

# Spring 2022 Data C100/C200 Midterm 2 Reference Sheet

## Ordinary Least Squares

Multiple Linear Regression Model:  $\hat{\mathbb{Y}} = \mathbb{X}\theta$  with design matrix  $\mathbb{X}$ , response vector  $\mathbb{Y}$ , and predicted vector  $\hat{\mathbb{Y}}$ . If there are  $p$  features plus a bias/intercept, then the vector of parameters  $\theta = [\theta_0, \theta_1, \dots, \theta_p]^T \in \mathbb{R}^{p+1}$ . The vector of estimates  $\hat{\theta}$  is obtained from fitting the model to the sample  $(\mathbb{X}, \mathbb{Y})$ .

Concept	Formula	Concept	Formula
Mean squared error	$R(\theta) = \frac{1}{n}   Y - X\theta  _2^2$	Normal equation	$\mathbb{X}^T \mathbb{X} \hat{\theta} = \mathbb{X}^T \mathbb{Y}$
Least squares estimate, if $\mathbb{X}$ is full rank	$\hat{\theta} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}$	Residual vector, $e$	$e = \mathbb{Y} - \hat{\mathbb{Y}}$
		Multiple $R^2$ (coefficient of determination)	$R^2 = \frac{\text{variance of fitted values}}{\text{variance of } y}$
Ridge Regression L2 Regularization	$\frac{1}{n}   Y - X\theta  _2^2 + \alpha   \theta  _2^2$	Squared L2 Norm of $\theta \in \mathbb{R}^d$	$  \theta  _2^2 = \sum_{j=1}^d \theta_j^2$
Ridge regression estimate (closed form)	$\hat{\theta}_{\text{ridge}} = (\mathbb{X}^T \mathbb{X} + n\alpha I)^{-1} \mathbb{X}^T \mathbb{Y}$		
LASSO Regression L1 Regularization	$\frac{1}{n}   Y - X\theta  _2^2 + \alpha   \theta  _1$	L1 Norm of $\theta \in \mathbb{R}^d$	$  \theta  _1 = \sum_{j=1}^d  \theta_j $

## Scikit-Learn

Suppose `sklearn.model_selection` and `sklearn.linear_model` are both imported packages.

Package	Function(s)	Description
<code>sklearn.linear_model</code>	<code>LinearRegression(fit_intercept=True)</code>	Returns an ordinary least squares Linear Regression model.
	<code>LassoCV(fit_intercept=True),</code> <code>RidgeCV(fit_intercept=True)</code>	Returns a Lasso (L1 Regularization) or Ridge (L2 regularization) linear model, respectively, and picks the best model by cross validation.
	<code>model.fit(X, y)</code>	Fits the scikit-learn <code>model</code> to the provided <code>X</code> and <code>y</code> .
	<code>model.predict(X)</code>	Returns predictions for the <code>X</code> passed in according to the fitted <code>model</code> .
	<code>model.coef_</code>	Estimated coefficients for the linear model, not including the intercept term.
	<code>model.intercept_</code>	Bias/intercept term of the linear model. Set to 0.0 if <code>fit_intercept=False</code> .
<code>sklearn.model_selection</code>	<code>train_test_split(*arrays,</code> <code>test_size=0.2)</code>	Returns two random subsets of each array passed in, with 0.8 of the array in the first subset and 0.2 in the second subset.

## Probability

Let  $X$  have a discrete probability distribution  $P(X = x)$ .  $X$  has expectation  $\mathbb{E}[X] = \sum_x xP(X = x)$  over all possible values  $x$ , variance  $\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$ , and standard deviation  $\text{SD}(X) = \sqrt{\text{Var}(X)}$ .

The covariance of two random variables  $X$  and  $Y$  is  $\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$ . If  $X$  and  $Y$  are independent, then  $\text{Cov}(X, Y) = 0$ .

Notes	Property of Expectation	Property of Variance
$X$ is a random variable.		$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$
$X$ is a random variable. $a, b \in \mathbb{R}$ are scalars.	$\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$	$\text{Var}(aX + b) = a^2\text{Var}$
$X, Y$ are random variables.	$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$	$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$
$X$ is a Bernoulli random variable that takes on value 1 with probability $p$ and 0 otherwise.	$\mathbb{E}[X] = p$	$\text{Var}(X) = p(1 - p)$
$Y$ is a Binomial random variable representing the number of ones in $n$ independent Bernoulli trials with probability $p$ of 1.	$E[Y] = np$	$\text{Var}(Y) = np(1 - p)$

Central Limit Theorem

Let  $(X_1, \dots, X_n)$  be a sample of independent and identically distributed random variables drawn from a population with mean  $\mu$  and standard deviation  $\sigma$ . The sample mean  $\overline{X}_n = \sum_{i=1}^n X_i$  is normally distributed, where  $\mathbb{E}[\overline{X}_n] = \mu$  and  $\text{SD}(\overline{X}_n) = \sigma/\sqrt{n}$ .

Parameter Estimation

Suppose for each individual with fixed input  $x$ , we observe a random response  $Y = g(x) + \epsilon$ , where  $g$  is the true relationship and  $\epsilon$  is random noise with zero mean and variance  $\sigma^2$ .

For a new individual with fixed input  $x$ , define our random prediction  $\hat{Y}(x)$  based on a model fit to our observed sample  $(\mathbb{X}, \mathbb{Y})$ . The model risk is the mean squared prediction error between  $Y$  and  $\hat{Y}(x)$ :

$$\mathbb{E}[(Y - \hat{Y}(x))^2] = \sigma^2 + \left(\mathbb{E}[\hat{Y}(x)] - g(x)\right)^2 + \text{Var}(\hat{Y}(x)).$$

Suppose that input  $x$  has  $p$  features and the true relationship  $g$  is linear with parameter  $\theta \in \mathbb{R}^{p+1}$ . Then  $Y = f_\theta(x) = \theta_0 + \sum_{j=1}^p \theta_j x_j + \epsilon$  and  $\hat{Y} = f_{\hat{\theta}}(x)$  for an estimate  $\hat{\theta}$  fit to the observed sample  $(\mathbb{X}, \mathbb{Y})$ .

Gradient Descent

Let  $L(\theta, \mathbb{X}, \mathbb{Y})$  be an objective function to minimize over  $\theta$ , with some optimal  $\hat{\theta}$ . Suppose  $\theta^{(0)}$  is some starting estimate at  $t = 0$ , and  $\theta^{(t)}$  is the estimate at step  $t$ . Then for a learning rate  $\alpha$ , the gradient update step to compute  $\theta^{(t+1)}$  is

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta L(\theta^{(t)}, \mathbb{X}, \mathbb{Y}),$$

where  $\nabla_\theta L(\theta^{(t)}, \mathbb{X}, \mathbb{Y})$  is the partial derivative/gradient of  $L$  with respect to  $\theta$ , evaluated at  $\theta^{(t)}$ .

SQL

SQLite syntax:

```
SELECT [DISTINCT]
    { * | expr [[AS] c_alias]
    {, expr [[AS] c_alias] ... } }
FROM tableref {, tableref}
[[INNER | LEFT ] JOIN table_name
    ON qualification_list]
[WHERE search_condition]
[GROUP BY colname {, colname...}]
[HAVING search_condition]
[ORDER BY column_list]
[LIMIT number]
[OFFSET number of rows];
```

Syntax	Description
<code>SELECT</code> <code>column_expression_list</code>	List is comma-separated. Column expressions may include aggregation functions ( <code>MAX</code> , <code>FIRST</code> , <code>COUNT</code> , etc). <code>AS</code> renames columns. <code>DISTINCT</code> selects only unique rows.
<code>FROM s INNER JOIN t ON cond</code>	Inner join tables <code>s</code> and <code>t</code> using <code>cond</code> to filter rows; the <code>INNER</code> keyword is optional.
<code>FROM s LEFT JOIN t ON cond</code>	Left outer join of tables <code>s</code> and <code>t</code> using <code>cond</code> to filter rows.
<code>FROM s, t</code>	Cross join of tables <code>s</code> and <code>t</code> : all pairs of a row from <code>s</code> and a row from <code>t</code>
<code>WHERE a IN cons_list</code>	Select rows for which the value in column <code>a</code> is among the values in a <code>cons_list</code> .
<code>ORDER BY RANDOM LIMIT n</code>	Draw a simple random sample of <code>n</code> rows.
<code>ORDER BY a, b DESC</code>	Order by column <code>a</code> (ascending by default) , then <code>b</code> (descending).
<code>CASE WHEN pred THEN cons</code> <code>ELSE alt END</code>	Evaluates to <code>cons</code> if <code>pred</code> is true and <code>alt</code> otherwise. Multiple <code>WHEN/THEN</code> pairs can be included, and <code>ELSE</code> is optional.
<code>WHERE s.a LIKE 'p'</code>	Matches each entry in the column <code>a</code> of table <code>s</code> to the text pattern <code>p</code> . The wildcard <code>%</code> matches at least zero characters.
<code>LIMIT number</code>	Keep only the first <code>number</code> rows in the return result.
<code>OFFSET number</code>	Skip the first <code>number</code> rows in the return result.