

Разбор летучки

Лекция 12

Методы восстановления регрессии

Екатерина Тузова

X – объекты в \mathbb{R}^n ; Y – ответы в \mathbb{R}

$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка

$y_i = y(x_i)$, $y : X \rightarrow Y$ – неизвестная зависимость

$a(x) = f(x, w)$ – модель зависимости,

$w \in \mathbb{R}^p$ – вектор параметров модели.

Метод наименьших квадратов (МНК):

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

К ближайших соседей

К ближайших соседей

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

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.

К ближайших соседей

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

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.

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

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

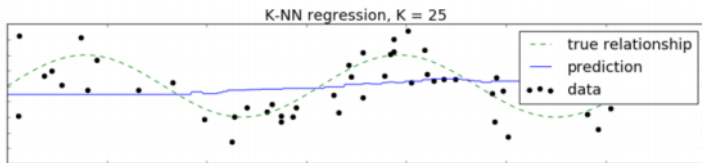
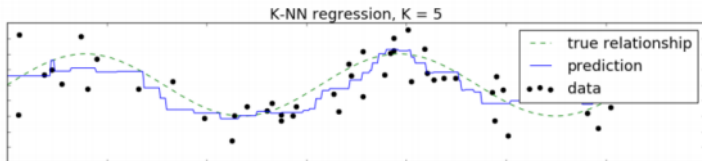
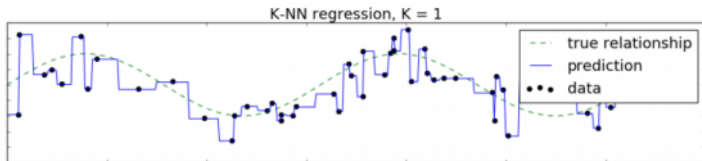
Как использовать для задачи
регрессии?

Идея: Усреднить характеристики k ближайших соседей.

Идея: Усреднить характеристики k ближайших соседей.

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

Пример

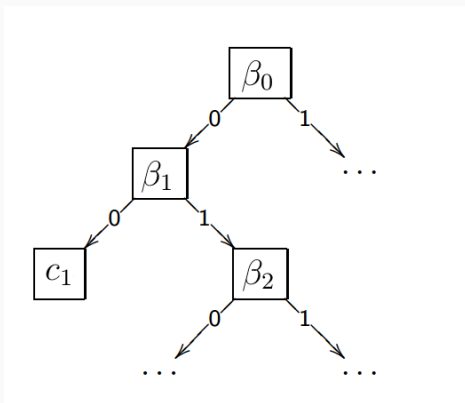


Деревья принятия решений

Бинарное решающее дерево

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

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



Алгоритм построения ID3

```
1 function LEARNID3( $U, \mathcal{B}$ )
2   if все объекты из  $U$  лежат в одном классе  $c \in Y$  then
3     return новый лист  $v$ ,  $c_v = c$ 
4    $\beta^* = \max_{\beta \in \mathcal{B}} I(\beta, U)$ 
5    $U_{left} = \{x \in U : \beta^*(x) = 0\}$ 
6    $U_{right} = \{x \in U : \beta^*(x) = 1\}$ 
7   if  $U_{left} = \emptyset$  или  $U_{right} = \emptyset$  then
8     return  $v$ ,  $c_v = \text{Majority}(U)$ 
9   Создать новую внутреннюю вершину  $v$ :  $\beta_v = \beta^*$ 
10   $L_v = \text{LearnID3}(U_{left}, \mathcal{B})$ 
11   $R_v = \text{LearnID3}(U_{right}, \mathcal{B})$ 
12  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.

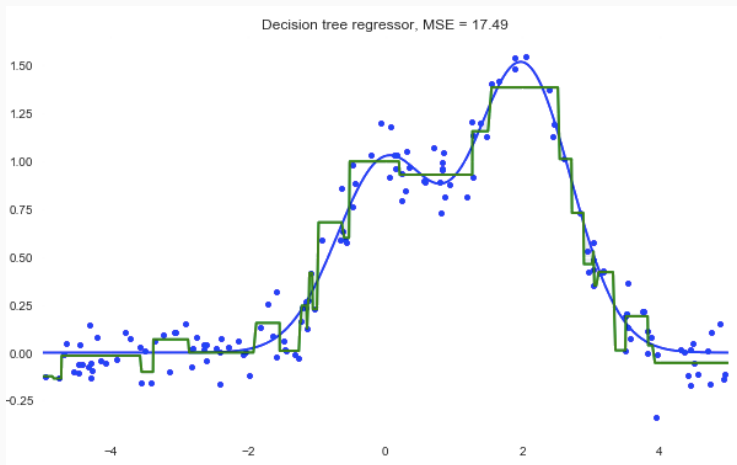
Decision tree regression

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.

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

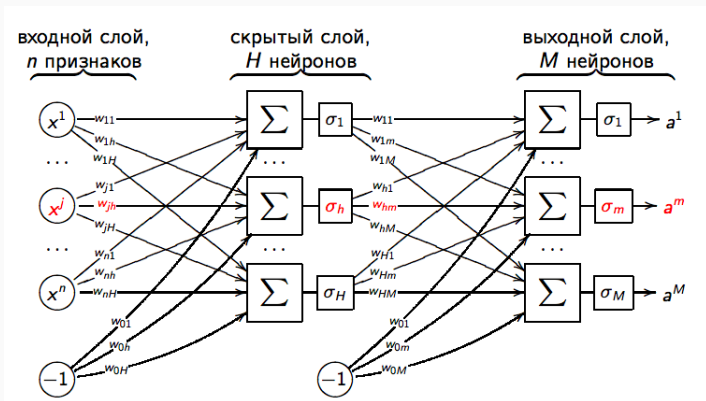
Пример



Нейронные сети

Многослойная нейронная сеть

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



Алгоритм обратного распространения ошибки

```
1 function BACKPROPAGATION( $X^l$ ,  $H$ ,  $\alpha$ ,  $\eta$ )  
2   ...  
3   repeat[пока  $Q$  не стабилизируются]  
4     Взять  $x_i$  из  $X^l$ 
```

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

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

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

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

6 $\left\{ \varepsilon_i^h = \sum_{m=1}^M \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \dots, H \right.$

Алгоритм обратного распространения ошибки

1 **function** BACKPROPAGATION(X^l, H, α, η)

2 ...

3 **repeat**[пока Q не стабилизируются]

4 Взять x_i из X^l

5

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

6

$$\left\{ \begin{array}{l} \varepsilon_i^h = \sum_{m=1}^M \varepsilon_i^m \sigma'_m w_{hm}, \quad h = 1, \dots, H \end{array} \right.$$

7

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

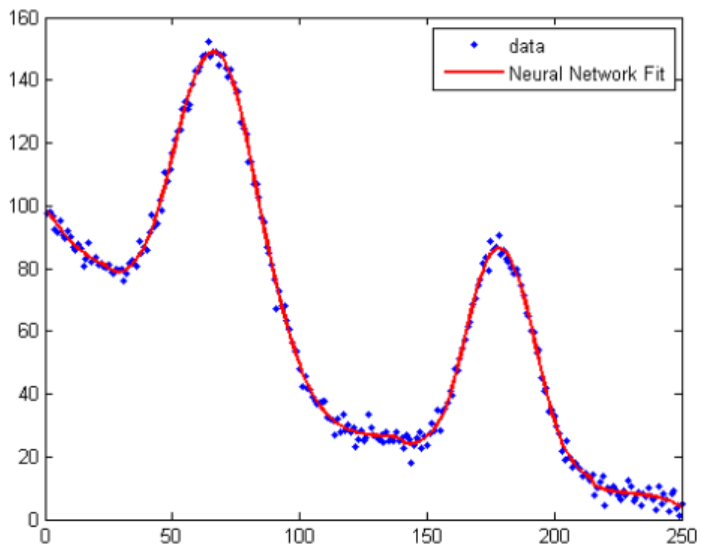
Как использовать для задачи
регрессии?

Идея: Использовать непрерывную функцию активации вместо ступенчатой.

Идея: Использовать непрерывную функцию активации вместо ступенчатой.

Идея: Использовать один нейрон выходного слоя.

Пример



Машина опорных векторов

Линейно разделяемая выборка:

$$\begin{cases} \frac{1}{2} \|\mathbf{w}\|^2 \rightarrow \min_{\mathbf{w}} \\ M_i(\mathbf{w}, w_0) \geq 1 \quad i = 1, \dots, l \end{cases}$$

Линейно разделяемая выборка:

$$\begin{cases} \frac{1}{2} \|\mathbf{w}\|^2 \rightarrow \min_{\mathbf{w}} \\ M_i(\mathbf{w}, w_0) \geq 1 \quad i = 1, \dots, l \end{cases}$$

Линейно неразделимая выборка – надо ослабить имеющиеся условия.

$$\begin{cases} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{\mathbf{w}, \xi} \\ M_i(\mathbf{w}, w_0) \geq 1 - \xi_i \quad i = 1, \dots, l \\ \xi_i \geq 0 \quad i = 1, \dots, l \end{cases}$$

Решение исходной задачи выражается через решение двойственной:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \\ w_0 = \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \end{cases}$$

Решение исходной задачи выражается через решение двойственной:

$$\begin{cases} \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \\ w_0 = \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \end{cases}$$

$$a(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^l \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle - w_0\right)$$

Как использовать для задачи
регрессии?

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

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

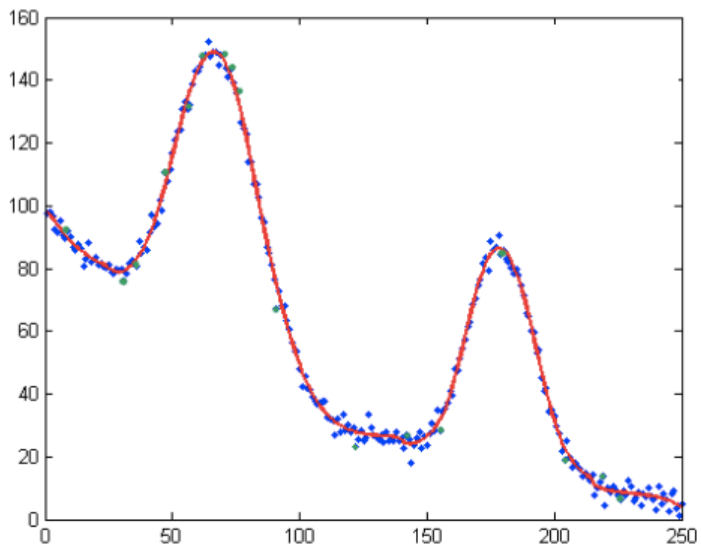
$$\begin{cases} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \hat{\xi}_i) \rightarrow \min_{\mathbf{w}, \xi} \\ y_i - \varepsilon - \xi_i \leq \langle \mathbf{w}, \mathbf{x}_i - w_0 \rangle \leq y_i + \varepsilon + \xi_i & i = 1, \dots, l \\ \xi_i, \hat{\xi}_i \geq 0 & i = 1, \dots, l \end{cases}$$

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

$$\begin{cases} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \hat{\xi}_i) \rightarrow \min_{\mathbf{w}, \xi} \\ y_i - \varepsilon - \xi_i \leq \langle \mathbf{w}, \mathbf{x}_i - w_0 \rangle \leq y_i + \varepsilon + \xi_i & i = 1, \dots, l \\ \xi_i, \hat{\xi}_i \geq 0 & i = 1, \dots, l \end{cases}$$

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

Пример



Вопросы?

На следующей лекции

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