a(x, \beta)$, задающийся бинарным деревом:

- $\forall v \in V_{inner} \to \beta_v : X \to \{0, 1\}$, $\beta \in \mathscr{B}$
- $\forall v \in V_{leaf} \to$ имя класса $c_v \in Y$



Алгоритм построения ID3

function $\textsc{LearnID3}(U, \mathscr{B})$

    if все объекты из U лежат в одном классе $c \in Y$ **then**

        return новый лист v, $c_v = c$

    $\beta^* = \max\limits_{\beta \in \mathscr{B}} I(\beta, U)$

    $U_{left} = \{x \in U : \beta^*(x) = 0\}$

    $U_{right} = \{x \in U : \beta^*(x) = 1\}$

    if $U_{left} = \varnothing$ или $U_{right} = \varnothing$ **then**

        return v, c_v = Majority(U)

    Создать новую внутреннюю вершину v: $\beta_v = \beta^*$

    L_v = LearnID3 (U_{left}, $\mathscr{B}$)

    R_v = LearnID3 (U_{right}, $\mathscr{B}$)

    return v

Как использовать для задачи регрессии?

Decision tree regression

Идея: В каждый узел дерева записать среднее значение целевой функции.

Decision tree regression

Идея: В каждый узел дерева записать среднее значение целевой функции.

Идея: Хотим, чтобы выборка разбивалась таким образом, что значения целевого признака в каждом листе примерно равны.

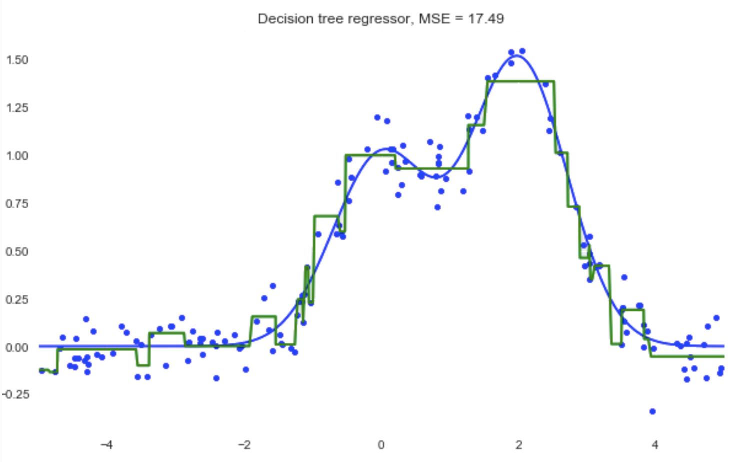
Decision tree regression

Идея: В каждый узел дерева записать среднее значение целевой функции.

Идея: Хотим, чтобы выборка разбивалась таким образом, что значения целевого признака в каждом листе примерно равны.

$$I(\beta, X^l) = \frac{1}{l}\sum_{i=1}^{l}(y_i - \frac{1}{l}\sum_{j=1}^{l} y_j)^2$$

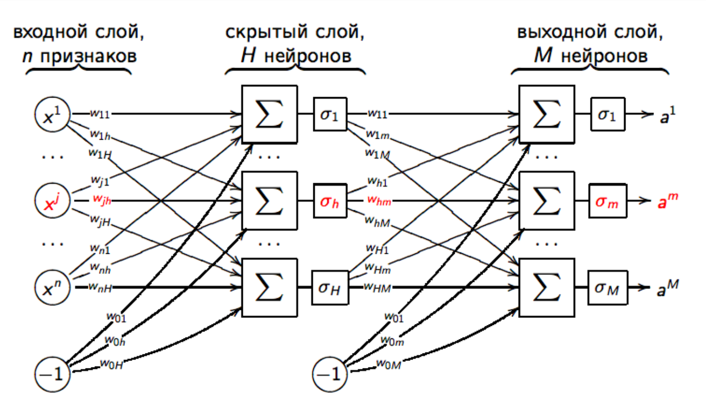
Пример



Decision tree regressor, MSE = 17.49
1.50
1.25
1.00
0.75
0.50
0.25
0.00
-0.25
−4
−2
0
2
4

Нейронные сети

Пусть $Y = \mathbb{R}^M$, два слоя в сети.



Алгоритм обратного распространения ошибки

function $\textsc{Backpropagation}(X^l, H, \alpha, \eta)$

    ...

    repeat[пока Q не стабилизируются]

        Взять x_i из X^l

Алгоритм обратного распространения ошибки

function $\textsc{Backpropagation}(X^l, H, \alpha, \eta)$

  ...

  repeat[пока Q не стабилизируются]

    Взять x_i из X^l

$$
\begin{cases}
u_i^h = \sigma_h(\sum\limits_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \dots, H \\
a_i^m = \sigma_m(\sum\limits_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \dots, M \\
\mathcal{L}_i = \frac{1}{2} \sum\limits_{m=1}^{M} (\varepsilon_i^m)^2
\end{cases}
$$

Алгоритм обратного распространения ошибки

function $\textsc{Backpropagation}(X^l,\ H,\ \alpha,\ \eta)$

    ...

    repeat[пока Q не стабилизируются]

        Взять x_i из X^l

$$
\begin{cases}
u_i^h = \sigma_h(\sum\limits_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \dots, H \\
a_i^m = \sigma_m(\sum\limits_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \dots, M \\
\mathcal{L}_i = \frac{1}{2} \sum\limits_{m=1}^{M} (\varepsilon_i^m)^2
\end{cases}
$$

$$
\begin{cases}
\varepsilon_i^h = \sum\limits_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \dots, H
\end{cases}
$$

Алгоритм обратного распространения ошибки

function $\textsc{Backpropagation}(X^l, H, \alpha, \eta)$

  ...

  repeat[пока Q не стабилизируются]

    Взять x_i из X^l

$$\begin{cases} u_i^h = \sigma_h(\sum\limits_{j=0}^{J} w_{jh} x_i^j), \quad h = 1, \ldots, H \\ a_i^m = \sigma_m(\sum\limits_{h=0}^{H} w_{hm} u_i^h), \quad \varepsilon_i^m = a_i^m - y_i^m, \quad m = 1, \ldots, M \\ \mathcal{L}_i = \frac{1}{2} \sum\limits_{m=1}^{M} (\varepsilon_i^m)^2 \end{cases}$$

$$\begin{cases} \varepsilon_i^h = \sum\limits_{m=1}^{M} \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \ldots, H \end{cases}$$

$$\begin{cases} w_{hm} = w_{hm} - \alpha \varepsilon_i^m \sigma'_m u_i^h, \quad h = 0, \ldots, H, \quad m = 1, \ldots, M \\ w_{jh} = w_{jh} - \alpha \varepsilon_i^h \sigma'_h x_i^j, \quad j = 0, \ldots, n, \quad h = 1, \ldots, H \\ Q = (1 - \eta) Q + \eta \mathcal{L}_i \end{cases}$$

Как использовать для задачи регрессии?

Neural Net Regression

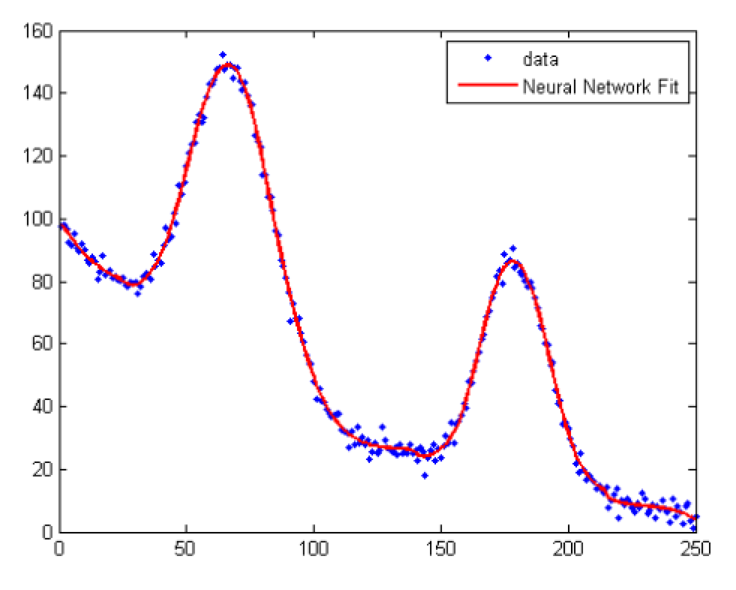
Идея: Использовать непрерывную функцию активации вместо ступенчатой.

Neural Net Regression

Идея: Использовать непрерывную функцию активации вместо ступенчатой.

Идея: Использовать один нейрон выходного слоя.

Пример



160
140
120
100
80
60
40
20
0
0
50
100
150
200
250
data
Neural Network Fit

Машина опорных векторов

Оптимальная разделяющая гиперплоскость

Линейно разделимая выборка:

$$\begin{cases} \frac{1}{2}\|\mathbf{w}\|^2 \to \min\limits_{\mathbf{w}} \\ M_i(\mathbf{w}, w_0) \geq 1 \qquad i = 1, \dots, l \end{cases}$$

Оптимальная разделяющая гиперплоскость

Линейно разделимая выборка:

$$\begin{cases} \frac{1}{2}\|\mathbf{w}\|^2 \to \min\limits_{\mathbf{w}} \\ M_i(\mathbf{w}, w_0) \geq 1 \qquad i = 1, \ldots, l \end{cases}$$

Линейно неразделимая выборка – надо ослабить имеющиеся условия.

$$\begin{cases} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum\limits_{i=1}^{l} \xi_i \to \min\limits_{\mathbf{w}, \xi} \\ M_i(\mathbf{w}, w_0) \geq 1 - \xi_i \qquad i = 1, \ldots, l \\ \xi_i \geq 0 \qquad i = 1, \ldots, l \end{cases}$$

Двойственная задача SVM

Решение исходной задачи выражается через решение двойственной:

$$\begin{cases} \mathbf{w} = \sum\limits_{i=1}^{l} \alpha_i y_i \mathbf{x_i} \\ w_0 = \langle \mathbf{w}, \mathbf{x_i} \rangle - y_i \end{cases}$$

Двойственная задача SVM

Решение исходной задачи выражается через решение двойственной:

$$\begin{cases} \mathbf{w} = \sum\limits_{i=1}^{l} \alpha_i y_i \mathbf{x_i} \\ w_0 = \langle \mathbf{w}, \mathbf{x_i} \rangle - y_i \end{cases}$$

$$a(\mathbf{x}) = \operatorname{sign}(\sum\limits_{i=1}^{l} \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle - w_0)$$

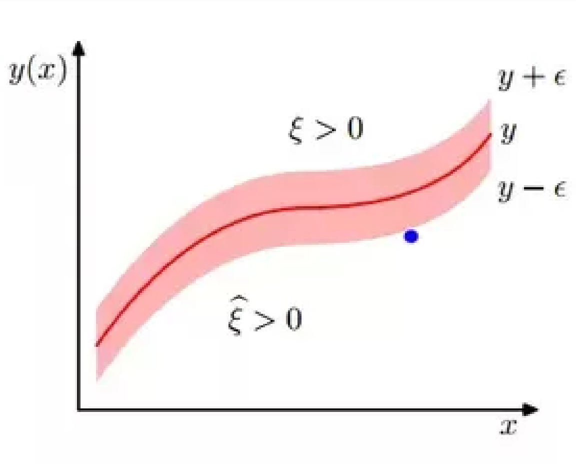
Как использовать для задачи регрессии?

SVR

Идея: Не считать за ошибку отклонение целевой функции меньше, чем на ε.

SVR

Идея: Не считать за ошибку отклонение целевой функции меньше, чем на ε.



SVR

$$\begin{cases} \min \frac{1}{2}\|\mathbf{w}\|^2 \\ y_i - \langle \mathbf{w}, \mathbf{x_i} \rangle + w_0 \leq \varepsilon \\ \langle \mathbf{w}, \mathbf{x_i} \rangle - w_0 - y_i \leq \varepsilon \end{cases}$$

SVR

$$\begin{cases} \min \frac{1}{2}\|\mathbf{w}\|^2 \\ y_i - \langle \mathbf{w}, \mathbf{x_i} \rangle + w_0 \leq \varepsilon \\ \langle \mathbf{w}, \mathbf{x_i} \rangle - w_0 - y_i \leq \varepsilon \end{cases}$$

$$\xi_i = \langle \mathbf{w}, \mathbf{x_i} \rangle - w_0 - y_i - \varepsilon$$
$$\hat{\xi}_i = -\langle \mathbf{w}, \mathbf{x_i} \rangle + w_0 + y_i - \varepsilon$$

SVR

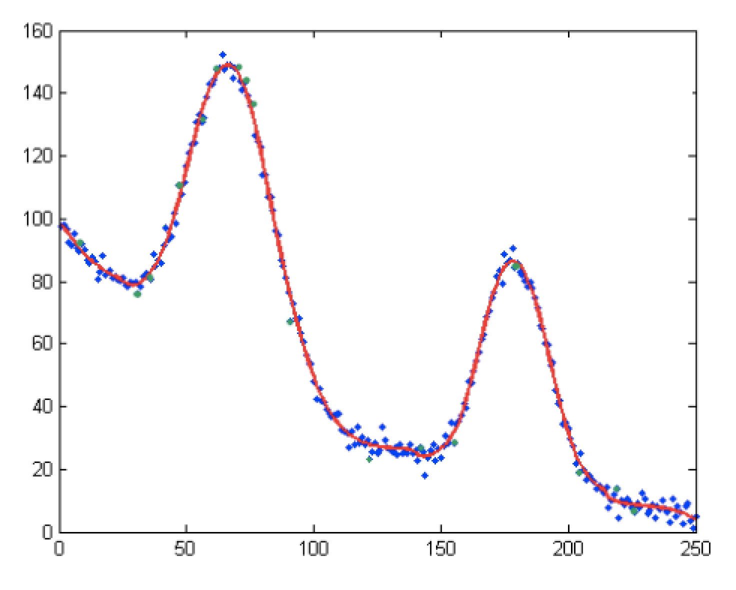
$$\begin{cases} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{l}(\xi_i + \hat{\xi}_i) \to \min\limits_{\mathbf{w},\xi} \\ y_i - \varepsilon - \hat{\xi}_i \leq \langle \mathbf{w}, \mathbf{x_i} \rangle - w_0 \leq y_i + \varepsilon + \xi_i \qquad i = 1, \dots, l \\ \xi_i, \hat{\xi}_i \geq 0 \qquad i = 1, \dots, l \end{cases}$$

SVR

$$\begin{cases} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \hat{\xi}_i) \to \min\limits_{\mathbf{w},\xi} \\ y_i - \varepsilon - \hat{\xi}_i \le \langle \mathbf{w}, \mathbf{x_i} \rangle - w_0 \le y_i + \varepsilon + \xi_i \qquad i = 1, \dots, l \\ \xi_i, \hat{\xi}_i \ge 0 \qquad i = 1, \dots, l \end{cases}$$

$$a(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i - \hat{\alpha}_i)\langle \mathbf{x_i}, \mathbf{x} \rangle - w_0$$

Пример



Вопросы?

На следующей лекции

- Voting
- Bootstrap
- Bagging
- Boosting
- Adaboost
- Bagboo
- Stacking