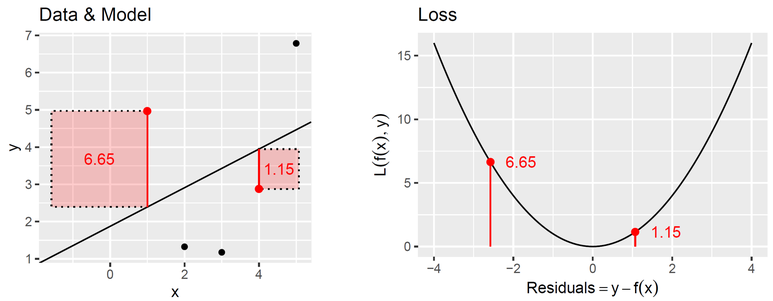
I2ML :: CHEAT SHEET

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

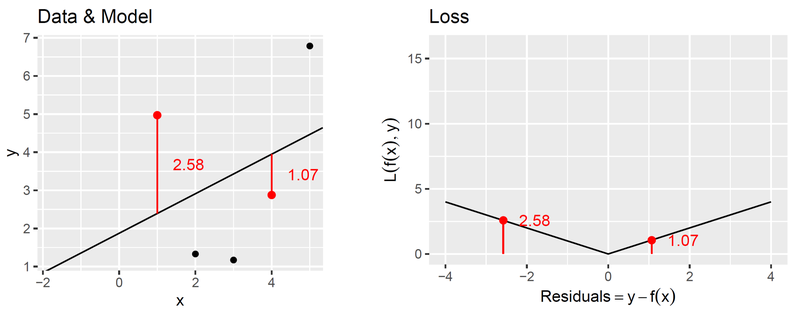
Basic Idea (L2 loss/ squared error):

1. $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ or $L(y, f(\mathbf{x})) = 0.5(y - f(\mathbf{x}))^2$
2. Convex and differentiable.
3. Tries to reduce large residuals (if residual is twice as large, loss is 4 times as large)



Basic Idea (L1 loss/ absolute error):

1. $L(y, f(\mathbf{x})) = |y - f(x)|$
2. Convex and more robust
3. No derivatives for $= 0$, $y = f(x)$, optimization becomes harder
4. $\hat{f}(x) =$ median of $y|x$



Linear Regression Models

Hypothesis Space:

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

This defines the hypothesis space $\mathcal{H}$ as the set of all linear functions in θ

Risk (corresponding to L2 Loss):

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

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: Linear functions $\mathbf{x}^T\boldsymbol{\theta}$ of features $\in \mathcal{X}$.

Risk: Any regression loss function.

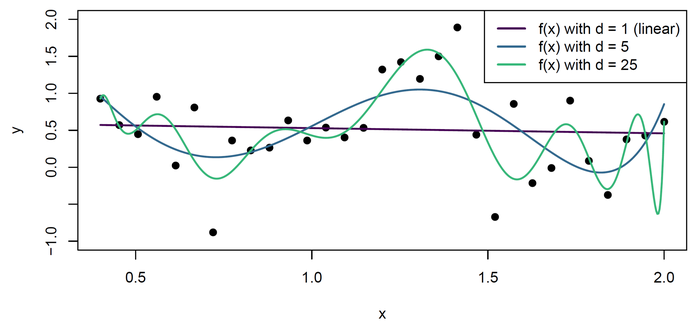
Optimization: Direct analytic solution for L2 loss, numerical optimization for L1 and others.

Polynomial Regression Models

Linear regression models can be made more flexible by using *polynomials* x_j^d – or any other *derived features* like $\sin(x_j)$ or $(x_j \cdot x_k)$ – as additional features. The optimization and risk of the learner remain the same.
Only the hypothesis space of the learner changes and includes linear functions of the derived features as well. e.g.:

$$f(\mathbf{x}^{(i)}) = \theta_0 + \sum_{k=1}^{d} \theta_{1k} \left(\mathbf{x}_1^{(i)}\right)^k + \sum_{k=1}^{d} \theta_{2k} \left(\mathbf{x}_2^{(i)}\right)^k + \ldots$$

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**

K-Nearest Neighbors Regression

K-NN can be used for both regression and classification.

Basic Idea:

1. It generates predictions $\hat{y}$ for a given x by comparing the k observations that are closest to x.
2. "Closeness" requires a distance or similarity measure.
3. The set containing the k closest points $\mathbf{x}^{(i)}$ to x in the training data is called the **k-neighborhood** $N_k(x)$ of x.
4. k-NN makes no assumptions about the underlying data distribution.
5. The smaller k, the less stable, less smooth and more "wiggly" the decision boundary becomes.

Distance Measures:

1. **The Euclidean distance**:

$$d_{Euclidean}(x, \tilde{x}) = \sqrt{\sum_{j=1}^{p} (x_j - \tilde{x}_j)^2}$$

2. **Manhattan distance:**

$$d_{manhattan}(x, \tilde{x}) = \sum_{j=1}^{p} |x_j - \tilde{x}_j|$$

3. **Mahalanobis distance:** (takes covariances in $\mathcal{X}$ into account)
4. **Gower distance:**

$$d_{gower}(x, \tilde{x}) = \frac{\sum_{j=1}^{p} \delta_{x_j, \tilde{x}_j} \cdot d_{gower}(x_j, \tilde{x}_j)}{\sum_{j=1}^{p} \delta_{x_j, \tilde{x}_j}}.$$

K-NN Summary:

Hypothesis Space: Step functions over tesselations of $\mathcal{X}$.
Hyperparameters: distance measure $d(\cdot, \cdot)$ on $\mathcal{X}$; size of neighborhod k.

Risk: Use any loss function for regression or classification.

Optimization: Not applicable/necessary.