DS

boosting your data exploration

Machine Learning Hands-on

KhanhPhamDinh, VinAI Applied, Ha Noi

my book: Machine Learning Algorithms to Practices
https://phamdinhkhanh.github.io/deepai-book/intro.html

Linear, Ridge and Lasso Regression

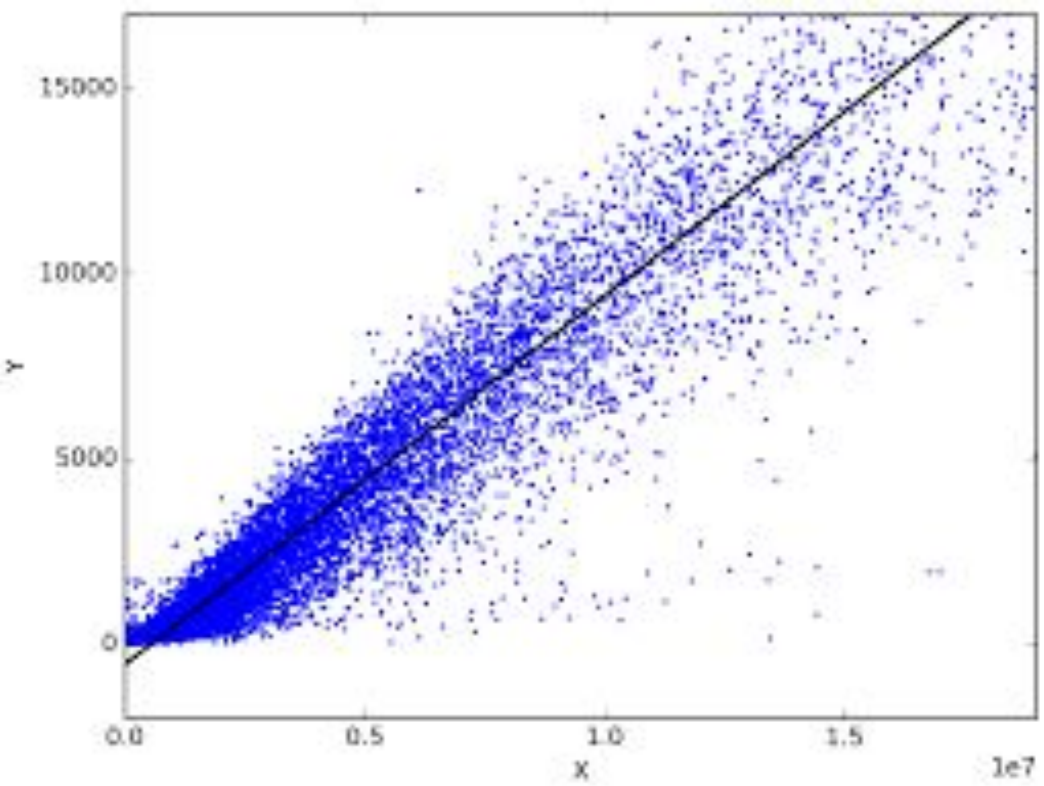
Content

- Target of Linear Regression
- One variable Linear Regression
- Multi-variable Linear Regression
- Evaluation of Linear Regression
- Overfitting Problem on Linear Regression
- Regularization
- Ridge Regression
- Lasso Regression
- Elastic Net Regression

Target of Linear Regression

- Linear Regression is a algorithms of Supervised Learning
- Find a linear function that fit the most with data
- Make a prediction of input Xi based on the regressor function

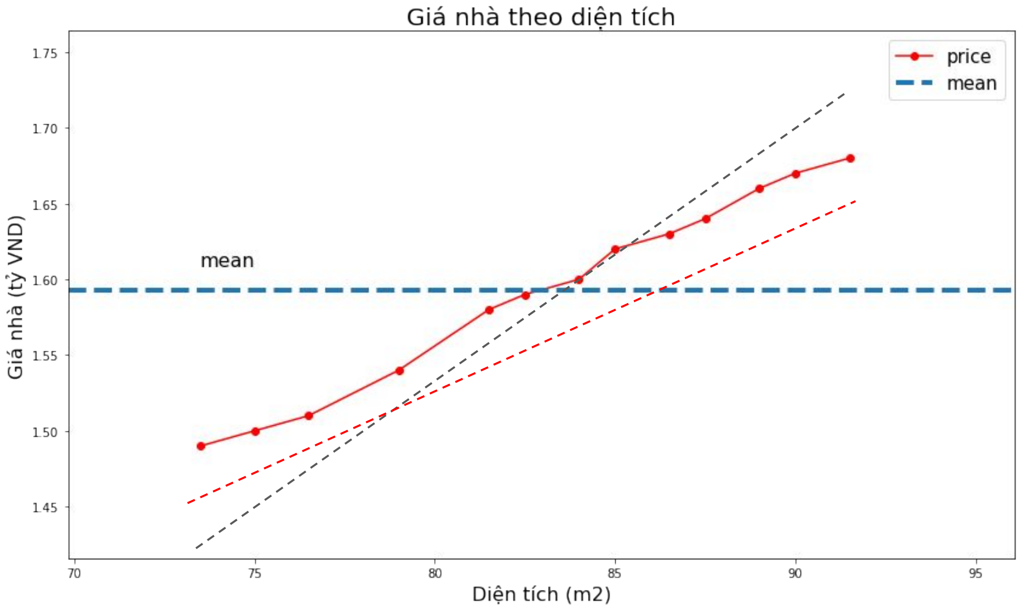
How Linear Regression and Classification are different?



One variable Linear Regression

- Input include only one variable
- Data = *{(x1, y1), (x2, y2),...,(xN, yN)}*
- Estimate target variable y:

$$y_i = \underbrace{w_0 + w_1 x_i}_{} + e_i$$
$$= f_{\mathbf{w}}(x_i) \quad + e_i$$



- Mean square Error:

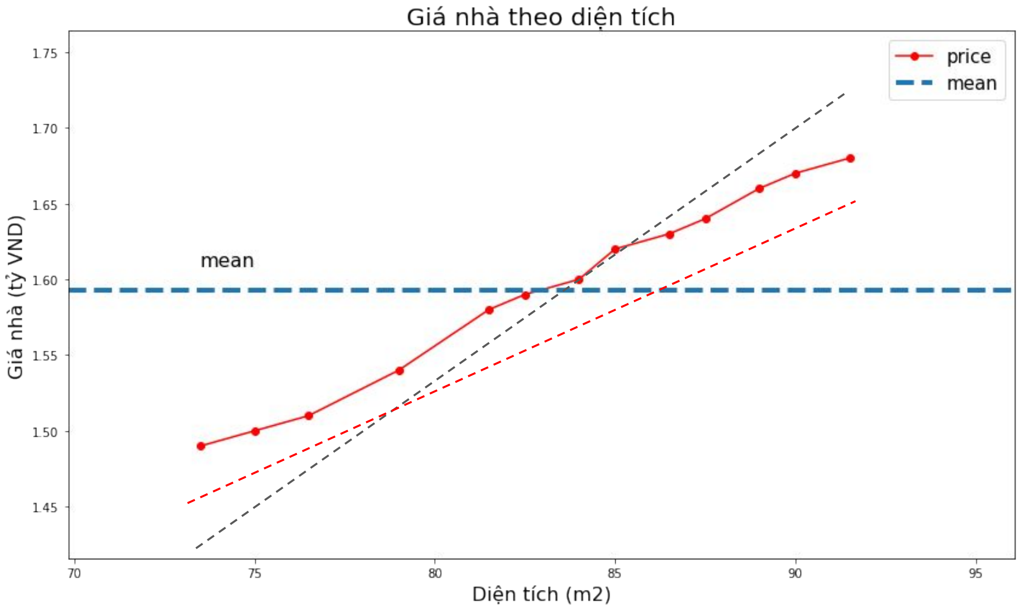
$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n} e_i^2 = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i)^2$$

- Target: Find *(w0, w1)* that best fit relationship between **x** and **y**.

One variable Linear Regression

- First order derivative according to *w0* and *w1*.

$$
\begin{aligned}
\frac{\delta \mathcal{L}(\mathbf{w})}{\delta w_0} &= \frac{-1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i) \\
&= \frac{-1}{n}\sum_{i=1}^{n} y_i + w_0 + w_1 \frac{1}{n}\sum_{i=1}^{n} x_i \\
&= -\bar{\mathbf{y}} + w_0 + w_1 \bar{\mathbf{x}} \\
&= 0
\end{aligned}
$$

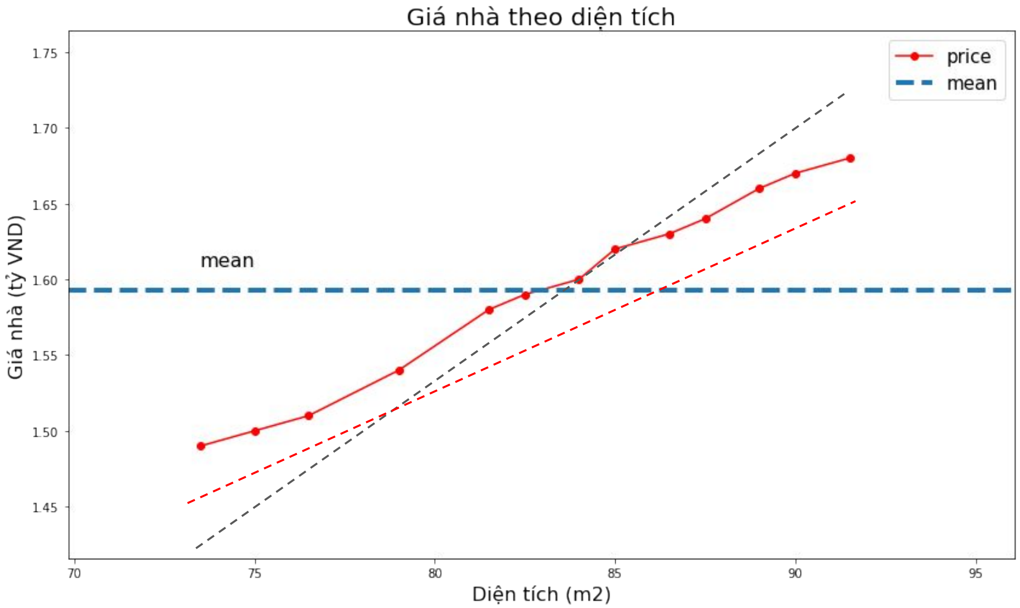


[Loss Function Linear Regression - Kan Pham]()

One variable Linear Regression

- First order derivative according to *w0* and *w1*.

$$\begin{aligned}\frac{\delta\mathcal{L}(\mathbf{w})}{\delta w_1} &= \frac{-1}{n}\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 * x_i) \\ &= \frac{-1}{n}\sum_{i=1}^{n} x_i y_i + w_0\frac{1}{n}\sum_{i=1}^{n} x_i + w_1\frac{1}{n}\sum_{i=1}^{n} x_i^2 \\ &= -\bar{\mathbf{x}\mathbf{y}} + w_0\bar{\mathbf{x}} + w_1\bar{\mathbf{x}^2} \\ &= 0\end{aligned}$$



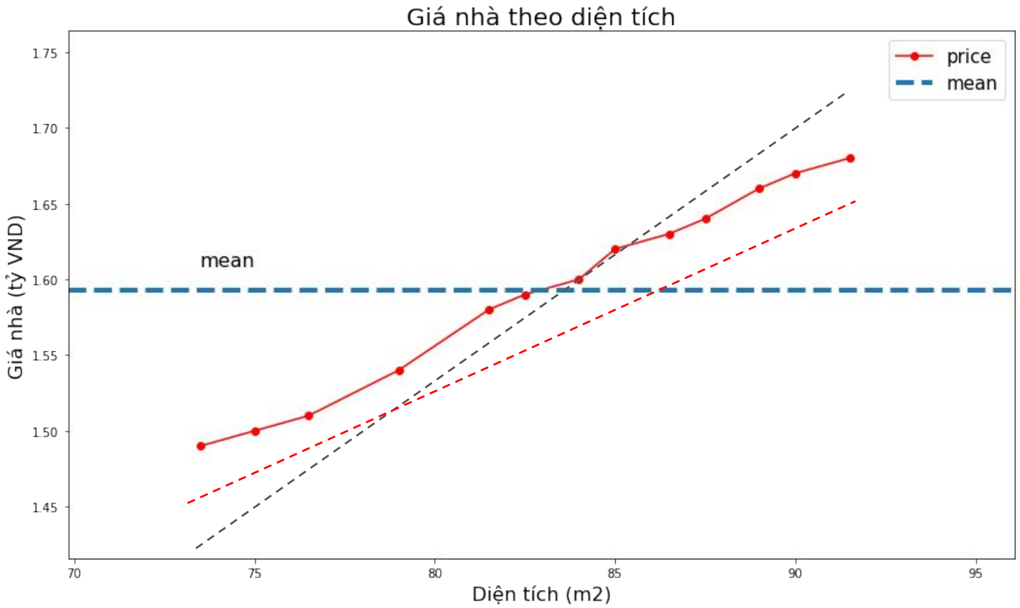
[Loss Function Linear Regression - Kan Pham]()

One variable Linear Regression

- Optimal values *(w0, w1)*:

$$w_0 = \bar{\mathbf{y}} - w_1\bar{\mathbf{x}}$$

$$w_1 = \frac{\bar{\mathbf{x}}\bar{\mathbf{y}} - \overline{\mathbf{x}\mathbf{y}}}{\bar{\mathbf{x}}^2 - \overline{\mathbf{x}^2}}$$



[Loss Function Linear Regression - Kan Pham]

One variable Linear Regression

Bài tập: chúng ta có 15 căn hộ với diện tích (đơn vị m2):

$$\mathbf{x} = [73.5, 75., 76.5, 79., 81.5, 82.5, 84., 85., 86.5, 87.5, 89., 90., 91.5]$$

Mức giá của căn hộ lần lượng là (đơn vị tỷ VND đồng):

$$\mathbf{y} = [1.49, 1.50, 1.51, 1.54, 1.58, 1.59, 1.60, 1.62, 1.63, 1.64, 1.66, 1.67, 1.68]$$

Xây dựng phương trình hồi qui tuyến tính đơn biến giữa diện tích và giá nhà.

[Practice Linear Regression - Kan Pham](#)

Multi-variable Linear Regression

- There are more than one variable as input.
- Data = $\{(x_{i1}, x_{i2}, ..., x_{ip}; y_i)\}_{i=1}^{N}$
- Estimation of target variable:

$$\hat{y}_i = f_{\mathbf{w}}(x_{i1}, x_{i2}, \ldots, x_{ip}) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_p x_{ip} = \mathbf{w}^\intercal \mathbf{x}_i$$

Multi-variable Linear Regression

- Estimation of target variable:

$$\hat{y}_i = f_{\mathbf{w}}(x_{i1}, x_{i2}, \ldots, x_{ip}) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_p x_{ip} = \mathbf{w}^\intercal \mathbf{x}_i$$

- Set vector and matrices:

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & \ldots & x_{1p} \\ x_{21} & \ldots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{np} \end{bmatrix}, \quad \bar{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix}$$

Multi-variable Linear Regression

- Estimation of target variable:

$$\hat{y}_i = f_{\mathbf{w}}(x_{i1}, x_{i2}, \ldots, x_{ip}) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_p x_{ip} = \mathbf{w}^\intercal \mathbf{x}_i$$

- Set vectors and matrices:

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & \ldots & x_{1p} \\ x_{21} & \ldots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{np} \end{bmatrix}, \quad \bar{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix}$$

$$\hat{\mathbf{y}} = f(\mathbf{X}) = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} = \bar{\mathbf{X}}\mathbf{w}$$

Multi-variable Linear Regression

- Error vector:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \bar{\mathbf{X}}\mathbf{w}$$

- Loss function:

$$\mathcal{L}(\mathbf{w}|\mathbf{x},\mathbf{y}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{2n}\mathbf{e}^\intercal\mathbf{e} = \frac{1}{2n}(\mathbf{y} - \bar{\mathbf{X}}\mathbf{w})^\intercal(\mathbf{y} - \bar{\mathbf{X}}\mathbf{w}) = \frac{1}{2n}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2$$

- Derivative:

$$\frac{\partial\mathcal{L}(\mathbf{w})}{\partial\mathbf{w}} = \bar{\mathbf{X}}^\intercal(\bar{\mathbf{X}}\mathbf{w} - \mathbf{y})$$

- Optimal values:

$$\mathbf{w} = (\bar{\mathbf{X}}^\intercal\bar{\mathbf{X}})^{-1}\bar{\mathbf{X}}^\intercal\mathbf{y} = (\mathbf{A}^{-1}\mathbf{b})$$

Evaluation of Linear Regression

- What is TSS, RSS, ESS:

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$ESS = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

- R-Squared: How many percentage of output variable can be explained by input variables

$$R^2 = 1 - \frac{RSS}{TSS}$$

Evaluation of Linear Regression

- MSE: Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- RMSE: Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

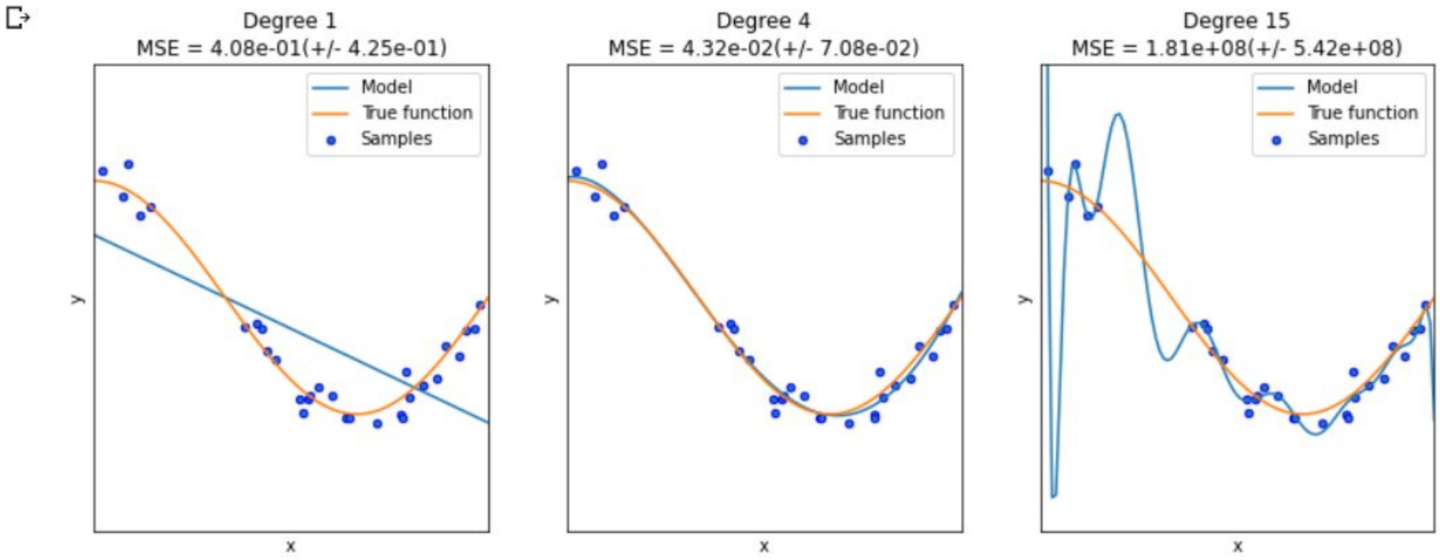
- MAE: Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- MAPE: Mean Absolute Percentage Error

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

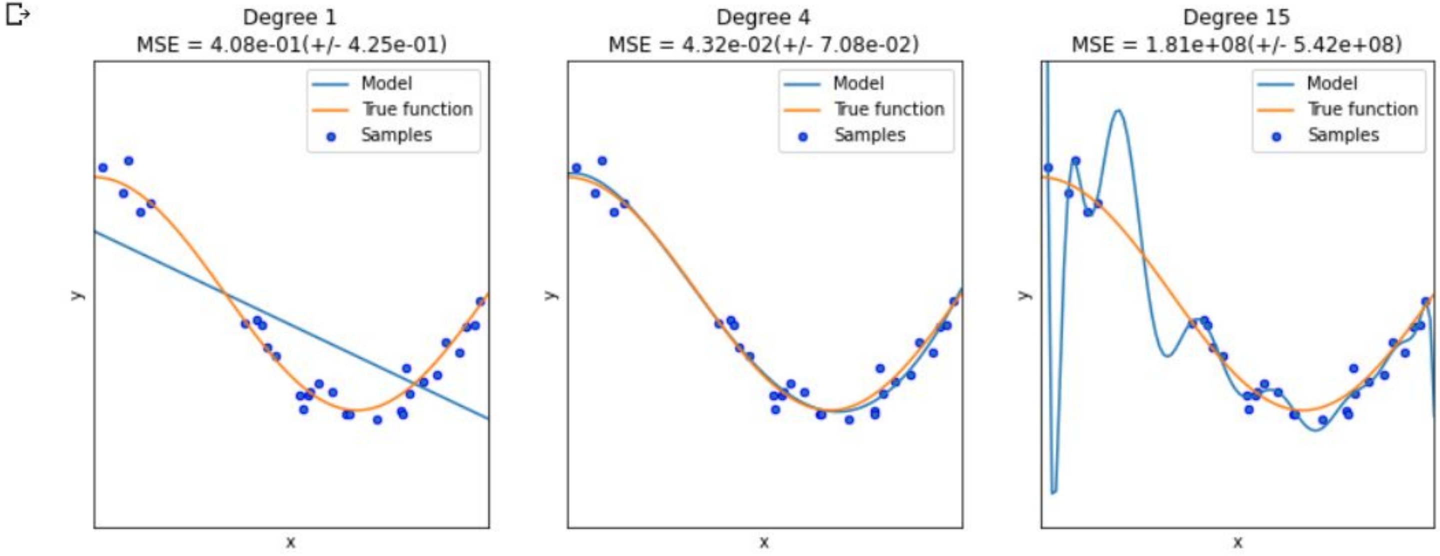
Overfitting Problem on Linear Regression



Dataset:

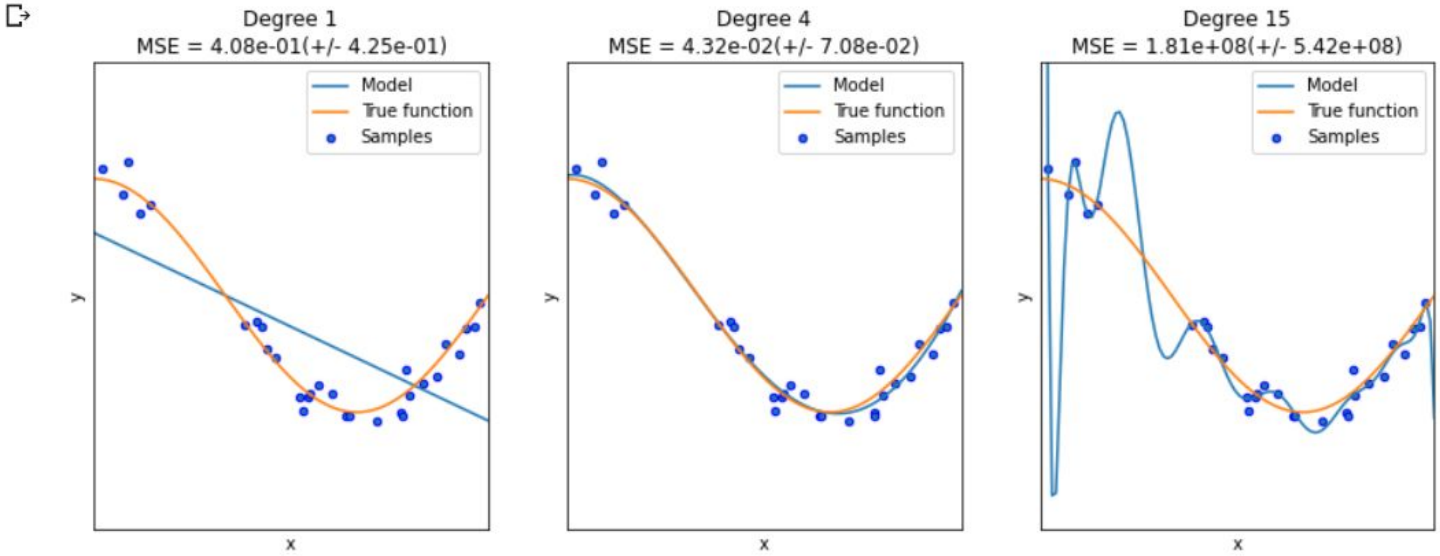
- **x:** total of labor force
- **y:** average cost per unit

Overfitting Problem on Linear Regression



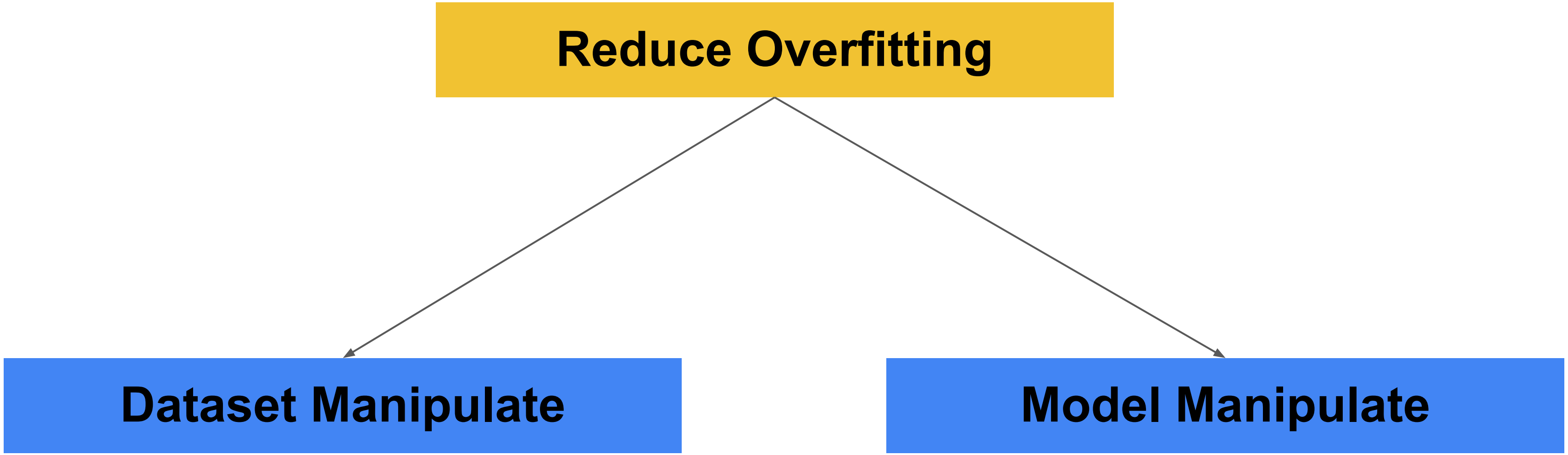
What is the model curve that is most general?

Overfitting Problem on Linear Regression



- Overfitting are the cases that your fitting curves are only exact on train dataset without test dataset.
- You can not apply your models into forecasting.
- Usually happen when:
  - Too many input variables.
  - Model is trained on small dataset.
  - Your models are completed.

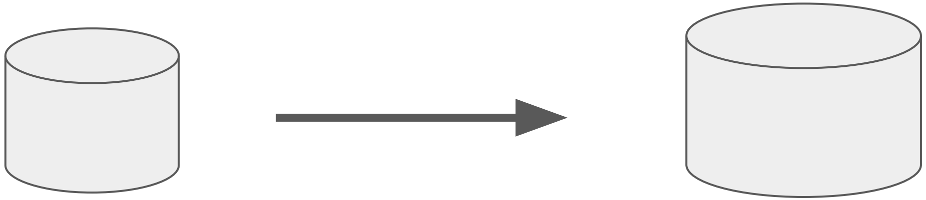
Reduce Overfitting



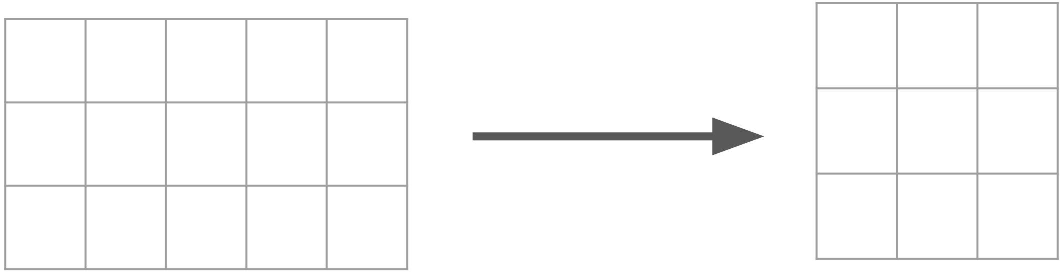
Reduce Overfitting

Dataset Manipulate

Collect more observations



Reduce total of inputs



- Correlation: Pearson, Kendall, Spearman
- Select k best
- SME: Subject Major Experts

Reduce Overfitting

Model Manipulate

Reduce model size

- **Number of layers**
- **Number of parameters**
- **Depth**
- **Total Nodes**
- **....**

Reduce weight values

- **Reduce unimportant weight into zero (Drop out)**
- **Reduce weight values (Regularization)**

Regularization

Regularized Loss = Loss + Regularization Term

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\intercal \mathbf{x}_i) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2$$

Difference between y and yhat

$$\alpha||\mathbf{w}||_2^2$$

Is a function of weights

Regularization - Loss function

Single variable model:

Regularized Loss = Loss + Regularization Term

- Dataset with one input variable:

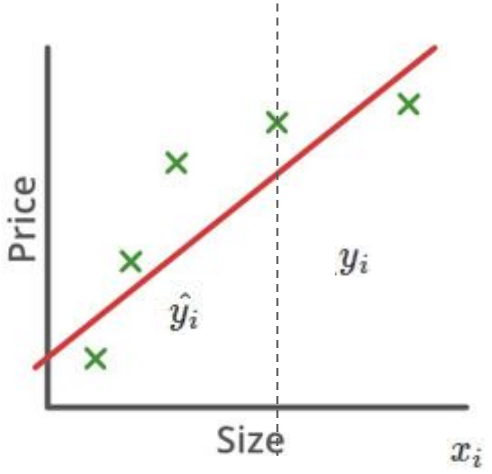
$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

- Prediction:

$$\hat{y_i} = f(x_i) = w_0 + w_1 * x_i$$

- Loss function:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i)^2$$



Regularization - Loss function

Multiple variable model:

Regularized Loss = Loss + Regularization Term

- Dataset with multiple input variables:

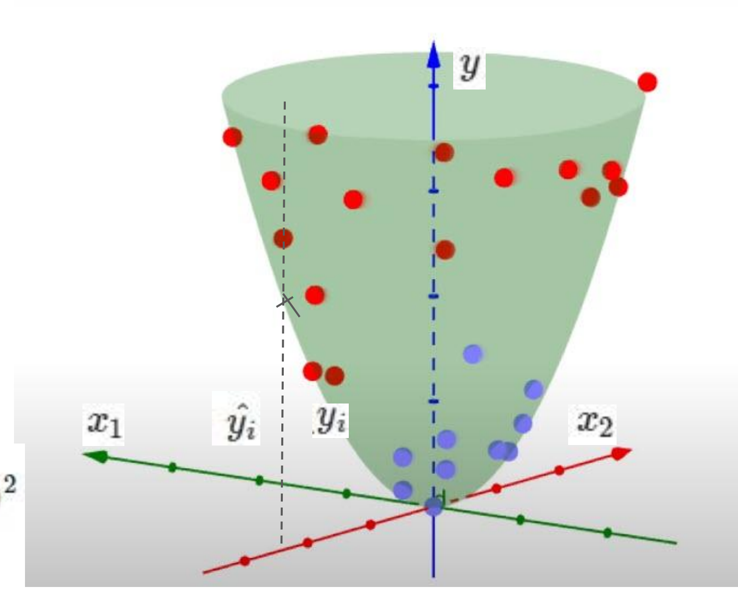
$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$
$$= \{([x_{11}, x_{12}], y_1), ([x_{21}, x_{22}], y_2), \ldots, ([x_{n1}, x_{n2}], y_n)\}$$

- Prediction:

$$\hat{y}_i = f_{\mathbf{w}}(\mathbf{x}_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} = \mathbf{w}^\top \mathbf{x}_i$$

- Loss function:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_{i1} + w_2 x_{i2}))^2$$
$$= \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \bar{\mathbf{x}}_i)^2$$



Regularization - Loss function

Multiple variable model:

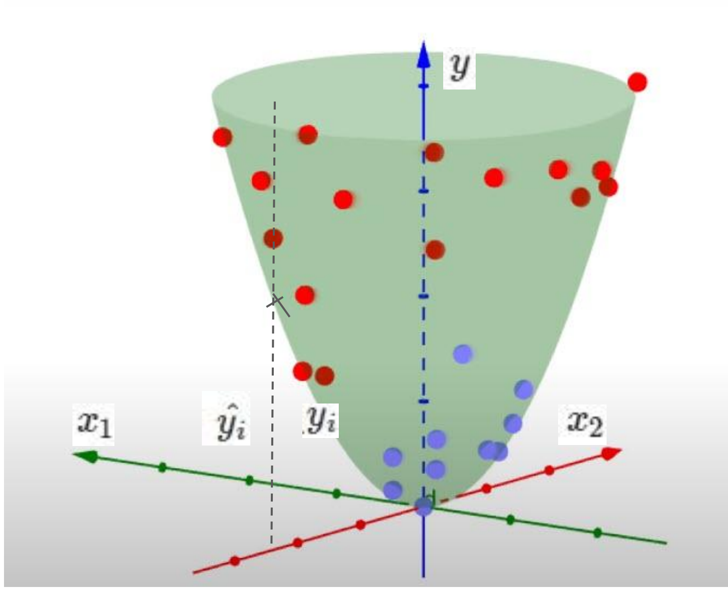
Regularized Loss = Loss + Regularization Term

- Prediction vector:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{x}}_1^\intercal \mathbf{w} \\ \bar{\mathbf{x}}_2^\intercal \mathbf{w} \\ \vdots \\ \bar{\mathbf{x}}_n^\intercal \mathbf{w} \end{bmatrix} = \bar{\mathbf{X}}\mathbf{w}$$

- Loss function:

$$\begin{aligned} \mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 &= \frac{1}{2n}(\mathbf{y} - \hat{\mathbf{y}})^\intercal(\mathbf{y} - \hat{\mathbf{y}}) \\ &= \frac{1}{2n}(\mathbf{y} - \bar{\mathbf{X}}\mathbf{w})^\intercal(\mathbf{y} - \bar{\mathbf{X}}\mathbf{w}) \\ &= \frac{1}{2n}||\mathbf{y} - \bar{\mathbf{X}}\mathbf{w}||_2^2 \end{aligned}$$



Regularization - Regularization Term

Multiple variable model:

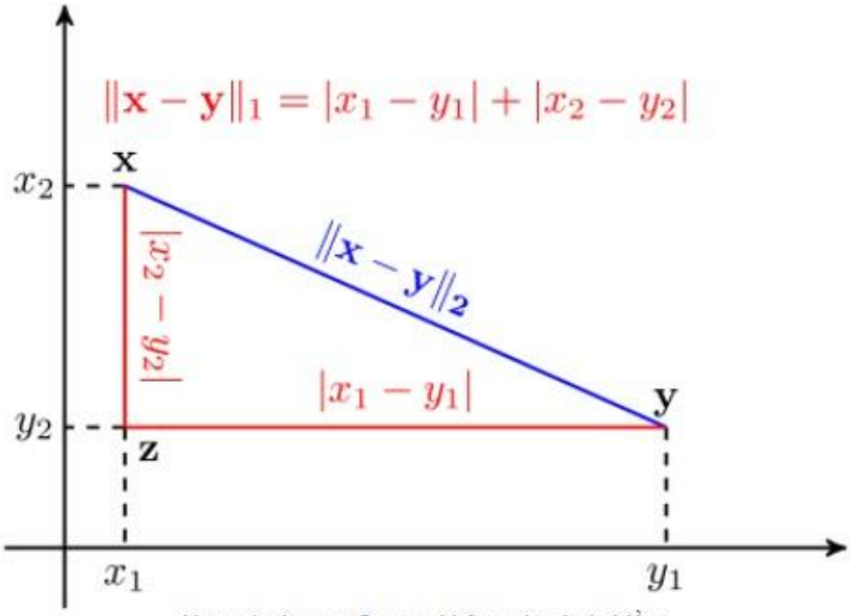
Regularized Loss = Loss + Regularization Term

- Regularization Term is usually a norm
- What is norm?

$$L_1 = \|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

$$L_2 = |\mathbf{x}|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$$L_p = \|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$



Norm 1 và norm 2 trong không gian hai chiều.

Regularization - Regularization Term

Multiple variable model:

Regularized Loss = Loss + Regularization Term

- Regularization Term is usually a norm
  - **w** is a vector (or sometime matrix) of all weights of model:
  - L2 norm:

$$\begin{aligned}\alpha R(\mathbf{w}) &= \alpha||\mathbf{w}||_2^2 \\ &= \alpha(w_0^2 + w_1^2 + \cdots + w_p^2)\end{aligned}$$

  - L1 norm:

$$\begin{aligned}\alpha R(\mathbf{w}) &= \alpha||\mathbf{w}||_1 \\ &= \alpha(|w_0| + |w_1| + \cdots + |w_p|)\end{aligned}$$

Regularization - Regularization Term

Multiple variable model:

Regularized Loss = Loss + Regularization Term

- The additional Regularization Term is added into normal Loss

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \underbrace{\alpha R(\mathbf{w})}_{\text{regularization term}}$$

- **Loss** term is to reduce difference between prediction and ground truth.
- **Regularization** term as a regulator controlling the values of **w,** avoiding too high.
  - Why it can control values of weight too high? https://www.youtube.com/watch?v=Xm2C_gTAI8c

Ridged Regression

Regularized Loss = Loss + Regularization Term (L2)

- Regularization term is L2-Norm:

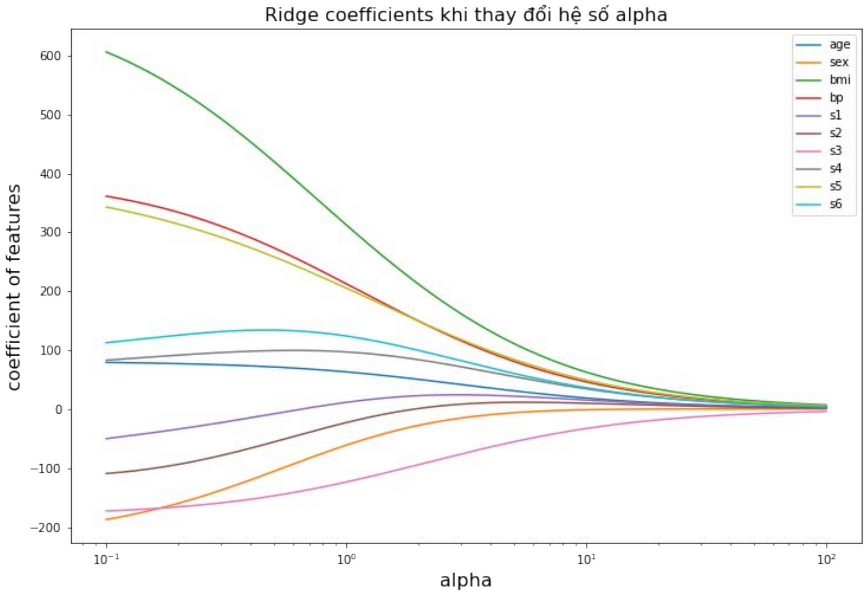
$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \alpha||\mathbf{w}||_2^2$$
$$= \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \underbrace{\alpha R(\mathbf{w})}_{\text{regularization term}}$$

- Alpha is parameter of regularization term.
  - alpha = 0: the regularization term is vanished and return to normal linear regression.
  - alpha big: The regularization term is important. The values of weights tends to be smaller and reduce model's complexity → reduce overfitting.
  - alpha small: The role of regularization term decrease. The values of weights are still be controlled but not much.

Ridged Regression

Regularized Loss = Loss + Regularization Term (L2)

- Experiment when we increase alpha:
  - Diabetes dataset
  - Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements
  - n = 442 diabetes patients
  - a quantitative measure of disease progression one year after baseline.
- **What do you see?**



Ridged Regression

Regularized Loss = Loss + Regularization Term (L2)

- Formulation of root:

$$\mathbf{w} = (\bar{\mathbf{X}}^{\intercal}\bar{\mathbf{X}} + N\alpha\mathbf{I})^{-1}\bar{\mathbf{X}}^{\intercal}\mathbf{y}$$

- Solution of Regularized Loss function: Ridge Regression Solve

- Ridged regression is always exists real root because of $\bar{\mathbf{X}}^{\intercal}\bar{\mathbf{X}} + N\alpha\mathbf{I}$ is non-singular matrix (always exist inverted matrix) when alpha > 0.

Ridged Regression - Tikhonov Regularization

Regularized Loss = Loss + Tikhonov Regularizer

- Regularization Term is Tikhonov Regularizer.

$$\lambda R(\mathbf{w}) = \|\Gamma \mathbf{w}\|_2^2$$

- Γ a square matrix.
- Γ is usually a diagonal matrix

$$\mathbf{\Gamma} = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_p \end{bmatrix}$$

- Tikhonov Regularizer is a general of Ridged Regression with diagonal matrix have diagonal is identity.

Lasso Regression

Regularized Loss = Loss + Regularization Term (L1)

- Regularization Term is L1 norm:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \underbrace{\alpha||\mathbf{w}||_1}_{\text{regularization term}}$$

- L1 norm equation:

$$\begin{aligned}\alpha R(\mathbf{w}) &= \alpha||\mathbf{w}||_1 \\ &= \alpha(|w_0| + |w_1| + \cdots + |w_p|)\end{aligned}$$

- **Question: What is the means of alpha relatively corresponding with zero, small, big?**

Lasso Regression - Selection variable

Regularized Loss = Loss + Regularization Term (L1)

- Lasso Regression usually returns an estimated sparse vector.
- Experiment of Lasso Regression spare weight-vetor return: [Lasso Regression Example]

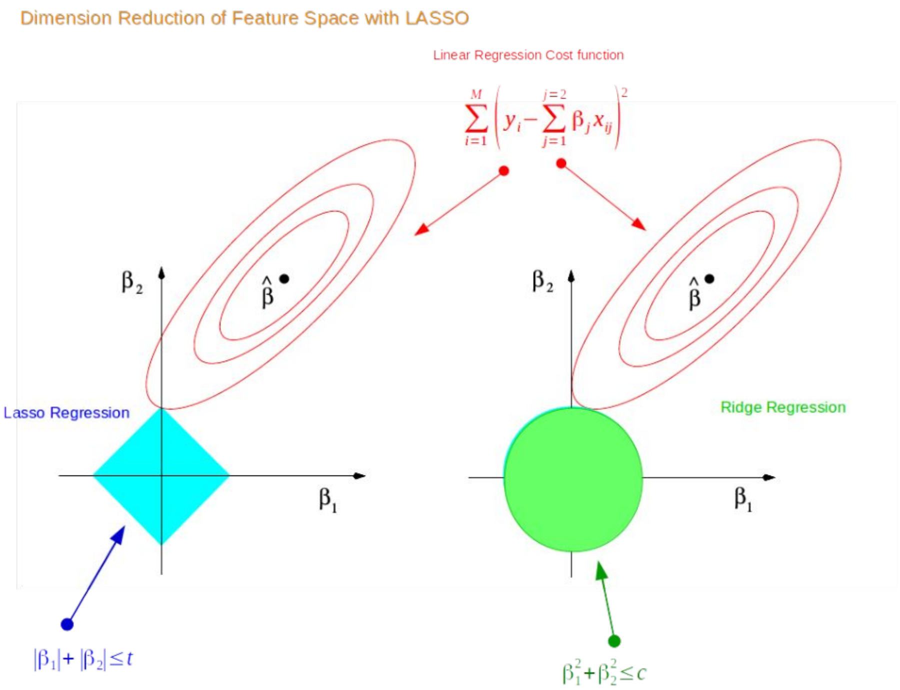
```
from sklearn.linear_model import LassoCV
from sklearn.datasets import make_regression
X, y = make_regression(noise=4, random_state=0)
reg_lasso_cv = LassoCV(cv=5, random_state=0).fit(X, y)
print(reg_lasso_cv.coef_)
print(reg_lasso_cv.intercept_)
```

```
[-4.21242132e-01 -0.00000000e+00  8.74020196e+00  0.00000000e+00
 -0.00000000e+00  0.00000000e+00  5.04074761e-02  7.46065852e+00
  0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 -0.00000000e+00 -1.72366886e-01  0.00000000e+00 -0.00000000e+00
 -0.00000000e+00  0.00000000e+00 -0.00000000e+00 -4.29663159e-01
```

- We use result of Lasso Regression to rank importances of variable.

Lasso Regression - Compare with Ridge Regression

- Constraints of Lasso Regression:
  - |B1| + |B2| <= t
  - Domain of the solution was a rhombus
  - Loss’s contour line tangent with Domain at angle which tend to be zero.
  -
- Constraints of Ridge Regression:
  - B1^2 + B2^2 <= C
  - Domain of the solution was a circle.
  - The contact points between two lines are not zero.



Elastic Net Regression

Regularized Loss = Loss + L1 + L2

- Regularization Term is a linear combination between Norm L1 and L2:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \alpha\left[\lambda||\mathbf{w}||_1 + \frac{(1-\lambda)}{2}||\mathbf{w}||_2^2\right]$$

- lambda is L1 ratio which has value between 0 and 1:
  - lambda = 0: vanish the L1 norm → Ridge Regression.
  - lambda = 1: vanish the L2 norm → Lasso Regression.
  - lambda great: L1 regularization term has more impact on loss.
  - lambda small: L2 has more impact.

Linear Regression Application

- Stock price forecast
  - https://www.kaggle.com/c/the-winton-stock-market-challenge
  - https://www.kaggle.com/c/two-sigma-financial-news
  - https://www.kaggle.com/c/jane-street-market-prediction
  - https://www.kaggle.com/faressayah/stock-market-analysis-prediction-using-lstm
- Weather forecast (humidity, temperature, wind speed)
  - https://www.kaggle.com/c/rainfall-prediction-challenge
  - https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
  - https://www.kaggle.com/c/dmbi-18-metro
  - https://www.kaggle.com/c/1056lab-temperature-forecasting/
  - https://www.kaggle.com/c/dmbi-18-metro/data
- Total consumption of customer
  - https://www.kaggle.com/c/ga-customer-revenue-prediction
  - https://www.kaggle.com/c/santander-customer-transaction-prediction
- Churn customer
  - https://www.kaggle.com/c/customer-churn-prediction-2020
  - https://www.kaggle.com/c/predict-the-churn-for-customer-dataset
  - https://www.kaggle.com/c/1056lab-credit-card-customer-churn-prediction

Linear Regression Application

- Income forecast
  - https://www.kaggle.com/c/ga-customer-revenue-prediction
  - https://www.kaggle.com/c/income-level-prediction
  - https://www.kaggle.com/c/ml-l1-finalcs
  - https://www.kaggle.com/c/tcdml1920-income-ind
- Inventory
  - https://www.kaggle.com/c/dsedelhi
  - https://www.kaggle.com/c/inventory-optimization
  - https://www.kaggle.com/c/demand-forecasting-kernels-only
  - https://www.kaggle.com/c/grupo-bimbo-inventory-demand
- Taxi demand prediction
  - https://www.kaggle.com/sohaibanwaar1203/taxi-demand-prediction
  - https://www.kaggle.com/c/new-york-city-taxi-fare-prediction

DS

boosting your data exploration

Question and Answering

Reference

https://phamdinhkhanh.github.io/deepai-book/ch_ml/index_RidgedRegression.html

https://www.youtube.com/watch?v=Xm2C_gTAl8c

https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

https://machinelearningmastery.com/ridge-regression-with-python/

Lecture Note Ridge Regression

Ridge regression for better usage

ridge and lasso regression guideline with scikit-learn

http://www.cs.cmu.edu/~ggordon/10725-F12/slides/08-general-gd.pdf