Linear Regression

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \text{ from} \]

\[ x = \{\# \text{ sq. ft., zip code, date of sale, etc.}\} \]

Training Data:

\[ \{(x_i, y_i)\}_{i=1}^{n} \]

\[ x_i \in \mathbb{R}^d \]

\[ y_i \in \mathbb{R} \]
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \quad \text{from} \]

\[ x = \{\text{# sq. ft., zip code, date of sale, etc.} \} \]

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^n \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_{w} \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]

\[ = \arg \min_{w} (y - X w)^T (y - X w) \]

\[ y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \]
The regression problem in matrix notation

\[ \hat{\mathbf{w}}_{LS} = \arg \min_{\mathbf{w}} \| \mathbf{y} - \mathbf{Xw} \|^2_2 \]

\[ = \arg \min_{\mathbf{w}} (\mathbf{y} - \mathbf{Xw})^T (\mathbf{y} - \mathbf{Xw}) \]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_w \|y - Xw\|_2^2 \]
\[ = (X^T X)^{-1} X^T y \]

What about an offset?

\[ \hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w,b} \sum_{i=1}^{n} (y_i - (x_i^T w + b))^2 \]
\[ = \arg \min_{w,b} \|y - (Xw + 1b)\|_2^2 \]
Dealing with an offset

$$\hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w,b} \| y - (Xw + 1b) \|^2_2$$
Dealing with an offset

\[ \hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w,b} \|y - (Xw + 1b)\|_2^2 \]

\[ X^T X \hat{w}_{LS} + \hat{b}_{LS} X^T 1 = X^T y \]
\[ 1^T X \hat{w}_{LS} + \hat{b}_{LS} 1^T 1 = 1^T y \]

If \( X^T 1 = 0 \) (i.e., if each feature is mean-zero) then

\[ \hat{w}_{LS} = (X^T X)^{-1} X^T Y \]

\[ \hat{b}_{LS} = \frac{1}{n} \sum_{i=1}^{n} y_i \]
The regression problem in matrix notation

\[ \hat{w}_{LS} = \arg \min_w ||y - Xw||^2 \]
\[ = (X^T X)^{-1} X^T y \]

But why least squares?

Consider \( y_i = x_i^T w + \epsilon_i \) where \( \epsilon_i \sim i.i.d. \mathcal{N}(0, \sigma^2) \)

\[ P(y|x, w, \sigma) = \]
Maximizing log-likelihood

Maximize:

$$\log P(D|w, \sigma) = \log\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \prod_{i=1}^{n} e^{-\frac{(y_i - x_i^T w)^2}{2\sigma^2}}$$
MLE is LS under linear model

\[ \hat{w}_{LS} = \arg \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]

\[ \hat{w}_{MLE} = \arg \max_w P(D | w, \sigma) \]

if \( y_i = x_i^T w + \epsilon_i \) and \( \epsilon_i \ i.i.d. \sim \mathcal{N}(0, \sigma^2) \)

\[ \hat{w}_{LS} = \hat{w}_{MLE} = (X^T X)^{-1} X^T Y \]
The regression problem

Given past sales data on zillow.com, predict:

\[ y = \text{House sale price} \quad \text{from} \]

\[ x = \{\# \text{ sq. ft., zip code, date of sale, etc.}\} \]

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^{n} \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]
The regression problem

Given past sales data on zillow.com, predict:
\[ y = \text{House sale price} \quad \text{from} \quad x = \{\# \text{ sq. ft., zip code, date of sale, etc.}\} \]

Training Data:
\[ \left\{ (x_i, y_i) \right\}_{i=1}^{n} \]

Hypothesis: linear
\[ y_i \approx x_i^T w \]

Loss: least squares
\[ \min_{w} \sum_{i=1}^{n} (y_i - x_i^T w)^2 \]
The regression problem

Training Data: \( \{ (x_i, y_i) \} \}_{i=1}^n \)

Hypothesis: linear

\( y_i \approx x_i^T w \)

Loss: least squares

\[
\min_w \sum_{i=1}^n \left( y_i - x_i^T w \right)^2
\]
The regression problem

Training Data: \( \{ (x_i, y_i) \}_{i=1}^n \)
- \( x_i \in \mathbb{R}^d \)
- \( y_i \in \mathbb{R} \)

Hypothesis: linear
- \( y_i \approx x_i^T w \)

Loss: least squares
- \( \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \)

Transformed data:
- \( h : \mathbb{R}^d \rightarrow \mathbb{R}^p \) maps original features to a rich, possibly high-dimensional space
- in \( d=1 \): \( h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} = \begin{bmatrix} x \\ x^2 \\ \vdots \\ x^p \end{bmatrix} \)
- for \( d>1 \), generate \( \{u_j\}_{j=1}^p \subset \mathbb{R}^d \)
  - \( h_j(x) = \frac{1}{1 + \exp(u_j^T x)} \)
  - \( h_j(x) = (u_j^T x)^2 \)
  - \( h_j(x) = \cos(u_j^T x) \)
The regression problem

Training Data: \[ \{(x_i, y_i)\}_{i=1}^n \]

Hypothesis: linear

\[ y_i \approx x_i^T w \]

Loss: least squares

\[ \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \]

Transformed data:

\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear

\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares

\[ \min_w \sum_{i=1}^n (y_i - h(x_i)^T w)^2 \]
The regression problem

Training Data:
\( \{(x_i, y_i)\}_{i=1}^{n} \)

\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]

Transformed data:
\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear
\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} (y_i - h(x_i)^T w)^2 \]

Sale Price

best linear fit

date of sale

date of sale

Sale Price
The regression problem

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^{n} \]

\[ x_i \in \mathbb{R}^d \]

\[ y_i \in \mathbb{R} \]

Transformed data:
\[ h(x) = \left[ \begin{array}{c} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{array} \right] \]

Hypothesis: linear
\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares
\[ \min_w \sum_{i=1}^{n} \left( y_i - h(x_i)^T w \right)^2 \]
The regression problem

Training Data:
\[ \{(x_i, y_i)\}_{i=1}^n \]
\[ x_i \in \mathbb{R}^d \]
\[ y_i \in \mathbb{R} \]

Transformed data:
\[ h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_p(x) \end{bmatrix} \]

Hypothesis: linear
\[ y_i \approx h(x_i)^T w \quad w \in \mathbb{R}^p \]

Loss: least squares
\[ \min_w \sum_{i=1}^n (y_i - h(x_i)^T w)^2 \]

What's going on here?
Bias-Variance Tradeoff

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
Statistical Learning

\[ P_{XY}(X = x, Y = y) \]
Statistical Learning

\[ P_{XY}(X = x, Y = y) \]

\[ P_{XY}(Y = y | X = x_0) \]

\[ P_{XY}(Y = y | X = x_1) \]
Statistical Learning

Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y | X = x]$$

$$P_{XY}(Y = y | X = x_0)$$

$$P_{XY}(Y = y | X = x_1)$$
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

But we only have samples:

$$(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n$$
Ideally, we want to find:

$$\eta(x) = \mathbb{E}_{XY}[Y|X = x]$$

But we only have samples:

$$(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n$$

and are restricted to a function class (e.g., linear) so we compute:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$
Ideally, we want to find:
\[ \eta(x) = \mathbb{E}_{XY}[Y|X = x] \]

But we only have samples:
\[(x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n\]

and are restricted to a function class (e.g., linear) so we compute:
\[ \widehat{f} = \operatorname*{arg\,min}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

We care about future predictions: \[ \mathbb{E}_{XY}[(Y - \widehat{f}(X))^2] \]
Ideally, we want to find:
\[ \eta(x) = \mathbb{E}_{XY}[Y|X = x] \]

But we only have samples:
\[ (x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n \]

and are restricted to a function class (e.g., linear) so we compute:
\[ \hat{f} = \arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

Each draw \( \mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n} \) results in different \( \hat{f} \)
Ideally, we want to find:
\[ \eta(x) = \mathbb{E}_{XY}[Y|X = x] \]

But we only have samples:
\[ (x_i, y_i) \overset{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \ldots, n \]

and are restricted to a function class (e.g., linear) so we compute:
\[ \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

Each draw \( D = \{(x_i, y_i)\}_{i=1}^{n} \) results in different \( \hat{f} \)
Bias-Variance Tradeoff

\[
\eta(x) = \mathbb{E}_{XY}[Y|X = x] \quad \quad \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2
\]

\[
\mathbb{E}_{Y|X=x}[\mathbb{E}_{D}[(Y - \hat{f}_D(x))^2]] = \mathbb{E}_{Y|X=x}[\mathbb{E}_{D}[(Y - \eta(x) + \eta(x) - \hat{f}_D(x))^2]]
\]
Bias-Variance Tradeoff

\[
\eta(x) = \mathbb{E}_{X,Y}[Y|X = x]
\]

\[
\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2
\]

\[
\mathbb{E}_{X,Y}[\mathbb{E}_{D}[(Y - \hat{f}_{D}(x))^2]|X = x] = \mathbb{E}_{X,Y}[\mathbb{E}_{D}[(Y - \eta(x) + \eta(x) - \hat{f}_{D}(x))^2]|X = x]
\]

\[
= \mathbb{E}_{X,Y}[\mathbb{E}_{D}[(Y - \eta(x))^2 + 2(Y - \eta(x))(\eta(x) - \hat{f}_{D}(x))]
\]

\[
+ (\eta(x) - \hat{f}_{D}(x))^2]|X = x]
\]

\[
= \mathbb{E}_{X,Y}[(Y - \eta(x))^2|X = x] + \mathbb{E}_{D}[(\eta(x) - \hat{f}_{D}(x))^2]
\]

**irreducible error**
Caused by stochastic label noise

**learning error**
Caused by either using too “simple” of a model or not enough data to learn the model accurately

©2017 Kevin Jamieson
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{XY}[Y|X = x] \quad \hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_{D}[(\eta(x) - \hat{f}_{D}(x))^2] = \mathbb{E}_{D}[(\eta(x) - \mathbb{E}_{D}[\hat{f}_{D}(x)]) + \mathbb{E}_{D}[\hat{f}_{D}(x)] - \hat{f}_{D}(x))^2] \]
Bias-Variance Tradeoff

\[ \eta(x) = \mathbb{E}_{XY}[Y | X = x] \]

\[ \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

\[ \mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2] = \mathbb{E}_D[(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)] + \mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]

\[ = \mathbb{E}_D[(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + 2(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x)) \]

\[ + (\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]

\[ = (\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + \mathbb{E}_D[(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2] \]

**biased squared** \quad **variance**
Bias-Variance Tradeoff

\[
\mathbb{E}_{XY} \left[ \mathbb{E}_D \left[ (Y - \hat{f}_D(x))^2 \right] | X = x \right] = \mathbb{E}_{XY} \left[ (Y - \eta(x))^2 | X = x \right]
\]

irreducible error

\[
+ (\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + \mathbb{E}_D[(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2]
\]

biased squared variance

Model too simple → high bias, cannot fit well to data

Model too complex → high variance, small changes in data change learned function a lot
Bias-Variance Tradeoff

$$\mathbb{E}_{XY} [\mathbb{E}_D [(Y - \hat{f}_D(x))^2 | X = x] = \mathbb{E}_{XY} [(Y - \eta(x))^2 | X = x]$$

**irreducible error**

$$+ (\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])^2 + \mathbb{E}_D[(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2]$$

**biased squared**

**variance**
Overfitting

Machine Learning – CSE546
Kevin Jamieson
University of Washington

Oct 5, 2017
Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class $\rightarrow$ less bias
  - More complex class $\rightarrow$ more variance

- But in practice??
Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class $\rightarrow$ less bias
  - More complex class $\rightarrow$ more variance

- But in practice??

- Before we saw how increasing the feature space can increase the complexity of the learned estimator:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots$$

$$\hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

Complexity grows as $k$ grows.
Training set error as a function of model complexity

\[ \mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad \mathcal{D} \sim \text{i.i.d.} \quad P_{XY} \]

\[ \hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2 \]

**TRAIN error:**

\[
\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_D^{(k)}(x_i))^2
\]

**TRUE error:**

\[
\mathbb{E}_{XY} [(Y - \hat{f}_D^{(k)}(X))^2]
\]
Training set error as a function of model complexity

\[ \mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad D \overset{i.i.d.}{\sim} P_{XY} \]

\[ \hat{f}_D^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - f(x_i))^2 \]

**TRAIN error:**

\[ \frac{1}{|D|} \sum_{(x_i, y_i) \in D} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

**TRUE error:**

\[ \mathbb{E}_{XY} [(Y - \hat{f}_D^{(k)}(X))^2] \]

**TEST error:**

\[ T \overset{i.i.d.}{\sim} P_{XY} \]

\[ \frac{1}{|T|} \sum_{(x_i, y_i) \in T} (y_i - \hat{f}_D^{(k)}(x_i))^2 \]

Important: \( D \cap T = \emptyset \)
Training set error as a function of model complexity

\[ \mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad \mathcal{D} \overset{i.i.d.}{\sim} P_{XY} \]

\[ \hat{f}^{(k)}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2 \]

**TRAIN error:**
\[ \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}^{(k)}_{\mathcal{D}}(x_i))^2 \]

**TRUE error:**
\[ \mathbb{E}_{XY} [(Y - \hat{f}^{(k)}_{\mathcal{D}}(X))^2] \]

**TEST error:**
\[ \mathcal{T} \overset{i.i.d.}{\sim} P_{XY} \quad \frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}^{(k)}_{\mathcal{D}}(x_i))^2 \]

Important: \( \mathcal{D} \cap \mathcal{T} = \emptyset \)
Training set error as a function of model complexity

$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots \quad \mathcal{D} \overset{i.i.d.}{\sim} \mathcal{P}_{XY}$

$\hat{f}_{D}^{(k)} = \arg \min_{f \in \mathcal{F}_k} \frac{1}{|\mathcal{D}|} \sum_{(x_i,y_i) \in \mathcal{D}} (y_i - f(x_i))^2$

**TRAIN error:**

$\frac{1}{|\mathcal{D}|} \sum_{(x_i,y_i) \in \mathcal{D}} (y_i - \hat{f}_{D}^{(k)}(x_i))^2$

**TRUE error:**

$\mathbb{E}_{XY}[(Y - \hat{f}_{D}^{(k)}(X))^2]$

**TEST error:**

$\mathcal{T} \overset{i.i.d.}{\sim} \mathcal{P}_{XY}$

$\frac{1}{|\mathcal{T}|} \sum_{(x_i,y_i) \in \mathcal{T}} (y_i - \hat{f}_{D}^{(k)}(x_i))^2$

**Important:** $\mathcal{D} \cap \mathcal{T} = \emptyset$

**TRAIN error** is optimistically biased because it is evaluated on the data it trained on. **TEST error** is unbiased only if $\mathcal{T}$ is never used to train the model or even pick the complexity $k$. 
Test set error

- Given a dataset, **randomly** split it into two parts:
  - Training data: $\mathcal{D}$
  - Test data: $\mathcal{T}$

- Use **training data** to learn predictor
  - e.g.,
    $$\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \hat{f}_{\mathcal{D}}^{(k)}(x_i))^2$$
  - use **training data** to pick complexity $k$ (next lecture)

- Use **test data** to report predicted performance

  $$\frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \hat{f}_{\mathcal{T}}^{(k)}(x_i))^2$$
Overfitting: a learning algorithm overfits the training data if it outputs a solution $\mathbf{w}$ when there exists another solution $\mathbf{w}'$ such that:

$$[\text{error}_{\text{train}}(\mathbf{w}) < \text{error}_{\text{train}}(\mathbf{w}')] \land [\text{error}_{\text{true}}(\mathbf{w}') < \text{error}_{\text{true}}(\mathbf{w})]$$
How many points do I use for training/testing?

- Very hard question to answer!
  - Too few training points, learned model is bad
  - Too few test points, you never know if you reached a good solution

- Bounds, such as Hoeffding’s inequality can help:

\[ P(|\hat{\theta} - \theta^*| \geq \epsilon) \leq 2e^{-2N\epsilon^2} \]

- More on this later this quarter, but still hard to answer

- Typically:
  - If you have a reasonable amount of data 90/10 splits are common
  - If you have little data, then you need to get fancy (e.g., bootstrapping)
Recap

- Learning is...
  - Collect some data
    - E.g., housing info and sale price
  - Randomly split dataset into TRAIN and TEST
    - E.g., 80% and 20%, respectively
  - Choose a hypothesis class or model
    - E.g., linear
  - Choose a loss function
    - E.g., least squares
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain estimator
  - Justifying the accuracy of the estimate
    - E.g., report TEST error