Linear Regression and Generalized Linear Models

Kevin Murphy

Presenter: Zhen Qiu - ANU - 2023

Introduction

The key property of the model is that the expected value of the output is assumed to be a linear function of the input, $\mathbb{E}[y \mid \boldsymbol{x}] = \boldsymbol{w}^\top \boldsymbol{x}$, where $\boldsymbol{w}$ are the weights, which makes the model easy to interpret, and easy to fit to data.

Introduction

The key property of the model is that the expected value of the output is assumed to be a linear function of the input, $\mathbb{E}[y \mid \boldsymbol{x}] = \boldsymbol{w}^\top \boldsymbol{x}$, where $\boldsymbol{w}$ are the weights, which makes the model easy to interpret, and easy to fit to data.

The "linear regression" usually refers to a model of the following form:

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\left(y \mid w_0 + \boldsymbol{w}^\top \boldsymbol{x}, \sigma^2\right)$$

where $\boldsymbol{\theta} = \left(w_0, \boldsymbol{w}, \sigma^2\right)$ are all the parameters of the model.

Introduction

The key property of the model is that the expected value of the output is assumed to be a linear function of the input, $\mathbb{E}[y \mid \boldsymbol{x}] = \boldsymbol{w}^\top \boldsymbol{x}$, where $\boldsymbol{w}$ are the weights, which makes the model easy to interpret, and easy to fit to data.

The "linear regression" usually refers to a model of the following form:

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\left(y \mid w_0 + \boldsymbol{w}^\top \boldsymbol{x}, \sigma^2\right)$$

where $\boldsymbol{\theta} = \left(w_0, \boldsymbol{w}, \sigma^2\right)$ are all the parameters of the model.

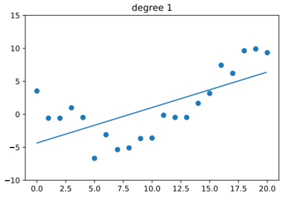
If the input is 1-D, simple linear regression:

$$f(\boldsymbol{x}; \boldsymbol{w}) = ax + b$$

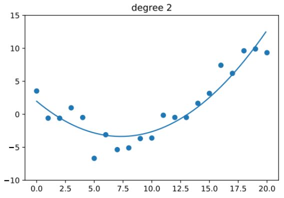
If the input is N-D, multiple/multivariate linear regression:

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{W}) = \prod_{j=1}^{J} \mathcal{N}\left(y_j \mid \boldsymbol{w}_j^\top \boldsymbol{x}, \sigma_j^2\right)$$

Comparing polynomial of degrees 1 and 2



(a)



(b)

Figure 11.1: Polynomial of degrees 1 and 2 fit to 21 datapoints. Generated by linreg_poly_vs_degree.ipynb.

In general, a straight line will not provide a good fit to most data sets. However, we can always apply a nonlinear transformation to the input features, by replacing x with $\sigma(x)$ to get

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\left(y \mid \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \sigma^2\right)$$

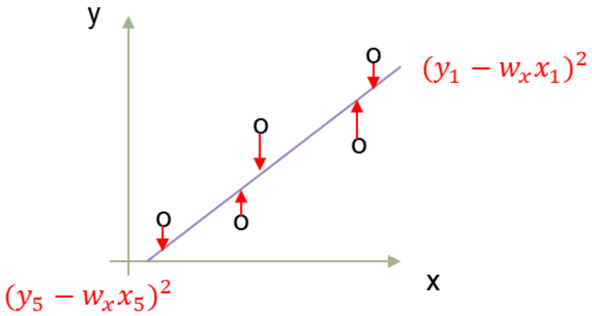
Least Square Estimation

$$\mathrm{NLL}\left(\boldsymbol{w}, \sigma^2\right) = -\sum_{n=1}^{N} \log \underbrace{\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\left(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n\right)^2\right)\right]}_{\textit{Gaussian}} \quad (1)$$

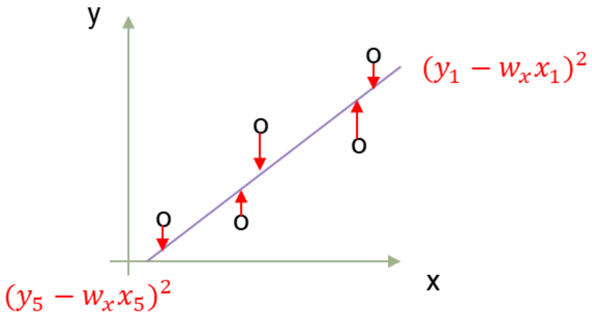
$$= \frac{1}{2\sigma^2} \sum_{n=1}^{N} \underbrace{\left(y_n - \hat{y}_n\right)^2}_{\textit{Error}} + \frac{N}{2} \log \underbrace{\left(2\pi\sigma^2\right)}_{\textit{Variance}} \quad (2)$$

The MLE is the point where $\nabla_{w,\sigma} NLL\left(w, \sigma^2\right) = 0$.
We can first optimize wrt w, and then solve for the optimal σ.

OLS-1D



OLS-1D

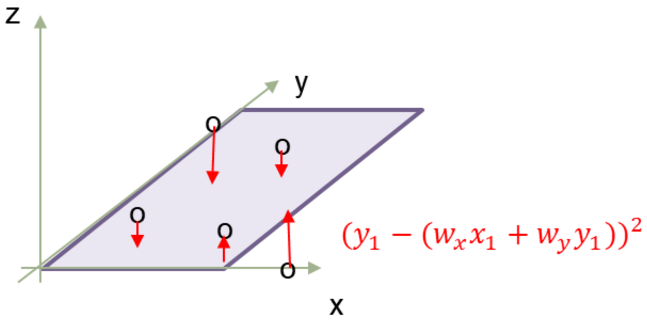


$$\mathrm{RSS}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n\right)^2 = \frac{1}{2}\|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 = \frac{1}{2}(\mathbf{X}\boldsymbol{w} - \boldsymbol{y})^\top(\mathbf{X}\boldsymbol{w} - \boldsymbol{y})$$

$$\begin{aligned}&\begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} - \begin{bmatrix} w_x \end{bmatrix}\begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \\ &= \begin{bmatrix} (y_1 - w_x x_1)^2 & \cdots & (y_n - w_x x_n)^2 \end{bmatrix}\end{aligned}$$

OLS-2D

Similarly, for 2D, we could have the figure below



$$\begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} - \begin{bmatrix} w_x \\ w_y \end{bmatrix}^T \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$
$$= \begin{bmatrix} (y_1 - (w_x x_1 + w_y y_1))^2 & \cdots & (y_n - (w_x x_n + w_y y_n))^2 \end{bmatrix}$$

How to get w

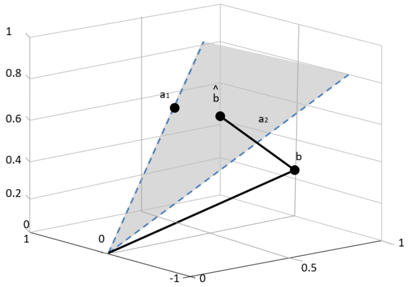


Figure 11.3: Graphical interpretation of least squares for $m = 3$ equations and $n = 2$ unknowns when solving the system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$. $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are the columns of $\mathbf{A}$, which define a 2d linear subspace embedded in $\mathbb{R}^3$. The target vector $\boldsymbol{b}$ is a vector in $\mathbb{R}^3$; its orthogonal projection onto the linear subspace is denoted $\hat{\boldsymbol{b}}$. The line from $\boldsymbol{b}$ to $\hat{\boldsymbol{b}}$ is the vector of residual errors, whose norm we want to minimize.

How to get w

We seek a vector $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ that lies in the linear subspace spanned by $\boldsymbol{X}$ and is as close as possible to $\boldsymbol{y}$, i.e.,

$$\underset{\hat{\boldsymbol{y}} \in \operatorname{span}\left(\left\{\boldsymbol{x}_{:,1}, \ldots, \boldsymbol{x}_{:,d}\right\}\right)}{\operatorname{argmin}} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2$$

where $x_{:,d}$ is the d'th column of $\boldsymbol{X}$.
Since $\hat{\boldsymbol{y}} \in \operatorname{span}(\mathbf{X})$, there exists some weight w such that

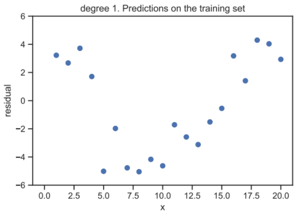
$$\hat{\boldsymbol{y}} = w_1 \boldsymbol{x}_{:,1} + \cdots + w_D \boldsymbol{x}_{:,D} = \mathbf{X}\boldsymbol{w}$$

To minimize the norm of the residual, $\boldsymbol{y} - \hat{\boldsymbol{y}}$, we want the residual vector to be orthogonal to every column of $\boldsymbol{X}$.

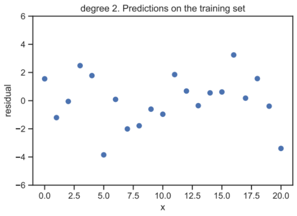
$$\boldsymbol{x}_{:,d}^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}) = 0 \Rightarrow \mathbf{X}^\top (\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) = \mathbf{0} \Rightarrow \boldsymbol{w} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{y}$$

Goodness of fit

- For 1d inputs, we can check the reasonableness of the model by plotting the residuals, $r_n = y_n - \hat{y}_n$, vs the input x_n. This is called a **residual plot**.



(a)

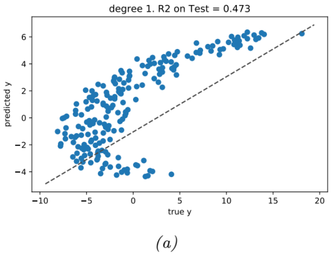


(b)

Figure 11.5: Residual plot for polynomial regression of degree 1 and 2 for the functions in Figure 1.7a(a-b). Generated by linreg_poly_vs_degree.ipynb.

Goodness of Fit

- To extend this approach to multi-dimensional inputs, we can plot predictions $\hat{y}_n$ vs the true output y_n, rather than plotting vs x_n. A good model will have points that lie on a diagonal line.



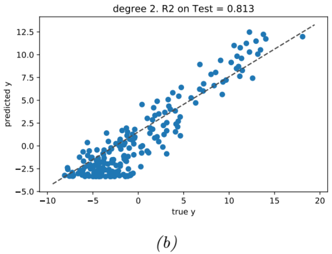


Figure 11.6: Fit vs actual plots for polynomial regression of degree 1 and 2 for the functions in Figure 1.7a(a-b). Generated by linreg_poly_vs_degree.ipynb.

Goodness of Fit

- We can assess the fit quantitatively by computing the RSS (residual sum of squares) on the dataset:

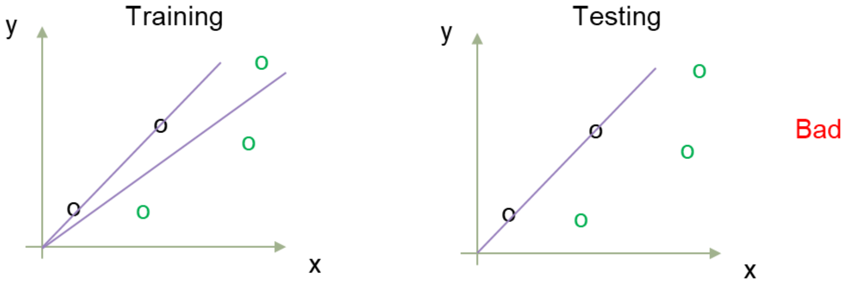
$$\mathrm{RSS}(\boldsymbol{w}) = \sum_{n=1}^{N} \left(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n \right)^2$$

- A model with lower RSS fits the data better. Another measure that is used is root mean squared error or RMSE:

$$R^2 \triangleq 1 - \frac{\sum_{n=1}^{N} (\hat{y}_n - y_n)^2}{\sum_{n=1}^{N} (\bar{y} - y_n)^2} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

where $\bar{y} = \frac{1}{N} \sum_{n=1}^{N} y_n$ is the empirical mean of the response, RSS is the residual sum of squares, and TSS is the total sum of squares.

Ridge Estimation



Ridge regression add a l_2 regularizer to avoid overfitting (avoid very high gradient).

Ridge Regression

Maximum likelihood estimation can result in overfitting. A simple solution to this is to use MAP estimation with a zero-mean Gaussian prior on the weights, $p(\boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{w} \mid \mathbf{0}, \lambda^{-1}\mathbf{I}\right)$.

Ridge Regression

Maximum likelihood estimation can result in overfitting. A simple solution to this is to use MAP estimation with a zero-mean Gaussian prior on the weights, $p(\boldsymbol{w}) = \mathcal{N}\left(\boldsymbol{w} \mid \mathbf{0}, \lambda^{-1}\mathbf{I}\right)$.

In details,

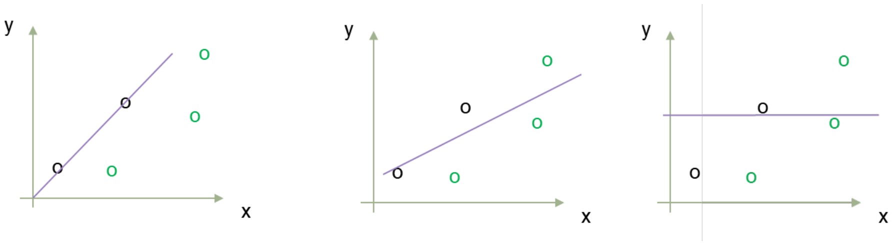
$$\begin{aligned}\hat{\boldsymbol{w}}_{\mathsf{map}} &= \operatorname{argmin} \frac{1}{2\sigma^2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) + \frac{1}{2\tau^2}\boldsymbol{w}^\top\boldsymbol{w} \\ &= \operatorname{argmin} \mathrm{RSS}(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_2^2\end{aligned}$$

where $\lambda \triangleq \frac{\sigma^2}{\tau^2}$ is proportional to the strength of the prior, and

$$\|\boldsymbol{w}\|_2 \triangleq \sqrt{\sum_{d=1}^{D} |w_d|^2} = \sqrt{\boldsymbol{w}^\top\boldsymbol{w}}$$

is the l_2 norm of the vector w.

Ridge Regression

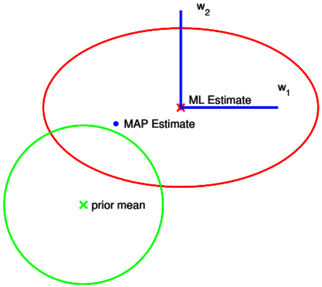


Zero $\lambda =$ Least Square   Big λ Very Big $\lambda = 10^5$

The traditional means of choosing λ is the ridge trace plot.
We could also use the cross validation.

Ridge Regression & PCA

Why ridge regression works well?



The horizontal w_1 parameter is not-well determined by the data (has high posterior variance), but the vertical w_2 parameter is well-determined. Hence $w_{map}(2)$ is close to $w_{mle}(2)$, but $w_{map}(1)$ is shifted strongly towards the prior mean, which is 0. In this way, ill-determined parameters are reduced in size towards 0. This is called shrinkage.

Ridge Regression & PCA

There is a related, but different, technique called principal components regression, which is a supervised version of PCA.

The idea is this: first use PCA to reduce the dimensionality to K dimensions, and then use these low dimensional features as input to regression. However, this technique does not work as well as ridge regression in terms of predictive accuracy.

The reason is that in PC regression, only the first K (derived) dimensions are retained, and the remaining D - K dimensions are entirely ignored. By contrast, ridge regression uses a “soft” weighting of all the dimensions.

Lasso

- Ridge allows parameters to be small but cannot reach zero.

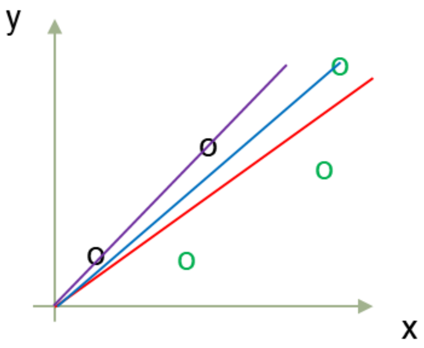
Lasso

- Ridge allows parameters to be small but cannot reach zero.
- Lasso allows parameters to be exactly zero.

Lasso

- Ridge allows parameters to be small but cannot reach zero.
- Lasso allows parameters to be exactly zero.
- This is useful because it can be used to perform feature selection, where the weight of certain features can be zero, which could make equations simpler

Lasso



Least square regression, $RSS \rightarrow$ less bias, high variance.
Ridge regression, $RSS + \lambda w_2 \rightarrow$ high bias, low variance.
Lasso regression, $RSS + \lambda|w|$

Lasso

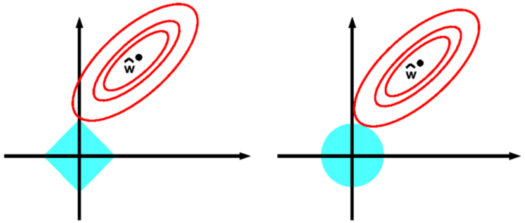


Figure: Illustration of l_1 (left) vs l_2 (right) regularization of a least squares problem.