National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
Institute of Physics and Technology

# Lecture 13

# Objects and Systems Identification Methods. Linear Regression

Dmytro Progonov,
PhD, Associate Professor,
Department Of Physics and Information Security Systems

# Content

- Model specification;
- MLE for linear regression;
- Robust linear regression;
- Ridge regression.

# Model specification (1/2)

Linear regression is a model of the form:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \varepsilon = \sum_{j=1}^{D} w_j x_j + \varepsilon$$

where $\mathbf{w}$ – model's weigh vector; $\varepsilon$ – residual vector.

There is often used assumption of Gaussian (normal) distribution of $\varepsilon$

$$\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$$

Therefore we can rewrite the model in the following form:

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mu(\mathbf{x}), \sigma^2)$$

Expected response can be represented as follows:

$$\mu(\mathbf{x}) = w_0 + w_1 x = \mathbf{w}^T\mathbf{x}$$

Then

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T\mathbf{x}, \sigma^2)$$

# Model specification (2/2)

Then final form of linear regression model is:

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T\mathbf{x}, \sigma^2)$$

Linear regression can be also made to model non-linear relationships by replacing $\mathbf{x}$ with some non-linear function of inputs $\phi(\mathbf{x})$ (***basis function expansion***):

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T\phi(\mathbf{x}), \sigma^2)$$

A simple example are polynomial basis functions:

$$\phi(\mathbf{x}) = [1, x, x^2, \cdots, x^d]$$

# MLE for linear regression (1/3)

The most common way to estimate the parameters of a statistical model is to compute the maximum likelihood estimate (MLE):

$$\widehat{\boldsymbol{\theta}} \triangleq \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log[p(\mathcal{D}|\boldsymbol{\theta})]$$

Instead of maximizing the likelihood estimate, we can equivalently minimize the ***negative log-likelihood*** (***NLL***)

$$NLL(\widehat{\boldsymbol{\theta}}) \triangleq -\sum_{i=1}^{N} \log[p(y_i|\mathbf{x}_i, \boldsymbol{\theta})]$$

Applying this method to linear regression we find that the log likelihood is given by:

$$\ell(\widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{N} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left[-\frac{1}{2\sigma^2}(y_i - \mathbf{w}^T\mathbf{x}_i)^2\right]\right]$$

# MLE for linear regression (2/3)

$$\ell(\widehat{\boldsymbol{\theta}}) = -\frac{1}{2\sigma^2} RSS(\mathbf{w}) - \frac{N}{2}\log[2\pi\sigma^2]$$

where $RSS(\mathbf{w}) \triangleq \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 = \sum_{i=1}^{N} {\varepsilon_i}^2$ – residual sum of squares

For determination the values of model's weigh vector, we rewrite the objective in a form that is more amenable to differentiation:

$$NLL(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \mathbf{w}^T(\mathbf{X}^T\mathbf{y})$$

where

$$\mathbf{X}^T\mathbf{X} = \sum_{i=1}^{N}\mathbf{x}_i{\mathbf{x}_i}^T = \sum_{i=1}^{N}\begin{pmatrix} {x_{i,1}}^2 & \cdots & x_{i,1}x_{i,D} \\ \vdots & \ddots & \vdots \\ x_{i,D}x_{i,1} & \cdots & {x_{i,D}}^2 \end{pmatrix}$$

$$\mathbf{X}^T\mathbf{y} = \sum_{i=1}^{N}\mathbf{x}_i y_i$$

# MLE for linear regression (3/3)

The gradient of $NLL(\mathbf{w})$ is given by

$$\mathbf{g}(\mathbf{w}) = [\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y}] = \sum_{i=1}^{N} \mathbf{x}_i(\mathbf{w}^T\mathbf{x}_i - y_i)$$

Equating to zero we get

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

This is known as the normal equation (orthogonal projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$). The corresponding solution $\hat{\mathbf{w}}$ to this linear system of equation is called the ***ordinary least square*** (***OLS***) solution:

$$\hat{\mathbf{w}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

# Robust linear regression (1/3)

It is very common to model the noise in regression model using a Gaussian distribution. However, if we have ***outliers*** in our data, this can result in a poor fit. This is because squared error penalizes deviations quadratically, so point far from the line have more affect on the fit than points near to the line.

One way to achieve robustness to outliers is to replace the Gaussian distribution for the response variable with a ***distribution that has heavy tails***, such as Laplace distribution. Then we get the follosing likelihood:

$$p(y|\mathbf{x}, \mathbf{w}, b) = Lap(y|\mathbf{w}^T\mathbf{x}, b) \propto \exp\left[-\frac{1}{b}|y - \mathbf{w}^T\mathbf{x}|\right]$$

# Robust linear regression (2/3)

One way to compute model's weigh vector under Laplace likelihood is to minimize the ***Huber loss function*** (Huber, 1964):

$$L_H(r,\delta) = \begin{cases} r^2/2 & if\ |r| \le \delta \\ \delta|r| - \delta^2/2 & if |r| > \delta \end{cases}$$

Alternative way consists in minimization of negative log-likelihood:

$$\ell(\mathbf{w}) = \sum_i |y_i - \mathbf{w}^T \mathbf{x}_i| = \sum_i |r_i(\mathbf{w})|$$

Unfortunately, this is a non-linear objective function, which is hard to optimize. Fortunately, we can convert the NLL to a linear objective, subject to linear constrains, using the ***split variable trick***:

$$r_i \triangleq r_i^+ + r_i^-, r_i^+ \ge 0, r_i^- \ge 0$$

# Robust linear regression (3/3)

Now the constrained objective becomes

$$\min_{\boldsymbol{w},\boldsymbol{r}^+,\boldsymbol{r}^-} \sum_i (r_i^+ + r_i^-), s.t. r_i^+ \geq 0, r_i^- \geq 0, \mathbf{w}^T\mathbf{x}_i + r_i^+ - r_i^- = y_i$$

This is an example of a ***linear program*** (***LP***) with $D + 2N$ unknowns and $3N$ constrains.

To solve a LP, we must write it in standard form:

$$\min_{\boldsymbol{\theta}} \boldsymbol{f}^T\boldsymbol{\theta} \;\; s.t. \mathbf{A}\boldsymbol{\theta} \leq \boldsymbol{b}, \mathbf{A}_{eq}\boldsymbol{\theta} = \boldsymbol{b}_{eq}, \boldsymbol{l} \leq \boldsymbol{\theta} \leq \boldsymbol{u}$$

where $\boldsymbol{\theta} = (\boldsymbol{w}, \boldsymbol{r}^+, \boldsymbol{r}^-)$, $\boldsymbol{f} = [\mathbf{0}, \mathbf{1}, \mathbf{1}]$, $\mathbf{A} = []$, $\boldsymbol{b} = []$, $\mathbf{A}_{eq} = [\boldsymbol{X}, \mathbf{I}, -\mathbf{I}]$, $\boldsymbol{b}_{eq} = \boldsymbol{y}$, $\mathbf{l} = [-\infty\mathbf{1}, \mathbf{0}, \mathbf{0}]$, $\boldsymbol{u} = []$.

# Ridge regression (1/2)

One problem with ML estimation is that it can result in overfitting – if we changed the data a little, the obtained coefficients would change a lot.

We can encourage the parameters of MLE to be small, thus resulting in a smoother curve, by using a zero-mean Gaussian prior:

$$p(\mathbf{w}) = \prod_j \mathcal{N}\left(w_j | 0, \tau^2\right)$$

where $1/\tau^2$ controls the strength of the prior. The corresponding MAP estimation problem becomes

$$\underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \log[\mathcal{N}(y_i | w_0 + \mathbf{w}^T \mathbf{x}, \sigma^2)] + \sum_{j=1}^{D} \log\left[\mathcal{N}\left(w_j | 0, \tau^2\right)\right]$$

# Ridge regression (2/2)

This is equivalent to minimizing the following:

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \mathbf{w}^T\mathbf{x}_i))^2 + \lambda\|\mathbf{w}\|_2{}^2$$

where $\lambda \triangleq \sigma^2/\tau^2$ is a complexity penalty and $\|\mathbf{w}\|_2{}^2 = \sum_j w_j{}^2 = \mathbf{w}^T\mathbf{w}$ is the squared two-norm. The corresponding solution is given by:

$$\hat{\mathbf{w}}_{ridge} = (\lambda\mathbf{I}_D + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

This technique is called ***penalized least squares***. In general, adding a Gaussian prior to the parameters of a model to encourage them to be small is called $\ell_2$ **regularization** or ***weight decay***.

# Conclusion

- Model specification for Linear Regression was shown;
- MLE for linear regression was considered;
- Special types of linear regression, such as robust and ridge regressions, were presented.