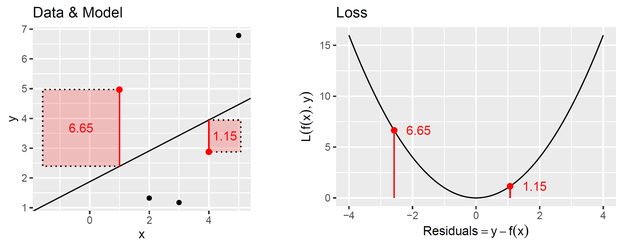
I2ML :: CHEAT SHEET

The **I2ML**: Introduction to Machine Learning course offers an introductory and applied overview of "supervised" Machine Learning. It is organized as a digital lecture.

Regression Losses

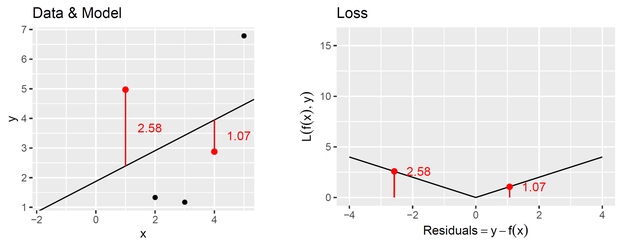
Basic Idea (L2 loss/ squared error):

- $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ or $L(y, f(\mathbf{x})) = 0.5(y - f(\mathbf{x}))^2$.
- Convex and differentiable.
- Tries to reduce large residuals (if residual is twice as large, loss is 4 times as large).
- Risk minimizer $f^*(\mathbf{x}) = \mathbb{E}_{y|x}[y \mid \mathbf{x}]$.
- Optimal constant model $\hat{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} y^{(i)}$.



Basic Idea (L1 loss/ absolute error):

- $L(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$.
- Convex and more robust.
- No derivatives for $= 0$, $y = f(\mathbf{x})$, optimization becomes harder.
- Risk minimizer $f^*(\mathbf{x}) = \text{med}_{y|x}[y \mid \mathbf{x}]$.
- Optimal constant model $\hat{f}(\mathbf{x}) = \text{med}(y^{(i)})$.



Linear Regression Models

Predict $y \in \mathbb{R}$ as **linear** combination of features:

$$\hat{y} = f(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p.$$

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta} = \begin{pmatrix} 1 & x_1^{(1)} & \ldots & x_p^{(1)} \\ 1 & x_1^{(2)} & \ldots & x_p^{(2)} \\ \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \ldots & x_p^{(n)} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix} = \begin{pmatrix} \theta_0 + \theta_1 x_1^{(1)} + \cdots + \theta_p x_p^{(1)} \\ \theta_0 + \theta_1 x_1^{(2)} + \cdots + \theta_p x_p^{(2)} \\ \vdots \\ \theta_0 + \theta_1 x_1^{(n)} + \cdots + \theta_p x_p^{(n)} \end{pmatrix}$$

$\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ is the **design matrix**.

Risk (corresponding to L2 Loss):

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \text{SSE}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2$$

Via normal equations $\frac{\partial \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$ we get ordinary-least-squares (OLS) estimator $\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$.

Risk (corresponding to L1 Loss):

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \sum_{i=1}^{n} \left|y^{(i)} - \theta^T \mathbf{x}^{(i)}\right|$$

L1 loss is harder to optimize, but the model is less sensitive to outliers.

Summary:

Hypothesis Space: Set of all linear functions in $\boldsymbol{\theta}$:

$$\mathcal{H} = \{\theta_0 + \boldsymbol{\theta}^T \mathbf{x} \mid (\theta_0, \boldsymbol{\theta}) \in \mathbb{R}^{p+1}\}$$

Risk: Any regression loss function.

Optimization: Direct analytic solution for L2 loss, numerical optimization for L1 and others.

Polynomial Regression Models

The linear model with **basis functions** ϕ_j:

$$f(\mathbf{x}) = \theta_0 + \sum_{j=1}^{p} \theta_j \phi_j(x_j) = \theta_0 + \theta_1 \phi_1(x_1) + \cdots + \theta_p \phi_p(x_p)$$

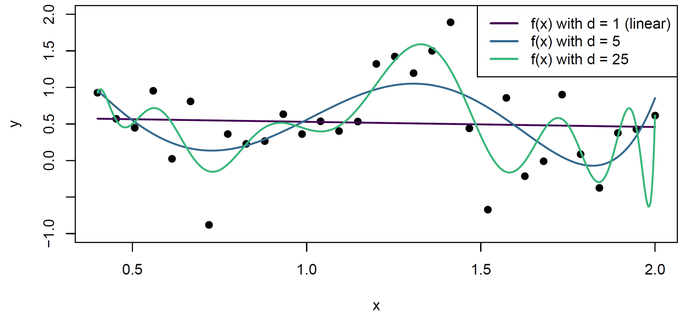
Design matrix:

$$\mathbf{X} = \begin{pmatrix} 1 & \phi_1(x_1^{(1)}) & \ldots & \phi_p(x_p^{(1)}) \\ \vdots & \vdots & & \vdots \\ 1 & \phi_1(x_1^{(n)}) & \ldots & \phi_p(x_p^{(n)}) \end{pmatrix}$$

Simple & flexible choice for basis functions: d-**polynomials** gives us **polynomial regression**:
Map x_j to (weighted) sum of its monomials up to order $d \in \mathbb{N}$

$$\phi^{(d)} : \mathbb{R} \to \mathbb{R}, x_j \mapsto \sum_{k=1}^{d} \beta_k x_j^k$$

Models of different *complexity*, i.e. of different polynomial order d, are fitted to the data:



The higher d is, the more **capacity** the learner has to learn complicated functions of x, but this also increases the danger of **overfitting**.