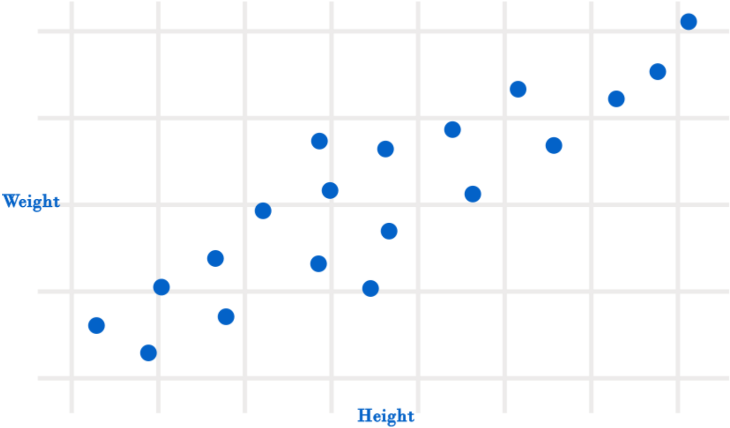
Linear Regression

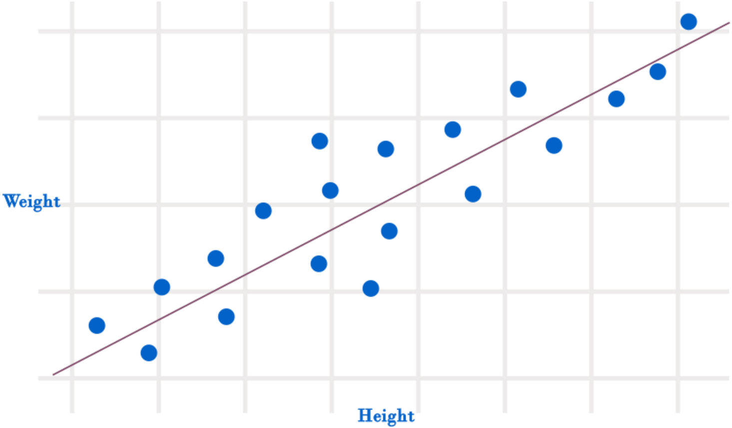
By Van Dinh Tran

January 27, 2025

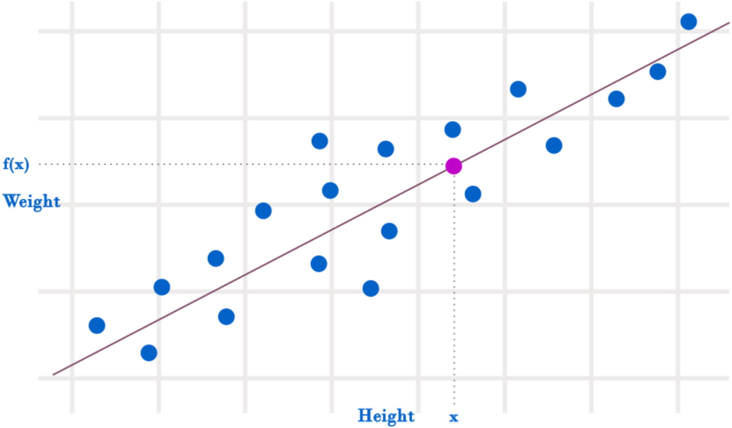
Motivation



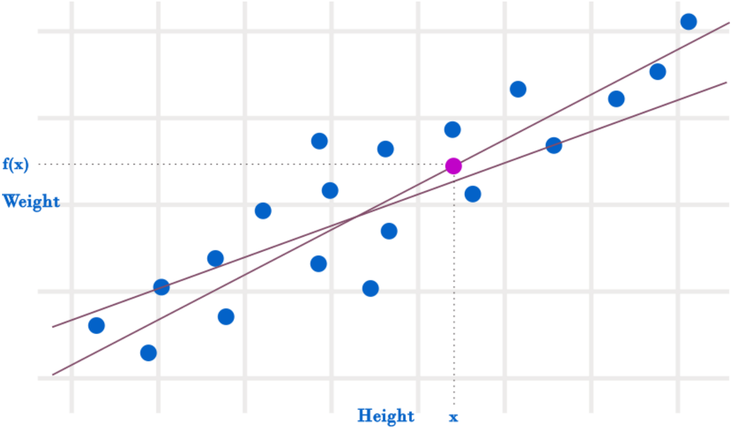
Motivation



Motivation



Motivation



Linear regression

- **Data:**

$$D = \{(x_i, y_i\}) \mid x_i \in X \subseteq R^d, y_i \in Y \subseteq R\}, |D| = N$$

x_i: input vector or instance; y_i: target (label)

- **Hypothesis space:**

$$\mathcal{H} = \{h_w(x) | h_w(x) = \mathbf{w}^\intercal x;\ h_w(x) : X \longrightarrow Y\}$$

($\mathcal{H}$ is a set of linear functions.)

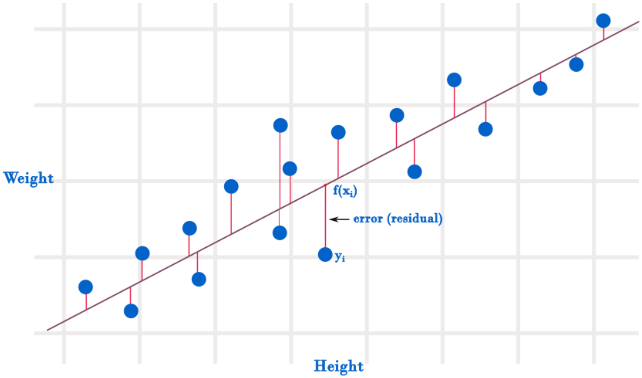
- **Loss function:**

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} \text{loss}(h_w(x_i), y_i)$$

- **Optimization:** Find w (hypothesis) that minimizes L(w)

$$w^\star = \arg\min_w L(w)$$

Ordinary Least Square (OLS) regression



Loss function:

$$L(w) = \frac{1}{N}\sum_{i=1}^{N}(w^{\intercal}x_i - y_i)^2$$

$$w^{\star} = \arg\min_{w} \left[\frac{1}{N}\sum_{i=1}^{N}(w^{\intercal}x_i - y_i)^2\right]$$

Closed form solution

$$w^\star = \arg\min_{w} \left[\frac{1}{N}\sum_{i=1}^{N}(w^\intercal x_i - y_i)^2\right]$$

Idea: Find $w^\star$ by setting partial derivatives to zero.

$$\frac{\partial L}{\partial w_j} = \frac{2}{N}\sum_{i=1}^{N} x_{ij}\left[w^\intercal x_i - y_i\right] = \frac{2}{N}\sum_{i=1}^{N} x_{ij}\left[\sum_{k=1}^{d} w_k x_{ik} - y_i\right]$$

- Let $\frac{\partial L}{\partial w_j} = 0$:

$$\sum_{k=1}^{d}\left[\sum_{i=1}^{N} x_{ij}x_{ik}\right] w_k - \sum_{i=1}^{N} x_{ij}y_i = 0$$

- Notice that It has the form of $Aw - b = 0$ with

$$A_{ij} = \sum_{i=1}^{N} x_{ij}x_{ik};\ b_i = \sum_{i=1}^{N} x_{ij}y_i$$

- Solving this linear system, we get w^*

Note that this solution is particular for this linear regression, not general linear regression.

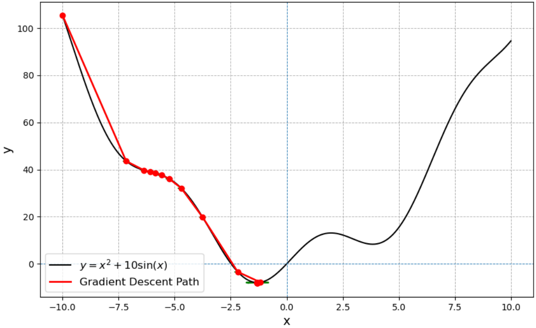
Closed form solution: Matrix form

$$
\begin{aligned}
L(w) &= \|Xw - y\|^2 \\
&= (Xw - y)^\intercal (Xw - y) \\
&= (w^\intercal X^\intercal - y^\intercal)(Xw - y) \\
&= w^\intercal X^\intercal X w - w^\intercal X^\intercal y - y^\intercal X w + y^\intercal y \\
\frac{\partial L}{\partial w} &= 2X^\intercal X w - X^\intercal y - X^\intercal y \\
&= 2X^\intercal X w - 2X^\intercal y
\end{aligned}
$$

$$2X^\intercal X w - 2X^\intercal y = 0 \implies w = (X^\intercal X)^{-1} X^\intercal y$$

- Utilizing SVD
- Note that $X^\intercal X$ is invertible if and only if it has full rank
- We can use gradient descent to solve

Gradient Descent



- Gradient descent is an iterative optimization algorithm to find a function's optimum.
- Commonly used in machine learning to optimize the parameters of complex models.
- Based on the property of gradient that it shows the direction of the steepest ascent of a function

Gradient Descent

$$\min_{w} L(w)$$

Gradient Descent:

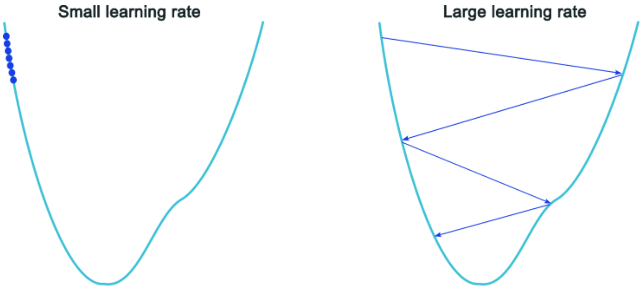
- Pick some value for w
- Iteratively update w until convergence:

$$w \leftarrow w - \alpha * \nabla L$$

where α is the learning rate and ∇L is defined as follows:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial w_d} \end{bmatrix}$$

Gradient Descent: learning rate



- Learning rate (α): a non-negative hyperparameter
- α is small: slow convergence, high precise convergence, and risk of getting stuck at local optima
- α is large: fast convergence, overshooting optimum, and instability

OLS: pros and cons

- **Pros:**
  - Simplicity
  - Interpretability
  - Efficiency
- **Cons:**
  - Linearity assumption
  - Sensitivity to outliers
  - Multicollinearity
  - No Automatic Feature Selection
  - Prone to overfitting
  - Assumption of Independence
  - Limited to Linear Relationships

OLS feature importance

- Standardization: Standardizing the predictors allows for direct comparison of coefficients
- t-statistics, p-value
- Adjusted R-squared contribution
- Combination of different methods

Note: "*statsmodels*" is a good Python package used in practice to interpret the importance of features.

Regression evaluation metrics

- Mean absolute error (MAE):

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

- Mean square error (MSE):

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

- Root mean square error (RMSE):

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

- R^2 error:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- Adjusted R^2 error:

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

Regularized linear regression

- Regularization refers to techniques used to calibrate machine learning models to minimize the adjusted loss function and prevent over-fitting or under-fitting.
- It is most commonly known for its use against over-fitting
- Linear regression with regularization:
  - Lasso: using L1 norm ($\|.\|_1$)
  - Ridge: using L2 norm ($\|.\|_2$)
  - Elastic Net: combining L1 and L2 ($\|.\|_1$ and $\alpha(\|.\|_2)$)
- Norms of $w = (w_1, w_2, \dots, w_d) \in R^d$:
  - Norm 1: $\|w\|_1 = |w_1| + |w_2| + \dots + |w_d|$
  - Norm 2: $\|w\|_2 = \sqrt{w_1^2 + w_2^2 + \dots + w_d^2}$
  - Norm infinity: $\|w\|_\infty = \mathrm{Max}(|w_1|, |w_2|, \dots, |w_d|)$

Ridge regression

- Ridge aims to shrink linear regression coefficients, i.e. parameters are pushed toward zero.
- It adds constraints to the objective function of the simple linear regression using the L2 norm.

$$L(w) = \|Xw - y\|_2^2 + \lambda\|w\|_2^2$$

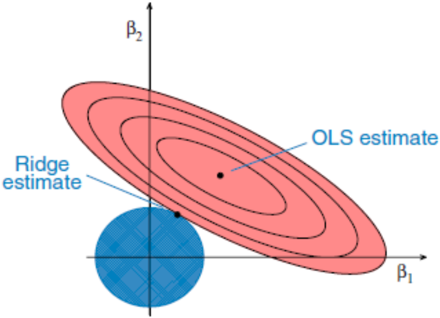
($\|w\|_2^2$: regularization term, λ: regularization strength hyperparameter)

$\frac{\partial L}{\partial w} = 2X^\intercal Xw - 2X^\intercal y + 2\lambda w = 0$

$\implies w^* = (X^\intercal X + \lambda I)^{-1} X^\intercal y$

Note that $(X^\intercal X + \lambda I)$ is is positive-definite, so is invertable.

Ridge regression



Geometric interpretation of Ridge regression

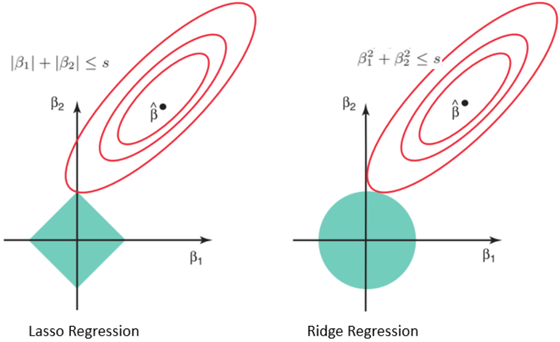
Lasso regression

- Like Ridge regression, it is a regularization technique that uses shrinkage
- The lasso procedure encourages simple, sparse models, i.e. models with fewer parameters. Thus, it has a feature selection capability
- It adds constraints to the objective function of the simple linear regression using the L1 norm.

$$\mathrm{L(w)} = \|\mathrm{Xw} - \mathrm{y}\|_2^2 + \lambda\|\mathrm{w}\|_1$$

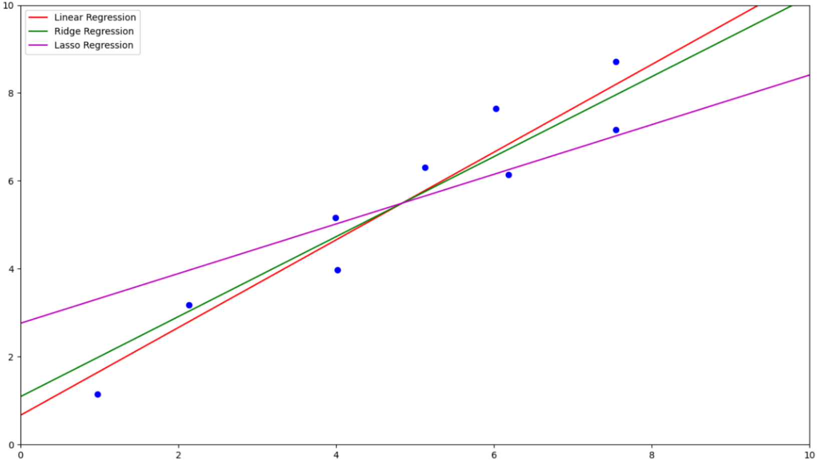
  - L(w) is not differentiable at zero (w=0).
  - Subgradient can be used to calculate zero gradients.
  - Subgradient descent is used instead of gradient descent.

Lasso regression



Ridge regression shrinks coefficients towards zero, while Lasso tends to give a subset of coefficients zero, leading to a sparse solution.

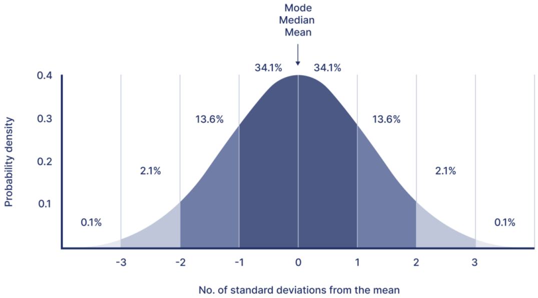
OLS, Ridge and Lasso



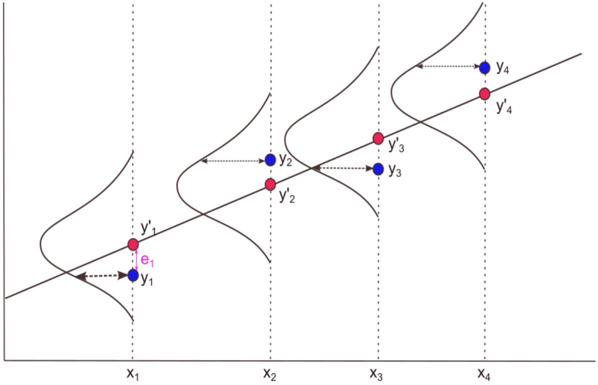
OLS, Ridge ($\lambda = 4$), Lasso ($\lambda = 2$)

Review Normal distribution (Gaussian distribution)

- Univariate:
  - $x \sim \mathcal{N}(\mu, \sigma^2)$
  - Pdf: $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- Multivariate:
  - $x \sim \mathcal{N}(\mu, \Sigma)$
  - Pdf: $f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$



Maximum likelihood regression



- Blue points are true values (labels)
- Red points are predicted values
- For each x_i, we estimate a Gaussian distribution over y_i values, $N(w^\intercal x_i, \sigma)$

Maximum likelihood regression

- Assuming that $y_i \sim \mathcal{N}(w^\intercal x_i, \sigma^2)$ and y_i are independent
- The likelihood of observing $\{y_1, y_2, \ldots, y_N\}$ is as follow:

$$\mathcal{L}(w, \sigma) = p(y|X, w, \sigma) = \prod_{i=1}^{N} p(y_i|x_i, w, \sigma)$$

$$\begin{aligned}
\mathcal{L}(w, \sigma) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - w^\intercal x_i)^2}{2\sigma^2}} \\
&= \frac{1}{(\sqrt{2\pi\sigma^2})^N} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w^\intercal x_i)^2} \\
&= \frac{1}{(\sqrt{2\pi\sigma^2})^N} e^{-\frac{1}{2\sigma^2} (y - Xw)^\intercal (y - Xw)} \quad \text{(Vector form)}
\end{aligned}$$

- Note that maximizing $\mathcal{L}(w, \sigma)$ is equivalent to minimizing the loss function, L, of linear regression.

$$\max_{w,\sigma} \mathcal{L}(w, \sigma) \iff \min_{w} L(w)$$

Maximum likelihood regression

$$\max_{w,\sigma} \mathcal{L}(w,\sigma) \iff \max_{w,\sigma} \mathrm{Log}(\mathcal{L}(w,\sigma))$$

$$\begin{aligned}
\mathrm{Log}(\mathcal{L}(w,\sigma)) &= \mathrm{Log}\big(\frac{1}{(\sqrt{2\pi\sigma^2})^N} e^{-\frac{1}{2\sigma^2}(y-Xw)^\intercal(y-Xw)}\big) \\
&= \mathrm{Log}\big(\frac{1}{(\sqrt{2\pi\sigma^2})^N}\big) + \mathrm{Log}\big(e^{-\frac{1}{2\sigma^2}(y-Xw)^\intercal(y-Xw)}\big) \\
&= -\frac{N}{2}\mathrm{Log}(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y-Xw)^\intercal(y-Xw) \\
&= -\frac{N}{2}\mathrm{Log}(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left[(y^\intercal y - y^\intercal Xw - w^\intercal X^\intercal y + w^\intercal X^\intercal Xw\right] \\
&= -\frac{N}{2}\mathrm{Log}(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left[(y^\intercal y - 2y^\intercal Xw + w^\intercal X^\intercal Xw\right]
\end{aligned}$$

Maximum likelihood regression

Note that $\frac{\partial Aw}{\partial w} = A^\intercal$ and $\frac{\partial w^\intercal Aw}{\partial w} = 2A^\intercal w$ (Matrix differentiation)

$$\frac{\partial \mathrm{Log}(\mathcal{L}(w, \sigma))}{\partial w} = 0 - 2X^\intercal y + 2XX^\intercal w$$
$$\frac{\partial \mathrm{Log}(\mathcal{L}(w, \sigma))}{\partial w} = 0$$
$$\implies w = (XX^\intercal)^{-1} X^\intercal y$$

$$\frac{\partial \mathrm{Log}(\mathcal{L}(w, \sigma))}{\partial \sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}(y - Xw)^2$$
$$\frac{\partial \mathrm{Log}(\mathcal{L}(w, \sigma))}{\partial \sigma} = 0$$
$$\implies \sigma^2 = \frac{1}{N}(y - Xw)^2$$

Prediction: $\hat{y} = \hat{X}w$

Maximum likelihood regression

The key advantage of maximum likelihood regression:

- Variance of estimators
- Handling different distributions: maximum likelihood regression can handle different error distributions, while other linear regression models assume normally distributed errors.
- Complex models: It can easily be extended to handle more complex models, including those with non-linear relationships