Linear Methods

Shyue Ping Ong

Aiiso Yufeng Li Family Department of Chemical and Nano Engineering
University of California, San Diego
http://materialsvirtuallab.org

Overview

Preliminaries

Preliminaries

- We will go very deep into linear models.
- Most of you probably have seen linear models in some form, but we will start from scratch to further illustrate key concepts such as bias and variance.
- Using linear examples, we will discuss the basic machine learning concepts of model selection, cross-validation, and loss functions.

Notation

- Capital letters, e.g., X denote variables.
- Lower-case letters e.g., x, denote observations.
- Dummy index j denotes different variables, e.g., X_j
- Dummy index i denotes different observations, e.g., x_i
- Bolded variables are vector/matrices, e.g., $\mathbf{y}$, $\mathbf{X}$

Linear regression

Simplest possible model between target and feature

$$Y = f(X_1, X_2, ..., X_p) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

X_j can be:

- Quantitative inputs
- Transformations of quantitative inputs, e.g., log, exp, powers, etc. Basis expansions, e.g., $X_2 = X_1^2$, $X_3 = X_1^3$
- Interactions between variables, e.g., $X_1 X_2$
- Encoding of levels of inputs

Supervised learning

- Given a set of paired observations $\{x_{ij}, y_i\}$, what are the model parameters (in this case, the coefficients β_j) that are "optimal"?
- "Optimal" is typically defined as minimization of some **loss function** (also known as **cost function**) that measures the error of the model.

Least squares regression

Consider the simple case of

$$Y = \beta_0 + \beta_1 X_1$$

In least squares regression, the loss function is defined as the sum squared error given the N observations:

$$\begin{aligned} L(Y, \hat{f}(X)) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_{i1})^2 \end{aligned}$$

What are the optimal parameters β_0 and β_1?

$$
\begin{aligned}
\frac{\partial L}{\partial \beta_0} &= \sum_{i=1}^{N} 2(y_i - \beta_0 - \beta_1 x_{i1})(-1) = 0 \\
\implies & \quad \sum_{i=1}^{N} y_i = N\beta_0 + \sum_{i=1}^{N} \beta_1 x_{i1} \\
\implies & \quad \beta_0 = \bar{y} - \beta_1 \bar{x_1} \\
\frac{\partial L}{\partial \beta_1} &= \sum_{i=1}^{N} 2(y_i - \beta_0 - \beta_1 x_{i1})(-x_{i1}) = 0 \\
\implies & \quad \beta_1 = \frac{\sum_{i=1}^{N} x_{i1} y_i - N\bar{x_1}\bar{y}}{\sum_{i=1}^{N} x_{i1}^2 - N\bar{x_1}^2}
\end{aligned}
$$

Reformulating the general multiple linear regression as a vector equation...

Considering N observations of

$$y_i = \beta_0 + \beta_1 x_{i1} + +\beta_2 x_{i2} + ... + \beta_p x_{ip}$$

Let

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_p \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & & & & \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix},$$

So,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$$

Note that $\mathbf{y}$ is a $N \times 1$ vector, $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector, and $\mathbf{X}$ is a $N \times (p+1)$ matrix.

Reformulating the general multiple linear regression as a vector equation...

$$L = RSS = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Assuming (for the moment) that $\mathbf{X}$ has full column rank, and hence $\mathbf{X}^T\mathbf{X}$ is positive definite, It can be shown using the same principles that the following unique solution for $\boldsymbol{\beta}$ is:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ \hat{\mathbf{y}} &= \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

Graphic representation of MLR with two dependent variables

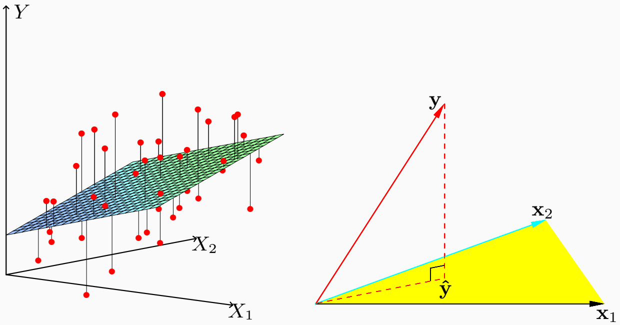


Figure 1: MLR minimizes sum square of residuals. The projection $\hat{\mathbf{y}}$ represents the vector of the least squares predictions onto the hyperplane spanned by the input vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. [1].

Validity of least squares criterion

- Observations are independently drawn at random.
- Variance of $\mathbf{y}$ is constant given by σ^2.

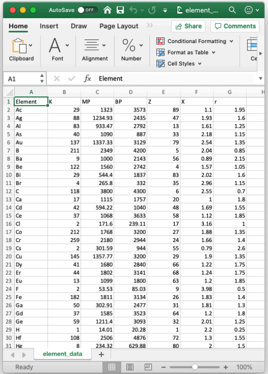
$$\mathrm{var}(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$$

- and σ is estimated using:

$$\sigma^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Example materials data

- Target: Bulk modulus of elements (from Materials Project)
- Candidate features:
    - Melting point (MP)
    - Boiling point (MP)
    - Atomic number (Z)
    - Electronegativity (χ)
    - Atomic radius (r)
- Question: Why these features?
- We will also add some transformations of these inputs, i.e., the square and square root of the electronegativity and the atomic radius.



Element	K	MP	BP	Z	X	r
Ac	29	1323	3573	89	1.1	1.95
Ag	88	1234.93	2435	47	1.93	1.6
Al	83	933.47	2792	13	1.61	1.25
As	40	1090	887	33	2.18	1.15
Au	137	1337.33	3129	79	2.54	1.35
B	211	2349	4200	5	2.04	0.85
Ba	9	1000	2143	56	0.89	2.15
Be	122	1560	2742	4	1.57	1.05
Bi	29	544.4	1837	83	2.02	1.6
Br	4	265.8	332	35	2.96	1.15
C	118	3800	4300	6	2.55	0.7
Ca	17	1115	1757	20	1	1.8
Cd	42	594.22	1040	48	1.69	1.55
Ce	37	1068	3633	58	1.12	1.85
Cl	2	171.6	239.11	17	3.16	1
Co	212	1768	3200	27	1.88	1.35
Cr	259	2180	2944	24	1.66	1.4
Cs	2	301.59	944	55	0.79	2.6
Cu	145	1357.77	3200	29	1.9	1.35
Dy	41	1680	2840	66	1.22	1.75
Er	44	1802	3141	68	1.24	1.75
Eu	13	1099	1800	63	1.2	1.85
F	2	53.53	85.03	9	3.98	0.5
Fe	182	1811	3134	26	1.83	1.4
Ga	50	302.91	2477	31	1.81	1.3
Gd	37	1585	3523	64	1.2	1.8
Ge	59	1211.4	3093	32	2.01	1.25
H	1	14.01	20.28	1	2.2	0.25
Hf	108	2506	4876	72	1.3	1.55
Hg	8	234.32	629.88	80	2	1.5

Using pandas for easy data manipulation

```python
import pandas as pd

# Read in data and set first column as index.
data = pd.read_csv("element_data.csv", index_col=0)
# Generate transformations as additional columns.
data["X^2"] = data["X"] ** 2
data["sqrt(X)"] = data["X"] ** 0.5
data["r^2"] = data["r"] ** 2
data["sqrt(r)"] = data["r"] ** 0.5
# Define our features, which is all the columns
# excluding K, which is the target.
features = [c for c in data.columns if c != "K"]
x = data[features]
y = data["K"]
```

MLR in scikit-learn

```
from sklearn import linear_model

reg = linear_model.LinearRegression()
reg.fit(x, y)
print(ref.coef_)
print(reg.intercept_)
```

- Note that x should contain the features only; there is no need to add a 1 column for the intercept. By default, the parameter `fit_intercept` in `sklearn.linear_model.LinearRegression` is True. You can set it to False to do a MLR without intercept.
- Documentation: link.

Hypothesis Testing for Coefficients

- To derive insights into a model, we often want to know which among the input parameters are the most relevant to the target.
- Under assumptions of the errors in y follow a Gaussian distribution $N(0, \sigma^2)$, the errors in $\hat{\boldsymbol{\beta}}$ also have a Gaussian distribution $N(\beta, (\mathsf{X}^T\mathsf{X})^{-1}\sigma^2)$
- Hypothesis testing can be carried out for whether a particular β_j is 0 using the following test statistic:

$$t_j = \frac{\hat{\boldsymbol{\beta}}_{\boldsymbol{j}}}{\sigma\sqrt{v_j}}$$

  where v_j is the jth diagonal element of $(\mathsf{X}^T\mathsf{X})^{-1}$. t_j has a t distribution with $N - p - 1$ degrees of freedom (dof).

Hypothesis Testing for Groups of Coefficients

- More often, we want to test groups of coefficient for significance. E.g., to the k levels of a categorical variable.
- We will use the following F statistic:

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/(p_1 - p_0)}{\mathrm{RSS}_1/(N - p_1 - 1)}$$

  where RSS_0 is the RSS of the larger model with $p_0 + 1$ parameters and RSS_1 is the RSS of the smaller model with $p_1 + 1$ parameters with $p_0 - p_1$ parameters set to zero. The F statistic has a distribution of $F_{p_1 - p_0, N - p_1 - 1}$.

Gauss-Markov Theorem

- Consider the estimator $\hat{\theta}$ for a variable θ.

$$\begin{aligned} \mathrm{MSE} &= E(\hat{\theta} - \theta)^2 \\ &= \mathrm{var}(\hat{\theta}) + [E(\hat{\theta}) - \theta]^2 \end{aligned}$$

- The MSE can be broken down into the variance of the estimate itself and the square of the bias.

Gauss-Markov Theorem

The least squares estimator has the smallest variance among all linear *unbiased* estimators.

- However, there can be estimators that are biased with smaller MSE.

Model selection

Model performance

- We will take a brief digression into model assessment and selection before continuing to other linear methods.
- Model performance is related to its performance on *independent test data*, i.e., one cannot simply report a model's performance on training data alone.
- Note that this section is deliberately limited to high-level concepts that are universally applicable to many different models.

Typical measures of model performance

- Mean squared error (MSE):

$$L(Y, \hat{f}(X)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

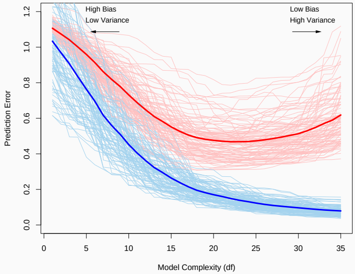
- Mean absolute error (MAE):

$$L(Y, \hat{f}(X)) = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(x_i)|$$

- Test error: L over independent test set.
- Training error: L over training set.

Training and test errors with model complexity

- Model complexity increases as the number of parameters increases (e.g., number of independent variables in MLR).
- Training errors **always** decrease with increasing model complexity.
- However, test errors do not have a monotonic relationship with model complexity. Test errors are high when model complexity is too low (underfitting) or too high (overfitting).



Under-fitting versus over-fitting

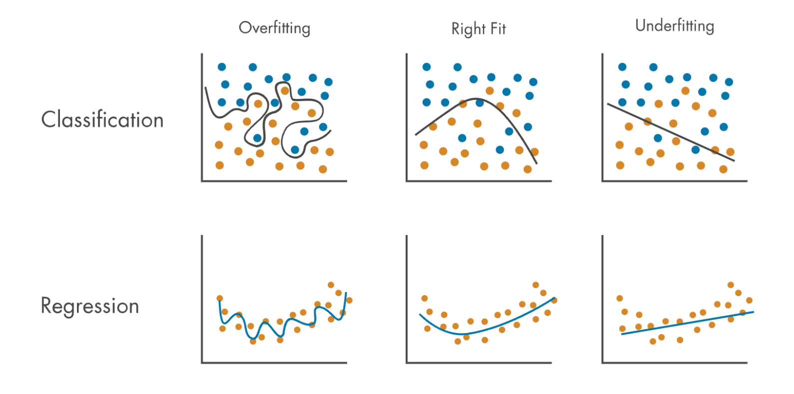


Figure 2: Source: Mathworks

Training, validation and test data

- **Model selection**: estimating the performance of different models in order to choose the best one.
- **Model assessment**: having chosen a final model, estimating its prediction error (generalization error) on new data.
- In a data-rich situation, divide data into three parts:
    - Training set: For training the model.
    - Validation set: For estimating prediction error to select the model.
    - Test set: For assessing the generalization error of the final model.
- Typical training:validation:test splits are 50:25:25 or 80:10:10, or 90:5:5.
- Note that at no point in the model fitting and selection process should the test set be "seen".

K-fold cross validation (CV)

- Simplest and most widely used approach for model validation.
- Data set is split into K buckets (usually by random).
- Typical values of K is 5 or 10. $K = N$ is known as "leave-one-out" CV.

Train	Train	Validate	Train	Train

- CV score is computed on the validate data set after training on the train data:

$$CV(\hat{f}^{-k(i)}, \alpha) = \frac{1}{N_{k(i)}} \sum_{i=1}^{N_{k(i)}} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha))$$

- assuming the k^{th} data bucket has $N_{k(i)}$ data points and $\hat{f}^{-k(i)}$ refers to the model fitted with the k^{th} data left out ($N - N_{k(i)}$ data in fitting).

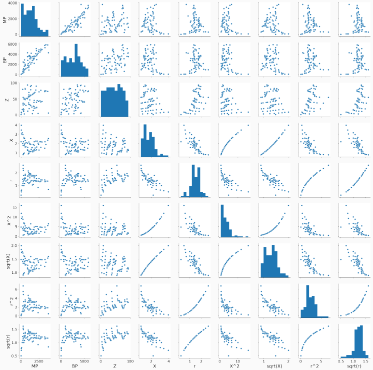
CV in scikit-learn

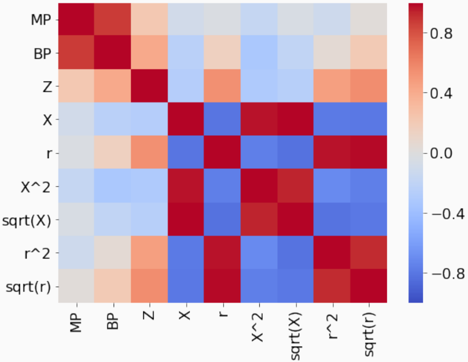
```
from sklearn.model_selection import cross_validate, KFold

kfold = KFold(n_splits=5, shuffle=True, random_state=42)
cv_results = cross_validate(ridge, z, y, cv=kfold)
```

- Note that we have customized the `KFold` object passed to the `cross_validate` method. The reason is that our element data is sorted by default, i.e., non-random. So we want to perform shuffling prior to doing the splits.
- Documentation: link.

Characteristics of the example materials dataset

- Before proceeding further, let us try to tease out some aspects of the dataset.
- Quite clearly, there are correlations between some sets of variables.
- In other words, the input features are **non-orthonormal** with each other.





Loss functions for regression

- We have thus far focused on the squared error loss $L(y, f(x)) = (y - f(x))^2$
- Another common loss function is the absolute error $L(y, f(x)) = |y - f(x)|$
- MSE penalizes outliers with large observed residuals severely, and hence is less robust in data with long-tailed distributions.
- MAE is more robust against outliers.
- Other criteria include the Huber loss:

$$L(y, f(x)) = \begin{cases} (y - f(x))^2 & |y - f(x)| \leq \delta \\ 2\delta(y - f(x) - \delta^2 & otherwise \end{cases}$$

Trevor Hastie, Robert Tibshirani, and Jerome Friedman.
The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.
Springer, New York, NY, 2nd edition edition, 2016.

The End