Guided Tour of Machine Learning in Finance

Linear Regression

Igor Halperin

NYU Tandon School of Engineering, 2017

Linear Regression as a ML task

Task: predict a scalar value $y \in \mathbb{R}$ from a vector of predictors ("**features**") $\mathbf{X} = \left(X_0, X_1, \ldots, X_{D-1}\right) \in \mathbb{R}^D$ ($D \geq 1$ is the dimension of the feature space, or the number of predictors).

Linear Regression as a ML task

Task: predict a scalar value $y \in \mathbb{R}$ from a vector of predictors ("**features**") $\mathbf{X} = \left(X_0, X_1, \ldots, X_{D-1}\right) \in \mathbb{R}^D$ ($D \geq 1$ is the dimension of the feature space, or the number of predictors).

Given: a dataset $(\mathbf{X}, y)_{data} = \left[(\mathbf{X}_{train}, y_{train}), (\mathbf{X}_{test}, y_{test})\right] \sim p_{data}$

Linear Regression as a ML task

Task: predict a scalar value $y \in \mathbb{R}$ from a vector of predictors ("**features**") $\mathbf{X} = \left(X_0, X_1, \ldots, X_{D-1}\right) \in \mathbb{R}^D$ ($D \geq 1$ is the dimension of the feature space, or the number of predictors).

Given: a dataset $\left(\mathbf{X}, y\right)_{data} = \left[\left(\mathbf{X}_{train}, y_{train}\right), \left(\mathbf{X}_{test}, y_{test}\right)\right] \sim p_{data}$

Architecture: Linear

$$\hat{\mathbf{y}} = \mathbf{X}\,\mathbf{W}$$

N predicted values

$\mathbf{X}$ is a N x D design matrix

$\mathbf{W} \in \mathbb{R}^D$ is a vector of parameters

Linear Regression as a ML task

Task: predict a scalar value $y \in \mathbb{R}$ from a vector of predictors ("**features**") $\mathbf{X} = \left(X_0, X_1, \ldots, X_{D-1} \right) \in \mathbb{R}^D$ ($D \geq 1$ is the dimension of the feature space, or the number of predictors).

Given: a dataset $(\mathbf{X}, y)_{data} = \left[(\mathbf{X}_{train}, y_{train}), (\mathbf{X}_{test}, y_{test}) \right] \sim p_{data}$

Architecture: Linear

$$\hat{\mathbf{y}} = \mathbf{X}\,\mathbf{W}$$

N predicted values

$\mathbf{X}$
is a N x D design matrix

$\mathbf{W} \in \mathbb{R}^D$
is a vector of parameters

Example: $\mathbf{y}$ is a vector of N daily returns of AMZN, and $\mathbf{X}$ is a $N \times D$ design matrix made of $D-1$ daily market index returns (S&P 500, NASDAQ, VIX, etc.)

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test} \mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test} \mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$?

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$?
Minimize MSE on the **training set**!
Why? Because both MSE_{train} and MSE_{test} estimate the same generalization (expected) error $\mathbb{E}\left[\left(\hat{\mathbf{y}} - \mathbf{y} \right)^2 \right]$ from the empirical distribution $\sim p_{data}$.

Note: In ML, we often minimize one function, while actually caring about minimization of another function. This makes ML different from optimization that typically focuses only on one function.

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test} \mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.
For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train} \mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.
For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\mathbf{W} - \mathbf{y}^{train} \right) = 0$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\,\mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.

For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\,\mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right) = 0$$

Simplify notation:
$\mathbf{X}^{train} \rightarrow \mathbf{X},\ \mathbf{y}^{train} \rightarrow \mathbf{y}$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\,\mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.
For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\,\mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right) = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{W}^T \mathbf{X}^T \mathbf{X}\,\mathbf{W} - 2\mathbf{W}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right)$$

Simplify notation:
$\mathbf{X}^{train} \to \mathbf{X},\ \mathbf{y}^{train} \to \mathbf{y}$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\,\mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.
For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\,\mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\,\mathbf{W} - \mathbf{y}^{train} \right) = 0$$

Simplify notation:
$\mathbf{X}^{train} \rightarrow \mathbf{X},\ \mathbf{y}^{train} \rightarrow \mathbf{y}$

$$\Rightarrow \nabla_W \left(\mathbf{W}^T \mathbf{X}^T \mathbf{X}\,\mathbf{W} - 2\mathbf{W}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right) = 2\left(\mathbf{X}^T \mathbf{X} \right) \mathbf{W} - 2\left(\mathbf{X}^T \mathbf{y} \right) = 0$$

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\, \mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.

For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\, \mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\, \mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\, \mathbf{W} - \mathbf{y}^{train} \right) = 0$$

Simplify notation:
$\mathbf{X}^{train} \to \mathbf{X},\ \mathbf{y}^{train} \to \mathbf{y}$

$$\Rightarrow \nabla_W \left(\mathbf{W}^T \mathbf{X}^T \mathbf{X}\, \mathbf{W} - 2\mathbf{W}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right) = 2\left(\mathbf{X}^T \mathbf{X} \right) \mathbf{W} - 2\left(\mathbf{X}^T \mathbf{y} \right) = 0$$

$$\Rightarrow \boxed{\mathbf{W} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y}}$$

Normal equation

Learning in Linear Regression

Performance measure P : mean square error (**MSE**) on the **test set**:

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\hat{y}_i^{test} - y_i^{test} \right)^2 = \frac{1}{N_{test}} \left\| \hat{\mathbf{Y}}^{test} - \mathbf{Y}^{test} \right\|_2^2 = \frac{1}{N_{test}} \left\| \mathbf{X}^{test}\, \mathbf{W} - \mathbf{Y}^{test} \right\|_2^2$$

How to optimize parameters $\mathbf{W} \in \mathbb{R}^D$? Minimize MSE on the **training set**! Why? Because both MSE_{test} and MSE_{train} estimate the generalization (expected) error.
For optimal weights $\mathbf{W}$, the gradient of MSE_{train} should be 0:

$$\nabla_W \mathrm{MSE}_{train} = \nabla_W \frac{1}{N_{train}} \left\| \hat{\mathbf{Y}}^{train} - \mathbf{Y}^{train} \right\|_2^2 = \frac{1}{N_{train}} \nabla_W \left\| \mathbf{X}^{train}\, \mathbf{W} - \mathbf{Y}^{train} \right\|_2^2 = 0$$

$$\Rightarrow \nabla_W \left(\mathbf{X}^{train}\, \mathbf{W} - \mathbf{y}^{train} \right)^T \left(\mathbf{X}^{train}\, \mathbf{W} - \mathbf{y}^{train} \right) = 0$$

Simplify notation:
$\mathbf{X}^{train} \to \mathbf{X},\ \mathbf{y}^{train} \to \mathbf{y}$

$$\Rightarrow \nabla_W \left(\mathbf{W}^T \mathbf{X}^T \mathbf{X}\, \mathbf{W} - 2\mathbf{W}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right) = 2\left(\mathbf{X}^T \mathbf{X} \right) \mathbf{W} - 2\left(\mathbf{X}^T \mathbf{y} \right) = 0$$

Here
$\mathbf{X} = \mathbf{X}^{train},\ \mathbf{y} = \mathbf{y}^{train}$

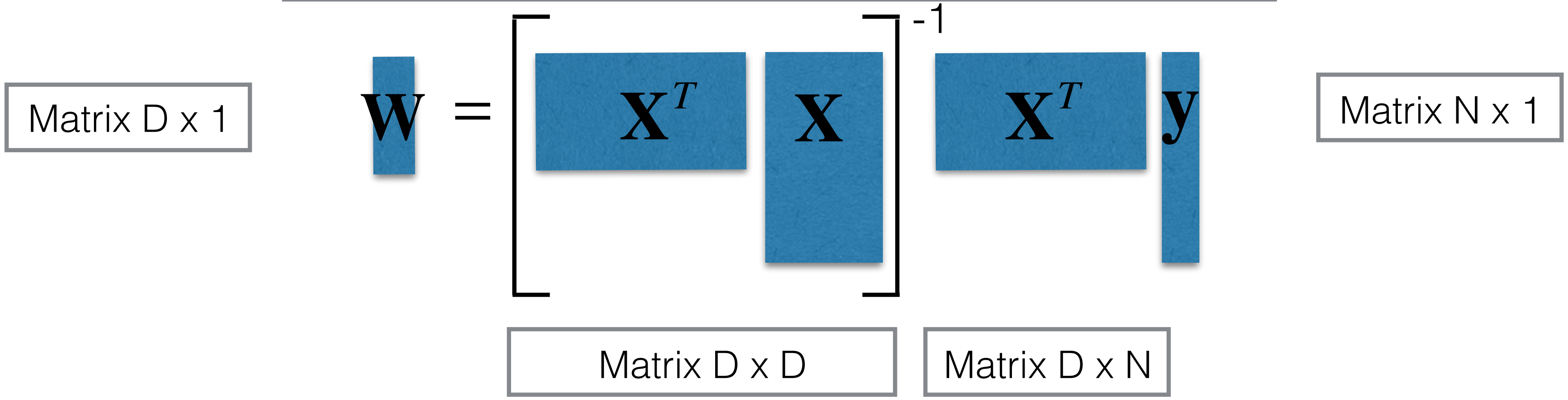
$$\Rightarrow \quad \boxed{\mathbf{W} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y}}$$

Normal equation

Learning in Linear Regression

Visualize the result for optimal weights $\mathbf{W}$ in the vector form

$$\boxed{\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}}$$



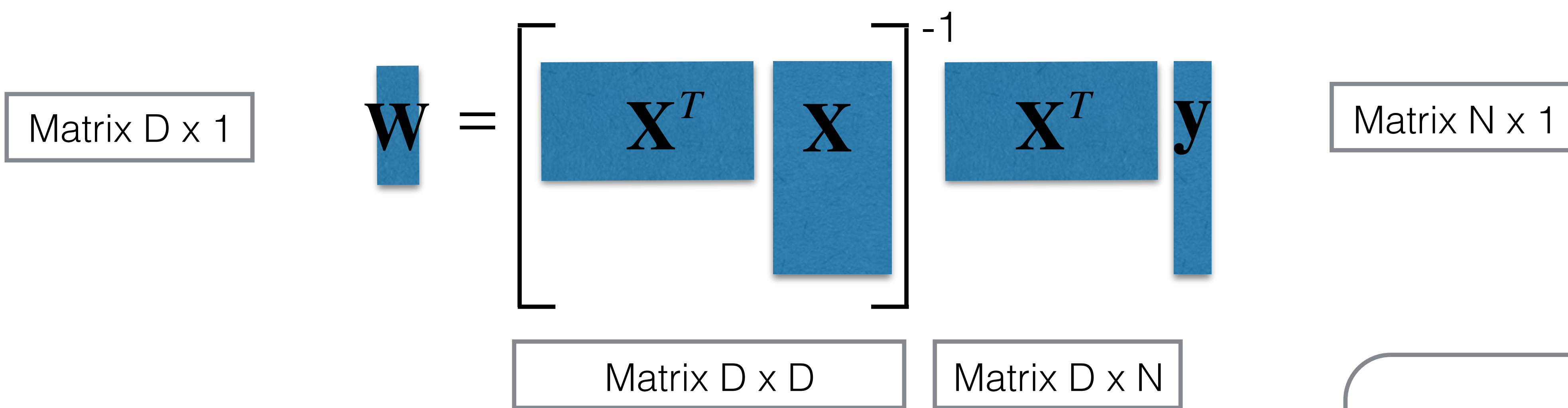
Prediction in-sample: $\hat{\mathbf{y}} = \mathbf{X}\mathbf{W} = \underbrace{\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T}_{=\mathbf{H}}\mathbf{y} \equiv \mathbf{H}\,\mathbf{y}$

The “hat” (projection) matrix

Learning in Linear Regression

Visualize the result for optimal weights $\mathbf{W}$ in the vector form

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$



Prediction in-sample:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{W} = \underbrace{\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T}_{=\mathbf{H}}\mathbf{y} \equiv \mathbf{H}\mathbf{y}$$

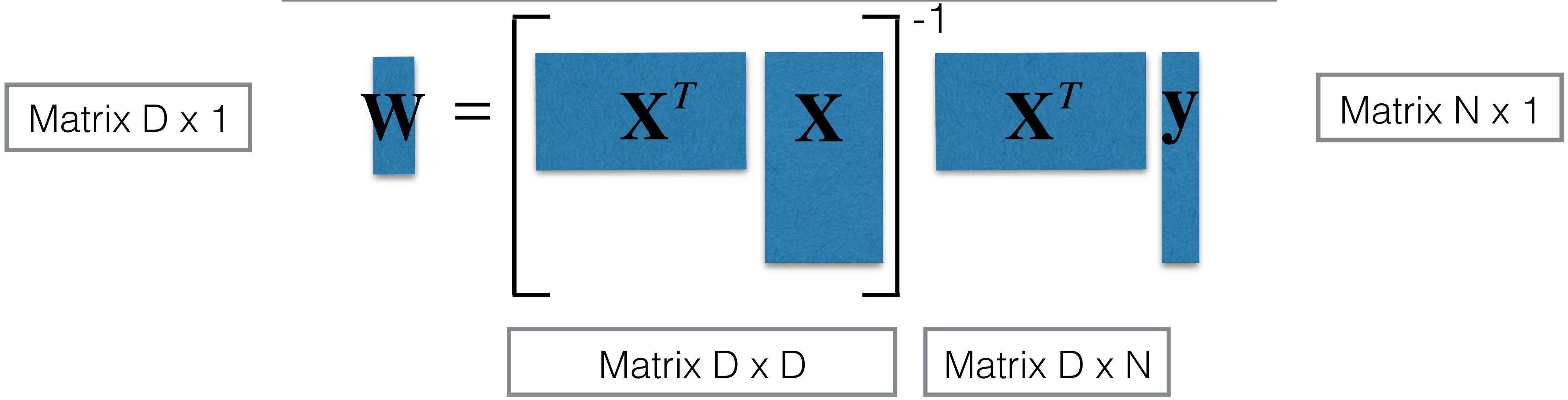
Symmetric and idempotent: $\mathbf{H}^2 = \mathbf{H}$

The "hat" (projection) matrix

Learning in Linear Regression

Visualize the result for optimal weights $\mathbf{W}$ in the vector form

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$



Prediction out-of-sample: $\hat{\mathbf{y}} = \mathbf{X}^{test}\mathbf{W}$

How to overfit with Linear Regression: add more features to the design matrix without a proper control!