# Regression (2)

COMP9417 Machine Learning and Data Mining

Term 2, 2022





## Acknowledgements

Material derived from slides for the book
"Elements of Statistical Learning (2nd Ed.)" by T. Hastie,
R. Tibshirani & J. Friedman. Springer (2009)
http://statweb.stanford.edu/~tibs/ElemStatLearn/

Material derived from slides for the book
"Machine Learning: A Probabilistic Perspective" by P. Murphy
MIT Press (2012)
http://www.cs.ubc.ca/~murphyk/MLbook

Material derived from slides for the book
"Machine Learning" by P. Flach
Cambridge University Press (2012)
http://cs.bris.ac.uk/~flach/mlbook

Material derived from slides for the book
"Bayesian Reasoning and Machine Learning" by D. Barber
Cambridge University Press (2012)
http://www.cs.ucl.ac.uk/staff/d.barber/brml

Material derived from slides for the book
"Machine Learning" by T. Mitchell
McGraw-Hill (1997)
http://www-2.cs.cmu.edu/~tom/mlbook.html

Material derived from slides for the course
"Machine Learning" by A. Srinivasan
BITS Pilani, Goa, India (2016)





**Optimization by gradient descent**





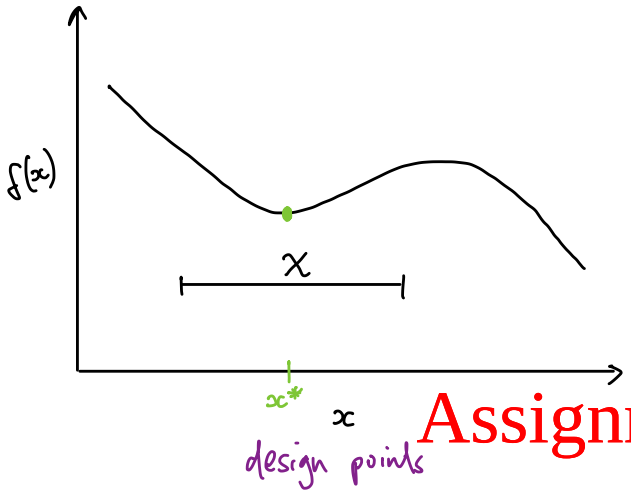
## Optimization

Basic problem[1]:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \end{aligned}$$

- The problem is to find some $\mathbf{x}$ which is called a *design point*, which is a $p$-dimensional vector of values for $p$ *design variables*.
- These values are manipulated by an optimization algorithm to find a *minimizer*, a solution $\mathbf{x}^*$ that minimizes the *objective function* $f$.
- A solution must be an element of the *feasible set* $\mathcal{X}$, and may be subject to additional constraints, called *constrained optimization*.
- Optimization is widely studied and applied in many fields such as engineering, science, economics, ...

[1] See: Kochenderfer and Wheeler (2019).






## Optimization

The formulation is general, for constrained or unconstrained optimization, since we can always replace a problem where we want to *maximize* the objective function $f$:

$$\underset{\mathbf{x}}{\text{maximize}} \quad f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{X}$$

by an equivalent *minimization* expression:

$$\underset{\mathbf{x}}{\text{minimize}} \quad -f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{X}$$







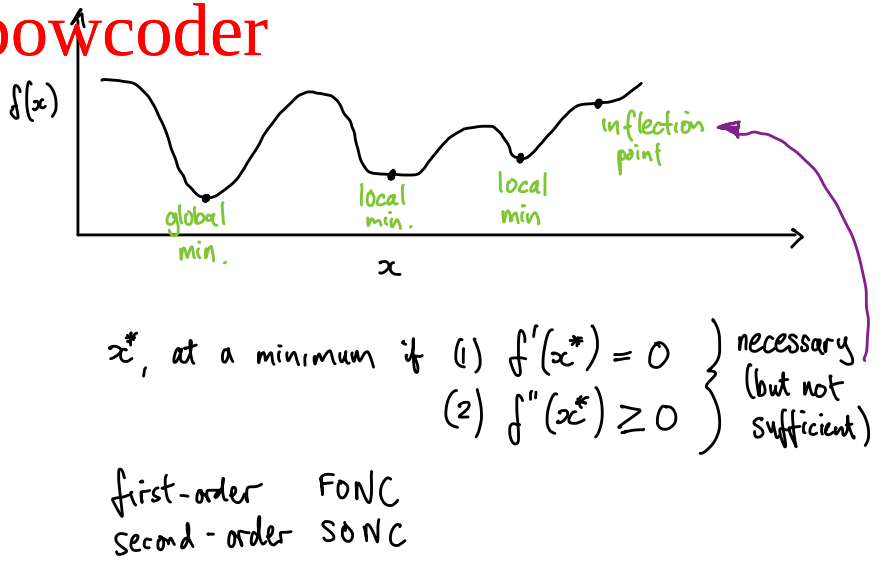
## Optimization

Usually, we would like an optimization algorithm to quickly reach an answer that is close to being the right one.

There are many possible approaches to optimization. In machine learning methods based on finding the derivative, or gradient, are widely used.

- typically, need to minimize a function
  - e.g., error or *loss*
  - optimization is known as *gradient descent* or *steepest descent*
- sometimes, need to maximize a function
  - e.g., probability or *likelihood*
  - optimization is known as *gradient ascent* or *steepest ascent*

Requires function to be *differentiable*.



## What is an optimization algorithm ?

- many kinds of optimization algorithm
- depends on objective function, constraints, data
- approaches based on derivatives or gradients are widely used
- derivatives provide the direction of the search for a minimizer
- may be obtained analytically or estimated numerically
- can also used automatic differentiation
- not all approaches require derivatives ...





## Optimization

A general iterative algorithm [2] to optimise some function $f$:

1. start with initial point $\mathbf{x} = \mathbf{x}_0$
2. select a search direction $\mathbf{g}$, usually to decrease $f(\mathbf{x})$
3. select a step length $\eta$ learning rate
4. set $\mathbf{s} = \eta\mathbf{g}$
5. set $\mathbf{x} = \mathbf{x} + \mathbf{s}$
6. go to step 2, unless convergence criteria are met

For example, could minimize a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$

Note: convergence criteria will be problem-specific.

[2] See: Ripley (1996).







## Least-Squares as Loss Minimization

- consider MSE as a *loss* function
- this is what we want to minimize in OLS regression
- finding the least-squares solution is in effect finding the value of $a$ and $b$ that minimizes $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where $\hat{y}_i = a + bx_i$
- we saw before that the minimum value can be obtained analytically by the usual process of differentiating and equating to 0
- an alternative is to apply an iterative gradient descent approach
- with MSE as a loss function, given data $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$

$$Loss(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(\mathbf{x_i}))^2$$

- note that $Loss$ is a function of $\theta$, and $\hat{y}_i = f_\theta(\mathbf{x_i})$, so here $\theta$ is the parameters $a$ and $b$

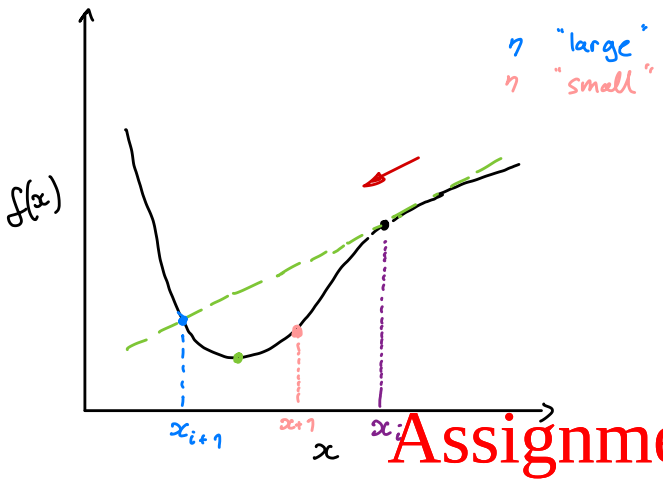




## Least-Squares as Loss Minimization

- the gradient descent alternative to the analytical approach is to take (small) steps that decreases the value of the loss function to be minimised, stopping when we reach a minimum
- recall that at a point the gradient vector points in the direction of greatest increase of a function. So, the opposite direction to the gradient vector gives the direction of greatest decrease
- for univariate linear regression, suppose $g_b$, $g_a$ give the gradient, then the iterative steps are:
  - $b_{i+1} = b_i - \eta \times g_b$
  - $a_{i+1} = a_i - \eta \times g_a$
  - Stop when $b_{i+1} \approx b_i$ and $a_{i+1} \approx a_i$

$0, \ldots, i-1, i, i+1, \ldots$ are the steps in iteration




**Multivariate linear regression**







## Many variables

- Often, we are interesting in modelling the relationship of $Y$ to several other variables
- In observational studies, the value of $Y$ may be affected by the values of several variables. For example, carcinogenicity may be gender-specific. A regression model that ignores gender may find that carcinogenicity to be related to some surrogate variable (height, for example)[3]
- Including more variables can give a narrower confidence interval on the prediction being made
- However, more complex models are not always better

[3] A surrogate variable is a variable for which we have data that replaces one that cannot be observed, or is otherwise not in the dataset.





## Multivariate linear model

- We now have multiple variables to estimate a model of the form $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$
- As before, this linear model is estimated from a sample by the equation $\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$
- With many variables, we often need to ensure the regression does not *overfit* the training data, so we control the $b_i$ by *regularisation*

**Regularisation**







## Parameter Estimation by Optimization

*Regularisation* is a general method to avoid overfitting by applying additional constraints to the weight vector. A common approach is to make sure the weights are, on average, small in magnitude: this is referred to as *shrinkage*.

Recall the setting for regression in terms of loss minimization.

- Can add penalty terms to a *loss* function, forcing coefficients to shrink to zero

$$Y = f_{\theta_0,\theta_1,\ldots,\theta_p}(X_1, X_2, \ldots, X_p) = f_\theta(\mathbf{X})$$

$\theta_j$ large, then $Y$ "sensitive to" $\theta_j$





## Parameter Estimation by Optimization

- MSE as a loss function, given data $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$:

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

and with a penalty function:

error terms; penalty terms; "ridge" L2

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(\mathbf{x_i}))^2 + \lambda\sum_j \theta_j^2$$

Sum: error + size of $\theta$'s; $\lambda$ trades off error and model complexity; user-supplied hyper-parameter

or:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(\mathbf{x_i}))^2 + \lambda\sum_j |\theta_j|$$

"lasso" L1

- Parameter estimation by optimisation will attempt to find values for $\theta_0, \theta_1, \ldots, \theta_p$ s.t. loss $\mathcal{L}(\theta)$ is minimized

variable selection





## Regularised regression

The multivariate least-squares regression problem is often written as an optimisation problem using vector notation:

Loss: $\mathcal{L}(\theta)$ $\mathcal{L}(w)$

$$\mathbf{w}^* = \underset{\mathbf{w}}{\arg\min}\ (\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\mathbf{w})$$

minimize; OLS; $(w_0, w_1, \ldots, w_p)$

where $\mathbf{w}$ represents the parameters as "weights".

The regularised version of this optimisation is then as follows:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\arg\min}\ (\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda||\mathbf{w}||^2$$

ridge

where $||\mathbf{w}||^2 = \sum_i w_i^2$ is the squared norm of the parameters or weights vector $\mathbf{w}$, or, equivalently, the dot product $\mathbf{w}^{\mathrm{T}}\mathbf{w}$; $\lambda$ is a scalar determining the amount of regularisation.

## Regularised regression

This regularised problem still has a closed-form solution:

$$\hat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

where $\mathbf{I}$ denotes the identity matrix. Regularisation amounts to adding $\lambda$ to the diagonal of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, a well-known trick to improve the numerical stability of matrix inversion. This form of least-squares regression is known as *ridge regression*.





## Regularised regression

The alternative "absolute value" form of regularised regression is provided by the *LASSO*, which stands for 'Least Absolute Shrinkage and Selection Operator'[4].

It replaces the ridge regularisation term $\sum_i w_i^2$ with the sum of absolute weights $\sum_i |w_i|$. The result is that some weights are shrunk, but others are set to 0, and so the lasso form of regularised regression favours *sparse solutions*. "some parameters are set to zero"

Unlike ridge regression, the lasso form of regularised regression has to be solved by an iterative optimization algorithm.

[4] (Tibshirani, 1996).







**Bias-Variance Decomposition**

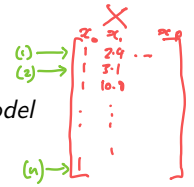




## The Bias-Variance Tradeoff

i i d independent, identically distributed

Linear regression assume $f$ is "linear"
Other models, different function $2 = f(x^{(1)})$



- we can view the issue of error using a *probabilistic model*

$$Y = f(\mathbf{X}) + \epsilon, \qquad \boxed{\epsilon \sim N(0, \sigma^2)}$$

$2 + \epsilon^{(1)}$

- briefly, this model assumes the data we observe was "generated" according to the target function $f$ and Gaussian or normally-distributed noise with zero mean and variance $\sigma^2$ was added to each data point
- this assumption leads to several useful properties maximum likelihood
- for regression, we assume that $f$ is the same as before

## The Bias-Variance Tradeoff

- When comparing unbiased estimators, we would like to select the one with minimum variance
- In general, we would be comparing estimators that have some bias and some variance
- We can combine the bias and variance of an estimator by obtaining the *mean square error* of the estimator, or MSE. This is the average value of squared deviations of an estimated value $V$ from the true value of the parameter $\theta$. That is:

$$\text{MSE} \;=\; \text{Avg. value of } (V-\theta)^2$$

- Now, it can be shown that:

$$\text{MSE} \;=\; (\text{variance}) \;+\; (\text{bias})^2$$

- If, as sample size increases, the bias and the variance of an estimator approaches 0, then the estimator is said to be *consistent*.





## The Bias-Variance Tradeoff

- Since

$$\text{MSE} \;=\; (\text{variance}) \;+\; (\text{bias})^2$$

the lowest possible value of MSE is 0

- In general, we may not be able to get to the ideal MSE of $0$. Sampling theory tells us the minimum value of the variance of an estimator. This value is known as the *Cramer-Rao* bound. So, given an estimator with bias $b$, we can calculate the minimum value of the variance of the estimator using the CR bound (say, $v_{min}$). Then:

$$\text{MSE} \;\geq\; v_{min} + b^2$$

The value of $v_{min}$ depends on whether the estimator is biased or unbiased (that is $b = 0$ or $b \neq 0$)

- It is not the case that $v_{min}$ for an unbiased ($b = 0$) estimator is less than $v_{min}$ for a biased estimator. So, the MSE of a biased estimator can end up being lower than the MSE of an unbiased estimator.







## Decomposition of MSE

"bias" overloaded ① bias term in a linear mod. ② bias part of error ③ inductive bias ④ unfairness in prediction

Suppose $f(\mathbf{x})$ is the value of the (unknown) target function for input $\mathbf{x}$, and $\hat{y} = g(\mathbf{x})$ is the prediction of the learned regression model.

Imagine evaluating predictions $\hat{y}$ of the model $g(\mathbf{x})$ trained on dataset $\mathcal{D}$ of size $n$ sampled at random from the target distribution, where error is based on the squared difference between predicted and actual values.

Averaged over all such datasets $\mathcal{D}$, the MSE can be decomposed like this:

$$\begin{aligned}\text{MSE} &= \mathbb{E}_{\mathcal{D}}[(\hat{y} - f(\mathbf{x}))^2] \\ &= \mathbb{E}_{\mathcal{D}}[(\hat{y} - \mathbb{E}_{\mathcal{D}}[\hat{y}])^2] + (\mathbb{E}_{\mathcal{D}}[\hat{y} - f(\mathbf{x})])^2\end{aligned}$$

variance term; bias² term

Note that the first term in the error decomposition (variance) does not refer to the actual value at all, although the second term (bias) does.





**Some further issues in learning linear regression models**

## What do the Coefficients $b_i$ Mean?

- Consider the two equations:

$$\hat{Y} = a + bX$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2$$

act independently

- $b$: change in $Y$ that accompanies a unit change in $X$
- $b_1$: change in $Y$ that accompanies a unit change in $X_1$ *provided* $X_2$ *remains constant*
- More generally, $b_j$ $(j > 0)$ is the change in $Y$ that accompanies a unit change in $X_j$ provided all other $X$'s are constant
- So: if all relevant variables are included, then we can assess the effect of each one in a controlled manner





## Categoric Variables: $X$'s

- "Indicator" variables are those that take on the values 0 or 1
- They are used to include the effects of categoric variables
- For example, if $D$ is a variable that takes the value 1 if a patient takes a drug and 0 if the patient does not. Suppose you want to know the effect of drug $D$ on blood pressure $Y$ keeping age $(X)$ constant

$$\hat{Y} = 70 + 5D + 0.44X$$

- So, taking the drug (a unit change in $D$) makes a difference of 5 units, provided age is held constant







## Categoric Variables: $X$'s

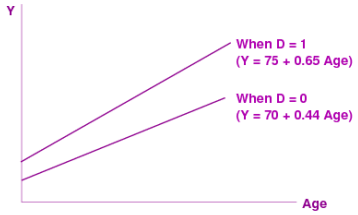
- How do we capture any interaction effect between age and drug intake?
- Introduce a new indicator variable $DX = D \times X$

$$\hat{Y} = 70 + 5D + 0.44X + 0.21DX$$

"feature engineering"

- construct a new feature: D * X
- human modeller based on knowledge

automated feature engineering








## Categoric Variables: $Y$ values

- Sometimes $Y$ values may just be one of two values (say, 0 and 1)
- We can't use the regression model as we described earlier, in which the $Y$'s can take any real value
- But, we can define a new linear regression model in which predicts not the value of $Y$, but what are called the *log odds* of $Y$:

$$\log \text{odds } Y = Odds = b_0 + b_1 X_1 + \cdots + b_p X_p$$

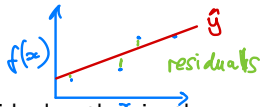
- Once $Odds$ are estimated, they can be used to calculate the probability of $Y$:

$$Pr(Y = 1) = \frac{e^{Odds}}{(1 + e^{Odds})}$$

We can then use the value of $Pr(Y = 1)$ to decide if $Y = 1$

- This procedure is called *logistic regression*

## Is the Model Appropriate ?




- If there is no systematic pattern to the residuals—that is, there are approximately half of them that are positive and half that are negative, then the line is a good fit
- It should also be the case that there should be no pattern to the residual scatter all along the line. If the average size of the residuals varies along the line (this condition is called *heteroscedasticity*) then the relationship is probably more complex than a straight line
- Residuals from a well-fitting line should show an approximate symmetric, bell-shaped frequency distribution with a mean of $0$





## Non-linear Relationships

**A question:** is it possible to do better than the line of best fit?

Maybe. Linear regression assumes that the $(\mathbf{x}_i, y_i)$ examples in the data are "generated" by the true (but unknown) function $Y = f(\mathbf{X})$.

So any training set is a sample from the true distribution $E(Y) = f(\mathbf{X})$.

But what if $f$ is non-linear ?

We may be able to reduce the mean squared error (MSE) value $\sum_i (y_i - \hat{y}_i)^2$ by trying a different function.







## Non-linear Relationships

"linear in the parameters" $\hat{Y} = b_0 + b_1 X_1 + b_2 X_2$

feature engineering

- Some non-linear relationships can be captured in a linear model by a transformation ("trick"). For example, the curved model $\hat{Y} = b_0 + b_1 X_1 + b_2 X_1^2$ can be transformed by $X_2 = X_1^2$ into a linear model. This works for polynomial relationships.
- Some other non-linear relationships may require more complicated transformations. For example, the relationship is $Y = b_0 X_1^{b_1} X_2^{b_2}$ can be transformed into the linear relationship

$$\log(Y) = \log b_0 + b_1 \log X_1 + b_2 \log X_2$$

- Other relationships cannot be transformed quite so easily, and will require full non-linear estimation (in subsequent topics in the ML course we will find out more about these)

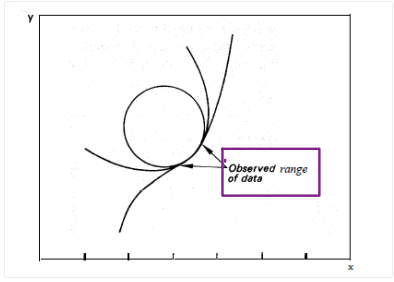




## Non-Linear Relationships

- Main difficulty with non-linear relationships is choice of function
  - How to learn ?
  - Can use a form of gradient descent to estimate the parameters
- After a point, almost any sufficiently complex mathematical function will do the job in a sufficiently small range

"inductive bias"




- Some kind of prior knowledge or theory is the only way to help here.
  - Otherwise, it becomes a process of trial-and-error, in which case, beware of conclusions that can be drawn

## Model Selection

- Suppose there are a lot of variables $X_i$, some of which may be representing products, powers, *etc*.
- Taking all the $X_i$ will lead to an overly complex model. There are 3 ways to reduce complexity:
  1. Subset-selection, by search over subset lattice. Each subset results in a new model, and the problem is one of model-selection
  2. Shrinkage, or *regularization* of coefficients to zero, by optimization. There is a single model, and unimportant variables have near-zero coefficients.
  3. Dimensionality-reduction, by projecting points into a lower dimensional space (this is different to subset-selection, and we will look at it later)





## Model Selection as Search I

variable subset selection

- The subsets of the set of possible variables form a lattice with $S_1 \cap S_2$ as the g.l.b. or meet and $S_1 \cup S_2$ as the l.u.b. or join
- Each subset refers to a model, and a pair of subsets are connected if they differ by just 1 element
- A lattice is a graph, and we know how to search a graph
  - A$^*$, greedy, randomised *etc*.
  - "Cost" of node in the graph: MSE of the model. The parameters (coefficients) of the model can be found

cross-validation

Greedy:

- Historically, model-selection for regression has been done using "forward-selection", "backward-elimination", or "stepwise" methods
  - These are greedy search techniques that either: (a) start at the top of the subset lattice and add variables; (b) start at the bottom of the subset lattice and remove variables; or (c) start at some interior point and proceed by adding or removing single variables (examining nodes connected to the node above or below)







## Model Selection as Search II

hyper-parameter selection

parameter => value within model
hyper-parameter => value for parameter used in training a model

- Greedy selection done on the basis of calculating the *coefficient of determination* (often denoted by $R^2$) which denotes the proportion of total variation in the dependent variable $Y$ that is explained by the model
- Given a model formed with a subset of variables $X$, it is possible to compute the observed change in $R^2$ due to the addition or deletion of some variable $x$
- This is used to select greedily the next best move in the graph-search

To set other *hyper-parameters*, such as shrinkage parameter $\lambda$, can use grid search

"grid search" iterate over hyper-parameters

$\lambda \in \{0, 3, 6, 10\}$

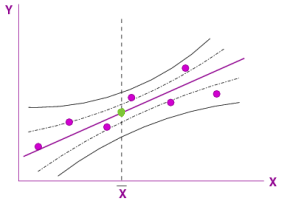
$\eta \in \{0.1, 0.5, 0.8\}$





## Prediction I

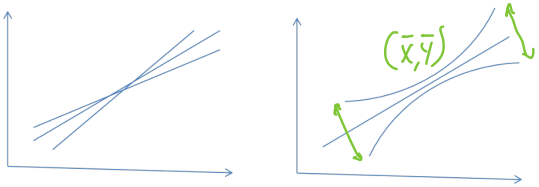
- It is possible to quantify what happens if the regression line is used for prediction:




- The intuition is this:
  - Recall the regression line goes through the mean $(\overline{X}, \overline{Y})$

## Prediction II

- If the $X_i$ are slightly different, then the mean is not going to change much. So, the regression line stays somewhat "fixed" at $(\overline{X}, \overline{Y})$ but with a different slope
- With each different sample of the $X_i$ we will get a slightly different regression line
- The variation in $Y$ values is greater further we move from $(\overline{X}, \overline{Y})$




- MORAL: Be careful, when predicting far away from the centre value
- ANOTHER MORAL: The model only works under approximately the same conditions that held when collecting the data





Model selection

① feature construction e.g., $X_1 * X_2$,

② feature subset selection

(1) hyper-parameter selection } iterative

**Summary**







## Summary

- Linear regression give us a glimpse into many aspects of Machine Learning
- Linear models are only one way to predict numerical quantities
  - Local regression: a nearest-neighbour approach
  - Trees and ensembles: non-parametric models, including regression
  - Neural networks: non-linear models, including regression





Kochenderfer, M. and Wheeler, T. (2019). *Algorithms for Optimization*. MIT Press.

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288.

COMP9417 ML & DM Regression (2) Term 2, 2022 40 / 40