Overfitting and underfitting of an ML model

Overfitting is a scenario when the predictive power of an ML model, as quantified by its performance on an unseen test set, is compromised due to overtraining of the model, which could still lead to an excellent performance on the training set.

→ Loss function can take very low values on training

Loss function would have a high value on the test

This occurs when the model selected is too complex and places too much importance on certain features.

eg. • model has an excessive number of parameters
• model architecture is too complex

Underfitting occurs when the chosen ML model is unable to capture the variations in data, such that it performs poorly on even the training set.

High values of loss function on both the training & test sets. Likely leading to low R^2 values.

(Model too complex)

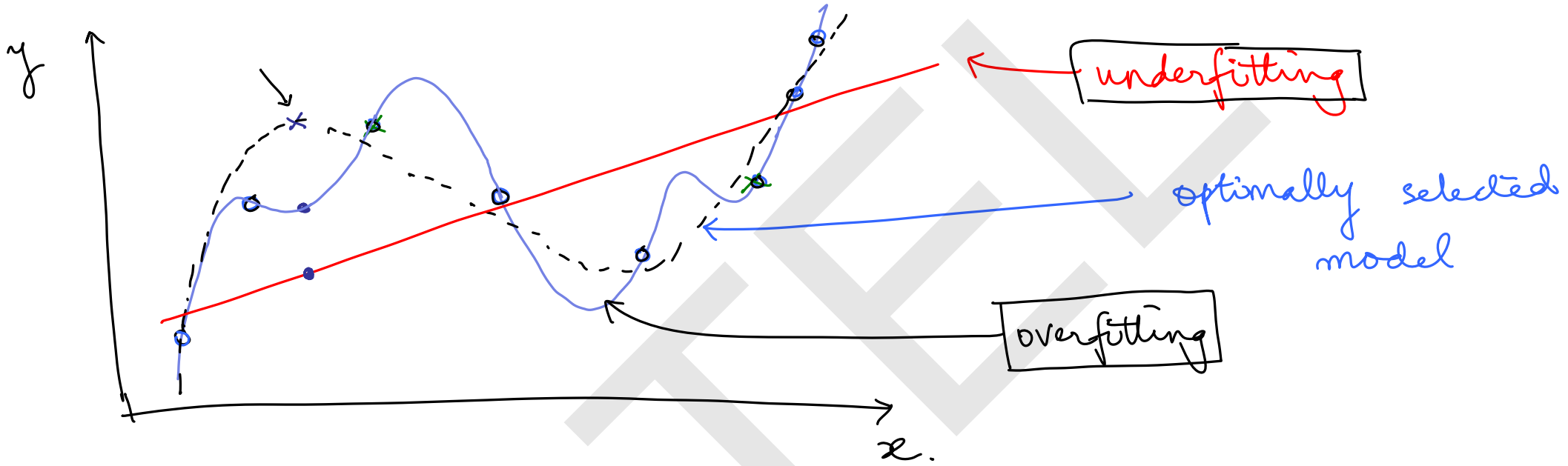
OVERFITTING

(Model just right)

comparable performance of the ML model on the training and the test sets as quantified by the chosen loss function.

(Model too simple)

UNDERFITTING



Dealing with the overfitting problem in linear regression

- Ridge regression
- LASSO
- Elastic net.

Ridge Regression

To avoid the problem of the model overfitting by assigning inordinately high values to the weights for some features, we introduce a penalty for very high weighting coefficients

$$L = \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j \right) \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

L — Loss function

$\sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j \right) \right)^2$ — usual loss function (SSE) for linear regression

$\lambda \sum_{j=1}^{p} \beta_j^2$ — penalty term for some weights to become too high

λ — Regularization parameter.

$$\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \left(L(\beta) \right)$$

Alternative implementation in terms of a constraint

$$\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right)^2 \right\}$$

subject to

$$\sum_{j=1}^{p} \beta_j^2 \leq t$$

$$\beta = [\beta_1 \ \beta_2 \ \cdots \ \beta_p]$$

$$\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \underbrace{\left[(Y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta \right]}_{L(\beta)}$$

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \Rightarrow -2X^T(Y - X\beta) + 2\lambda\beta = 0$$

$$(X^T X + \lambda I)\beta = X^T Y$$

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y$$

if $\lambda = 0$

this expression reduces to $(X^T X)^{-1} X^T Y$.

Advantages of ridge regression:

1. It can prevent over reliance on some parameters to improve the generalizability of the model.
2. λ can be tuned to achieve comparable/best performance on both train and test sets.
3. If X^TX is singular, ie. $(X^TX)^{-1}$ does not exist, regularization can still enable the calculation of $(X^TX + \lambda I)^{-1}$.