VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Fundamentals of Machine Learning

Extending of Linear Model

Jan Platos

December 1, 2021

Extending of Linear Model

Extending of Linear Model - Linear Regression

- The goal is to find a linear model that is able to predict the true value y from the input vector x.
- The expected value $\bar{y}$ is expressed using linear coefficient.

$$\bar{y} = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_k x_k$$

- x_0 is always 1 and represents the bias.
- $x_1, x_2, \ldots, x_k$ are the attribute values,
- $w_0, w_1, \ldots, w_k$ are the weights.

Extending of Linear Model - Linear Regression

- The weights are calculated from the training data.
- The prediction for the i-th instance is calculated as:

$$w_0 x_0^{(i)} + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots + w_k x_k^{(i)} = \sum_{j=0}^{k} w_j x_j^{(i)}$$

- The important is the difference between the true value y and the predicted one $\overline{y}$.

Extending of Linear Model - Linear Regression

- The error function is defined as:

$$\sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2$$

- The goal is to find the weights to minimize the error.

$$\min_{w} \left\{ \sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2 \right\}$$

Extending of Linear Model - Linear Regression

$$\min_{w} \left\{ \sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2 \right\}$$

- The solution may be find using:
  - Ordinary Least Squares algorithm.
  - Gradient Descent (a learning rate need to be set and iterative approach is processed).

Extending of Linear Model - Linear Regression - Regularization

- The weights computed by the optimization algorithm may exceeds some limits and/or may contains many small numbers.
- Such weights means over-fitting - too big specialization to the training data.

Extending of Linear Model - Linear Regression - Regularization

- Lasso regression
  - Minimizes the sum of weights.
  - Eliminates small weight in favor to more important ones.

$$\min_{w} \left\{ \sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2 + \alpha \sum_{j=0}^{k} |w_i| \right\}$$

Extending of Linear Model - Linear Regression - Regularization

- Ridge regression
  - Minimizes the sum of squares of the weights (a norm o the weight vector).
  - Suppress large values in favor of smaller and more universal ones.

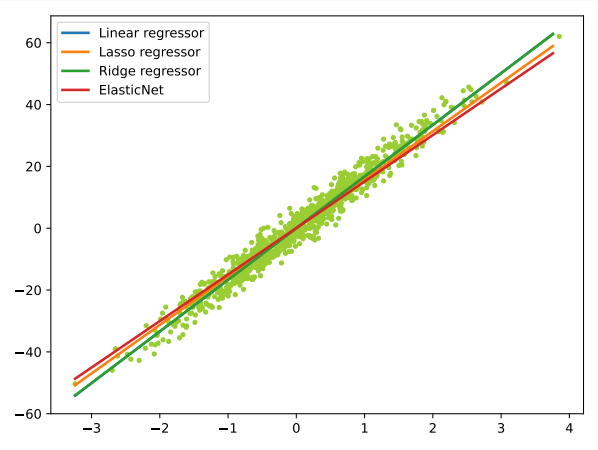
$$\min_{w} \left\{ \sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2 + \beta \sum_{j=0}^{k} |w_i|^2 \right\}$$

Extending of Linear Model - Linear Regression - Regularization

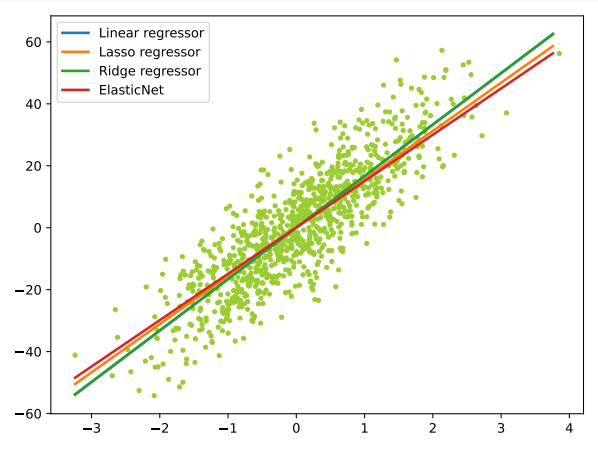
- Elastic Net
  - Combines both regularization to gain benefit from them.

$$\min_{w} \left\{ \sum_{i=0}^{n} \left(y^{(i)} - \sum_{j=0}^{k} w_j x_j^{(i)} \right)^2 + \alpha \sum_{j=0}^{k} |w_i| + \beta \sum_{j=0}^{k} |w_i|^2 \right\}$$

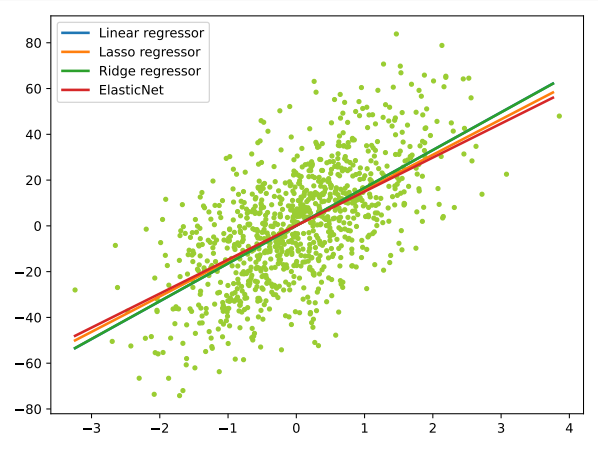
Extending of Linear Model - Linear Regression



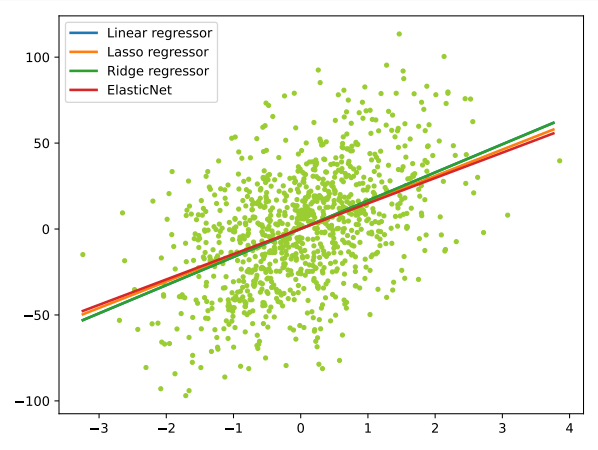
Extending of Linear Model - Linear Regression



Extending of Linear Model - Linear Regression



Extending of Linear Model - Linear Regression



Extending of Linear Model - Multi-layer Neural Network

- The perceptron, with only one computational neuron produces only a linear model.
- Multi-layer perceptron adds a hidden layer beside the input and output layer.
- The hidden layer itself may consist of different type of topology (e.g. several layers).

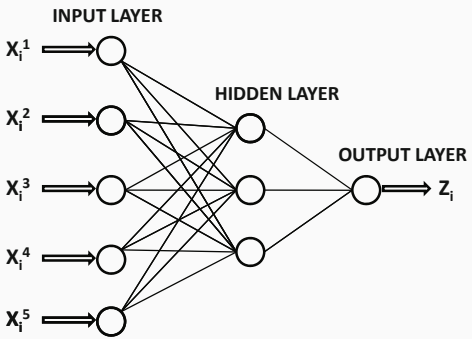


Figure 1: Multi-layer neural network

Extending of Linear Model - Multi-layer Neural Network

- The output of nodes in one layer feed the inputs of the nodes in the next layer - this behavior is called *feed-forward network*.
- The nodes in one layer are fully connected to the neurons in the previous layer.

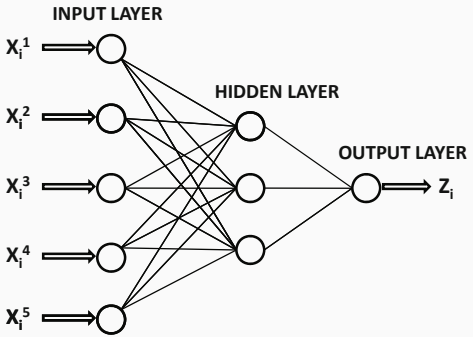


Figure 2: Multi-layer neural network

Extending of Linear Model - Multi-layer Neural Network

- The topology of the multi-layer feed-forward network is determined automatically.
- The perceptron may be considered as a single-layer feed-forward neural network.
- The number of layers and the number of nodes in each layer have to be determined manually.
- Standard multi-layer network uses only one hidden layer, i.e. this is considered as a two-layer feed forward neural network.
- The activation function is not limited to linear signed weighted sum, other functions such as logistic, sigmoid or hyperbolic tangents are allowed.

Extending of Linear Model - Multi-layer Neural Network

Sigmoid/Logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$

TanH $\tanh(x) = \frac{\left(e^{x}-e^{-x}\right)}{\left(e^{x}+e^{-x}\right)}$

ReLU (Rectified linear unit) $f(x) = \begin{cases} 0 & for x \leq 0 \\ x & for x \geq 0 \end{cases}$

Sinc $f(x) = \begin{cases} 1 & for x = 0 \\ \frac{sin(x)}{x} & for x \neq 0 \end{cases}$

Gaussian $f(x) = e^{x^2}$

Softmax $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$

Extending of Linear Model - Multi-layer Neural Network

- The learning phase is more complicated than the one in perceptron.
- The biggest problem is the get the error in the hidden layer, because the direct class label is not defined on this level.
- Some kind of *feedback* is required from the nodes in the forward layer to the nodes in earlier layers about the *expected* outputs and corresponding errors.
- This principle is realized in the *back-propagation* algorithm.

Extending of Linear Model - Multi-layer Neural Network

Back-propagation algorithm

- *Forward phase:*
  - The input is fed into input neurons.
  - The computed values are propagated using the current weights to the next layers.
  - The final predicted output is compared with the class label and the error is determined.

Extending of Linear Model - Multi-layer Neural Network

Back-propagation algorithm

- *Backward phase:*
  - The main goal is to learn weights in the backward direction by providing the error estimation from later layers to the earlier layers.
  - The estimation in the hidden layer is computed as a function of the error estimate and weight is the layers ahead.
  - The error is estimated again using the gradient method.
  - The process is complicated by the using of non-linear functions n the inner nodes.

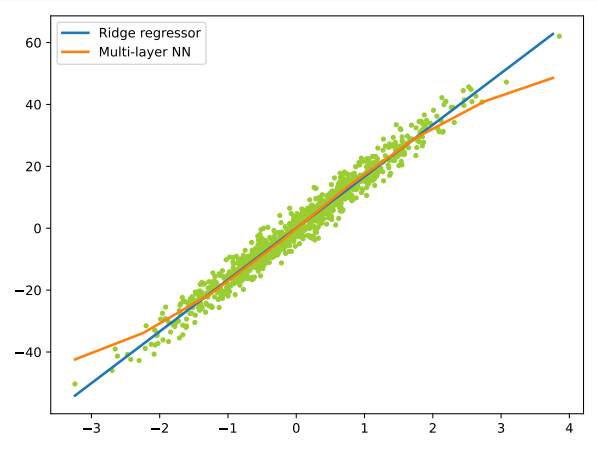
Extending of Linear Model - Multi-layer Neural Network

- It has ability not only to capture decision boundaries of arbitrary shapes, but also non-contiguous class distribution with different decision boundaries in different regions.
- With increasing number of nodes and layers, virtually any function may be approximated.
- **The neural networks are universal function approximate**.

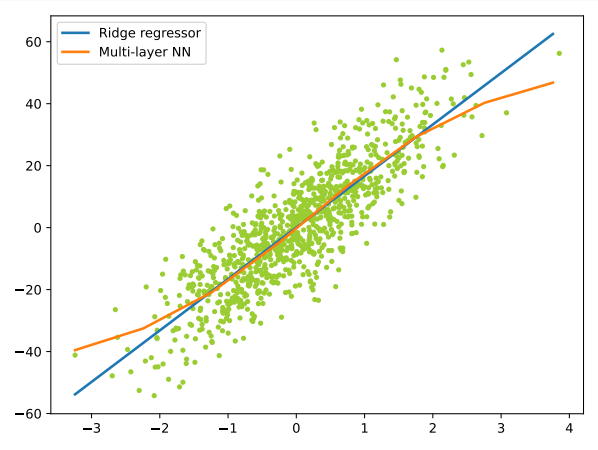
Extending of Linear Model - Multi-layer Neural Network

- This generality brings several challenges that have to be dealt with:
  - The design of the topology presents many trade=off challenges for the analyst.
  - Higher number of nodes and layers provides greater generality but also the risk of over-fitting.
  - There is very little guidance provided from the data.
  - The neural network has poor interpretability associated with the classification process.
  - The learning process is very slow and sensitive to the noise.
  - Larger networks has very slow learning process.

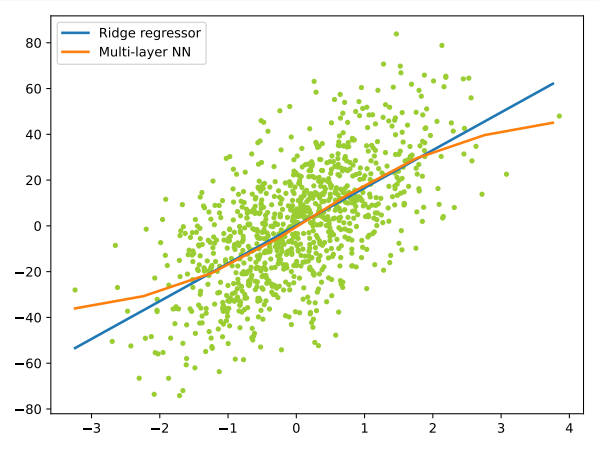
Extending of Linear Model - Multi-layer Neural Network



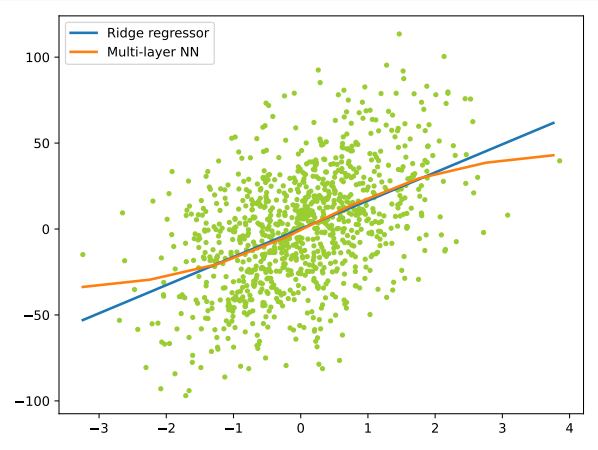
Extending of Linear Model - Multi-layer Neural Network



Extending of Linear Model - Multi-layer Neural Network



Extending of Linear Model - Multi-layer Neural Network



Extending of Linear Model- Regression Trees

- In reality, local linear regression may be quite effective even when the relationships is nonlinear.
- This is used in Regression Trees.
- Each test instance is classified with its locally optimized linear regression by determining its appropriate partition.
- The partition is determined using split criteria in the internal nodes, i.e. the same as the Decision trees.

Extending of Linear Model- Regression Trees

- The general strategy of tree construction is the same as for Decision Trees.
- The splits are univariate (single variable/axis parallel).
- The changes are done in splitting criterion determination and in the pruning.
- The number of points used for training need to be high to avoid over-fitting

Extending of Linear Model- Regression Trees

Splitting criterion

- Due to numeric nature of the class variable, error-based measure have to be used instead of entropy or Gini index.
- The regression modeling is applied on each child resulting from potential split.
- The aggregated squared error of prediction of all training points is computed.

Extending of Linear Model- Regression Trees

Splitting criterion

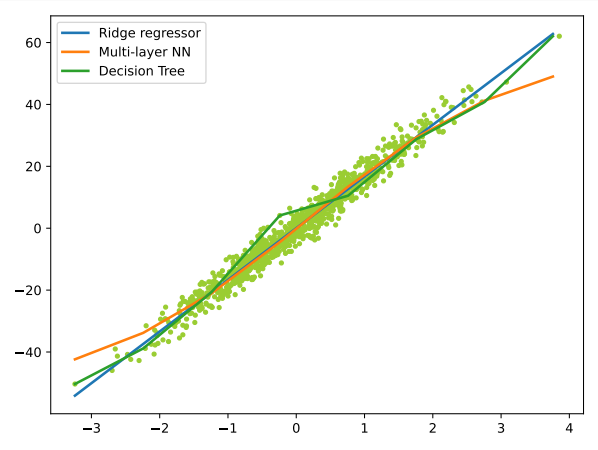
- The split point with the minimum aggregated error is selected.
- The complete regression modeling is computationally very expensive.
- An average variance of the numeric class variable may be used instead.
- The linear regression models are constructed at the leaf nodes after the tree is created.
- This results in larger trees but it its computational expensiveness is much lower.

Extending of Linear Model- Regression Trees

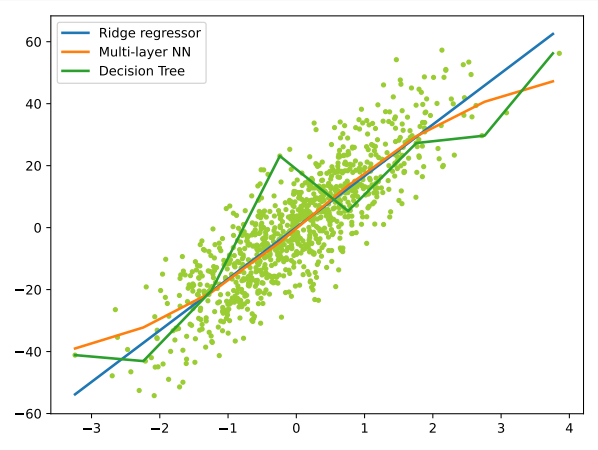
Pruning criterion

- A portion of the training data is not used during construction phase.
- This set is used for evaluation of the squared error of the prediction.
- Leaf nodes are iteratively removed if the accuracy not decreases.

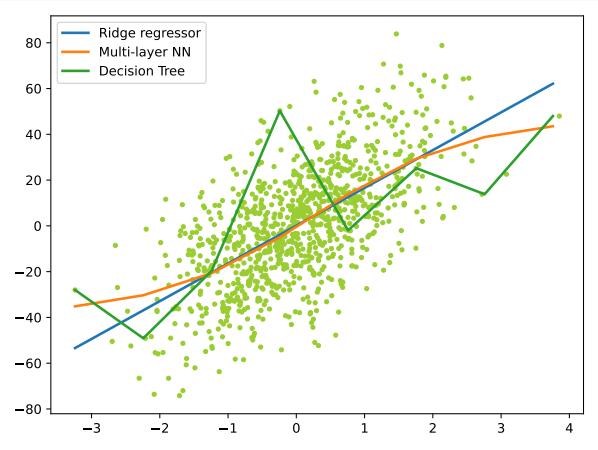
Extending of Linear Model - Regression Trees



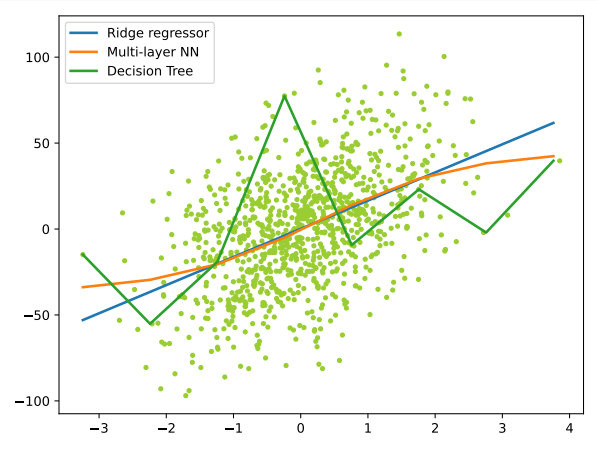
Extending of Linear Model - Regression Trees



Extending of Linear Model - Regression Trees



Extending of Linear Model - Regression Trees



Extending of Linear Model- Assessing Model Effectiveness

- **Mean Absolute Error (MAE)** - is the average of the absolute difference between the predicted and actual value. It is highly affected by outliers.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - g(\overline{X}_i)|$$

Extending of Linear Model- Assessing Model Effectiveness

- **Mean Squared Error (MSE)** - is the average of the squared difference between the predicted and actual value. It is differentiable and may be used for optimization.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - g(\overline{X_i})\right)^2$$

- **Root Mean Squared Error (RMSE)** - is the square root of the average of the squared difference of the predicted and actual value. The root mean is able penalize large errors.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - g\left(\overline{X_i}\right)\right)^2}$$

Extending of Linear Model- Assessing Model Effectiveness

- The effectiveness of the linear regression models can be evaluated with a measure known as **R^2-*statistics*** or *coefficient of determination*.
- The standard Sum of Squared Error is defined for a model $g(\overline{X})$ as:

$$SSE = \sum_{i=1}^{n} \left(y_i - g(\overline{X_i})\right)^2$$

- The Squared Error of the response variable about its mean is defined as:

$$SST = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \frac{y_j}{n} \right)^2 = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

Extending of Linear Model- Assessing Model Effectiveness

- The R^2-*statistics* is then defined as:

$$R^2 = 1 - \frac{SSE}{SST}$$

- The value is always between 0 and 1 and higher are more desirable.
- For high dimension data, **adjusted** version is more accurate:

$$R^2 = 1 - \frac{(n-d)SSE}{(n-1)SST}$$

- The R^2-*statistics* is not applicable on the nonlinear models.
- The nonlinear regression may be evaluated using pure SSE.

Extending of Linear Model- Assessing Model Effectiveness

- **Mean Average Percentage Error (MAPE)** - is the average percentage error between the predicted and actual value.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - g(\overline{X_i})}{y_i} \right|$$

- **Symmetric Mean Average Percentage Error (SMAPE)** - is the symmetric average percentage error between the predicted and actual value.

$$SMAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{\left| y_i - g(\overline{X_i}) \right|}{\frac{|y_i| + \left| g(\overline{X_i}) \right|}{2}}$$

Questions