Lecture 8

Non-linear regression

Machine Learning

Andrey Filchenkov

20.10.2017

Lecture plan

- Regression problem and non-linear regression
- Second order methods
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

- The presentation is prepared with materials of the K.V. Vorontsov’s course “Machine Leaning”.
- Slides are available online: **goo.gl/Wkif2w**

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Problem formalization

X is an object set, Y is an answer set,

$y\colon X \to Y$ is an unknown dependency, $Y \in \mathbb{R}$

$X^\ell = \{x_1, \dots, x_\ell\}$ is training sample,

$T^\ell = \{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ is set of instances.

$a(x) = f(x, \theta)$ is a dependency model, $\theta \in \mathbb{R}^t$.

Ordinary least squares:

$$Q(a, T^\ell) = \sum_{i=1}^{\ell} w_i (f(x_i, \theta) - y_i)^2 \to \min_\theta$$

Non-parametric regression

Basic idea: let think, that $\theta(x) = \theta$ nearby $x \in X$:

$$Q(\theta, T^\ell) = \sum_{i=1}^{\ell} w_i(x)(\theta - y_i)^2 \to \min_{\theta \in \mathbb{R}}.$$

Main idea: let use kernel smoothing:

$$w_i(x) = K\left(\frac{\rho(x_i, x)}{h}\right),$$

where h is window width.

Linear regression

Model of multidimensional linear regression:

$$f(x,\theta) = \sum_{j=1}^{n} \theta_j f_j(x), \quad \theta \in \mathbb{R}^n.$$

Quality in matrix notation:

$$Q(\theta, T^\ell) = \sum_{i=1}^{\ell} (f(x_i,\theta) - y_i)^2 = ||F\theta - y||^2 \to \min_{\theta\in\mathbb{R}}.$$

Non-linear regression

Its general case:

$$Q(a, T^\ell) = \sum_{i=1}^{\ell} w_i (f(x_i, \theta) - y_i)^2 \to \min_{\theta}$$

$a(x) = f(x, \theta)$ is a dependency model.

Given dependency model, we can simply search the optimum with **gradient descent** or any other optimization method.

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Newton-Raphson method

$$Q\left(a, T^{\ell}\right) = \sum_{i=1}^{\ell} (f(x_i, \theta) - y_i)^2 \to \min_{\theta \in \mathbb{R}^p}.$$

1. Choose an initial guess $\theta^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_p^{(0)}\right)$.
2. Repeat iteratively:

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t \left(Q''\left(\theta^{(t)}\right)\right)^{-1} Q'\left(\theta^{(t)}\right),$$

where $Q'\left(\theta^{[t]}\right)$ is gradient of Q in $\theta^{(t)}$;

$Q''\left(\theta^{[t]}\right)$ is a hessian Q in $\theta^{(t)}$;

η_t is step (usually $\eta_t = 1$).

Gradient and hessian

jth element of gradient:

$$\frac{\partial Q(\theta)}{\partial \theta_j} = 2\sum_{i=1}^{\ell}(f(x_i,\theta) - y_i)\frac{\partial f(x_i,\theta)}{\partial \theta_j}.$$

(j,k)th element of hessian:

$$\frac{\partial^2 Q(\theta)}{\partial \theta_j \delta \theta_k} = 2\sum_{i=1}^{\ell}\frac{\partial f(x_i,\theta)}{\partial \theta_j}\frac{\partial f(x_i,\theta)}{\partial \theta_k} -$$

$$-2\sum_{i=1}^{\ell}(f(x_i,\theta) - y_i)\frac{\partial^2 f(x_i,\theta)}{\partial \theta_j \partial \theta_k}.$$

Problem

It is very inconvenient to compute hessian each time in each point.

To avoid this, **quasi-netwon methods** are used to use approximate estimation of hessian.

Newton-Gauss method

Main idea is **linearization**:

$$f(x_i, \theta) \approx f\left(x_i, \theta^{(t)}\right) + \sum_{j=1}^{p} \left(\theta_j - \theta_j^{(t)}\right) \frac{\partial f\left(x_i, \theta_j^{(t)}\right)}{\partial \theta_j} +$$

$$+ o\left(\theta_j - \theta_j^{(t)}\right).$$

$F_t = F_t = \left(\frac{\partial f_i}{\partial \theta_j}\left(x_i, \theta^{(t)}\right)\right)_{j=1..\ell}^{j=1..p}$ is first derivatives matrix.

$f_t = \left(f\left(x_i, \theta^{[t]}\right)\right)_{i=1..\ell}$ is vector of f values.

Newton-Gauss as linear regression series

$$\theta^{(t+1)} = \theta^{(t)} - h_t\left(F_t^\top F_t\right)^{-1} F_t\left(f^{(t)} - y\right),$$

$\beta = \left(F_t^\top F_t\right)^{-1} F_t\left(f^{(t)} - y\right)$ is a solution for the problem

$$\left\|F_t\beta - \left(f^{(t)} - y\right)\right\|^2 \to \min_{\beta}.$$

This is a series of linear regression problems.

It converges with the same speed as Netwon-Raphson method.

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Logistic regression

Constraint: $Y = \{-1, +1\} = \{y_{-1}, y_{+1}\}$

Linear classifier:

$$a_w(x, T^\ell) = \text{sign}\left(\sum_{i=1}^{n} w_i f_i(x) - w_0\right).$$

where $w_1, \dots, w_n \in \mathbb{R}$ are features weights.

$$a_w(x, T^\ell) = \text{sign}(\langle w, x\rangle).$$

$$\Pr(y|x) = \sigma(\langle w, x\rangle y),$$

where $\sigma(s) = \frac{1}{1+e^{-s}}$, which is **logistic (sigmoid) function**

Logarithmic loss function

$$\widetilde{Q_w}(a, T^\ell) = \sum_{i}^{\ell} \ln(1 + \exp(-\langle w, x \rangle y)) \to \min_w .$$

We can apply Newton-Raphson method:

$$w^{(t+1)} = w^{(t)} - \eta_t \left(Q''\left(w^{(t)}\right) \right)^{-1} Q'\left(w^{(t)}\right).$$

Newton-Raphson application

jth element of gradient:

$$\frac{\partial Q(w)}{\partial w_j} = -\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i)\,,$$

(j,k)th element of hessian:

$$\frac{\partial^2 Q(w)}{\partial w_j \delta w_k} = \sum_{i=1}^{\ell}(1-\sigma_i)\sigma_i f_j(x_i) f_k(x_i)\,,$$

where $\sigma_i = \sigma(y_i w^\top x_i)$.

Newton-Raphson application

$F_{\ell \times n} = (f_i(x_i))$ is features-objects matrix;

$\Gamma_{\ell \times \ell} = \mathrm{diag}\Big(\sqrt{(1-\sigma_i)\sigma_i}\Big)$;

$\tilde{F} = \Gamma F$ is weighted features-objects matrix;

$\tilde{y}_i = y_i \sqrt{(1-\sigma_i)\sigma_i}$, $(\tilde{y}_i)_{i=1}^{\ell}$ is a weighted answer vector.

$$(Q''(w))^{-1} Q'(w) = -\left(F^\top \Gamma^2 F\right)^{-1} F^\top \Gamma \tilde{y} =$$
$$= -\left(\tilde{F}^\top \tilde{F}\right)^{-1} \tilde{F}^\top \tilde{y} = -\tilde{F}^{+} \tilde{y}.$$

Logistic regression solution

$$Q(w) = \|\tilde{F}w - \tilde{y}\|^2 =$$

$$= \sum_{i=1}^{\ell} (1 - \sigma_i)\sigma_i \left(w^\top x - \frac{y_i}{\sigma_i} \right)^2 \to \min_w .$$

σ_i is a probability of true classification.

$(1 - \sigma_i)\sigma_i$ is degree of “sureness” of object classification, which is margin.

This solution is performed in a way if we apply regression to solve classification.

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Feature transformation

Basic assumption: regression model is a sum of different functions of features (but monoms).

$$f(x, \theta) = \sum_{j=1}^{n} \varphi_j\big(f_j(x)\big).$$

Idea: sequentially approximate functions φ_j.

Backfitting method

$$Q\left(\varphi_j, T^\ell\right) =$$

$$= \sum_{i=1}^{\ell} \left(\varphi_j(f_j(x_i)) - \left(y_i - \sum_{k=1, k \neq j}^{n} \varphi_k(f_k(x_i)) \right) \right)^2 .$$

$$z_i := y_i - \sum_{k=1, k \neq j}^{n} \varphi_k(f_k(x_i)) = \text{const}\left(\varphi_j\right).$$

This is problem of single variable optimization.

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Group Method of Data Handling

1. Start with fix set of models (for example, linear).
2. Estimate the best parametrization.
3. Increase model complexity, until it improves solution.

Usually Kolmogorov-Gabor polynoms are used:

$$y = w_0 + \sum_{i=1}^{\ell} w_i f_i(x) + \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} w_{i,j} f_i(x) f_j(x) + \cdots$$

Lecture plan

- Regression problem and non-linear regression
- Second derivative tests
- Rethinking logistic regression
- Feature transformation
- Group method of data handling
- Outlier management

Outliers handling

Problem: we need to clean **outliers** (which can be understood as noise).

Weighting objects

$\varepsilon_i = \mathrm{LOO}(x_i) = L(a(x_i; T^\ell \setminus \{x_i\}))$.

This can be used as a weight in general approach:

$$Q(a, T^\ell) = \sum_{i=1}^{\ell} K(\varepsilon_i)(f(x_i, \theta) - y_i)^2 \to \min_\theta .$$

LOWESS method (LOcally WEighted Scatter plot Smoothing) for nonparametric regression:

use $\varepsilon_i = |a - y_i|$ as a loss function;

use kernel function $K(\varepsilon_i) = K_Q\left(\frac{\varepsilon_i}{6\mathrm{med}\varepsilon_i}\right)$.

Robust regression

Regression model:

$$a(x) = f(x, \theta).$$

Meshalkin function:

$$L(x) = 1 - \exp(x^2).$$