Machine Learning

Regression Methods

Topics of previous lecture

- ✓ Ingredients of Machine Learning
- ✓ Classification Basics
- ✓ Basic Linear Classifier
- ✓ K-Nearest Neighbours Classifier
- ✓ Naive Bayes Classifier
- ✓ Linear and Quadratic Discriminant Analysis
- ✓ Support Vector Machines (SVM)
- ✓ Decision Trees
- ✓ Ensemble Methods (Bagging, Weighted Voting, Stacking)

Topics of today's lecture

- Main concepts in regression
- Linear Regression
- Ordinary Least Squares (OLS)

Regression

- Regression task is the same as classification task, except that we must predict a continuous variable (instead of a categorical class label)

Regression

- Regression task is the same as classification task, except that we must predict a continuous variable (instead of a categorical class label)
- For example:

Regression

- Regression task is the same as classification task, except that we must predict a continuous variable (instead of a categorical class label)
- For example:
  - predict the salary given the info about a person

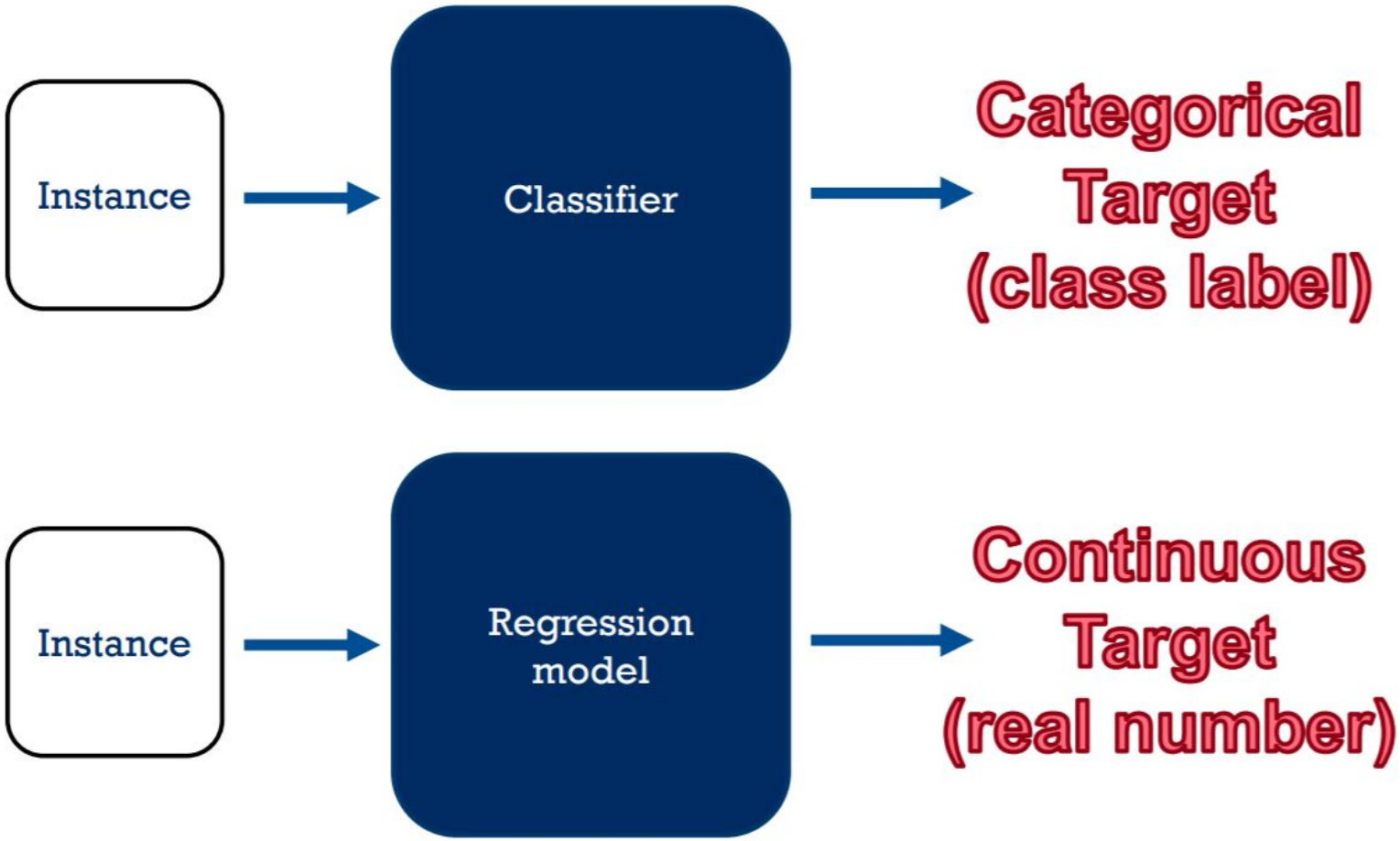
Regression

- Regression task is the same as classification task, except that we must predict a continuous variable (instead of a categorical class label)
- For example:
  - predict the salary given the info about a person
  - predict the risk of a policyholder for insurance

Regression

- Regression task is the same as classification task, except that we must predict a continuous variable (instead of a categorical class label)
- For example:
  - predict the salary given the info about a person
  - predict the risk of a policyholder for insurance
  - predict the expected number of days that a patient will stay in a hospital
  - ...

Classification vs Regression model



Mathematical notation of classification

- $\mathbb{X}$ - input space (set of all possible instances)
- $\mathbb{Y}$ - output space (all possible labels)
- $f : \mathbb{X} \to \mathbb{Y}$ - any such function is a classifier
- $\mathbf{x} \in \mathbb{X}$ - instance
- $y \in \mathbb{Y}$ - actual / true label of instance $\mathbf{x}$
- $\hat{y} = f(\mathbf{x})$- predicted label of instance $\mathbf{x}$

Mathematical notation of ~~classification~~ **regression**

- $\mathbb{X}$ - input space (set of all possible instances)
- $\mathbb{Y}$ - output space (all possible ~~labels~~ **real numbers**)
- $f : \mathbb{X} \to \mathbb{Y}$ - any such function is a ~~classifier~~ **regression model**
- $\mathbf{x} \in \mathbb{X}$ - instance
- $y \in \mathbb{Y}$ - actual / true ~~label~~ **target value** of instance $\mathbf{x}$
- $\hat{y} = f(\mathbf{x})$- predicted ~~label~~ **target value** of instance $\mathbf{x}$

Learning Problem in Regression

- Suppose that there exists an actual / true function, mapping the features to target variable $f^* : \mathbb{X} \rightarrow \mathbb{R}$

Learning Problem in Regression

- Suppose that there exists an actual / true function, mapping the features to target variable $f^* : \mathbb{X} \rightarrow \mathbb{R}$
- In regression the task is to learn a function approximator $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$ such that $\hat{f} \approx f^*$

Learning Problem in Regression

- Suppose that there exists an actual / true function, mapping the features to target variable $f^* : \mathbb{X} \to \mathbb{R}$
- In regression the task is to learn a function approximator $\hat{f} : \mathbb{X} \to \mathbb{R}$ such that $\hat{f} \approx f^*$
- For this we are given training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{X} \times \mathbb{R}$

Learning Problem in Regression

- Do we want to learn $\hat{f}$ such that: $\hat{f}(\mathbf{x}_1) \approx y_1, \ldots, \hat{f}(\mathbf{x}_n) \approx y_n$?

Learning Problem in Regression

- Do we want to learn $\hat{f}$ such that: $\hat{f}(\mathbf{x}_1) \approx y_1, \ldots, \hat{f}(\mathbf{x}_n) \approx y_n$?
- No! This would mean good predictions on training data, which is not our main goal!

Learning Problem in Regression

- Do we want to learn $\hat{f}$ such that: $\hat{f}(\mathbf{x}_1) \approx y_1, \ldots, \hat{f}(\mathbf{x}_n) \approx y_n$?
- No! This would mean good predictions on training data, which is not our main goal!
- We want to predict well on (future) test data!

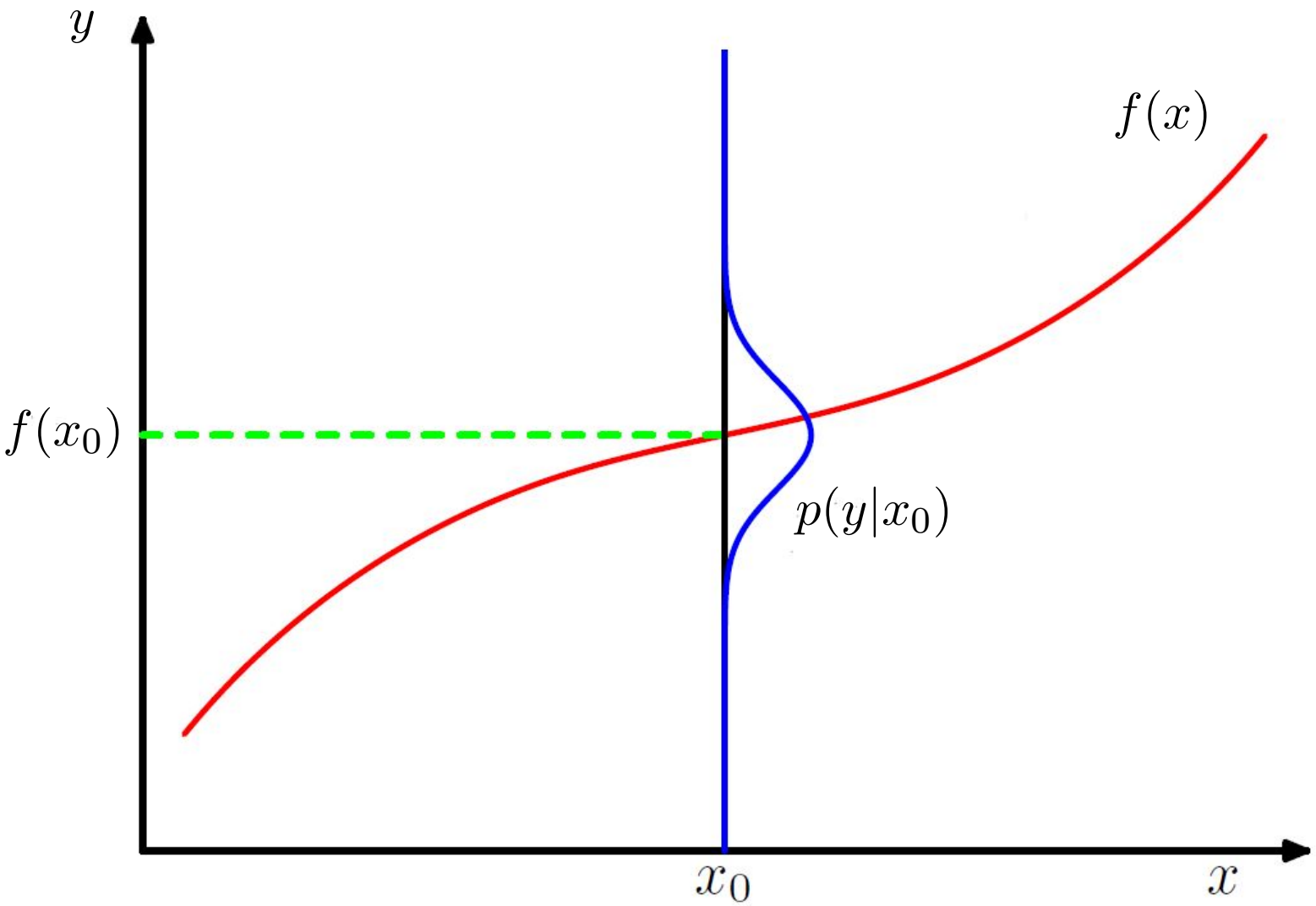
Learning Problem in Regression

- Do we want to learn $\hat{f}$ such that: $\hat{f}(\mathbf{x}_1) \approx y_1, \ldots, \hat{f}(\mathbf{x}_n) \approx y_n$?
- No! This would mean good predictions on training data, which is not our main goal!
- We want to predict well on (future) test data!
- On any future instance $\mathbf{X} \in \mathbb{X}$ with true (hidden) target Y we want $\hat{f}(\mathbf{X}) \approx Y$

Learning Problem in Regression

- Do we want to learn $\hat{f}$ such that: $\hat{f}(\mathbf{x}_1) \approx y_1, \ldots, \hat{f}(\mathbf{x}_n) \approx y_n$?
- No! This would mean good predictions on training data, which is not our main goal!
- We want to predict well on (future) test data!
- On any future instance $\mathbf{X} \in \mathbb{X}$ with true (hidden) target Y we want $\hat{f}(\mathbf{X}) \approx Y$
- What does this really mean?

Learning Problem in Regression



Derivation of the Regression function

Definition

For X and Y continuous random variables, the conditional expectation is

$$\mathbb{E}(X|Y) = \int_{x \in \mathcal{X}} x p(x|y) dx,$$

where $p(x|y) = \frac{p(x,y)}{p(y)}$ is the conditional density function of X given Y.

Derivation of the Regression function

Definition

For X and Y continuous random variables, the conditional expectation is

$$\mathbb{E}(X|Y) = \int_{x \in \mathcal{X}} x p(x|y) dx,$$

where $p(x|y) = \frac{p(x,y)}{p(y)}$ is the conditional density function of X given Y.

- When choosing a particular estimate function $f(\mathbf{x})$ for the true value y we encounter some loss $L(y, f(\mathbf{x}))$

Derivation of the Regression function

Definition

For X and Y continuous random variables, the conditional expectation is

$$\mathbb{E}(X|Y) = \int_{x \in \mathcal{X}} x p(x|y) dx,$$

where $p(x|y) = \frac{p(x,y)}{p(y)}$ is the conditional density function of X given Y.

- When choosing a particular estimate function $f(\mathbf{x})$ for the true value y we encounter some loss $L(y, f(\mathbf{x}))$
- The average or expected loss is given by

$$\mathbb{E}(L) = \iint L(y, f(\mathbf{x})) p(\mathbf{x}, y) d\mathbf{x} dy$$

Derivation of the Regression function

- The average or expected loss is given by $\mathbb{E}(L) = \int\int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$

Derivation of the Regression function

- The average or expected loss is given by $\mathbb{E}(L) = \int\int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$
- A common choice of loss function in regression is the squared loss $L(y, f(\mathbf{x})) = (f(\mathbf{x}) - y)^2$. The average loss will be

$$\mathbb{E}(L) = \int\int (f(\mathbf{x}) - y)^2 p(\mathbf{x}, y)d\mathbf{x}dy$$

Derivation of the Regression function

- The average or expected loss is given by $\mathbb{E}(L) = \int\int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$
- A common choice of loss function in regression is the squared loss $L(y, f(\mathbf{x})) = (f(\mathbf{x}) - y)^2$. The average loss will be

$$\mathbb{E}(L) = \int\int (f(\mathbf{x}) - y)^2 p(\mathbf{x}, y)d\mathbf{x}dy$$

- Our goal is to choose $f(\mathbf{x})$ such that $\mathbb{E}(L)$ will be minimized:

$$\frac{\partial \mathbb{E}(L)}{\partial f(\mathbf{x})} = 2\int (f(\mathbf{x}) - y)p(\mathbf{x}, y)dy = 0$$

Derivation of the Regression function

- The average or expected loss is given by $\mathbb{E}(L) = \int\int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$
- A common choice of loss function in regression is the squared loss $L(y, f(\mathbf{x})) = (f(\mathbf{x}) - y)^2$. The average loss will be

$$\mathbb{E}(L) = \int\int (f(\mathbf{x}) - y)^2 p(\mathbf{x}, y)d\mathbf{x}dy$$

- Our goal is to choose $f(\mathbf{x})$ such that $\mathbb{E}(L)$ will be minimized:

$$\frac{\partial \mathbb{E}(L)}{\partial f(\mathbf{x})} = 2\int (f(\mathbf{x}) - y)p(\mathbf{x}, y)dy = 0$$

- Solving for $f(\mathbf{x})$ gives

$$f(\mathbf{x}) = \frac{\int yp(\mathbf{x}, y)dy}{p(\mathbf{x})} = \int yp(y|\mathbf{x})dy = \mathbb{E}(Y|X)$$

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y
- Suppose we want to predict the weight of a person from height

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y
- Suppose we want to predict the weight of a person from height
- In reality, there are many people with the same height but different weight

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y
- Suppose we want to predict the weight of a person from height
- In reality, there are many people with the same height but different weight
- Height does not determine the weight uniquely

Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y
- Suppose we want to predict the weight of a person from height
- In reality, there are many people with the same height but different weight
- Height does not determine the weight uniquely
- Relationship between height and weight is non-deterministic

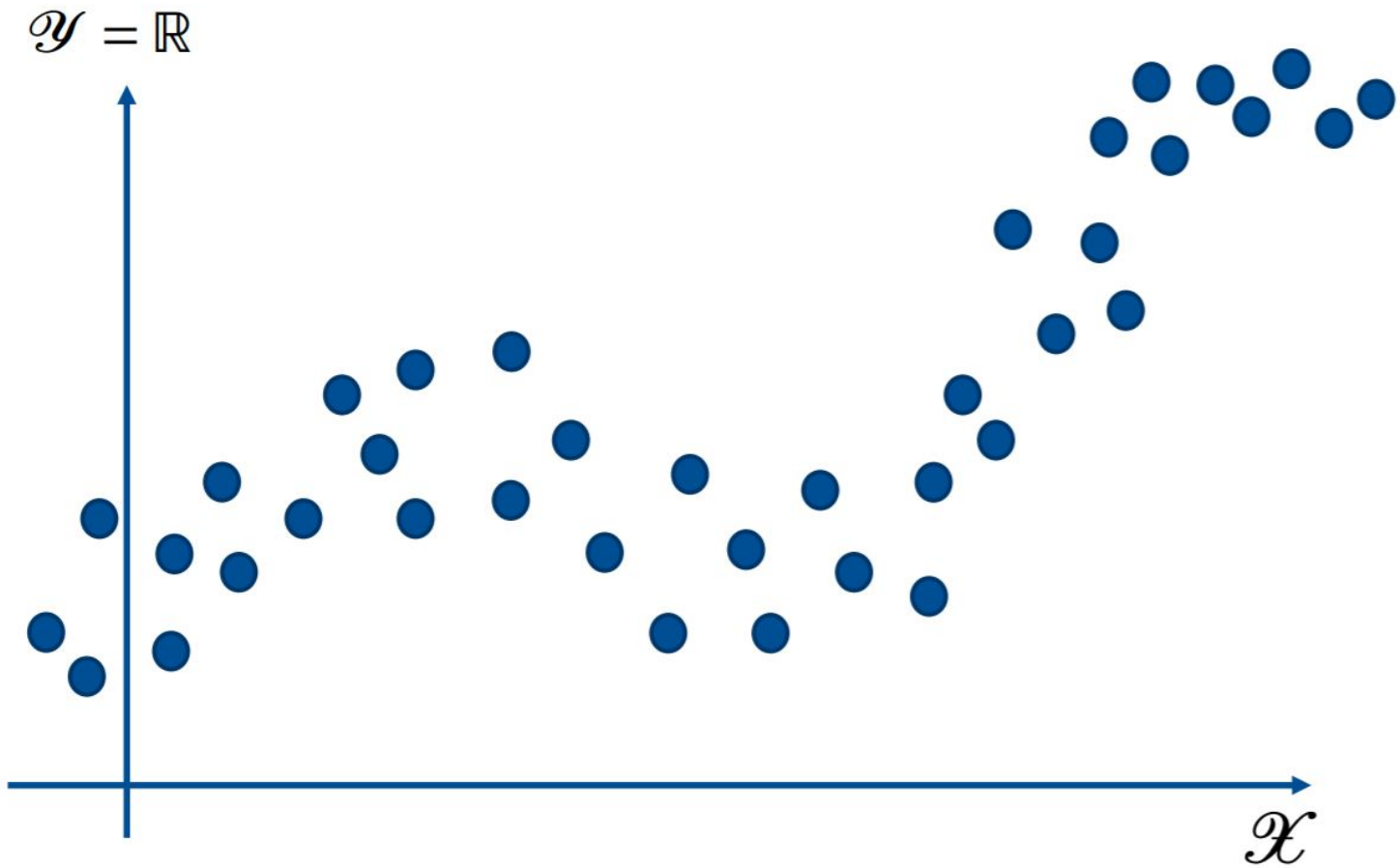
Mean Regression

- In mean regression (the usual type of regression) the task is taken to predict the conditional expectation:

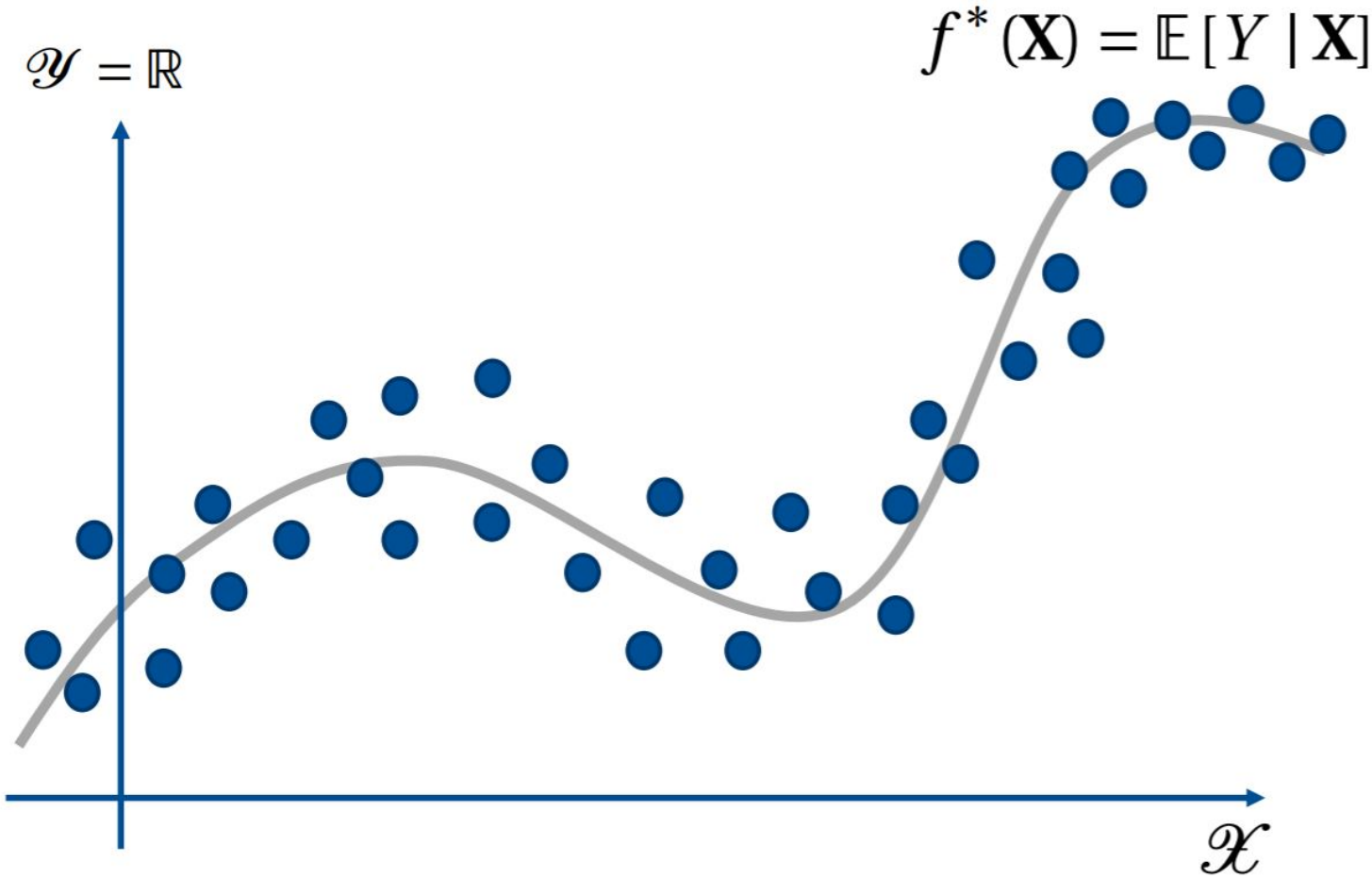
$$f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$$

- Hence, for any $\mathbf{X}$ we want to estimate the expected value of the corresponding Y
- Suppose we want to predict the weight of a person from height
- In reality, there are many people with the same height but different weight
- Height does not determine the weight uniquely
- Relationship between height and weight is non-deterministic
- In case of mean regression, we predict the expected weight of people with a given height

Mean Regression



Mean Regression



Alternatives to mean regression

- Median regression:
  Example: for a given height, predict the weight such that approximately half of the people with this height would be heavier than this

Alternatives to mean regression

- Median regression:
  Example: for a given height, predict the weight such that approximately half of the people with this height would be heavier than this
- Quantile regression:
  Median regression generalized to any other quantile (median is the 50%-quantile)

Evaluation of regression

- How do we evaluate how well $\hat{f}$ approximates $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$?

Evaluation of regression

- How do we evaluate how well $\hat{f}$ approximates $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$?
- Even if we have hold-out test data, we still do not know the true $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$.

Evaluation of regression

- How do we evaluate how well $\hat{f}$ approximates $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$?
- Even if we have hold-out test data, we still do not know the true $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$.
- $f^*(\mathbf{X})$ minimizes the expected **squared error** on future data

$$f^*(\mathbf{X}) = \underset{y \in \mathbb{R}}{\operatorname{argmin}} \, \mathbb{E}[(y - Y)^2 | \mathbf{X}] = \mathbb{E}[Y|\mathbf{X}]$$

Evaluation of regression

- How do we evaluate how well $\hat{f}$ approximates $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$?
- Even if we have hold-out test data, we still do not know the true $f^*(\mathbf{X}) = \mathbb{E}(Y|\mathbf{X})$.
- $f^*(\mathbf{X})$ minimizes the expected **squared error** on future data

$$f^*(\mathbf{X}) = \underset{y \in \mathbb{R}}{\operatorname{argmin}} \, \mathbb{E}[(y - Y)^2 | \mathbf{X}] = \mathbb{E}[Y|\mathbf{X}]$$

- Therefore, the most usual evaluation measure in (mean) regression is **mean squared error (MSE)** on test data:

$$\frac{1}{|Te|} \sum_{(\mathbf{x},y) \in Te} (\hat{f}(\mathbf{x}) - y)^2$$

Noise in regression

- **Noise** is the difference between the true label and the prediction $f^* = \mathbb{E}[Y|\mathbf{X}]$

$$\epsilon = Y - f^*(\mathbf{X})$$

Noise in regression

- **Noise** is the difference between the true label and the prediction $f^* = \mathbb{E}[Y|\mathbf{X}]$

$$\epsilon = Y - f^*(\mathbf{X})$$

- This relationship is usually presented as:

$$Y = f^*(X) + \epsilon$$

Noise in regression

- **Noise** is the difference between the true label and the prediction $f^* = \mathbb{E}[Y|\mathbf{X}]$

$$\epsilon = Y - f^*(\mathbf{X})$$

- This relationship is usually presented as:

$$Y = f^*(X) + \epsilon$$

- The task in mean regression is to predict $f^*(X)$ and the noise cannot be (and should not attempted to be) predicted from the features X

Example on noise

- Suppose we want to predict a person's weight from height

Example on noise

- Suppose we want to predict a person's weight from height
- What is the noise here?

Example on noise

- Suppose we want to predict a person's weight from height
- What is the noise here?
- Noise is the difference between the actual weight and the average weight among people with the same height

More on evaluation of regression

- Two common measures:

More on evaluation of regression

- Two common measures:
  - MSE - mean squared error

$$MSE = \frac{1}{|Te|} \sum_{(\mathbf{x},y) \in Te} (\hat{f}(\mathbf{x}) - y)^2$$

More on evaluation of regression

- Two common measures:
  - MSE - mean squared error

$$MSE = \frac{1}{|Te|} \sum_{(\mathbf{x},y)\in Te} (\hat{f}(\mathbf{x}) - y)^2$$

  - RMSE - root mean squared error

$$RMSE = \sqrt{MSE}$$

More on evaluation of regression

- Two common measures:
  - MSE - mean squared error

$$MSE = \frac{1}{|Te|} \sum_{(\mathbf{x},y)\in Te} (\hat{f}(\mathbf{x}) - y)^2$$

  - RMSE - root mean squared error

$$RMSE = \sqrt{MSE}$$

- The advantage of RMSE: easier to interpret because it is measured in the same units as the target variable (whereas MSE is measured in these units squared)

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$
- In linear regression:

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$
- In linear regression:
  - We assume that the features are all numeric:

$$\mathbb{X} \in \mathbb{R}^d$$

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$
- In linear regression:
  - We assume that the features are all numeric:

$$\mathbb{X} \in \mathbb{R}^d$$

  - We must learn a linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$
- In linear regression:
  - We assume that the features are all numeric:

$$\mathbb{X} \in \mathbb{R}^d$$

  - We must learn a linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

    - w_0 - intercept

Linear regression

- In regression the task is to learn a function approximator: $\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$
- In linear regression:
  - We assume that the features are all numeric:

$$\mathbb{X} \in \mathbb{R}^d$$

  - We must learn a linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

    - w_0 - intercept
    - $\mathbf{w}$ - coefficients (coefficient vector)

Univariate linear regression

- Linear regression with a single feature

Univariate linear regression

- Linear regression with a single feature
- We must learn a univariate linear function:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + w_1 x_1,$$

where w_0 is the intercept and w_1 is the slope of the line

Univariate linear regression

- Linear regression with a single feature
- We must learn a univariate linear function:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + w_1 x_1,$$

  where w_0 is the intercept and w_1 is the slope of the line
- We have training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathbb{R}^2$

Univariate linear regression

- Linear regression with a single feature
- We must learn a univariate linear function:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + w_1 x_1,$$

  where w_0 is the intercept and w_1 is the slope of the line
- We have training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathbb{R}^2$
- Task is to learn w_0 and w_1 such that $\hat{f}$ minimizes future squared error

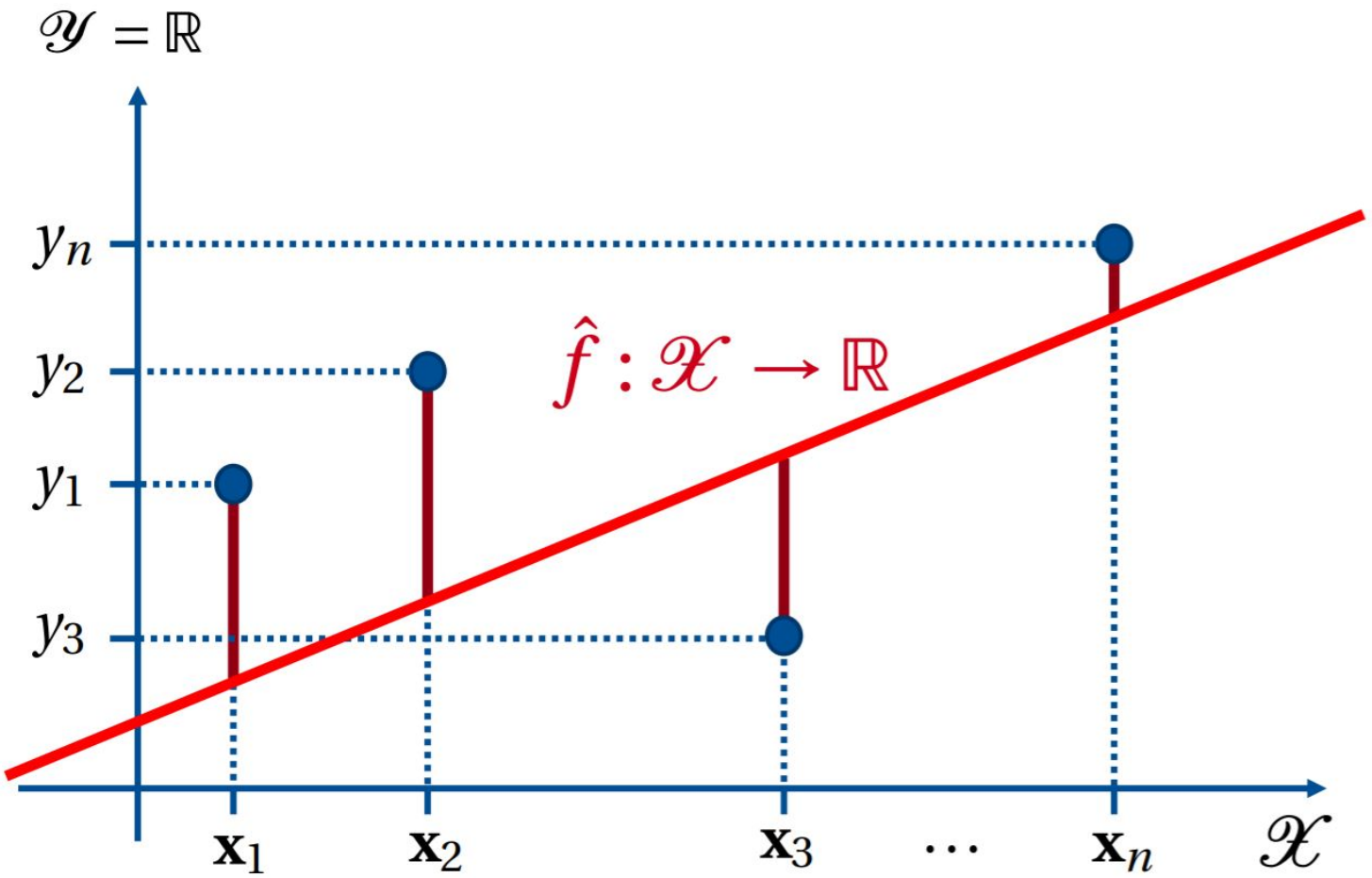
Univariate linear regression

- Linear regression with a single feature
- We must learn a univariate linear function:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + w_1 x_1,$$

  where w_0 is the intercept and w_1 is the slope of the line
- We have training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathbb{R}^2$
- Task is to learn w_0 and w_1 such that $\hat{f}$ minimizes future squared error
- One of the popular methods is the Ordinary Least Squares (OLS) method

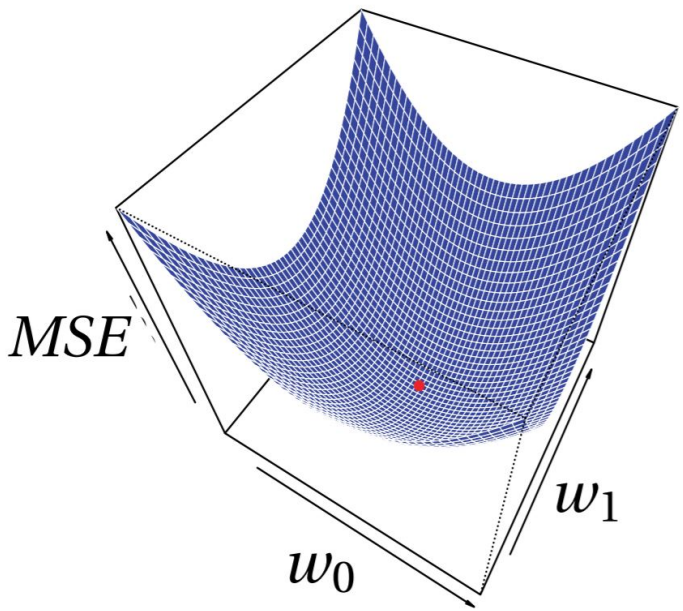
Ordinary Least Squares (OLS)



Univariate OLS

Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{w}_1 = \underset{w_0, w_1}{\operatorname{argmin}} MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$



Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

- Calculating the gradient with respect to w_0

$$\frac{\partial MSE}{\partial w_0} =$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

- Calculating the gradient with respect to w_0

$$\frac{\partial MSE}{\partial w_0} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y) = 0$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

- Calculating the gradient with respect to w_0

$$\frac{\partial MSE}{\partial w_0} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y) = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} y = w_0 + w_1 \frac{1}{|Tr|} \sum_{(x,y) \in Tr} x$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

- Calculating the gradient with respect to w_0

$$\frac{\partial MSE}{\partial w_0} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y) = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} y = w_0 + w_1 \frac{1}{|Tr|} \sum_{(x,y) \in Tr} x$$

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

where $\bar{x} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} x$ and $\bar{y} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} y$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} =$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

$$w_1 \sum_{(x,y) \in Tr} (x - \bar{x}) \cdot x = \sum_{(x,y) \in Tr} (y - \bar{y}) \cdot x$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

$$w_1 \sum_{(x,y) \in Tr} (x - \bar{x}) \cdot x = \sum_{(x,y) \in Tr} (y - \bar{y}) \cdot x$$

$$w_1 \Big(\sum (x - \bar{x}) \cdot x - \sum (x - \bar{x}) \cdot \bar{x} \Big) = \sum (y - \bar{y}) \cdot x - \sum (y - \bar{y}) \cdot \bar{x}$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

$$w_1 \sum_{(x,y) \in Tr} (x - \bar{x}) \cdot x = \sum_{(x,y) \in Tr} (y - \bar{y}) \cdot x$$

$$w_1 \Big(\sum (x - \bar{x}) \cdot x - \sum (x - \bar{x}) \cdot \bar{x} \Big) = \sum (y - \bar{y}) \cdot x - \sum (y - \bar{y}) \cdot \bar{x}$$

$$w_1 \sum (x - \bar{x})^2 = \sum (y - \bar{y})(x - \bar{x}) \Rightarrow$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

$$w_1 \sum_{(x,y) \in Tr} (x - \bar{x}) \cdot x = \sum_{(x,y) \in Tr} (y - \bar{y}) \cdot x$$

$$w_1 \Big(\sum (x - \bar{x}) \cdot x - \sum (x - \bar{x}) \cdot \bar{x} \Big) = \sum (y - \bar{y}) \cdot x - \sum (y - \bar{y}) \cdot \bar{x}$$

$$w_1 \sum (x - \bar{x})^2 = \sum (y - \bar{y})(x - \bar{x}) \Rightarrow \quad w_1 = \frac{\sum (y - \bar{y})(x - \bar{x})}{\sum (x - \bar{x})^2} =$$

Calculating the partial derivatives

$$MSE(w_0, w_1) = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + w_1 x) - y)^2$$

Calculating the gradient with respect to w_1

$$\frac{\partial MSE}{\partial w_1} = \frac{1}{|Tr|} \sum_{(x,y) \in Tr} 2(w_0 + w_1 x - y)x = 0$$

$$\frac{1}{|Tr|} \sum_{(x,y) \in Tr} (\bar{y} - w_1 \bar{x} + w_1 x - y)x = 0$$

$$w_1 \sum_{(x,y) \in Tr} (x - \bar{x}) \cdot x = \sum_{(x,y) \in Tr} (y - \bar{y}) \cdot x$$

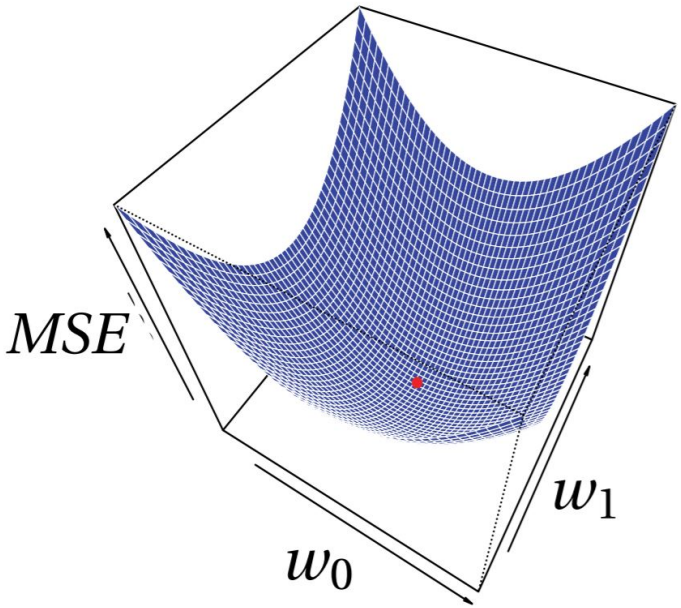
$$w_1 \Big(\sum (x - \bar{x}) \cdot x - \sum (x - \bar{x}) \cdot \bar{x} \Big) = \sum (y - \bar{y}) \cdot x - \sum (y - \bar{y}) \cdot \bar{x}$$

$$w_1 \sum (x - \bar{x})^2 = \sum (y - \bar{y})(x - \bar{x}) \Rightarrow \quad w_1 = \frac{\sum (y - \bar{y})(x - \bar{x})}{\sum (x - \bar{x})^2} = \frac{\hat{Cov}(x,y)}{\hat{Var}(x)}$$

Univariate OLS

Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{w}_1 = \underset{w_0, w_1}{\operatorname{argmin}} MSE(w_0, w_1) = \begin{cases} \hat{w}_1 & = \dfrac{\hat{Cov}(x, y)}{\hat{Var}(x)} \\ \hat{w}_0 & = \bar{y} - \hat{w}_1 \bar{x} \end{cases}$$



Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x
  - sample mean of y

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x
  - sample mean of y
  - sample covariance of x and y

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x
  - sample mean of y
  - sample covariance of x and y
- Predicting for a new instance x:

$$\hat{y} = \hat{f}(x) = \hat{w}_0 + \hat{w}_1 x$$

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x
  - sample mean of y
  - sample covariance of x and y
- Predicting for a new instance x:

$$\hat{y} = \hat{f}(x) = \hat{w}_0 + \hat{w}_1 x$$

- If before learning the regression model we **standardise** both the feature and the target variable (zero mean and unit variance), then

Univariate OLS

- Fitting:
  Calculate $\hat{w}_0, \hat{w}_1$ based on the
  - sample mean and variance of feature values x
  - sample mean of y
  - sample covariance of x and y
- Predicting for a new instance x:

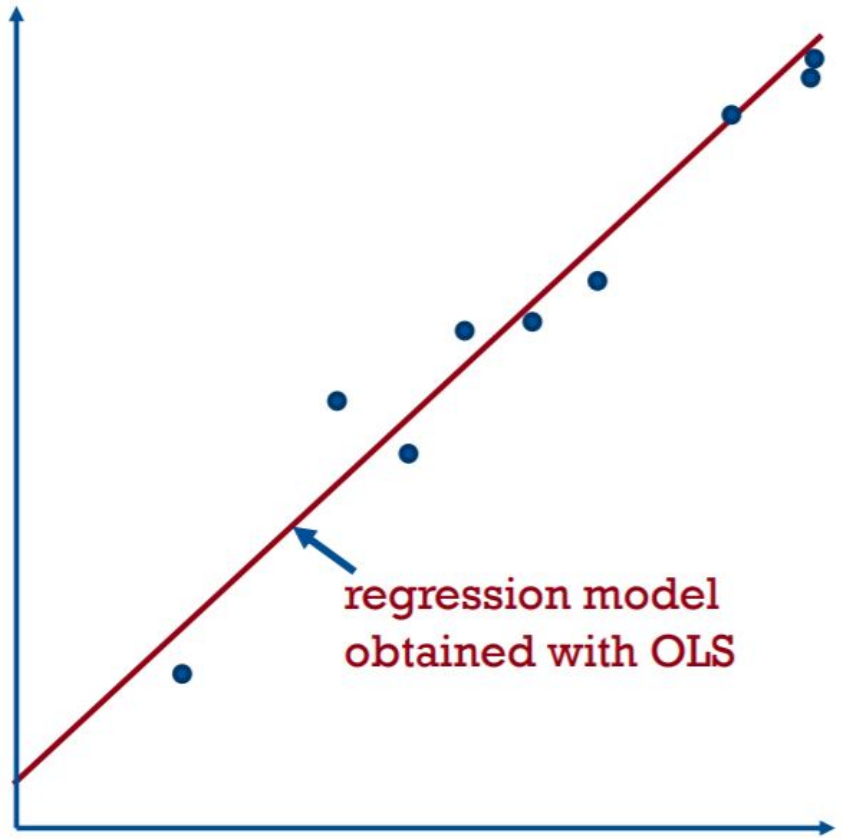
$$\hat{y} = \hat{f}(x) = \hat{w}_0 + \hat{w}_1 x$$

- If before learning the regression model we **standardise** both the feature and the target variable (zero mean and unit variance), then

$$\hat{w}_0, \hat{w}_1 = \underset{w_0, w_1}{\operatorname{argmin}} MSE(w_0, w_1) = \begin{cases} \hat{w}_1 = \frac{\hat{Cov}(x,y)}{\hat{Var}(x)} = \hat{Corr}(x,y) \\ \hat{w}_0 = \bar{y} - w_1 \bar{x} = 0 \end{cases}$$

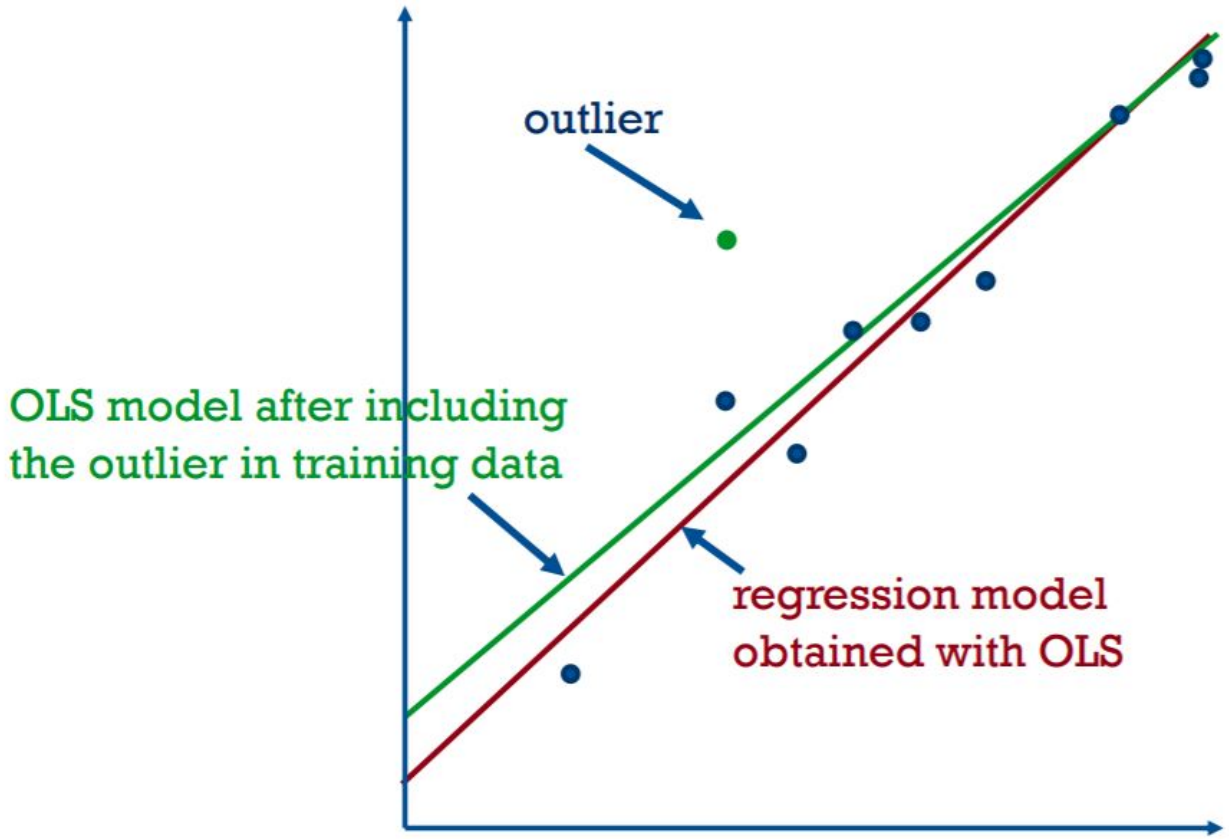
Ordinary Least Squares

- OLS is very sensitive to outliers
- A single faraway point can significantly shift the predictions



Ordinary Least Squares

- OLS is very sensitive to outliers
- A single faraway point can significantly shift the predictions



Multivariate OLS

- Linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

Multivariate OLS

- Linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

- Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{\mathbf{w}} = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \, MSE(w_0, \mathbf{w}) = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + \mathbf{w} \cdot \mathbf{x}) - y)^2$$

Multivariate OLS

- Linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

- Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{\mathbf{w}} = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \, MSE(w_0, \mathbf{w}) = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + \mathbf{w} \cdot \mathbf{x}) - y)^2$$

- Let's add a feature, which is always 1

$$\mathbf{x} = (x_1, \ldots, x_d) \rightarrow \mathbf{x}' = (1, x_1, \ldots, x_d) = (x_0, x_1, \ldots, x_d)$$

Multivariate OLS

- Linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

- Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{\mathbf{w}} = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \, MSE(w_0, \mathbf{w}) = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + \mathbf{w} \cdot \mathbf{x}) - y)^2$$

- Let's add a feature, which is always 1

$$\mathbf{x} = (x_1, \ldots, x_d) \rightarrow \mathbf{x}' = (1, x_1, \ldots, x_d) = (x_0, x_1, \ldots, x_d)$$

$$\hat{f}(\mathbf{x}') = \mathbf{w}' \cdot \mathbf{x}' = \sum_{i=0}^{d} w_i x_i$$

Multivariate OLS

- Linear model:

$$\hat{f}(\mathbf{x}) = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_{i=1}^{d} w_i x_i$$

- Ordinary least squares (OLS) regression learns the weights by minimizing MSE on training data:

$$\hat{w}_0, \hat{\mathbf{w}} = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \, MSE(w_0, \mathbf{w}) = \underset{w_0, \mathbf{w}}{\operatorname{argmin}} \frac{1}{|Tr|} \sum_{(x,y) \in Tr} ((w_0 + \mathbf{w} \cdot \mathbf{x}) - y)^2$$

- Let's add a feature, which is always 1

$$\mathbf{x} = (x_1, \ldots, x_d) \rightarrow \mathbf{x}' = (1, x_1, \ldots, x_d) = (x_0, x_1, \ldots, x_d)$$

$$\hat{f}(\mathbf{x}') = \mathbf{w}' \cdot \mathbf{x}' = \sum_{i=0}^{d} w_i x_i$$

from now on we will skip the ' for simplicity

OLS in matrix form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, MSE(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{|Tr|} \sum_{(\mathbf{x},y)\in Tr} (\mathbf{w}\cdot\mathbf{x} - y)^2$$

OLS in matrix form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, MSE(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{|Tr|} \sum_{(\mathbf{x},y)\in Tr} (\mathbf{w}\cdot\mathbf{x} - y)^2$$

- Denote the training instances by $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^{d+1} \times \mathbb{R}$

OLS in matrix form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, MSE(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{|Tr|} \sum_{(\mathbf{x},y)\in Tr} (\mathbf{w}\cdot\mathbf{x} - y)^2$$

- Denote the training instances by $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^{d+1} \times \mathbb{R}$
- Denote the **residual errors** on instances by: $e_1 = y_1 - \mathbf{x}_1 \cdot \mathbf{w}, \ldots, e_n = y_n - \mathbf{x}_n \cdot \mathbf{w}$

OLS in matrix form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, MSE(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{|Tr|} \sum_{(\mathbf{x},y)\in Tr} (\mathbf{w}\cdot\mathbf{x} - y)^2$$

- Denote the training instances by $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^{d+1} \times \mathbb{R}$
- Denote the **residual errors** on instances by: $e_1 = y_1 - \mathbf{x}_1 \cdot \mathbf{w}, \ldots, e_n = y_n - \mathbf{x}_n \cdot \mathbf{w}$
- In matrix form: $\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} e_i^2 = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbf{e}\cdot\mathbf{e} = \underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{w})\cdot(\mathbf{y} - \mathbf{X}\mathbf{w})$$

OLS in matrix form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, MSE(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}}\, \frac{1}{|Tr|} \sum_{(\mathbf{x},y)\in Tr} (\mathbf{w}\cdot\mathbf{x} - y)^2$$

- Denote the training instances by $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^{d+1} \times \mathbb{R}$
- Denote the **residual errors** on instances by: $e_1 = y_1 - \mathbf{x}_1 \cdot \mathbf{w}, \ldots, e_n = y_n - \mathbf{x}_n \cdot \mathbf{w}$
- In matrix form: $\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} e_i^2 = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbf{e}\cdot\mathbf{e} = \underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{w})\cdot(\mathbf{y} - \mathbf{X}\mathbf{w})$$

- To solve this we equate the gradient to zero:

$$\frac{\partial (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})}{\partial \mathbf{w}} = 0$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial \mathbf{w}} =$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} = -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = 0$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} = -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} = -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

Calculating the partial derivatives

Based on the properties of matrix derivatives:

$$\frac{\partial(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{\partial\mathbf{w}} = -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

For multivariate OLS there exists a **closed-form solution** (i.e. can be explicitly calculated without numerical optimization):

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}(\mathbf{y}-\mathbf{X}\mathbf{w})\cdot(\mathbf{y}-\mathbf{X}\mathbf{w}) = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

Multivariate OLS

Pro: if the number of instances is much bigger than the number of features, then OLS works quite well

Multivariate OLS

Pro: if the number of instances is much bigger than the number of features, then OLS works quite well

Con: otherwise OLS tends to overfit the noise, particularly if there is a lot of noise in the data

Multivariate OLS

Pro: if the number of instances is much bigger than the number of features, then OLS works quite well

Con: otherwise OLS tends to overfit the noise, particularly if there is a lot of noise in the data

Con: if many features are collinear (highly correlated) then OLS tends to overfit

What have we learned today?

- ✓ Main concepts in regression
- ✓ Linear Regression
- ✓ Ordinary Least Squares (OLS)