Theory and Concepts of Regression

Abstract:

× We make the connection between linear regression and a Gaussian distribution

× We show that the MSE is a maximum likelihood estimation procedure

Source:

- Kevin P. Murphy: Machine Learning: A Probabilistic Perspective
- Runkler, Skiena

Properties of Maximum Likelihood:

Consider $\mathbb{X} = \{\vec{x}_1, \dots, \vec{x}_N\}$ N examples drawn from $p_{data}(\vec{x})$. Let $p_{MODEL}(\vec{x}, \vec{\theta})$ be a parametric model used to estimate $p_{data}(\vec{x})$

The maximum likelihood estimator for $\vec{\theta}$ is then defined as

$$\vec{\theta}_{ML} = \arg\max_{\vec{\theta}} \left[p_{MODEL}(\mathbb{X}; \vec{\theta}) \right]$$
$$= \arg\max_{\vec{\theta}} \left[\prod_{i=1}^{N} p_{MODEL}(\vec{x}_i; \vec{\theta}) \right]$$

For convenience take the log (does not change its arg max):

$$\vec{\theta}_{ML} = \arg\max \left[\sum_{i=1}^{N} \log p_{MODEL}(\vec{x}_i; \vec{\theta}) \right]$$

<u>Interpretation</u>: Divide the formula by N to arrive at a expectation value:

$$\vec{\theta}_{ML} = \arg\max_{\vec{\theta}} \mathbb{E}[\log p_{MODEL}(\vec{x}, \vec{\theta})]$$

One way to interpret it is to view it as minimizing the dissimilarity between the empirical distribution $\hat{p}_{DATA}$ and the model distribution p_{MODEL}:

$$\underbrace{\mathbb{E}[\log \hat{p}_{DATA}(\vec{x}) - \log p_{MODEL}(\vec{x})]}$$

minimize the cross-entropy between the distributions

The mean squared error (MSE) is the cross-entropy between empirical data and a Gaussian model!

Asymptotically ($N \to \infty$) the MLE is the best estimator: ~~When~~ If the number of training examples would approach infinity, the MLE of a parameter converges to the true value of the parameter.

1.1 Linear regression

Linear regression formula:

$$y(\vec{x}) = \vec{\omega}^T \cdot \vec{x} + \varepsilon =$$

$$= \sum_{j=1}^{D} \omega_j x_j + \varepsilon$$

with $\vec{\omega}$ the weight vector,

ε the residual error (between our linear predictions and the true response)

$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$

We often assume $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$ Gaussian distributed

To make the connection between linear regression and Gaussians more explicit, we rewrite the model as

$$p(y|\vec{x}, \vec{\theta}) = \mathcal{N}(y|\overbrace{\mu(\vec{x})}^{\text{mean of the Gaussian}}, \sigma^2(\vec{x}))$$

with $\mu = \vec{\omega}^T \cdot \vec{x}$, $\sigma^2(\vec{x}) = \sigma^2$,
$\vec{\theta} = (\vec{\omega}, \sigma^2)$

For 1-dimensional input:

$$\mu(\vec{x}) = \underbrace{\omega_0}_{\text{intercept/bias term}} + \underbrace{\omega_1}_{\text{slope}} \cdot x = \vec{\omega}^T \cdot \vec{x}$$

Non-linear relationships by replacing $\vec{x}$ with some non-linear function $\phi(\vec{x})$:

$$p(y \mid \vec{x}, \vec{\theta}) = \mathcal{N}(y \mid \vec{\omega}^T \cdot \phi(\vec{x}), \sigma^2)$$

basis function

constant variance

Example: Polynomial regression

$$\phi(x) = [1, x, x^2, \ldots, x^d]$$

1.2. Logistic regression

Two changes for binary classification:

a) Gaussian $\longrightarrow$ Bernoulli distribution
$\uparrow$ tossing a coin

$$p(y|\vec{x},\vec{\omega}) = \mathrm{Ber}(y|\mu(\vec{x}))$$

where $\mu(\vec{x}) = \mathbb{E}(y(\vec{x})) = p(y=1|\vec{x})$

b) Pass $\vec{\omega}^T \vec{x}$ through a function that ensures $0 \leq \mu(\vec{x}) \leq 1$

$$\mu(\vec{x}) = \mathrm{sigm}(\vec{\omega}^T \cdot \vec{x}),$$

where sigm = sigmoid/logistic/logit is defined as

$$\mathrm{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{e^{\eta}+1}$$

Putting these two steps together we get

$$p(y|\vec{x},\vec{\omega}) = Ber(y|\,sigm(\vec{\omega}^T\cdot\vec{x}))$$

This is called logistic regression due to its similarity to linear regression.

If we choose a threshold 0.5, we can induce a decision rule (linear decision boundary)

$$\hat{y}(x) = 1 \iff p(y=1|\vec{x}) > 0.5$$

2. Maximum Likelihood Estimation

As discussed

$$p(y|\vec{x},\vec{\theta}) = \mathcal{N}(y|\underbrace{\vec{\omega}^T\cdot\vec{x}}_{\mu(\vec{x})},\sigma^2)$$

To estimate the parameters $\vec{\theta}$ we define the MLE (= maximum likelihood estimation):

$$\hat{\vec{\theta}} = \arg\max_{\vec{\theta}} [\log p(y|\vec{\theta})]$$

We can rewrite the log-likelihood as follows (assuming that the training examples are independent and identically distributed)

$$\ell(\vec{\theta}) = \log p(y|\vec{\theta}) = \sum_{i=1}^{N} \log p(y_i|\vec{x}_i,\vec{\theta})$$

Instead of maximizing the log-likelihood, we can minimize the negative log likelihood or NLL:

more convenient for numerical evaluation

$$NLL(\vec{\theta}) = -\sum_{i=1}^{N} \log p(y_i | \vec{x}_i, \vec{\theta})$$

Now we insert the definition of the Gaussian into the log-likelihood $\ell(\vec{\theta})$:

$$\ell(\vec{\theta}) = \sum_{i=1}^{N} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left(-\frac{1}{2\sigma^2}(y_i - \vec{\omega}^T \cdot \vec{x}_i)^2\right)\right]$$

$$= -\frac{1}{2\sigma^2} \underbrace{\sum_{i=1}^{N} (y_i - \vec{\omega}^T \cdot \vec{x}_i)^2}_{RSS} - \frac{N}{2}\log(2\pi\sigma^2)$$

RSS = residual sum of squares

$RSS(\omega) = |\varepsilon|_2^2$ (ℓ_2-norm)

$$MSE = \frac{RSS}{N}$$ mean squared error

The MLE (maximum likelihood estimation) for $\vec{\omega}$ is the one that minimizes the RSS, so this method is known as least squares. $\Rightarrow$ MSE is a maximum likelihood estimation procedure

Next we derive the maximum likelihood estimation (MLE):

$$NLL(\vec{\omega}) = \frac{1}{2}(\vec{y} - \vec{X}\vec{\omega})^T(\vec{y} - \vec{X}\cdot\vec{\omega})$$

$$= \frac{1}{2}\vec{\omega}^T\vec{X}^T\vec{X}\vec{\omega} - \vec{\omega}^T(\vec{X}^T\cdot\vec{y})$$

$$+ \frac{1}{2}\vec{y}^2$$

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

sum of squares matrix

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

The gradient $g(\vec{\omega})$ is given as

$$g(\vec{\omega}) = [\vec{X}^T \cdot \vec{X} \cdot \vec{\omega} - \vec{X}^T \cdot \vec{y}] =$$

$$= \sum \vec{x}_i (\vec{\omega}^T \cdot \vec{x}_i - y_i)$$

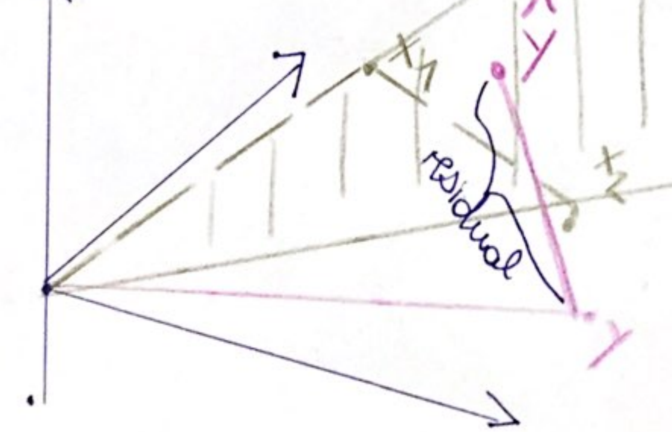
Equation to zero ($g(\vec{\omega}) = 0$) gives

$$\boxed{\vec{X}^T \cdot \vec{X} \cdot \vec{\omega} = \vec{X}^T \cdot \vec{y}}$$ normal equation

The solution to this equation is

$$\boxed{\hat{\omega}_{OLS} = (\vec{X}^T \cdot \vec{X})^{-1} \cdot \vec{X}^T \cdot \vec{y}}$$ ordinary least squares solution

Graphical interpretation:



- x_1 and x_2 define a 2D plane
- y does not lie on that plane
- the orthogonal projection of y on that plane is $\hat{y}$

1-dim example:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \omega_0 - \omega_1 x_i \right)^2$$

$$\frac{\partial MSE}{\partial \omega_0} = -\frac{2}{N} \sum_{i=1}^{N} \left(y_i - \omega_0 - \omega_1 x_i \right) \stackrel{!}{=} 0$$

$$\Rightarrow \omega_0 = \bar{y} - \omega_1 \bar{x}$$

$$\Rightarrow MSE = \frac{1}{N} \sum \left(y_i - \bar{y} - \omega_1 (x_i - \bar{x}) \right)^2$$

$$\frac{\partial MSE}{\partial \omega_1} = -\frac{2}{N} \sum (x_i - \bar{x}) \left(y_i - \bar{y} - \omega_1 (x_i - \bar{x}) \right) \stackrel{!}{=} 0$$

$$\Rightarrow \boxed{\omega_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{c_{ij}}{c_{jj}}}$$

Calculation example:

$$y = \begin{bmatrix} 6 \\ 2 \\ 0 \\ 0 \\ 2 \end{bmatrix} \qquad X = \begin{bmatrix} 4 & -2 \\ 1 & -1 \\ 0 & 0 \\ 1 & 1 \\ 4 & 2 \end{bmatrix}$$

$$\bar{y} = \frac{6+2+2}{5} = 2$$

$$\vdots$$

$$w_0 = 2 \;, \; w_1 = 1 \;, \; w_2 = -1$$

3. Robust linear regression

To model the noise in regression models using a Gaussian distribution can be a bad choice when we have outliers in our fit. This is because squared error penalizes deviations quadratically, so points far from the line have more effect on the fit than points near the line.

Idea: Replace the Gaussian for the response variable with a distribution that has heavy tails.

One possibility is to use the Laplace distrib:

$$p(y|\vec{x},\vec{\omega},b) = \mathrm{Lap}(y|\vec{\omega}^T\cdot\vec{x},b)$$

$$\sim \exp\left(-\frac{1}{b}|y-\vec{\omega}^T\cdot\vec{x}|\right)$$

$$\left(\frac{1}{2b}e^{-\frac{|x-\mu|}{b}}\right)$$

It assigns higher likelihood to outliers.

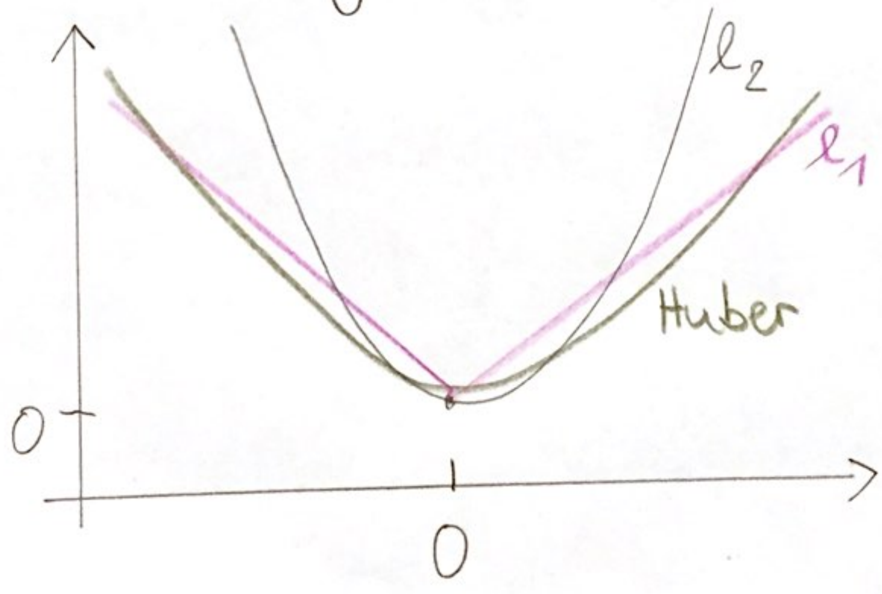
The robustness arises from using $|y - \vec{\omega}^T \cdot \vec{x}|$ instead of $(y - \vec{\omega}^T \cdot \vec{x})^2$.

$\ell(\vec{\omega}) = \sum |y_i - \vec{\omega}^T \vec{x}_i|$ log-likelihood

However, this non-linear objective function is hard to optimize. An alternative is to minimize the Huber loss

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

This is equivalent to ℓ_2 for errors smaller than δ and ℓ_1 for larger errors. This loss function is everywhere differentiable.



4. Ridge Regression

Changing the log-likelihood $\ell(\vec{\theta})$ to:

$$\underset{\vec{\omega}}{\arg\max}\left\{\sum_{i=1}^{N}\log\mathcal{N}(y_i\,|\,\omega_0+\vec{\omega}^T\cdot\vec{x}_i,\sigma^2)+\underbrace{\sum_{j=1}^{D}\log\mathcal{N}(\omega_j\,|\,0,\tau^2)}_{\text{zero-mean Gaussian prior}}\right\}$$

$P(A)\cdot P(B|A) = P(B)\cdot P(A|B)$

zero-mean Gaussian prior
(to encourage parameters to be small)

This is equivalent to minimizing

$$J(\vec{\omega})=\frac{1}{N}\sum_{i=1}^{N}\underbrace{\left(y_i-(\omega_0+\vec{\omega}^T\cdot\vec{x}_i)\right)^2}_{\text{MSE}}+\underbrace{\lambda\|\vec{\omega}\|_2^2}_{\text{penalty}}$$

with $\lambda=\frac{\sigma^2}{\tau^2}$ and $\|\omega\|_2^2=\vec{\omega}^T\omega=\sum_{j=1}^{D}\omega_j^2$ squared two-norm.

The corresponding solution is given as

$$\boxed{\hat{\omega}_{Ridge} = (\lambda \mathbb{1}_D + \vec{X}^T \cdot \vec{X})^{-1} \vec{X}^T \cdot \vec{y}}$$ penalized least squares

Adding a Gaussian prior to the parameters of the model is called Ridge / ℓ_2 regular. or weight decay.

Note: $(\lambda \mathbb{1}_D + \vec{X}^T \cdot \vec{X})$ is easier to invert than $\vec{X}^T \cdot \vec{X}$, at least for large enough λ.

Note: In theory ridge regression is a biased MLE. A more "effective" approach is to increase the training data N. The approximation error should go to zero for $N \to \infty$.

5. Logistic Regression

$$p(y \mid \vec{x}, \vec{\omega}) = \mathrm{Ber}(y \mid \mathrm{sigm}(\vec{\omega}^T \cdot \vec{x}))$$

$$NLL(\vec{\omega}) = -\sum_{i=1}^{N} \log\left[\mu_i^{\mathbb{1}(y_i=1)} \cdot (1-\mu_i)^{\mathbb{1}(y_i=0)}\right]$$

$$= -\sum_{i=1}^{N} \left[y_i \log \mu_i + (1-y_i)\log(1-\mu_i)\right]$$

Unlike for linear regression we can no longer write down the MLE in closed form. We need an optimization algorithm to compute it.

⇝ gradient descent