CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering
Department of Cybernetics

Linear regression

Petr Pošík

Linear regression

Linear regression

Regression task is a supervised learning task, i.e.

- a training (multi)set $T = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \dots, (\boldsymbol{x}^{(|T|)}, y^{(|T|)})\}$ is available, where
- the labels $y^{(i)}$ are *quantitave*, often continuous (as opposed to classification tasks where $y^{(i)}$ are nominal).
- Its purpose is to model the relationship between independent variables (inputs) $\boldsymbol{x} = (x_1, \dots, x_D)$ and the dependent variable (output) y.

Linear regression

Regression task is a supervised learning task, i.e.

- a training (multi)set $T = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(|T|)}, y^{(|T|)})\}$ is available, where
- the labels $y^{(i)}$ are *quantitave*, often continuous (as opposed to classification tasks where $y^{(i)}$ are nominal).
- Its purpose is to model the relationship between independent variables (inputs) $\boldsymbol{x} = (x_1, \ldots, x_D)$ and the dependent variable (output) y.

Linear regression is a particular regression model which assumes (and learns) linear relationship between the inputs and the output:

$$\widehat{y} = h(\boldsymbol{x}) = w_0 + w_1 x_1 + \ldots + w_D x_D = w_0 + \langle \boldsymbol{w}, \boldsymbol{x} \rangle = w_0 + \boldsymbol{x}\boldsymbol{w}^T,$$

where

- $\widehat{y}$ is the model *prediction* (*estimate* of the true value y),
- $h(\boldsymbol{x})$ is the linear model (a *hypothesis*),
- $w_0, \ldots, w_D$ are the coefficients of the linear function, w_0 is the *bias*, organized in a row vector $\boldsymbol{w}$,
- $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ is a *dot product* of vectors $\boldsymbol{w}$ and $\boldsymbol{x}$ (scalar product),
- which can be also computed as a matrix product $\boldsymbol{x}\boldsymbol{w}^T$ if $\boldsymbol{w}$ and $\boldsymbol{x}$ are row vectors.

Notation remarks

Homogeneous coordinates: If we add "1" as the first element of $\boldsymbol{x}$ so that $\boldsymbol{x} = (1, x_1, \ldots, x_D)$, then we can write the linear model in an even simpler form (without the explicit bias term):

$$\widehat{y} = h(\boldsymbol{x}) = w_0 \cdot 1 + w_1 x_1 + \ldots + w_D x_D = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \boldsymbol{x}\boldsymbol{w}^T.$$

Notation remarks

Homogeneous coordinates: If we add "1" as the first element of $\boldsymbol{x}$ so that $\boldsymbol{x} = (1, x_1, \ldots, x_D)$, then we can write the linear model in an even simpler form (without the explicit bias term):

$$\widehat{y} = h(\boldsymbol{x}) = w_0 \cdot 1 + w_1 x_1 + \ldots + w_D x_D = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \boldsymbol{x}\boldsymbol{w}^T.$$

Matrix notation: If we organize the data into matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$, such that

$$\boldsymbol{X} = \begin{pmatrix} 1 & \boldsymbol{x}^{(1)} \\ \vdots & \vdots \\ 1 & \boldsymbol{x}^{(|T|)} \end{pmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(|T|)} \end{pmatrix},$$

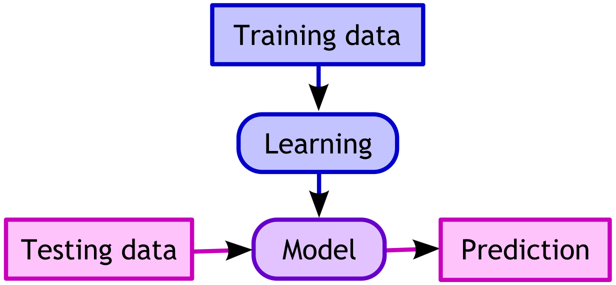
and similarly with $\widehat{\boldsymbol{y}}$, then we can write a batch computation of predictions for all data in $\boldsymbol{X}$ as

$$\widehat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w}^T.$$

Two operation modes

Any ML model has 2 operation modes:

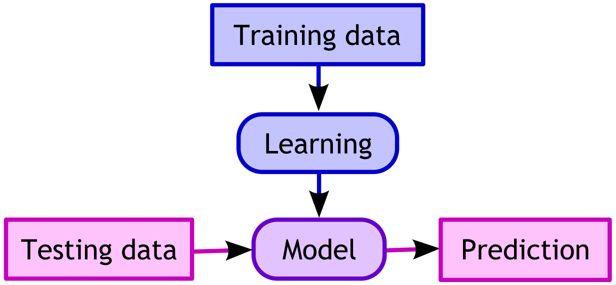
1. learning (training, fitting) and
2. application (testing, making predictions).



Two operation modes

Any ML model has 2 operation modes:

1. learning (training, fitting) and
2. application (testing, making predictions).

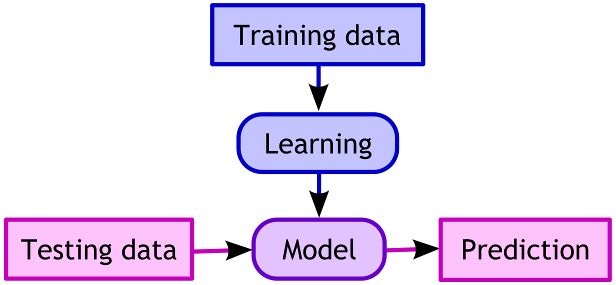


The model h can be viewed as a function of 2 variables: $h(\boldsymbol{x}, \boldsymbol{w})$.

Two operation modes

Any ML model has 2 operation modes:

1. learning (training, fitting) and
2. application (testing, making predictions).



The model h can be viewed as a function of 2 variables: $h(\boldsymbol{x}, \boldsymbol{w})$.

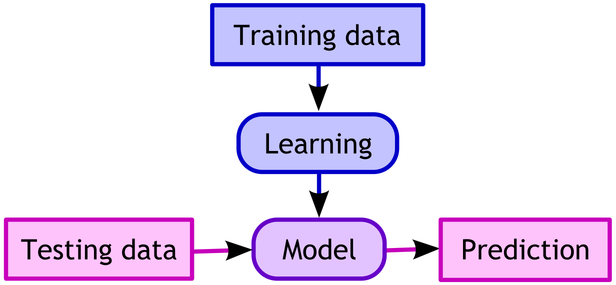
Model application: If the model is given ($\boldsymbol{w}$ is fixed), we can manipulate $\boldsymbol{x}$ to make predictions:

$$\widehat{y} = h(\boldsymbol{x}, \boldsymbol{w}) = h_{\boldsymbol{w}}(\boldsymbol{x}).$$

Two operation modes

Any ML model has 2 operation modes:

1. learning (training, fitting) and
2. application (testing, making predictions).



The model h can be viewed as a function of 2 variables: $h(\boldsymbol{x}, \boldsymbol{w})$.

Model application: If the model is given ($\boldsymbol{w}$ is fixed), we can manipulate $\boldsymbol{x}$ to make predictions:

$$\widehat{y} = h(\boldsymbol{x}, \boldsymbol{w}) = h_{\boldsymbol{w}}(\boldsymbol{x}).$$

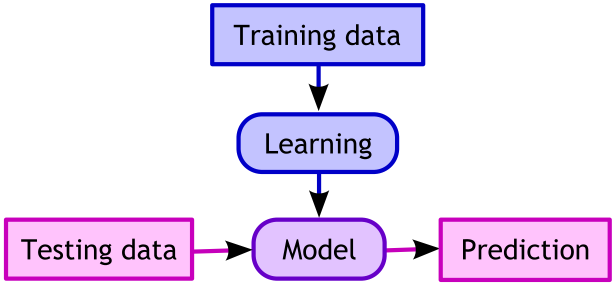
Model learning: If the data is given (T is fixed), we can manipulate the model parameters $\boldsymbol{w}$ to fit the model to the data:

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}} J(\boldsymbol{w}, T).$$

Two operation modes

Any ML model has 2 operation modes:

1. learning (training, fitting) and
2. application (testing, making predictions).



The model h can be viewed as a function of 2 variables: $h(\boldsymbol{x}, \boldsymbol{w})$.

Model application: If the model is given ($\boldsymbol{w}$ is fixed), we can manipulate $\boldsymbol{x}$ to make predictions:

$$\widehat{y} = h(\boldsymbol{x}, \boldsymbol{w}) = h_{\boldsymbol{w}}(\boldsymbol{x}).$$

Model learning: If the data is given (T is fixed), we can manipulate the model parameters $\boldsymbol{w}$ to fit the model to the data:

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}} J(\boldsymbol{w}, T).$$

How to train the model?

Simple (univariate) linear regression

Simple (univariate) regression deals with cases where $\boldsymbol{x}^{(i)} = x^{(i)}$, i.e. the examples are described by a single feature (they are 1-dimensional).

Simple (univariate) linear regression

Simple (univariate) regression deals with cases where $\boldsymbol{x}^{(i)} = x^{(i)}$, i.e. the examples are described by a single feature (they are 1-dimensional).

Fitting a line to data:

- find parameters w_0, w_1 of a linear model $\hat{y} = w_0 + w_1 x$
- given a traning (multi)set $T = \{(x^{(i)}, y^{(i)})\}_{i=1}^{|T|}$.

Simple (univariate) linear regression

Simple (univariate) regression deals with cases where $\boldsymbol{x}^{(i)} = x^{(i)}$, i.e. the examples are described by a single feature (they are 1-dimensional).

Fitting a line to data:

- find parameters w_0, w_1 of a linear model $\hat{y} = w_0 + w_1 x$
- given a traning (multi)set $T = \{(x^{(i)}, y^{(i)})\}_{i=1}^{|T|}$.

How to fit depending on the number of training examples:

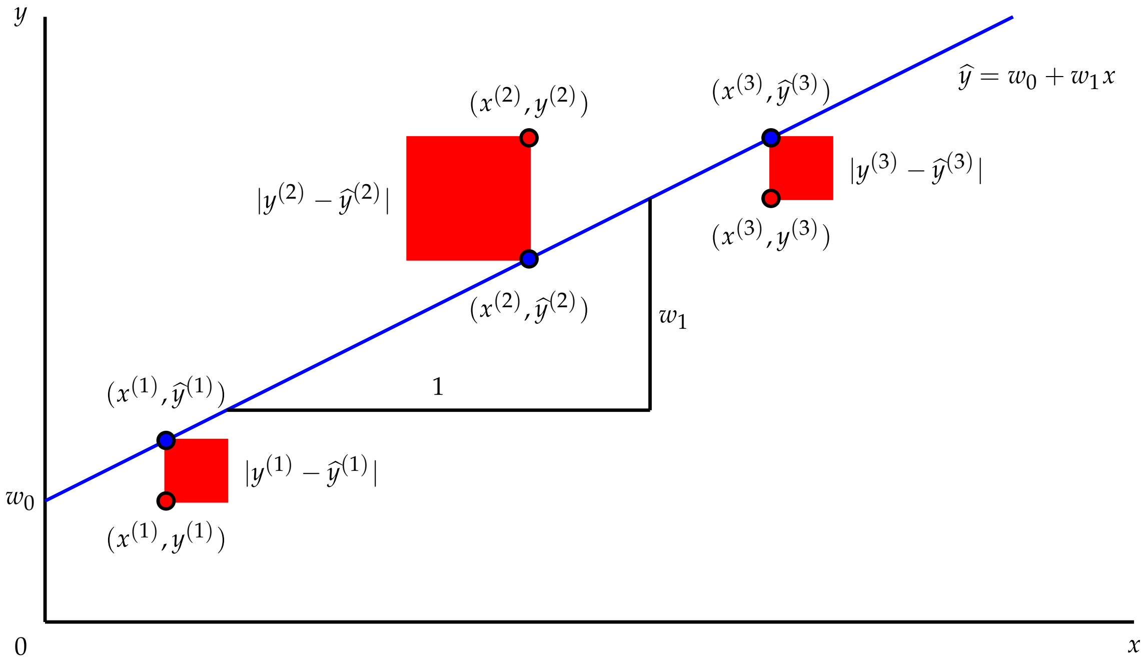
- Given a single example (1 equation, 2 parameters)
  $\Rightarrow$ infinitely many linear function can be fitted.
- Given 2 examples (2 equations, 2 parameters)
  $\Rightarrow$ exactly 1 linear function can be fitted.
- Given 3 or more examples ($>$ 2 equations, 2 parameters)
  $\Rightarrow$ no line can be fitted without any error
  $\Rightarrow$ a line which minimizes the "size" of error $y - \widehat{y}$ can be fitted:

$$\boldsymbol{w}^* = (w_0, w_1) = \operatorname*{argmin}_{w_0, w_1} J(w_0, w_1, T).$$

The least squares method

The **least squares method (LSM)** suggests to choose such parameters $\boldsymbol{w}$ which minimize the *mean squared error*

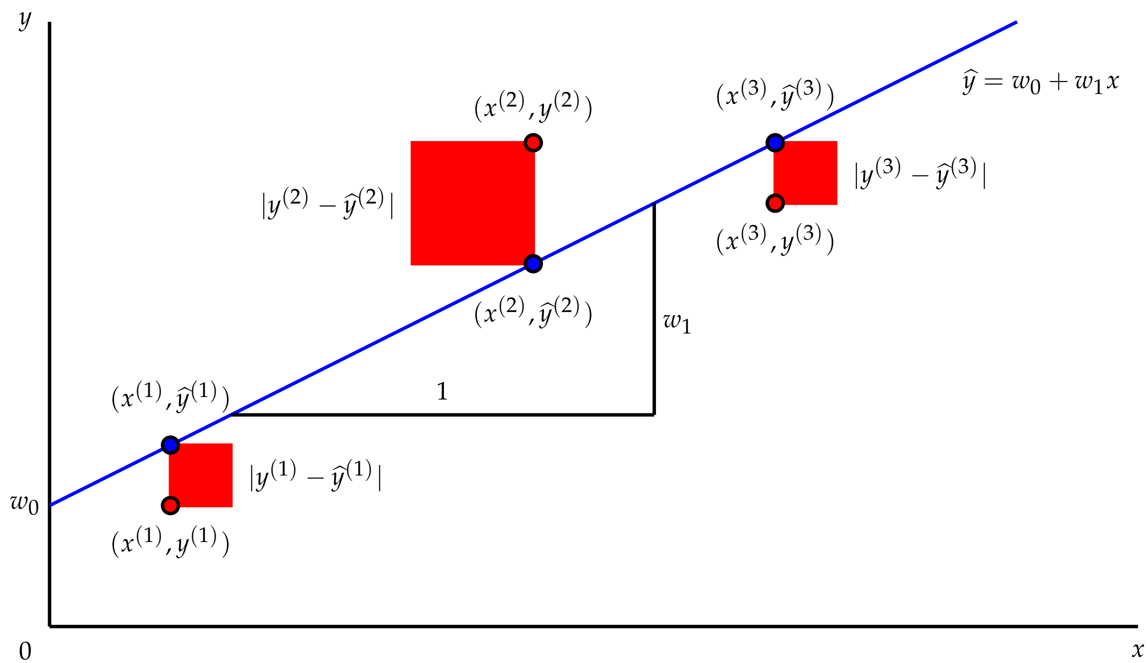
$$J(\boldsymbol{w}) = \frac{1}{|T|}\sum_{i=1}^{|T|}\left(y^{(i)} - \widehat{y}^{(i)}\right)^2 = \frac{1}{|T|}\sum_{i=1}^{|T|}\left(y^{(i)} - h_{\boldsymbol{w}}(\boldsymbol{x}^{(i)})\right)^2.$$



The least squares method

The **least squares method (LSM)** suggests to choose such parameters $\boldsymbol{w}$ which minimize the *mean squared error*

$$J(\boldsymbol{w}) = \frac{1}{|T|}\sum_{i=1}^{|T|}\left(y^{(i)} - \widehat{y}^{(i)}\right)^2 = \frac{1}{|T|}\sum_{i=1}^{|T|}\left(y^{(i)} - h_{\boldsymbol{w}}(\boldsymbol{x}^{(i)})\right)^2.$$

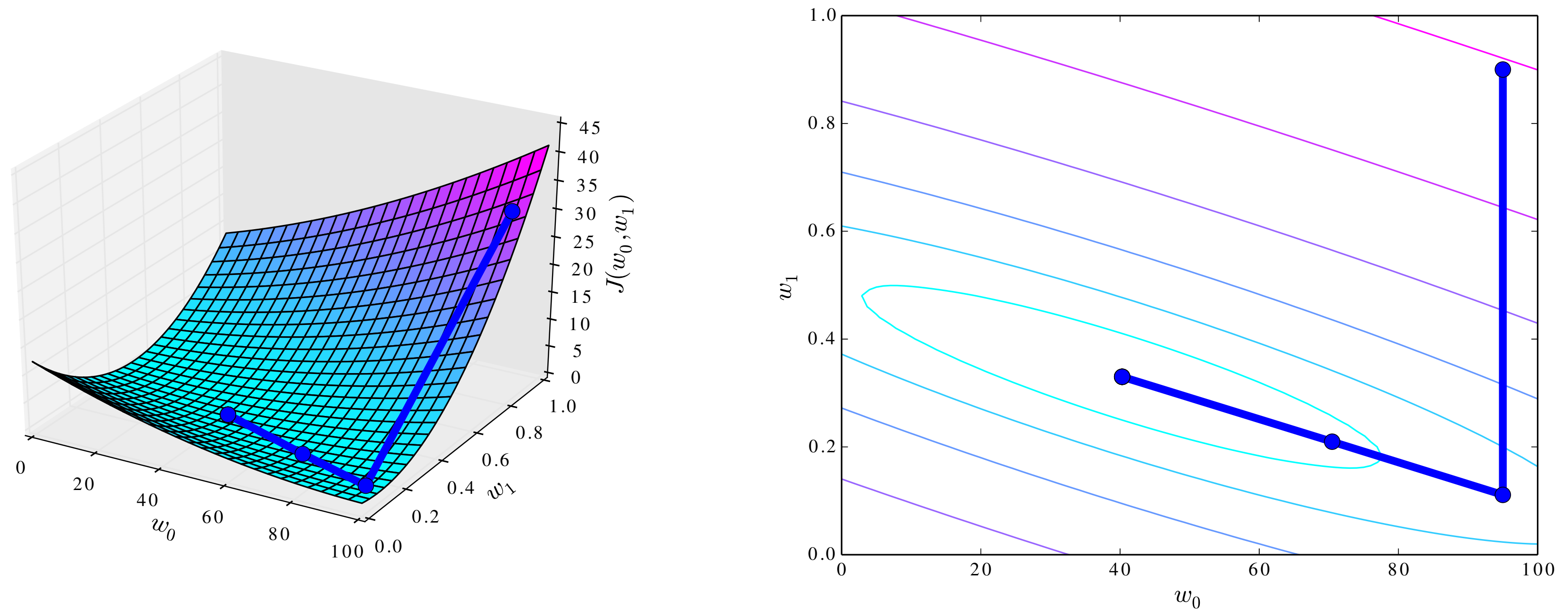


Explicit solution:

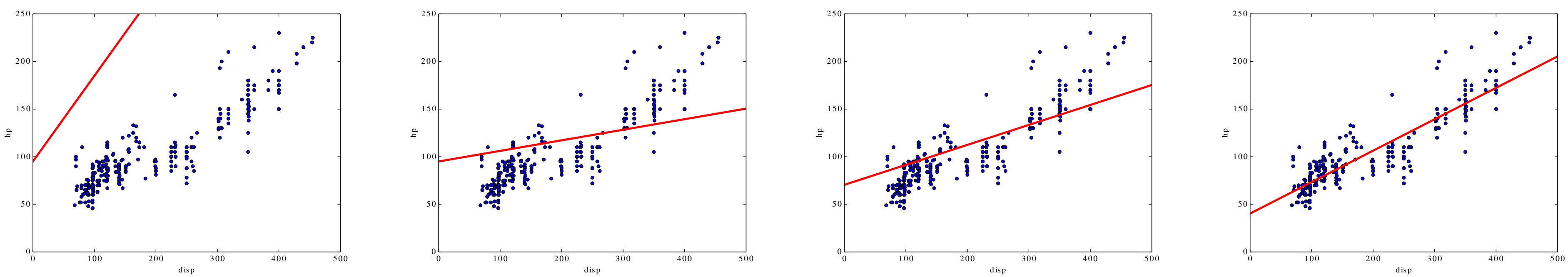
$$w_1 = \frac{\sum_{i=1}^{|T|}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^{|T|}(x^{(i)} - \bar{x})^2} = \frac{s_{xy}}{s_x^2} \qquad w_0 = \bar{y} - w_1\bar{x}$$

Universal fitting method: minimization of cost function J

The landscape of J in the space of w_0 and w_1:



Gradually better linear models found by an optimization method (BFGS):



Multivariate linear regression

Multivariate linear regression deals with cases where $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})$, i.e. the examples are described by more than 1 feature (they are D-dimensional).

Multivariate linear regression

Multivariate linear regression deals with cases where $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})$, i.e. the examples are described by more than 1 feature (they are D-dimensional).

Model fitting:

- find parameters $\boldsymbol{w} = (w_1, \ldots, w_D)$ of a linear model $\widehat{y} = \boldsymbol{x}\boldsymbol{w}^T$
- given the training (multi)set $T = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{|T|}$.
- The model is a *hyperplane* in the $D+1$-dimensional space.

Multivariate linear regression

Multivariate linear regression deals with cases where $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})$, i.e. the examples are described by more than 1 feature (they are D-dimensional).

Model fitting:

- find parameters $\boldsymbol{w} = (w_1, \ldots, w_D)$ of a linear model $\widehat{y} = \boldsymbol{x}\boldsymbol{w}^T$
- given the training (multi)set $T = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{|T|}$.
- The model is a *hyperplane* in the $D+1$-dimensional space.

Fitting methods:

1. Numeric optimization of $J(\boldsymbol{w}, T)$:
   - Works as for simple regression, it only searches a space with more dimensions.
   - Sometimes one need to tune some parameters of the optimization algorithm to work properly (learning rate in gradient descent, etc.).
   - May be slow (many iterations needed), but works even for very large D.
2. **Normal equation:**

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

   - Method to solve for the optimal $\boldsymbol{w}^*$ analytically!
   - No need to choose optimization algorithm parameters.
   - No iterations.
   - Needs to compute $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$, which is $O(D^3)$. Slow, or intractable, for large D.