Linear Regression

Quách Đình Hoàng

2022/09/22

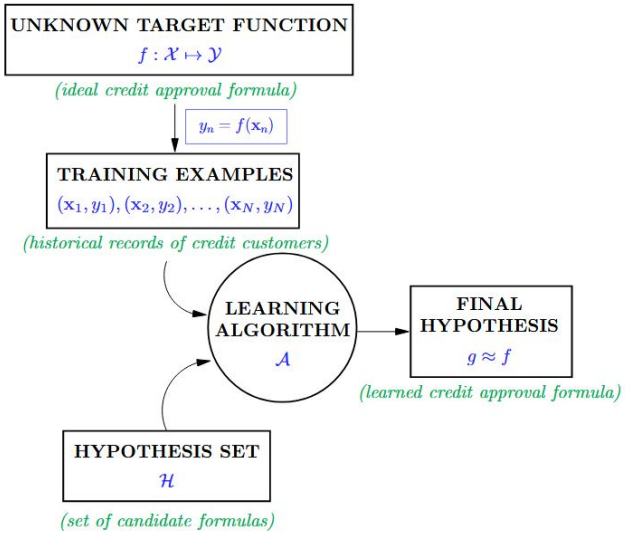
Contents

Machine learning review

Machine learning definition

- "Machine learning as a field of study that gives computers the ability to learn without being explicitly programmed" - Arthur Samuel (1959)
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." - Tom M. Mitchell (1997)
- In general, machine learning is about designing algorithms that learn from data.

Learning problem setup



Source: Yaser S. Abu-Mostafa, et al., Learning From Data, AMLBook 2012.

Fundamental assumption of learning

- The dataset (sample) that we use represents all the data (population) from which it was generated.
- iid assumption: the objects selected into the dataset are independent and have the same distribution (draw from the sam joint probability distribution, $p(x, y)$)

$$(x_i, y_i) \overset{iid}{\sim} P(X, Y), \forall i = 1, 2, \ldots, n$$

- There is no free lunch
  - No algorithm outperforms any other on all tasks $\rightarrow$ we must make assumptions in order to learn.

Empirical risk minimization

- A loss function $l : R^2 \to R$ measure how well $\hat{y} = f(x)$ approximate y. It is loss (error) of predicting $\hat{y} = \hat{f}(x)$ when the actual value is y.
- Empirical risk is average loss on the whole dataset $\{(x_i, y_i)\}_{i=1..n}$

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(\hat{y}_i, y_i) = \frac{1}{n}\sum_{i=1}^{n} l(\hat{f}_\theta(x_i), y_i)$$

- ERM: Choose θ to minimize empirical risk $L(\theta)$
- This optimization problem usually has no explicit solution, and the numerical optimization algorithm is often used to find θ that minimizes $L(\theta)$.

Regularized empirical risk minimization

- We measure the complexity of model f_θ using a regularizer function $r : \mathbb{R}^d \to \mathbb{R}$
- We want the model to predict well on test data, i.e. small empirical risk

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{n} l(\theta^T x_i, y_i)$$

- We want also the model is not too complex/sensitive, i.e. $r(\theta)$ is small
- To balance these two goals, we optimize regularized empirical risk

$$L(\theta) + \lambda r(\theta)$$

  - $\lambda > 0$ is regularization parameter (also known as hyper-parameter)
- Regularized empirical risk minimization (RERM): choose θ to minimize regularized empirical risk

Split dataset

- If we evaluate too many models on the test set, we are using a test set like traning set $\rightarrow$ test set is no longer good to simulate our performance model again.
- To overcome this, we will split the data into 3 sets
  - Training set: for train model
  - Validation set: for choose hyper-parameter
  - Test set: for estimate performance of the model on future data

Hyperparameter selection and model evaluation

- Suppose we want to compare multiple hyperparameter settings $\theta_1, \theta_2, \cdots, \theta_k$
- For $i = 1, 2, \cdots, k$
  - Train a model on D_{train} using θ_i
- Evaluate each model on D_{val} and find the best hyperparameter setting, θ_i^*
- Compute the error of a model trained with θ_i^* on D_{test}

Regression problem

Regression problem

- We think that $y \in R$ and $x \in R^d$ are approximated by:

$$y \approx f(x)$$

  - x is called independence variables or input or features
  - y is called dependence variable or output or response
  - Usually, y is the variable we want to predict.
- We don't know the actual relationship between y and x, the function $f(.)$ is just an approximation.

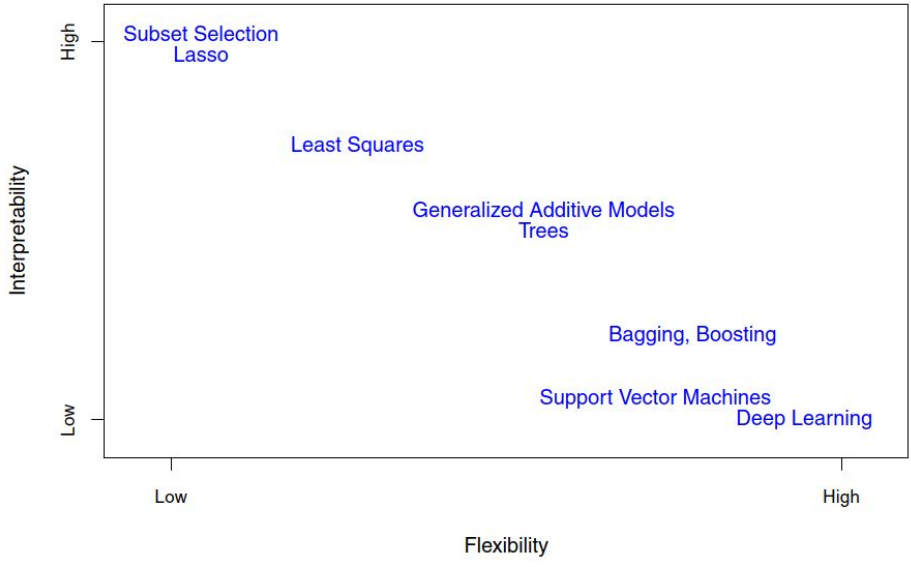
Explanatory vs. predictive modeling with regression

	Explanatory Modeling (Statistical approach)	Predictive Modeling (machine learning approach)
General goal	Explain the relationship between input x and output y.	Predict output y from input x.
Modeling	Find the data generation model (distribution $p(x, y)$).	Find function f (blackbox) to predict y from x.
Model validaion	Use the whole dataset to perform the "goodness-of-fit" test: R^2, residual analysis, p-values, ...	Split dataset into train/test set. Train model on train set and evaluate model on test set

References:

Leo Breiman, *Statistical Modeling: The Two Cultures, Statistical Science*, Vol. 16, No. 3, 199-231, 2001.

Prediction Accuracy and Model Interpretability



Regression algorithms

- Linear regression
- Linear model selection
  - Best subset selection
  - Forward/Backward stepwise selection
  - Ridge/Lasso/ElasticNet regression
- Linear regression extensions
  - Splines and smoothing splines
  - Local regression
  - Generalized additive models
- Non-linear regression
  - Polynomial regression
  - k-nn regression
  - Regression tree
    - Bagging for regression
    - Random forest for regression
- Support vector regression
- Neural network regression

References:

1. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, An introduction to statistical learning, Second edition, Springer, 2021.
2. AJ Smola and B Schölkopf, A tutorial on support vector regression, Statistics and computing, 14, 199-222, 2004.

Linear regression

Linear regression

- The simplest and most common model of f is linear function in terms of x.
- Linear regression model:

$$\widehat{y} = f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d = \theta^T x$$

  - $\theta^T = (\theta_0, \theta_1, \cdots, \theta_d) \in R^{d+1}$: the parameters of the model
  - $x = (1, x_1, \cdots, x_d)^T \in R^{d+1}$: input of the model.
  - We assume x is a column vector, x^T is a row vector.
  - We can write $f_\theta(x)$ to emphasis the dependence of f on θ.
  - θ_0 is the prediction of the model when all features are 0.

Interpretation of regression coefficients

$$\widehat{y}_i = f(x_i) = \theta_0 + \theta_1 x_i + \cdots \theta_d x_d$$

- $\theta_i (i \neq 0)$ is the degree to which $\widehat{y} = f(x)$ increases when x_i increases by one unit
- $\theta_i = 0$ implies that $\widehat{y} = f(x)$ does not depend on x_i
- θ small implies that the model insensitive to the change of x

$$|f(x) - f(x')| = |\theta^T x - \theta^T x'| = |\theta^T (x - x')| \leq ||\theta|| ||x - x'||$$

Loss function

- Loss function $l : R \times R \to R$ determine how close $\hat{y}$ is approximate y
  - $l(\hat{y}, y) \geq 0, \forall \hat{y}, y$
  - $l(\hat{y}, y)$ small shows that $\hat{y}$ is a good approximation of y
- Two common loss functions:
  - Quadratic/square loss (L_2): $l(\hat{y}, y) = (\hat{y} - y)^2$
  - Absolute loss (L_1): $l(\hat{y}, y) = |\hat{y} - y|$

Empirical risk

- Empirical risk is average loss on the whole dataset $\{(x_i, y_i)\}_{i=1..N}$

$$L = \frac{1}{n}\sum_{i=1}^{n} l(\hat{y}_i, y_i) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)$$

  - If L small, model predict well on given data
  - When the model is parameterized by θ, we write

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i), y_i)$$

    to show the dependence of the model on θ

Mean square error

- When loss function is L_2: $l(\hat{y}, y) = (\hat{y} - y)^2$ then empirical risk is mean square error (MSE)

$$L = MSE = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

  - However, we often use root-mean-square error, $RMSE = \sqrt{MSE}$, since it has the same units as y_i

Mean absolute error

- When loss function is L_1: $l(\hat{y}, y) = |\hat{y} - y|$ then empirical risk is mean absolute error (MAE)

$$L = MAE = \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|$$

  - MAE has the same units as y_i

Empirical risk minimization

- Empirical risk minimization (ERM) is a general method for selecting parameter θ for the model $f_\theta(x)$
- ERM chooses θ such that empirical risk $L(\theta)$ is minimized
- In general, there is no analytic solution for this optimization problem. Therefore, we often have to use numerical optimization methods to find θ such that $L(\theta)$ is minimized.

Least square linear regression

- Linear regression model

$$\hat{y} = f_\theta(x) = \theta^T x$$

  - $\theta \in R^{d+1}$ is the parameters of the model
  - $x \in R^{d+1}$ is input of the model
- We use loss function $l(\hat{y}, y) = (\hat{y} - y)^2$
- Empirical risk is mean square error (MSE)

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\theta^T x_i - y_i)^2$$

- We estimate θ use empirical risk minimization (ERM) method
  - Choose θ such that $L(\theta)$ is minimized

Least square linear regression

- MSE is writen as matrix form

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^T x_i - y_i)^2 = \frac{1}{n}||X\theta - y||^2$$

  - $\theta = (\theta_0, \theta_1, \cdots, \theta_d)^T \in R^{d+1}$
  - $X \in R^{n\times(d+1)}, y \in R^n$

$$X = \begin{bmatrix} (x_1)^T \\ (x_2)^T \\ \vdots \\ (x_n)^T \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad (1)$$

- We need to choose θ such that $||X\theta - y||^2$ is minimized.

Least square linear regression - analytical solution

$$L(\theta) = ||X\theta - y||^2 = \sum_{i=1}^{n}(\sum_{j=0}^{d} x_{ij}\theta_j - y_i)^2$$

- The optimal solution $\hat{\theta}$ satisfies

$$\frac{\delta L}{\delta \theta_j}(\hat{\theta}) = \nabla L(\hat{\theta})_j = 0, j \in \{1, \cdots, d\}$$

- Take the partial derivative on θ_j: $\nabla L(\theta)_j = (2X^T(X\theta - Y))_j$
- Write as matrix form: $\nabla L(\hat{\theta}) = 2X^T(X\hat{\theta} - Y) = 0$
- $\hat{\theta}$ need to satisfy the equation: $(X^T X)\hat{\theta} = X^T Y$
- Therefore: $\hat{\theta} = (X^T X)^{-1} X^T Y$ (if $X^T X$ is invertible)

Least square linear regression

- Problem: Choose θ such that $||X\theta - y||^2$ is minimized.
- If the columns (or rows) of X are linearly independent then $X^T X$ is invertible, problem has unique solution

$$\theta^* = (X^T X)^{-1} X^T Y = X^\dagger Y$$

- If the columns (and rows) of X are linearly dependent then $X^T X$ is not invertible, pseudo-inverse matrix of $X^T X$ with formula $X^T (X X^T)^{-1}$ can be used.

Least square linear regression - gradient descent

$t = 0$

Initialize $\theta_j^{(t)}$ randomly

repeat

- $partial[j] = 0$ for all $0 \leq j \leq d$
- foreach data point $i = 1, 2, \ldots, n$
  - foreach parameter $j = 0, 1, \ldots, d$
    - $partial[j]$ += $\left(-x_i(y_i - x_i^T \theta_j^{(t)})\right)$
- $\theta^{(t+1)} = \theta^{(t)} - \eta \cdot partial[j]$
- $t \leftarrow t + 1$

until $||\theta^{(t)} - \theta^{(t-1)}|| \leq \delta$

Least square linear regression - stochastic gradient descent

$t = 0$

Initialize $\theta^{(t)}$ randomly

repeat

- foreach $i = 1, 2, \ldots, n$ (random order)
  - $partial[i] = -x_i(y_i - x_i^T \theta_i^{(t)})$
  - $\theta^{(t+1)} = \theta^{(t)} - \eta \cdot partial[i]$
- $t \leftarrow t + 1$

until $||\theta^{(t)} - \theta^{(t-1)}|| \leq \delta$

Least square linear regression - mini-batch gradient descent

$t = 0$

Initialize $\theta_j^{(t)}$ randomly

repeat

- Split dataset into k mini_batch with size l randomly $(k * l = n)$
- foreach random $mini\_batch$
  - $partial[j] = 0$ for all $0 \leq j \leq d$
  - foreach parameter $j = 0, 1, \ldots, d$
    - $partial[j] \mathrel{+}= \sum_{(x_i, y_i) \in mini\_batch} \left(-x_i(y_i - x_i^T \theta_j^{(t)})\right)$
  - $\theta^{(t+1)} = \theta^{(t)} - \eta \cdot partial[j]$
  - $t \leftarrow t + 1$

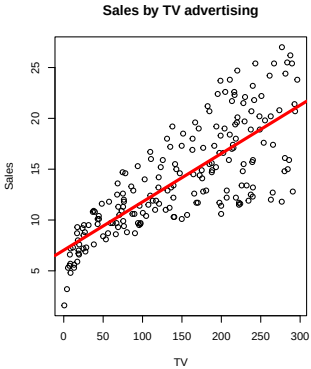
until $||\theta^{(t)} - \theta^{(t-1)}|| \leq \delta$

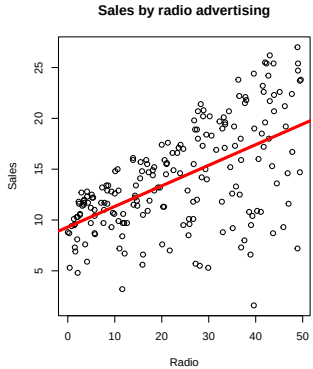
Example: Advertising data

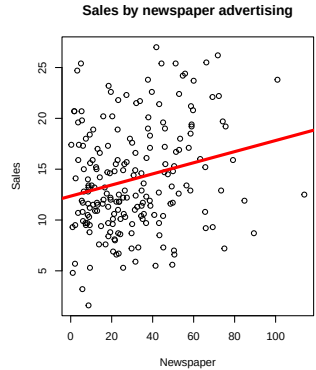
- Input u are variables $TV, radio, newspaper$ describe the amount of advertising for these types.
- Output v is the variable $sales$ describe the company's revenue.
- We would like to answer some of the following questions:
  - Is there a relationship between money spent on advertising and sales?
  - Is the relationship strong?
  - Is the relationship linear?
  - Which type of advertising contributes more to revenue?
  - Can we predict future sales based on the amount of money spent on advertising?

Linear regression model for advertising data

- Linear regression model for each variable on advertising data

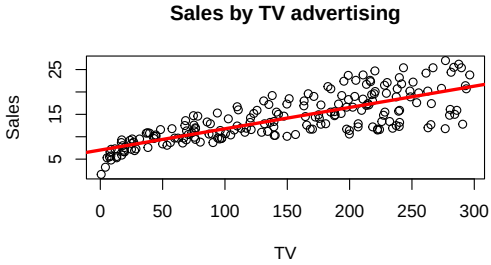


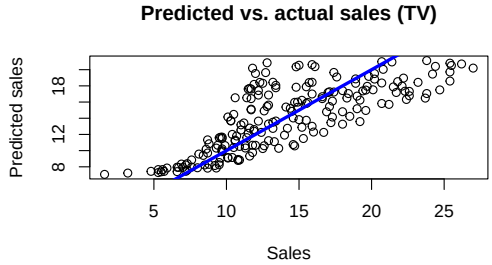




- The best predictor is TV, $MSE = 10.51$
  - $radio$ has $MSE = 18.09$
  - $newspaper$ has $MSE = 25.67$

Predict sales with TV advertising





- The figure on the left is a linear regression model with the variable TV

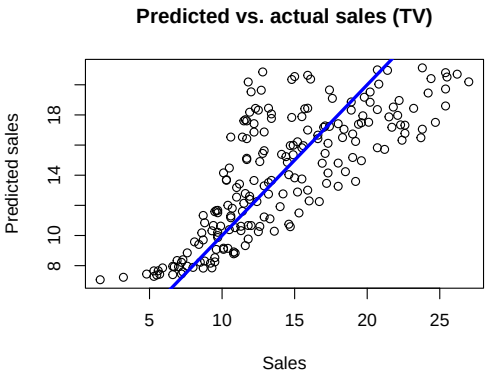
$$sales \approx 7.03 + 0.05 \times TV$$

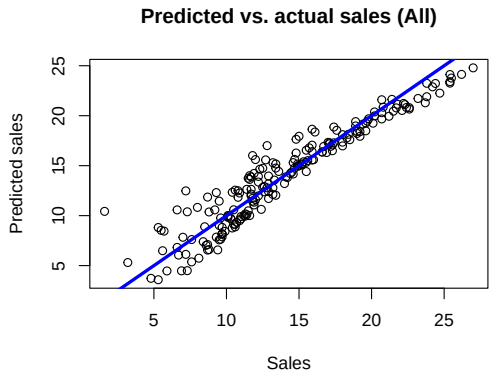
- The picture on the right is the predicted sales and the actual sales
  - Ideally every point is on the blue line

Predict sales with TV advertising

```
##
## Call:
## lm(formula = sales ~ TV, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
```

Predict sales with all variables





- The figure on the left is a model using only the variable TV, $MSE = 10.51$
- The figure on the right shows the model using all the variables, $MSE = 2.78$

$$sales \approx 2.94 + 0.05 \times TV + 0.19 \times radio - 0.001 \times newpaper$$

Predict sales with all variables

```
##
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

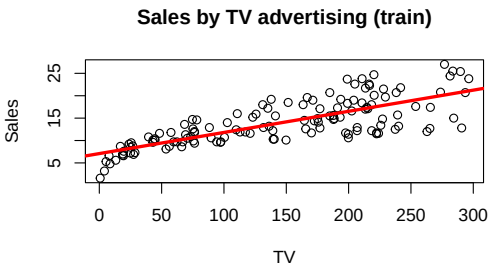
Validation

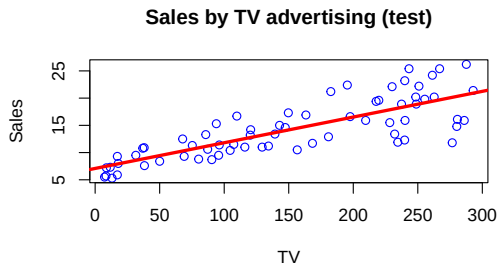
Generalization

- Generalization is the ability that a model predicts well on new data
  - Predict well on training data is not the end goal
- We build the model based on training data or in-sample data
- We expect the model to also predict well on test data or out-of-sample data
- If it does not predict well on new data, we say the model does not generalize (failure to generalize).

Example on advertising data

- We train model use $2/3$ dataset to predict on the remaining $1/3$.





- MSE on train set is 10.65, MSE on test set is 10.64
- We can conclude that the model is generalize
  - The difference of MSE on the train set and the test set is not big

Out-of-sample validation

- We use validation/test set to test the ability of the model on new (unseen) data.
  - This is called the principle out-of-sample validation
- Two popular evaluation methods are based on out-of-sample validation principle:
  - Holdout method
  - Cross-validation

Holdout validation

Holdout validation is the simplest form of out-of-sample validation.

- Idea: uses a part of the dataset as unseen data and assumes that future data will be similar.
  - We split the dataset into two sets, train và test randomly
  - Use train set to build the model
  - Use test set to evaluate the model
- This is a way for us to simulate the predictive abilit of the model on unseen data.
  - Usually, we only have one dataset, new (unseen) data is often difficult to collect.

Holdout validation

- Test error is what we care about
  - Train error is not important
- We split the dataset into two sets, train và test randomly
  - The train/test ratio is usually 80/20 or 90/10
  - When the data is a lot, can we divide by the ratio 50/50 or 60/40
- Test error usually larger train error a little
- If test error much larger than train error, we say the model is overfit.
- Random sampling is a variation of holdout
  - Repeat holdout k times
  - Use average test error over k times to evaluate the model

Holdout validation

- Train/test error results can have the following cases

test/train	small train error	large train error
small test error	generalizes (performs well)	lucky (or fraud)
large test error	fails to generalize (overfit)	generalizes (underfit)

Example on advertising data

features	train error	test error
TV	10.74	10.06
radio	17.01	20.29
newspaper	24.54	28.04
TV + radio	2.68	2.98
TV + newspaper	2.68	9.17
radio + newspaper	2.68	20.77
all	2.68	3.07

- The model with only two attributes TV and $radio$ gives the same good prediction results as when using all attributes (all).

Overfitting

- I just tested many models
- We can choose the model best fit with training data.
- But this can lead to bad predictive model on test data
- This is called overfitting

Example - polynomial fit

- Suppose the raw input is $u \in R$
- We use features that are polynomials according to u (polynomial features)

$$x = \phi(u) = (1, u, u^2, \cdots, u^k)^T$$

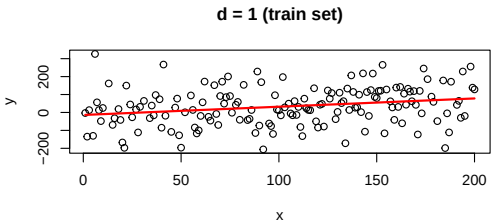
  and linear regression model $f(x) = \theta^T x$
- $f(x)$ is a polynomial of degree d with respect to u

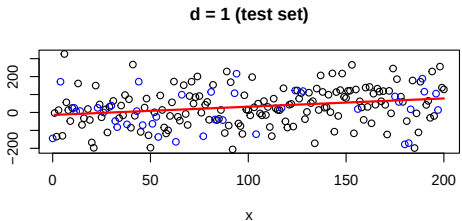
$$\hat{y} = f(x) = \theta^T x = \theta_0 + \theta_1 u + \theta_2 u^2 + \cdots \theta_d u^d$$

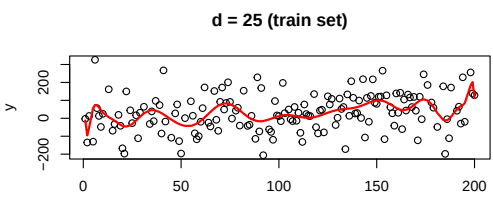
- We choose θ use ERM with lost function is $l(\hat{y}, y) = (\hat{y} - y)^2$

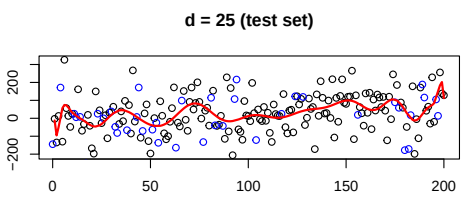
Example - polynomial fit

- 200 data point is divided into 2 train/test sets with the ratio 80/20



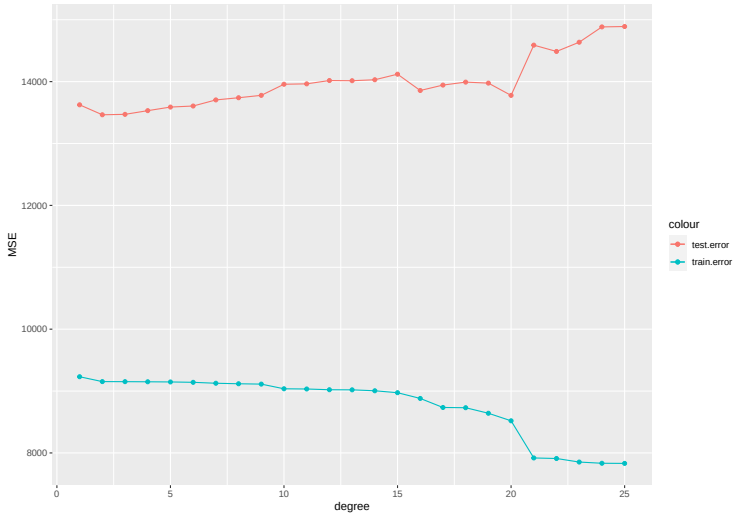






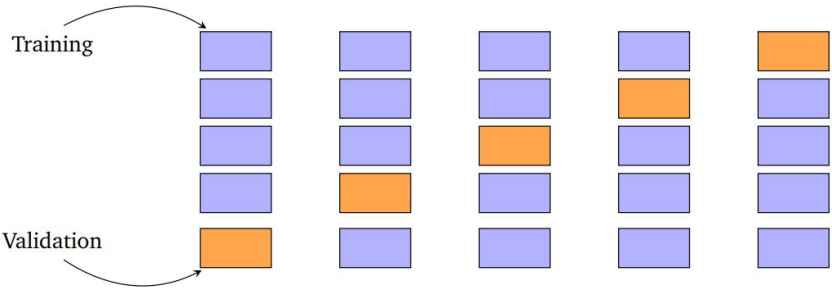
Example - polynomial fit

- 200 data point is divided into 2 train/test sets with the ratio 80/20



Cross validation

- This is another popular method for model evaluation
  - Split the dataset into k folds that are approximately the same
  - For each fold i, build the model on other folds except i
  - Evaluate the model on fold i
  - Use everage test error over all folds to valuate the model



Cross validation

There are two special cases of cross validation

- Leave-one-out cross validation
  - Divide the data into n folds, where n is the number of objects in the dataset
  - This method is useful for small data sets
- Stratified cross-validation
  - The folds are stratified so that the class distribution in each section approximates the class distribution of the dataset
  - This method is useful for data sets with unequal class distribution

Linear model selection

Best subset selection

- Let M_0 is null model, contains no variables.
  - This model uses sample mean to predict for each data point
- for $k = 1, 2, \ldots, d$
  - Fit all $\begin{pmatrix} d \\ k \end{pmatrix}$ models that contain k variables
  - Choose best model (has minimum $L(\theta)$) in $\begin{pmatrix} d \\ k \end{pmatrix}$ models, and call it M_k.
- Choose best model (has minimum validation error) from $M_0, \ldots M_d$ use cross validation.

Best Subset Selection (cont)

- Best subset selection hard to apply when the dimension d is large.
- Best subset selection may lead to overfiting and high variance when d is large.
  - The search space is too large $\rightarrow$ more likely to find a model with good performance on the training set but bad on the test set.
- Therefore, the stepwise selection methods, which limit the search space, are used instead of best subset selection.

Forward stepwise selection

- Let M_0 is null model, contains no variables.
  - This model uses sample mean to predict for each data point
- for $k = 0, 2, \ldots, d-1$
  - Consider all $d-k$ models created by adding a variable to M_k
  - Choose best model (has minimum $L(\theta)$) from $d-k$ models, call it M_{k+1}.
- Choose best model (has minimum validation error) from $M_0, \ldots M_d$ use cross validation.

Forward stepwise selection (cont)

- Forward stepwise selection has a much smaller search space than best subset selection
  - The number of cases to be considered is $1 + d(d+1)/2 = O(d^2)$ compared to $O(2^d)$
- Forward stepwise selection is a greedy method, so it is not guaranteed to find the best model like best subset selection

Backward stepwise selection

- Let M_d is full model, contains all the variables.
- for $k = d, d-1, \ldots, 1$
  - Consider all k models created by removing a variable from M_k
  - Choose best model (has minimum $L(\theta)$) from k models, call it M_{k-1}.
- Choose best model (has minimum validation error) from $M_0, \ldots M_d$ use cross validation.

Backward stepwise selection (cont)

- Similar to forward stepwise selection, backward stepwise selection also has a much smaller search space than best subset selection.
  - The number of cases to be considered is $1 + d(d+1)/2 = O(d^2)$ compared to $O(2^d)$
- Backward stepwise selection is also a greedy method, so it is not guaranteed to find the best model like best subset selection
- Backward stepwise selection needs $n > d$ (the number of data points n is greater than the number of variables d) to fit full model
  - If $n < d$, we can use forward stepwise selection.

Regularization

Sensitivity and regularization

- Suppose we have a model $\hat{y} = f_\theta(x) = \theta^T x$
- Is θ_i is large, then $\hat{y}_i$ will be very sensitive with x_i
  - A small change in x_i lead to a large change in $\hat{y}_i$
- Large sensitivity can lead to overfit and poor generalization
- We measure the magnitude of θ use a regularizer function $r : \mathbb{R}^d \to \mathbb{R}$
- $r(\theta)$ is a measure for the magnitude of θ
  - Square regularizer (l_2)

$$r(\theta) = ||\theta||_2 = \theta_1^2 + \cdots + \theta_d^2$$

  - Absolute regularizer (l_1)

$$r(\theta) = ||\theta||_1 = |\theta_1| + \cdots + |\theta_d|$$

Regularized empirical risk minimization

- The model should fit with the given data well, i.e. empirical risk $L(\theta)$ is small

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(\theta^T x_i, y_i)$$

- The model should also not be too sensitive, i.e. $r(\theta)$ is small
- To balance these two goals, we use regularized empirical risk

$$L(\theta) + \lambda r(\theta)$$

  In there: $\lambda > 0$ is regularization parameter (or hyper-parameter)

- Regularized empirical risk minimization (RERM): choose θ to regularized empirical risk is minimized

Regularized empirical risk minimization (cont)

- With $\lambda = 0$, RERM becomes ERM
- With $\lambda = \infty$, $\theta = 0$
- RERM generate a family of models for different λ
- We will choose several (dozen) values of θ, usually logarithmic distances over a large range of values.
- We use cross validation to choose the best model
- In general, we will choose the maximum λ value for test error near the minimum value (so that it is less sensitive and generalizes well)

Ridge regression (L_2 regularization)

- Choose λ to minimize

$$L(\theta) + \lambda||\theta||_2^2$$

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{n}(y_i - \theta^T x_i)^2 \\ &= (y - X\theta)^T(y - X\theta) \end{aligned}$$

$$cost(\theta) = L(\theta) + \lambda||\theta||_2^2 = (y - X\theta)^T(y - X\theta) + \lambda\theta^T\theta$$

Gradient of ridge regression cost

$$
\begin{aligned}
\nabla[cost(\theta)] &= \nabla[L(\theta) + \lambda||\theta||_2^2] \\
&= \nabla[(y - X\theta)^T(y - X\theta) + \lambda\theta^T\theta] \\
&= \nabla[(y - X\theta)^T(y - X\theta)] + \lambda\nabla[\theta^T\theta] \\
&= -2X^T(y - X\theta) + 2\lambda\theta \\
&= -2X^T(y - X\theta) + 2\lambda I_{d+1}\theta
\end{aligned}
$$

Analytical solution for ridge regression

- The optimal solution $\hat{\theta}$ satisfies:

$$\nabla[cost(\hat{\theta})] = -2X^T(y - X\hat{\theta}) + 2\lambda I_{d+1}\hat{\theta} = 0$$

$$-X^T y + X^T X\hat{\theta} + \lambda I_{d+1}\hat{\theta} = 0$$

$$X^T X\hat{\theta} + \lambda I_{d+1}\hat{\theta} = X^T y$$

$$\hat{\theta} = (X^T X + \lambda I_{d+1})^{-1} X^T y$$

Analytical solution for ridge regression (cont)

$$\hat{\theta} = (X^T X + \lambda I_{d+1})^{-1} X^T y$$

- If $\lambda = 0$, $\hat{\theta}^{ridge} = (X^T X)^{-1} X^T y = \hat{\theta}^{LS}$
  - $(X^T X)$ is the matrix of $(d+1) \times (d+1)$
  - The complexity of calculating $(X^T X)^{-1}$ is $O((d+1)^3) = O(d^3)$
- If $\lambda = \infty$, $\hat{\theta}^{ridge} = 0$
  - With $\lambda > 0$, $(X^T X + \lambda I_{d+1})$ is always invertible

Stochastic gradient descent for ridge regression

$t = 0$

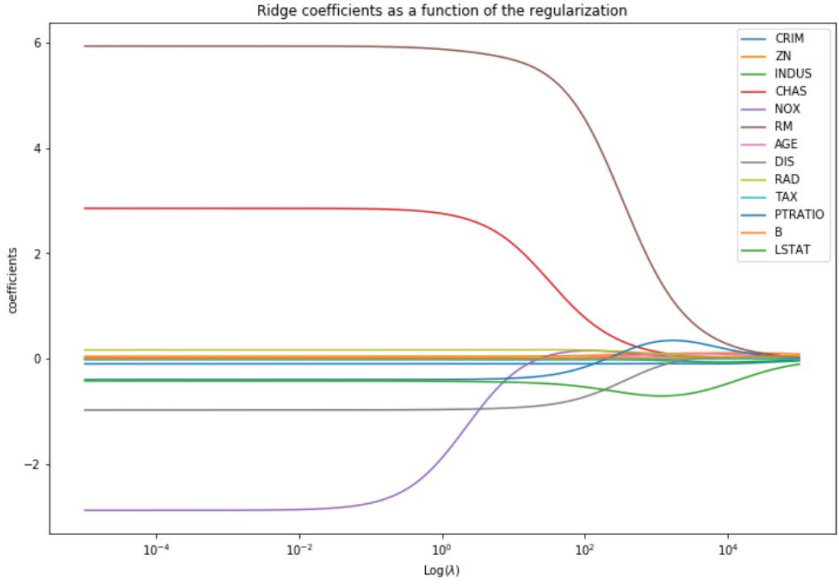
Initialize $\theta^{(t)}$ randomly

repeat

- foreach $i = 1, 2, \ldots, n$ (random order)
  - $partial[i] = -x_i(y_i - x_i^T \theta_i^{(t)}) + \frac{\lambda}{n}\theta_i^{(t)}$
  - $\theta^{(t+1)} = \theta^{(t)} - \eta \cdot partial[i]$
- $t \leftarrow t + 1$

until $||\theta^{(t)} - \theta^{(t-1)}|| \leq \delta$

Ridge regression on housing data



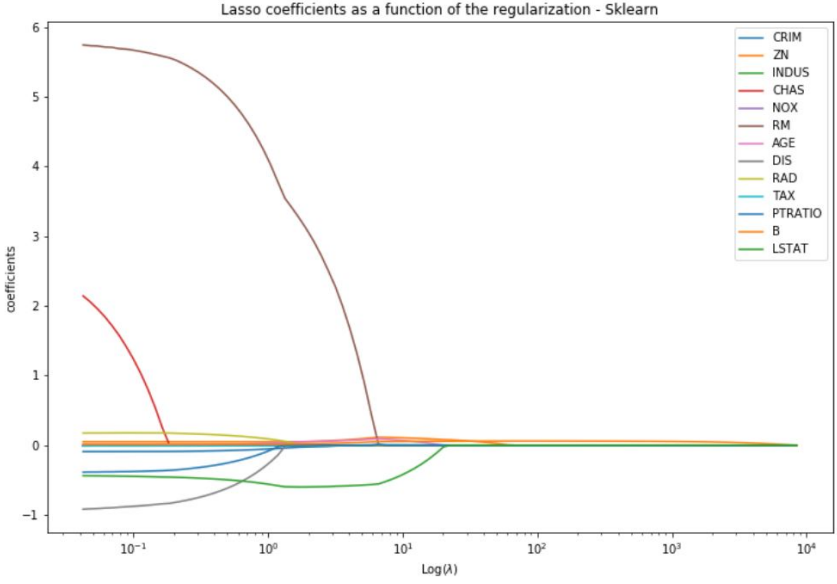
Lasso regression (L_1 regularization)

- Choose λ to minimize

$$L(\theta) + \lambda ||\theta||_1$$

- Lasso regression has no analytical solution
- Some algorithms for lasso regression:
  - Least angle regression (LARS): "Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, Least angle regression, Annals of Statistics, Volume 32, Number 2 (2004), 407-499."
  - Coordinate descent: "Jerome Friedman, Trevor Hastie, and Robert Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics, Volume 9, Issue 3 (2008), 432-441"
  - ...
- L_1 regularization often leads to a sparse solution, so it is considered a feature selection method.

Lasso regression on housing data



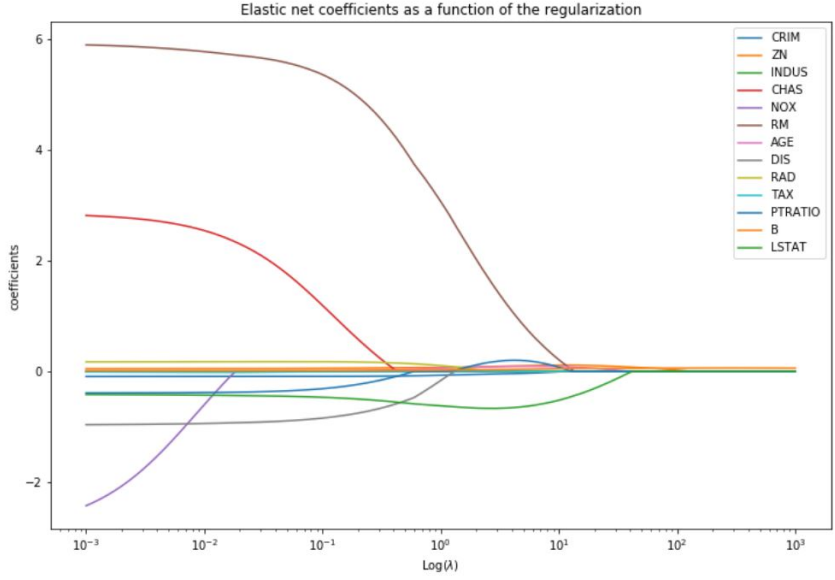
Elastic net regression (kết hợp L_1 và L_2 regularization)

- Choose λ, α to minimize

$$L(\theta) + \lambda(\frac{1-\alpha}{2}||\theta||_2^2 + \alpha||\theta||_1)$$

- Elastic net regression also has no analytical solution
- Algorithm for elastic net regression:
  - LARS-EN: "Hui Zou and Trevor Hastie, Regularization and variable selection via the elastic net, Journal of the royal statistical society: series B (statistical methodology), Volume 67, Issue 2 (2005), 301-320."

ElasticNet regression on housing data



Summary

- Linear regression
- Validation
  - Holdout
  - Cross validation
- Linear model selection
  - Best subset selection
  - Forward/Backward stepwise selection
- Regularization
  - Ridge regression
  - Lasso regression
  - ElasticNet regression

References

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, Statistical Learning, In An Introduction to Statistical Learning with Applications in R, 2nd Edition, Springer, 2021.