CS 559: Linear Regression

Lecture 4

In Jang

ijang@stevens.edu

Outline:

- Regression - Supervised Learning
- Linear Regression
  - Example
  - Linear Regression Result Interpretation
  - Linear Regression Limitation
- Empirical Risk
  - Lasso and Ridge
  - Overfitting vs. Underfitting
  - Bias-Variance Tradeoff

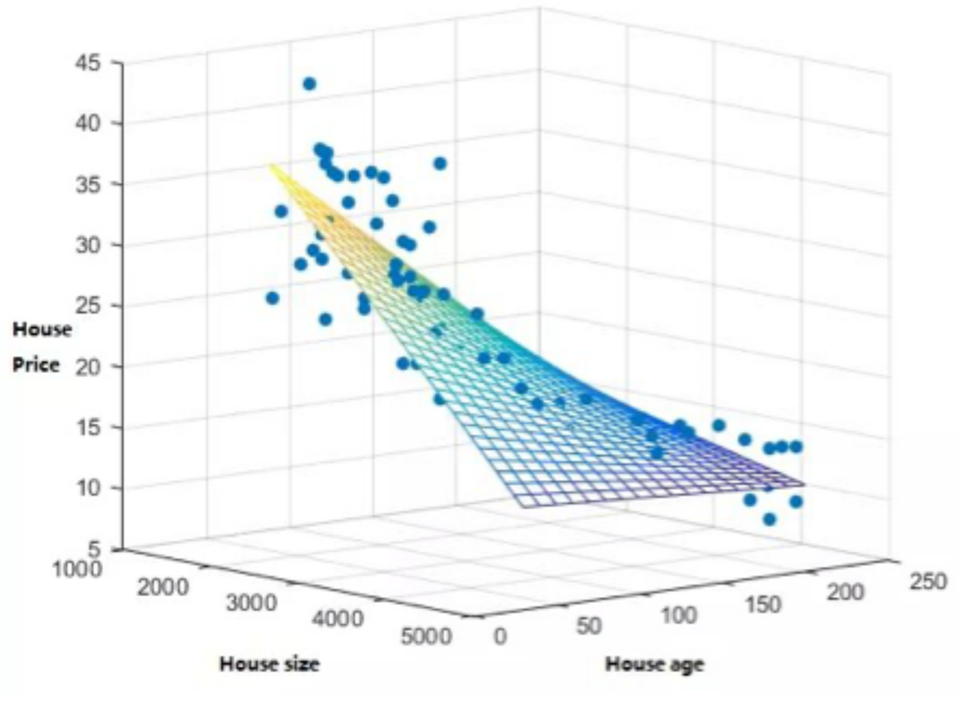
Linear Regression

Regression

- Regression: a ML technique to summarize relationships between continuous variables
  - One is regarded as response, outcome, dependent, or target variable
  - The rest is regarded as predictor(s), explanatory(ies), features(s) or independent variable(s)
- Simple linear regression
  - Simple: study only one predictor variable (one target – one sample)
  - Linear: the relationship is linear and features are independent to each other.
- The ML model to predict the response

$$y \approx H(w,x) = w_0 + w_1 x$$

where y is the observed outcome, e.g., target variable, x is the explanatory variable, $w = [w_0, w_1]^T$ is the model parameters to be determined from ML training, and *H(w,x)* is the linear regression model that would give the predicted target variable.



Linear Regression

The linear regression model involves a **linear combination** of the input variables.

- The simplest model

Formation:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \cdots + w_n x_n \qquad (1)$$

- Assume we have a vector with D features, $\boldsymbol{x} = (x_1, \cdots, x_D)^T$
- The key terms are the parameters $\boldsymbol{w} = (w_0, w_1, \cdots, w_D)$, also known as weight.
- The predicted value function $y(\boldsymbol{x}, \boldsymbol{w})$ predicts the target variable, t. Then

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon$$

where ϵ is a noise that has a zero mean Gaussian random variable with *inverse variance* β, $\mathcal{N}(0, \beta^{-1})$.

Linear Regression

Eqn. (1) imposes significant limitations on the model.
Often, we extend (1) by considering linear combinations of fixed nonlinear functions of $\boldsymbol{x}$ using basis function, $\phi_j(\boldsymbol{x})$:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1\phi_1 + \cdots + w_D\phi_n = w_0 + \sum_{j=1}^{M-1} w_j\phi_j(\boldsymbol{x}) \qquad (2)$$

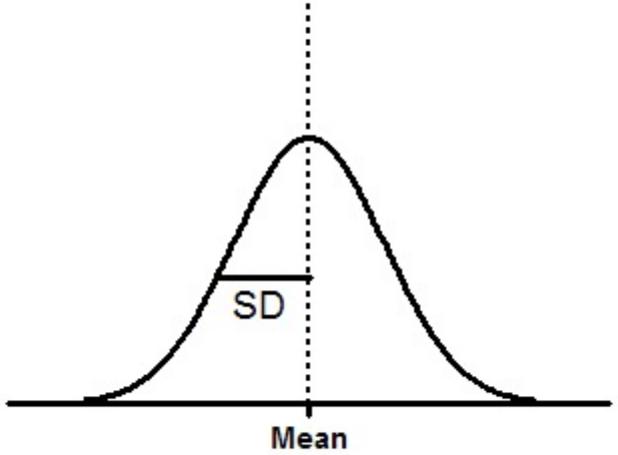
- If we let $\phi_0(\boldsymbol{x}) = 1$, then (2) becomes

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} w_j\phi_j(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{x})$$

- Is this function linear or non-linear?
  - Non-linear function ϕ
  - But the function is linear in $\boldsymbol{w}$**.**

Gaussian Distribution

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



Normalized: $\int_{-\infty}^{\infty}\mathcal{N}(x|\mu,\sigma^2)dx = 1$

Average: $\mathbb{E}[x] = \int_{-\infty}^{\infty}\mathcal{N}(x|\mu,\sigma^2)xdx = \mu$

Second order moment: $\mathbb{E}[x^2] = \int_{-\infty}^{\infty}\mathcal{N}(x|\mu,\sigma^2)x^2dx = \mu^2+\sigma^2$

Variance: $\mathbb{E}[x^2] - \mathbb{E}[x] = \sigma^2$

Probability: $p(x|\mu,\sigma^2) = \prod_{n=1}^{N}\mathcal{N}(x|\mu,\sigma^2)$

The log likelihood function:

$$\ln p(x|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi) \qquad (3)$$

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N}x_n$$

$$\sigma_{ML}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{ML})^2$$

Linear Regression

The target t probability is then

$$p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1})$$

Consider a data set:
Input: $\boldsymbol{X} = \{\boldsymbol{x_1}, \cdots, \boldsymbol{x_N}\}$
Target: $\boldsymbol{t} = \{t_1, \cdots, t_N\}$
t is from a single observation and each data points are drawn independently from the distribution.
The likelihood function

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \middle| \boldsymbol{w^T} \boldsymbol{\phi}(\boldsymbol{x_n}), \beta^{-1}\right)$$

So (3) simply because (4)

$$\ln p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \frac{\beta}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{w^T} \boldsymbol{\phi}(\boldsymbol{x_n})\}^2$$

where $\beta = \sigma^{-2}$.

Estimation with MLE

- Independence & chain rule of probability.
- x_i is independent of w & $P(x_i)$ is constant and can be dropped.
- $P(t_i, x_i|w) = P(t_i|x_i, w)P(x_i|w) = P(t_i|x_i, w)P(x_i) = P(t_i|x_i, w)$

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P((t_1, x_1), \ldots, (t_n, x_n)|\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(t_i, x_i|w)$$

- Log is a monotonic function

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(t_i|x_i, \boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log[P(t_i|x_i, w)]$$

- $P \sim N(x^T w, \sigma^2)$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \log\left(\exp\left(-\frac{\left(x_i^T w - t_i\right)^2}{2\sigma^2}\right)\right)\right]$$

- The 1st term is a constant

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(x_i^T w - t_i\right)^2 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(x_i^T w - t_i\right)^2$$

- $\frac{1}{n}$ makes the lost interpretable (average squared error).

* This function is also known as the squared loss or Ordinary Least Square (OLS) which can be optimized with gradient descent, Newton's method, or in closed from: $w = (XX^T)^{-1}Xt^T$

Estimation with Maximum A Posteriori(MAP) from Bayes Rule

Recall,

- Bayesian view: probabilities provide a quantification of uncertainty. Before observing the data, the assumptions about w are captured in the form of a prior probability distribution $P(\boldsymbol{w})$. The effect of the observed data $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$ is expressed by $P(D|\boldsymbol{w})$.
- Bayes' theorem:

$$P(\boldsymbol{w}|D) = \frac{P(D|\boldsymbol{w})P(\boldsymbol{w})}{P(D)}$$

- Bayes' theorem in words: posterior $\propto$ likelihood × prior

Estimation with Maximum A Posteriori(MAP) from Bayes Rule

- Independence & chain rule of probability.
- x_i is independent of w & $P(\cdot)$ constant and can be dropped.
- $P(t_i, x_i|w) = P(t_i|x_i, w)P(x_i|w) = P(t_i|x_i, w)P(x_i) = P(t_i|x_i, w)$

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(w|(t_1, x_1), \dots (t_n, x_n)) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{P\big(w\big|(t_1, x_1), \dots (t_n, x_n)\big)P(w)}{P\big((t_1,x_1),\dots(t_n,x_n)\big)}$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \left[\prod_{i=1}^{n} P(t_i|x_i, \boldsymbol{w})\right] P(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log[P(t_i|x_i, w)] + \log P(w)$$

- $w \sim N(0, \tau^2)$ assumption

$$= \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i^T w - t_i) + \frac{1}{\tau^2} w^T w = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (x_i^T w - y_i)^2 + \lambda ||w||^2$$

- $\lambda = \sigma^2 / n\tau^2$

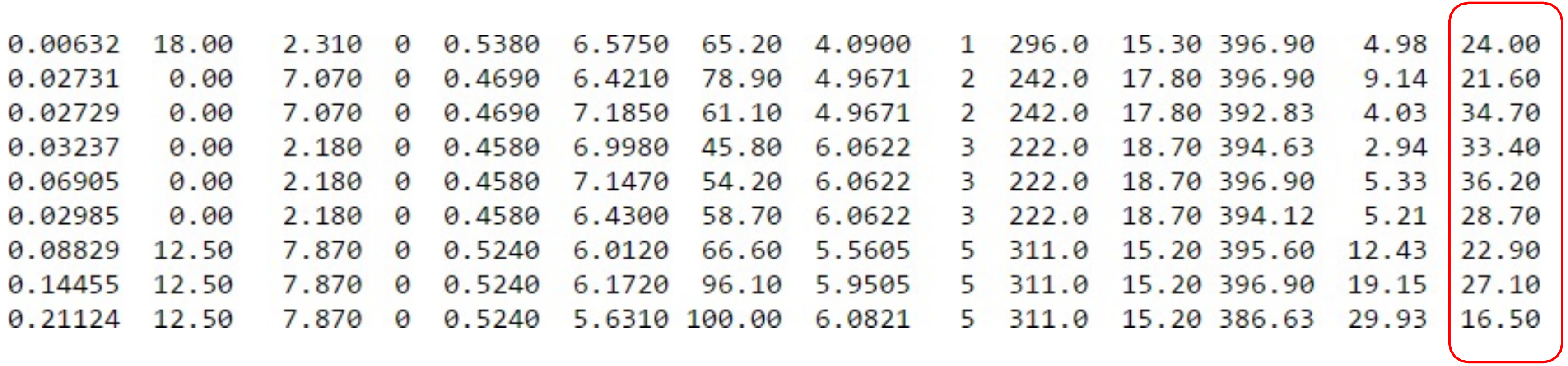
* This objective is known as Ridge Regression.

Linear Regression Example

- https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
  - 1. CRIM: per capita crime rate by town
  - 2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
  - 3. INDUS: proportion of non-retail business acres per town
  - 4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  - 5. NOX: nitric oxides concentration (parts per 10 million)
  - 6. RM: average number of rooms per dwelling
  - 7. AGE: proportion of owner-occupied units built prior to 1940
  - 8. DIS: weighted distances to five Boston employment centres
  - 9. RAD: index of accessibility to radial highways
  - 10. TAX: full-value property-tax rate per $10,000
  - 11. PTRATIO: pupil-teacher ratio by town
  - 12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - 13. LSTAT: % lower status of the population
  - 14. MEDV: Median value of owner-occupied homes in$1000's

Two simple but often ignored questions before applying any ML algorithms

- First, we need to identify what data feature to predict from the model
    - Assume we're interested in predicting the housing prices



```
0.00632  18.00   2.310  0  0.5380  6.5750  65.20  4.0900   1  296.0  15.30 396.90   4.98  24.00
0.02731   0.00   7.070  0  0.4690  6.4210  78.90  4.9671   2  242.0  17.80 396.90   9.14  21.60
0.02729   0.00   7.070  0  0.4690  7.1850  61.10  4.9671   2  242.0  17.80 392.83   4.03  34.70
0.03237   0.00   2.180  0  0.4580  6.9980  45.80  6.0622   3  222.0  18.70 394.63   2.94  33.40
0.06905   0.00   2.180  0  0.4580  7.1470  54.20  6.0622   3  222.0  18.70 396.90   5.33  36.20
0.02985   0.00   2.180  0  0.4580  6.4300  58.70  6.0622   3  222.0  18.70 394.12   5.21  28.70
0.08829  12.50   7.870  0  0.5240  6.0120  66.60  5.5605   5  311.0  15.20 395.60  12.43  22.90
0.14455  12.50   7.870  0  0.5240  6.1720  96.10  5.9505   5  311.0  15.20 396.90  19.15  27.10
0.21124  12.50   7.870  0  0.5240  5.6310 100.00  6.0821   5  311.0  15.20 386.63  29.93  16.50
```

- So the last column MEDV becomes our target variable (and it's numerical)

- Second, what explanatory variables do we use for ML
    - These could be determined/given by the application in hand as well
        - What explanatory variables are easy to get in practice?
    - We can also use whatever data features we have (all other features)
        - Will computation become an issue? Are all of them good quality data?
    - We can explore the data to draw a sensible set of features (e.g., feature selection)

Exploratory data analysis

- Let's visualize pair-wise scatterplots correlations between features

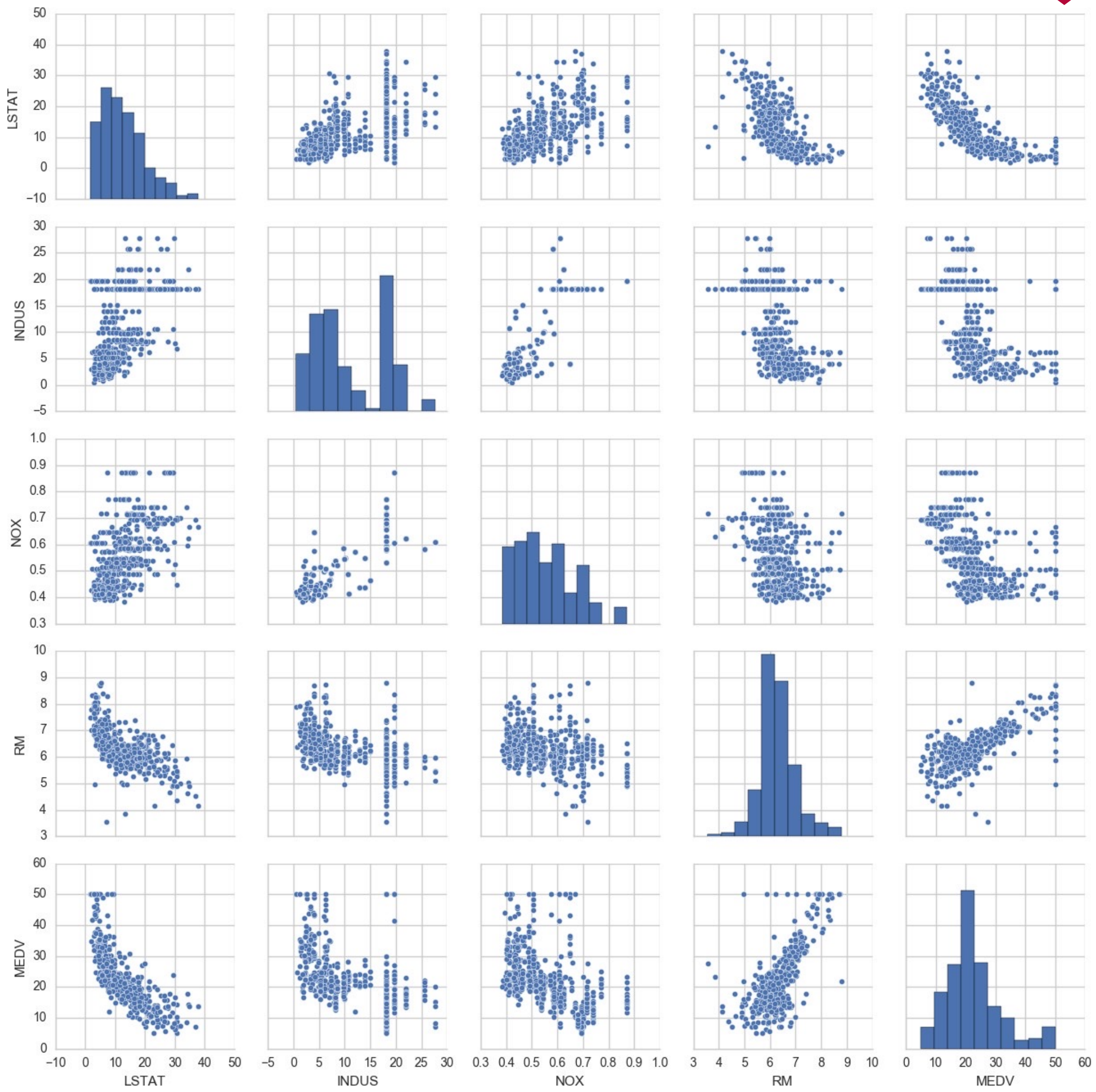
LSTAT: % lower status of the population

INDUS: proportion of non-retail business acres per town

NOX: nitric oxides concentration (parts per 10 million)

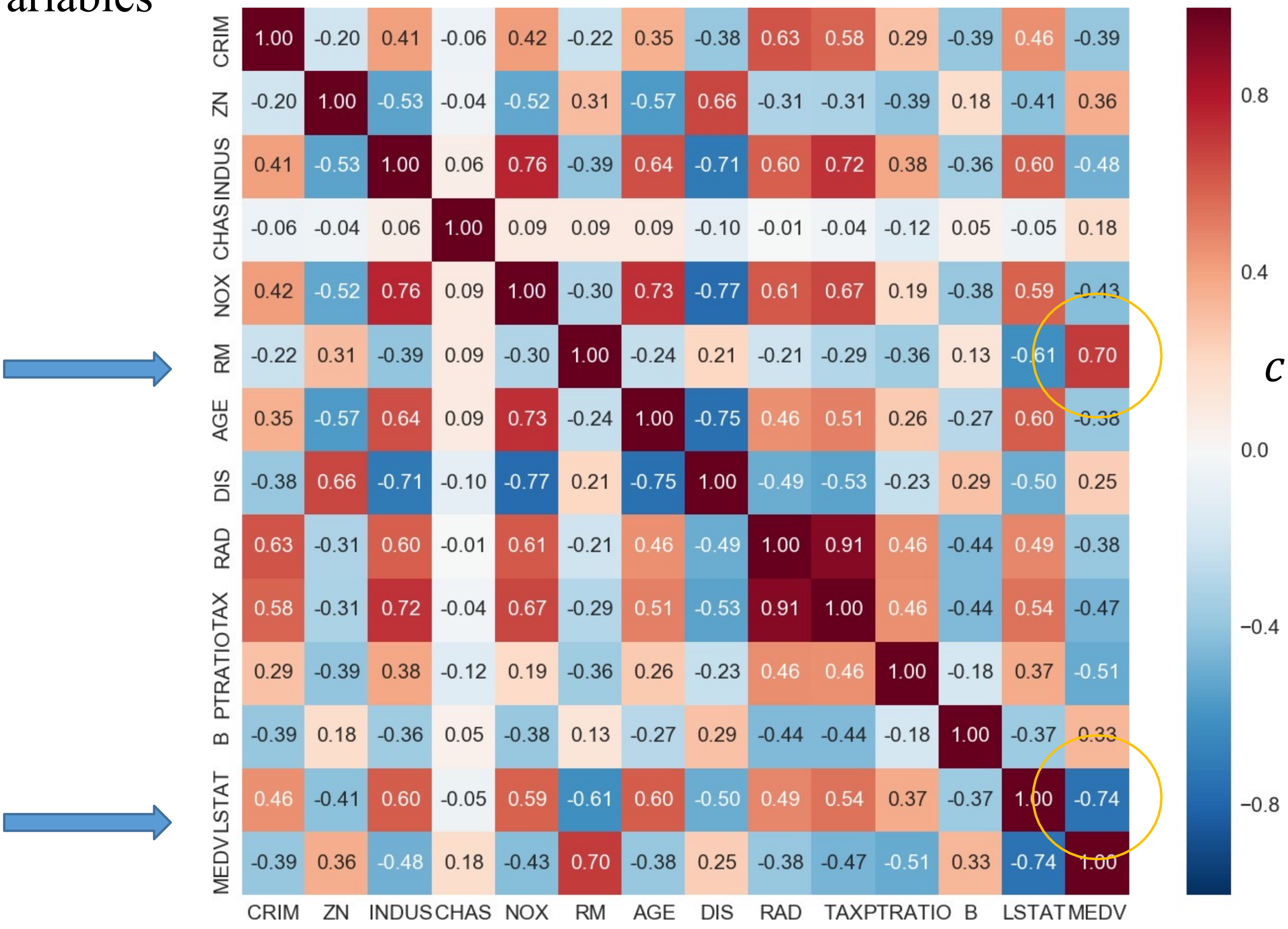
RM: average number of rooms per dwelling

MEDV: Median value of owner-occupied homes in $1000's



Correlation matrix for the data

- Two features (RM and LSAT) have higher correlation with MEDV, implying high chances of being explanatory variables



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1.00	-0.20	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.63	0.58	0.29	-0.39	0.46	-0.39
ZN	-0.20	1.00	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.31	-0.31	-0.39	0.18	-0.41	0.36
INDUS	0.41	-0.53	1.00	0.06	0.76	-0.39	0.64	-0.71	0.60	0.72	0.38	-0.36	0.60	-0.48
CHAS	-0.06	-0.04	0.06	1.00	0.09	0.09	0.09	-0.10	-0.01	-0.04	-0.12	0.05	-0.05	0.18
NOX	0.42	-0.52	0.76	0.09	1.00	-0.30	0.73	-0.77	0.61	0.67	0.19	-0.38	0.59	-0.43
RM	-0.22	0.31	-0.39	0.09	-0.30	1.00	-0.24	0.21	-0.21	-0.29	-0.36	0.13	-0.61	0.70
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1.00	-0.75	0.46	0.51	0.26	-0.27	0.60	-0.38
DIS	-0.38	0.66	-0.71	-0.10	-0.77	0.21	-0.75	1.00	-0.49	-0.53	-0.23	0.29	-0.50	0.25
RAD	0.63	-0.31	0.60	-0.01	0.61	-0.21	0.46	-0.49	1.00	0.91	0.46	-0.44	0.49	-0.38
TAX	0.58	-0.31	0.72	-0.04	0.67	-0.29	0.51	-0.53	0.91	1.00	0.46	-0.44	0.54	-0.47
PTRATIO	0.29	-0.39	0.38	-0.12	0.19	-0.36	0.26	-0.23	0.46	0.46	1.00	-0.18	0.37	-0.51
B	-0.39	0.18	-0.36	0.05	-0.38	0.13	-0.27	0.29	-0.44	-0.44	-0.18	1.00	-0.37	0.33
LSTAT	0.46	-0.41	0.60	-0.05	0.59	-0.61	0.60	-0.50	0.49	0.54	0.37	-0.37	1.00	-0.74
MEDV	-0.39	0.36	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.38	-0.47	-0.51	0.33	-0.74	1.00

$$correlation = \rho = \frac{Cov(x, y)}{\sigma_x \sigma_y}$$

Simple linear regression example

- The explanatory variable is chosen as RM (average number of rooms per dwelling)
- The target variable is chosen as MEDV (Median value of owner-occupied homes in $1000's)
- The ML model to predict the response

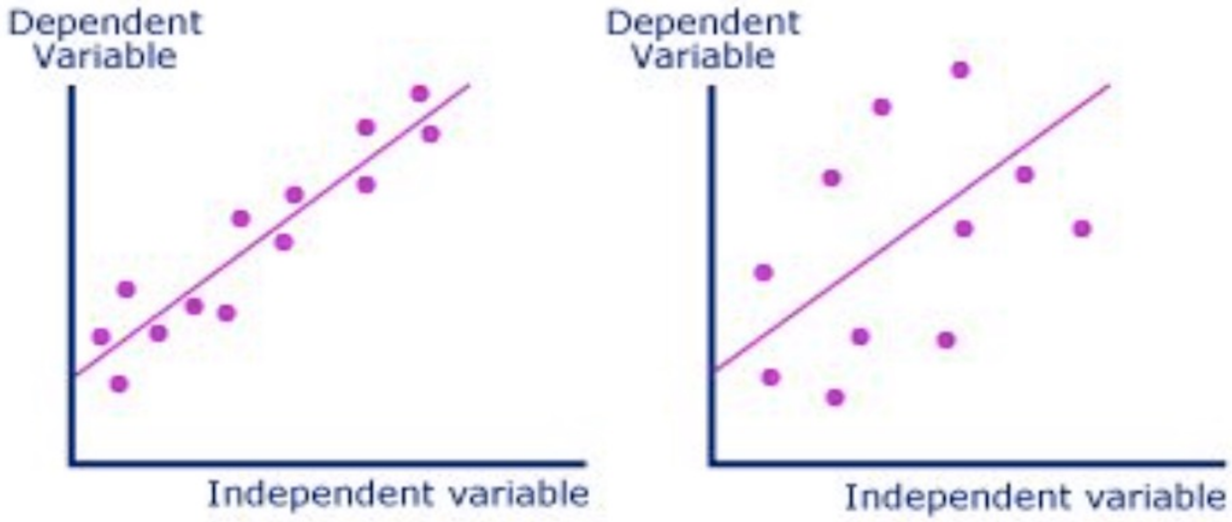
$$H(w, x) = w_0 + w_1 x$$

where y is the observed outcome, e.g., target variable, x is the explanatory variable, $w = [w_0, w_1]^T$ is the model parameters to be determined from ML training

What is the best fit?

Once a linear model is fitted, we assess the overall accuracy of the model.

Two graphs below have fitted with the same model.



Graph A

Graph B

What is the best fit?

The usual way to measure the overall accuracy is to use the coefficient of determination, R^2.
It measures how well data fits a model or how well the model describes the data:

$$R^2 = 1 - \frac{RSS}{TSS}$$

where TSS is the total sum of squares, which measures the total variance of the output data y and RSS is the residual sum of squares.

Given a dataset, TSS is completely determined and the fitted linear model has the minimum RSS.
$R^2 = 1$ indicates that the regression line perfectly fits the data.
$R^2 = 0$ indicates that the line does not fit the data at all.

What is the best fit?

- Prediction Error

$$e^i = y^i - H(w, x^i)$$

where (x^i, y^i) represents the i-th observed data (RM, MEDV)

- Problem formulation: minimize Sum of Squared Errors (SSE aka Least Square Problem)

$$Minimize_w\ f(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - H\left(w, x^i\right)\right)^2$$

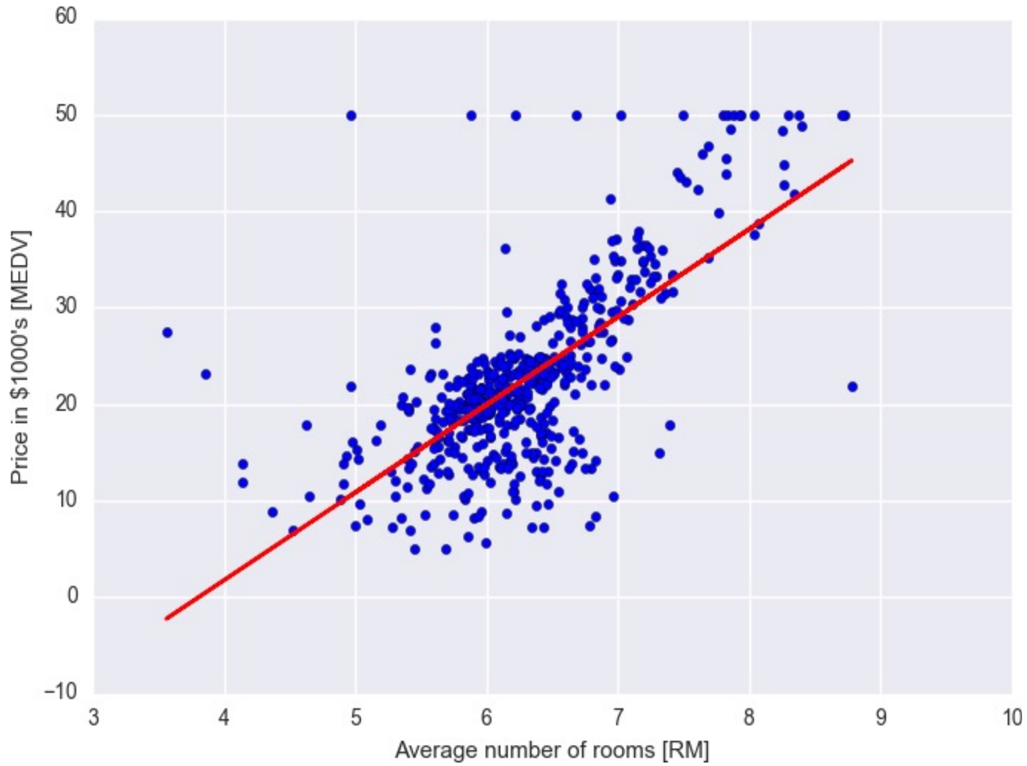
where n is the number of total observations used for training

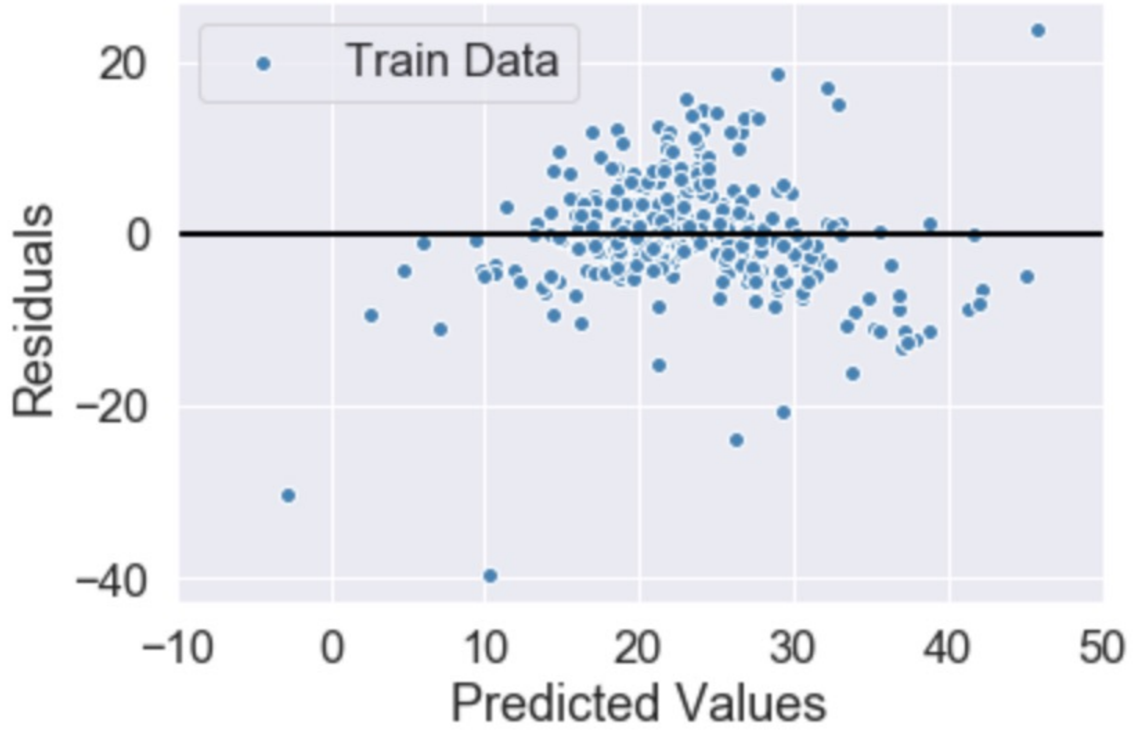
- This is a typical unconstrained optimization problem, more specifically, quadratic programming problem

$$Minimize_w\ f(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - w_0 - w_1 x^i\right)^2$$

What is the best model?

- Is the model good enough?





Extensions of linear regression formulations

- Multiple linear regression model

$$H(w, x) = w_0 + \sum w_i x_i$$

- Polynomial regression

$$H(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

- Nonlinear transformation of some data features
  - For example, log transform the LSAT data

$$H(w, x) = w_0 + w_1 x_1 + w_2 \log x_2$$

- Since the unknown model parameters are linear, they all end up with the same quadratic-like formulation

Nonlinear regression

- So far our regression models are all linear w.r.t. model parameters

$$H(w,x) = w_0 + w_1 x_1 + \cdots + w_m x_m$$
$$H(w,x) = w_0 + w_1 x + w_2 x^2 \ldots + w_d x^d$$
$$H(w,x) = w_0 + w_1 x_1 + w_2\, log(x_2) + w_3\, log^2(x_2)$$

- Nonlinear regression is simply a replacement of the model with a nonlinear function w.r.t. parameters

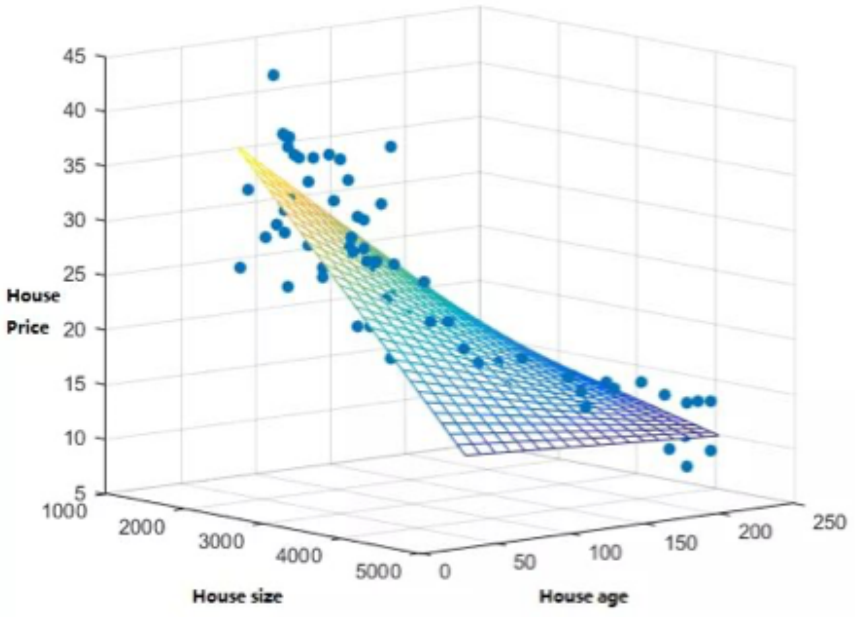
$$H(w,x) = e^{w_0 + w_1 x_1} + w_2 x_2$$

$$H(w,x) = \frac{w_0 + w_1 x_1}{1 + w_2 x_2}$$

- Note some nonlinear form can be transformed to be a linear problem

$$y = H(w,x) = e^{w_0 + w_1 x_1} \rightarrow \ln(y) = w_0 + w_1 x_1$$

Linear Regression Assumptions and Limitations



Assumptions:

No **multicollinearity** between the predictor variables.

- The relationship among independent variables.
- Redundancies within predictors.
- Difficulties in making inferences between predictors and target.
- The standard errors of estimates will be inflated.
- The power and reliability of coefficients will decrease.
- The needs of a larger sample size will be necessary.

Ways to overcome?

- Find multicollinearity and do model selection.
- We can do stepwise procedures. (Is this the best way?)
  - Regularization
  - Bias-Variance Tradeoff
  - Model Selection

Empirical Risk Minimization

- Regularization
- Overfit vs. Underfit
- Bias-Variance Tradeoff

Empirical Risk Minimization

Consider a linear regression:

- Data D(X,y) where y is the target
- We need a function mapping $f: X \to y$
- Model $h: X \to y$

Empirical implies the minimum error based on a sample set S from the set X, with D being the distribution over X.

The **true error** from X:

$$L_{D,f}(h,y) = P_{x \sim D}[h(x) \neq f(x)] = D(\{x: h(x) \neq f(x)\})$$

Since we use the subset of training examples, we cannot access to the true error, but we can access to the **empirical error** (**generalization error**).

$$L_s(h,y) = \frac{|\{i \in [m]: h(x_i) \neq y_i\}|}{m}$$

This error is also called the **risk** by generalizing the error.

Regularization

Recall from Equation (1), $t = y(x, w) + \epsilon$, where $\epsilon = \mathcal{N}(0, \tau^2)$

$$\min_{w} \sum_{i=1}^{N} l(y(x, w), t) + \lambda r(w) \qquad (2)$$

$$\Leftrightarrow \min_{w} \sum_{i=1}^{N} l(y(x, w), t) \;\; s.t. r(w) \leq B$$

- We **regularize** the model – make the model be acceptable by applying a penalty term λ to reduce the error.
- $l(y(x, w), t)$ is the error function (cost function) – we use the squared error function for Linear Regression.
- Equation (2) indicates that for each penalty, $\lambda \geq 0$, the regularization function $r(w)$ is less than or equal to a constant number B.

Regularization – Lasso and Ridge

l_2 Regularization:

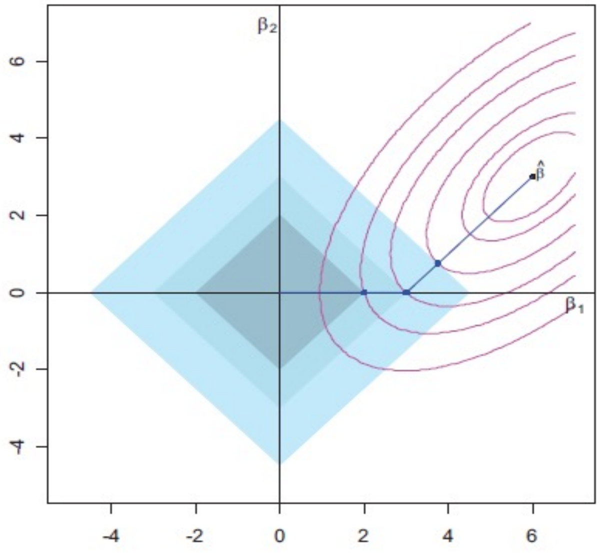
$$r(w) = w^T w = ||w||^2$$

- Strictly convex and differentiable
- Uses weights on all features – dense solutions

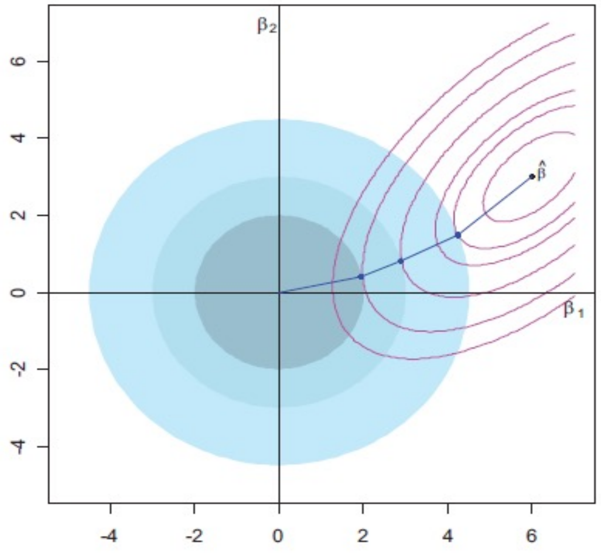
l_1 Regularization:

$$r(w) = ||w||$$

- Convex but not strictly
- Not differentiable at 0
- Sparse Solutions



l_1



l_2

Elastic Net Regularization

$$\lambda||w|| + (1 - \lambda)||w||^2$$

Loss and Regularization	Comments
Least Squares $\frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$	• Square Loss • No Regularization • Closed form solution: $w = (XX^T)^{-1}Xy^T$ • $X = [x_1, \dots, x_n], y = [y_1, \dots, y_n]$
Ridge Regression $\min_w \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda\|\|w\|\|^2$	• Squared Loss • l_2 regularization • $w = (XX^T + \lambda I)^{-1}Xy^T$
Lasso Regression $\min_w \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda\|\|w\|\|$	• Sparsity inducing – good for feature selection • Convex but not strict (no unique solution) • Not differentiable at 0 • Solve with sub-gradient descent
Elastic Net $\min_w \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2 + \lambda\|\|w\|\| + (1-\lambda)\|\|w\|\|^2$	• Unique solution and sparsity inducing • Dual of squared-loss SVM • Non-differentiable

Overfitting vs. Underfitting

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n} \underset{\substack{\uparrow \\ \text{Loss}}}{l(y(w,x),t)} + \lambda \underset{\substack{\uparrow \\ \text{Regularization}}}{r(w)}$$

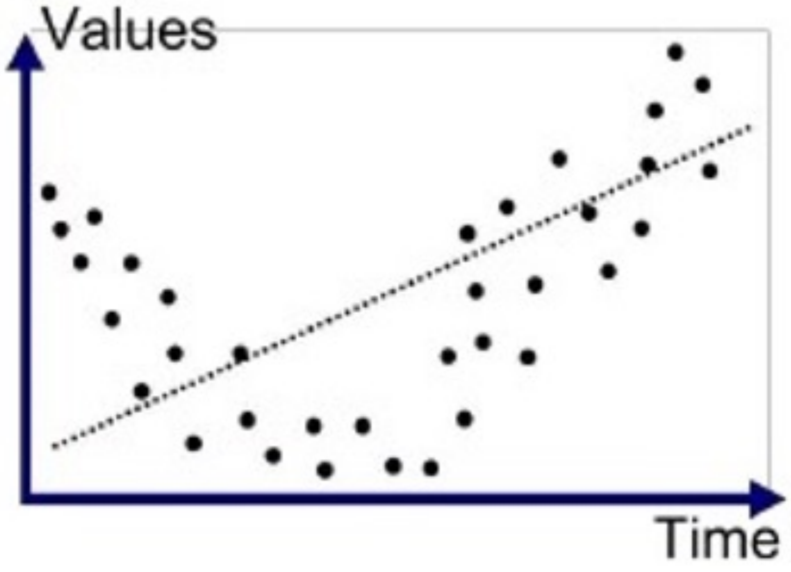
A question rise is on λ – how should we handle it?

Leads to two possible scenarios: **Underfitting** or **Overfitting**

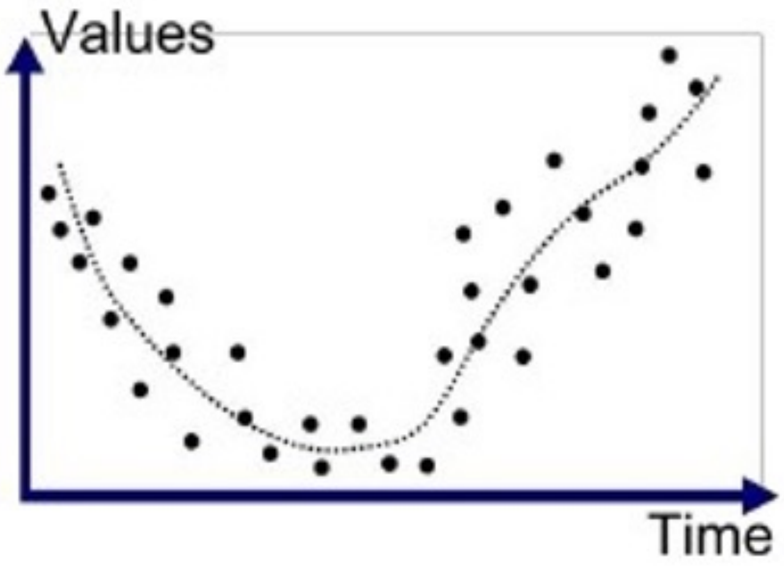
Underfitting – The model does not learn enough through the training process. Both errors on training and test will be high. (The model is relatively simpler than what it supposed to be.)

Overfitting – The model learns too much from training data set. The test error may rise because the model will reflects on the exact patterns of training data set in the test set. (The model is relatively complex than what it should be.)

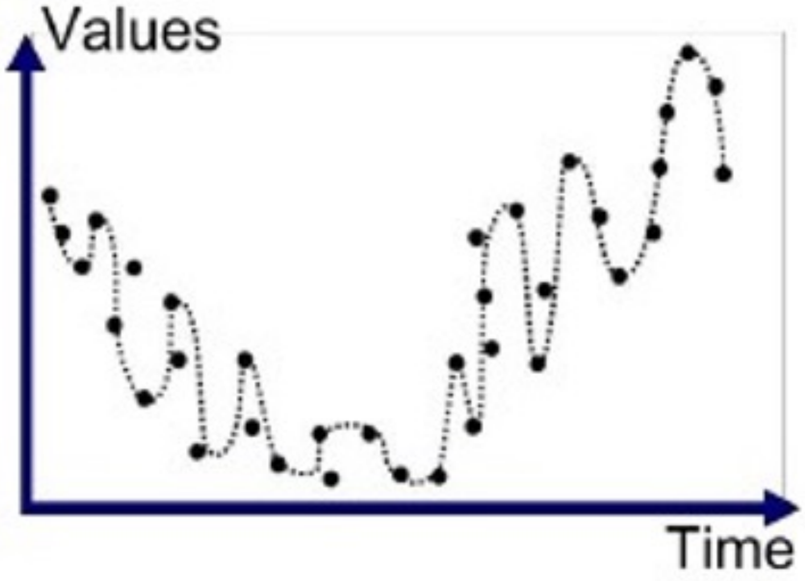
Overfitting vs. Underfitting



Underfitted

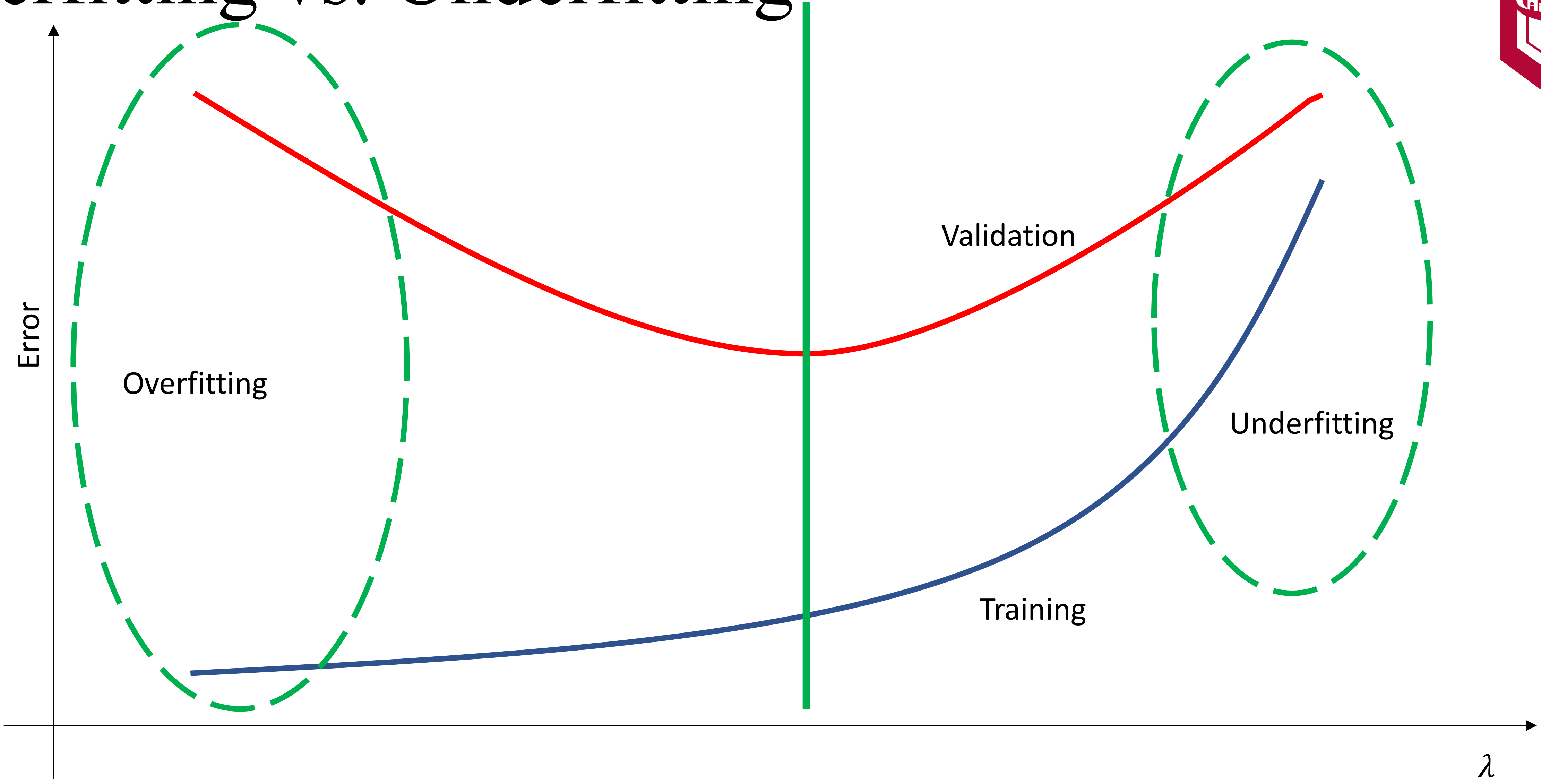


Good Fit/Robust



Overfitted

Overfitting vs. Underfitting



Overfitting vs. Underfitting

The big question is…. **HOW do we find the exact point to stop?**

Typical range: $10^{-5}, 10^{-4}, 10^{-3}, \cdots, 10^{1}, 10^{2}, 10^{3}, \cdots$

K-fold Cross Validation

- Divide the training data k portion equally.
- Train on k-1 of them and leave kth one for the validation.
- Do this k times and find the best λ that gives the smallest error on the validation set.
  - When the data is too small, you leave only one observation for the validation.

For a specific search on λ

- Do with $\lambda = \{\cdots, 10^{-3}, 10^{-2}, \cdots, 10^{1}, 10^{2}, \cdots\}$ to find the best magnitude.
- Then do for the specific value of λ value in the best chosen magnitude
- Example: Let's say you learned that $\lambda = 10^{-2}$ is the best magnitude on the 1st try. Then do with $\lambda = \{0.01, 0.015, 0.02, \cdots\}$ on the 2nd try.

Bias-Variance Tradeoff

$$E_{x,y,D}[(h_D(x) - y)^2] = E_{x,y,D}\left[\left[\left(h_D(x) - \bar{h}(x)\right) + \left(\bar{h}(x) - y\right)\right]^2\right]$$

$$= E_{x,D}\left[\left(h_D(x) - \bar{h}(x)\right)^2\right] + 2E_{x,y,D}\left[\left(h_D(x) - \bar{h}(x)\right)\left(\bar{h}(x) - y\right)\right] + E_{x,y}\left[\left(\bar{h}(x) - y\right)^2\right] \qquad (1)$$

2nd Term

$$E_{x,y,D}\left[\left(h_D(x) - \bar{h}(x)\right)\left(\bar{h}(x) - y\right)\right] = E_{x,y}[E_D[h_D(x) - \bar{h}(x)](\bar{h}(x) - y)]$$

$$= E_{x,y}\left[\left(E_D[h_D(x)] - \bar{h}(x)\right)\left(\bar{h}(x) - y\right)\right]$$

$$= E_{x,y}[0]$$

$$E_D[h_D(x)] = \bar{h}(s)$$

3rd Term

$$E\left[\left(\bar{h}(x) - y\right)^2\right] = E\left[\left(\left(\bar{h}(x) - \bar{y}(x)\right) + (\bar{y}(x) - y)\right)^2\right]$$

$$= E[(\bar{y}(x) - y)^2] + E\left[\left(\bar{h}(x) - \bar{y}(x)\right)^2\right] + \cancel{2E\left[\left(\bar{h}(x) - \bar{y}(x)\right)(\bar{y}(x) - y)\right]}$$

$$E_{x,y,D}[(h_D(x) - y)^2] = E_{x,D}\left[\left(h_D(x) - \bar{h}(x)\right)^2\right] + E_{x,y}\left[\left(\bar{h}(x) - \bar{y}(x)\right)^2\right] + E_x[(\bar{y}(x) - y)^2] \qquad (2)$$

$Var(x)$ =Variance Bias^2 ϵ =Noise

Bias-Variance Tradeoff

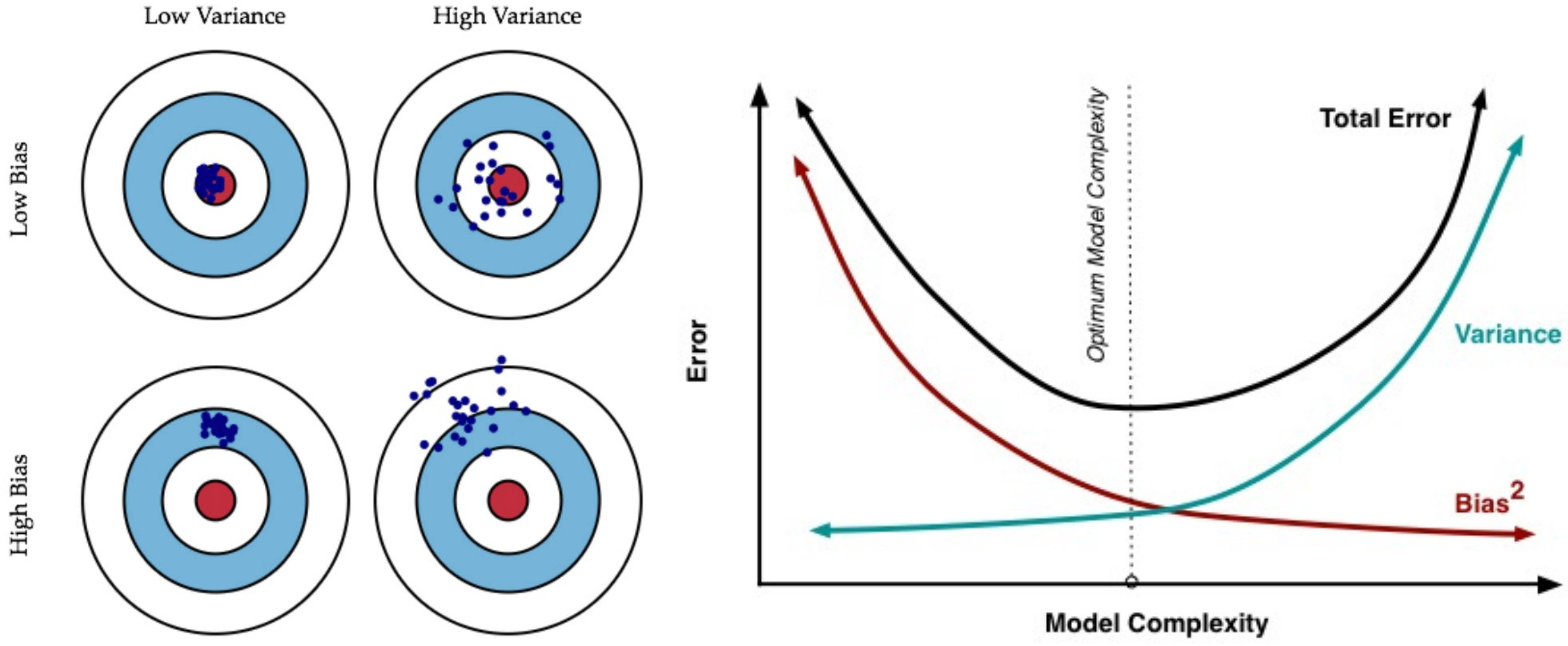


Figure: Graphical illustration of bias vs. variance

Summary

- Linear Regression:
  - One of easiest regression algorithm in ML
  - Easy to implement and apply
  - 3 Conditions:
    - Linear relationship – the target has a roughly linear relationship with features/variables.
    - Homoscedasticity – the distribution of residuals has the same variance.
    - Independent errors – errors should be uncorrelated
  - Things to consider
    - Outliers
    - Normal Distributed residuals
- To overcome LR,
  - Regularization – Lasso and Ridge or Elastic Net
  - Bias – Variance Tradeoff
  - Overfit vs. Underfit