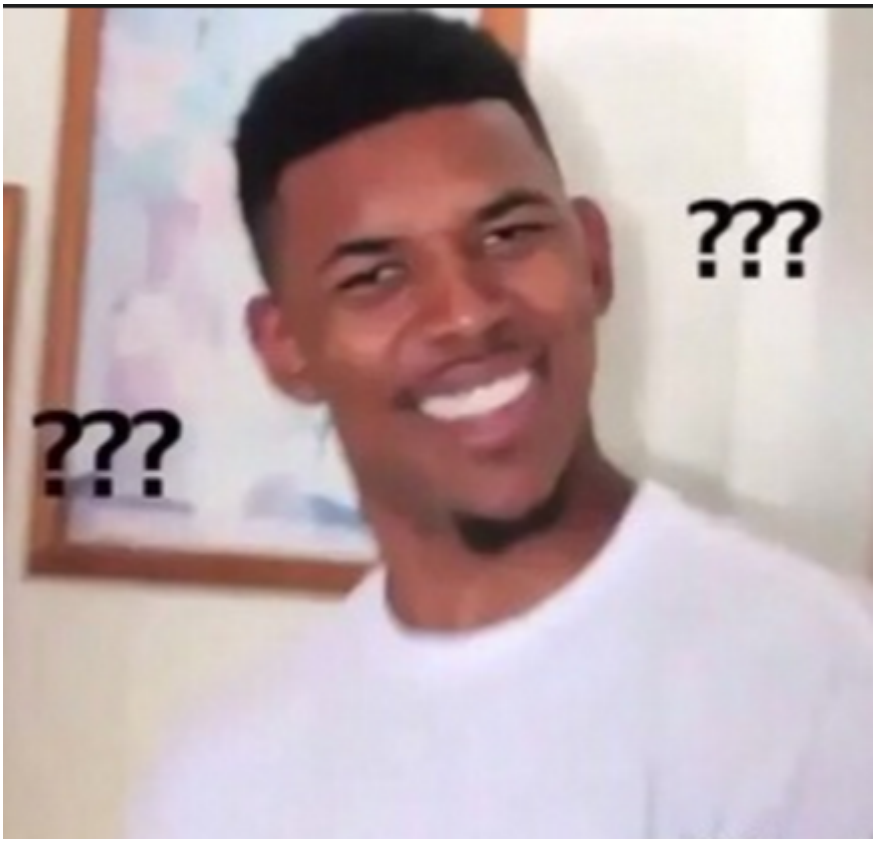
COMP9417 Machine Learning and Data Mining Notes

- **Author: Yunqiu Xu**
- Overview:
  - Notes for UNSW COMP9417 Machine Learning and Data Mining
  - Based on resources from course website
  - Maybe this note is not suitable for fresh meat for that some details are omitted, you can find them in other notes or reference books
  - Related notes can be found in my github
    - Stanford Statistics Learning
    - Stanford CS229 Machine Learning
    - Stanford CS231n Convolutional Neural Networks for Visual Recognition
  - Books I think may be useful
    - Friedman J, Hastie T, Tibshirani R. The elements of statistical learning[M]. Springer, Berlin: Springer series in statistics, 2001.
    - Christopher M.. Bishop. Pattern recognition and machine learning[M]. Springer, 2006.
    - Murphy K P. Machine learning: a probabilistic perspective[M]. MIT press, 2012.
    - 李航. 统计学习方法[J]. 清华大学出版社, 北京, 2012.
    - 周志华. 机器学习, 清华大学出版社, 北京, 2016.

Week 1 Introduction to Machine Lea rning

1.1 What is M(ake) L(ove)



- Supervised Learning V.S. Unsupervised Learning V.S. Semi-supervised Learning
- Models defined by intuition:
    - Geometric models: hyper-planes, linear transformations, distance metrics
    - Probabilistic models: try to reducing uncertainty modelled by means of probability distributions
    - Logical models: easily interpretable logical expressions
    - Note that models can also be defined by algorithmic properties(e.g. Regression, Classification, NN)

- Tasks for ML

The most common machine learning tasks are *predictive*, in the sense that they concern predicting a target variable from features.

- Binary and multi-class classification: categorical target
- Regression: numerical target
- Clustering: hidden target
- Dimensionality reduction: intrinsic structure

Exploratory or *descriptive* tasks are concerned with exploiting underlying structure in the data.

- Deduction V.S. Induction

Deduction: derive specific consequences from general theories

Induction: derive general theories from specific observations

Deduction is well-founded (mathematical logic).

Induction is (philosophically) problematic – induction is useful since it often seems to work – an inductive argument !

Generalisation - the key objective of machine learning

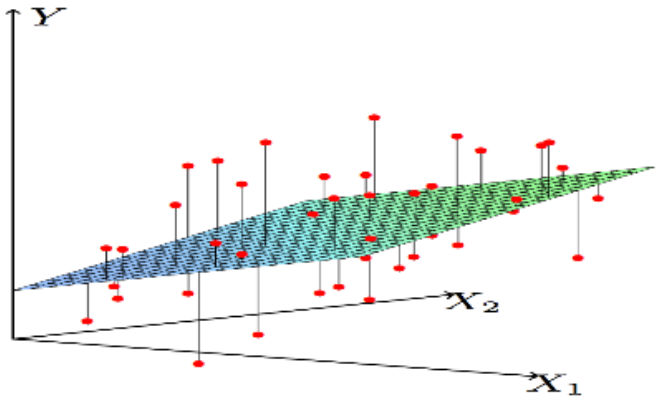
What we are really interested in is *generalising* from the sample of data in our training set. This can be stated as:

The inductive learning hypothesis

> *Any hypothesis found to approximate the target (true) function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.*

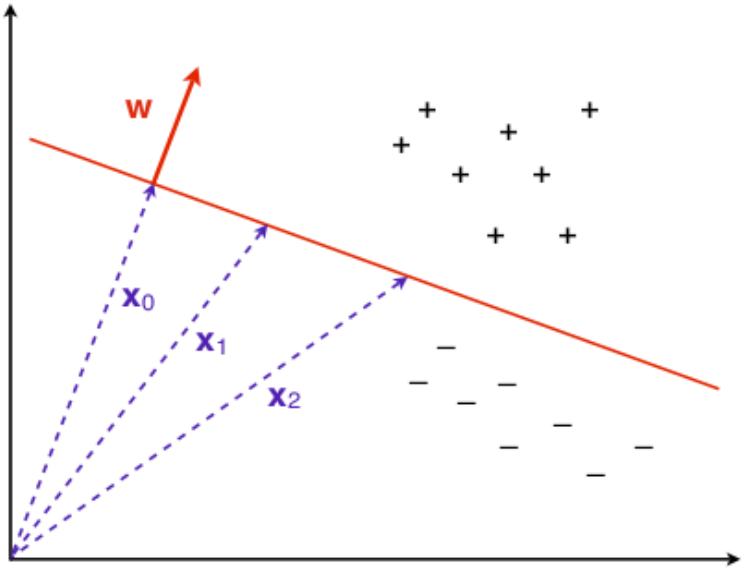
1.2 Different Kinds of Models

- Linear regression: try to minimize MSE



Learning here is by minimizing MSE, i.e., the average of the squared vertical distances of values of Y from the learned function $\hat{Y} = \hat{f}(\mathbf{X})$.

- Nearest Neighbour
- Linear classification:
  + Homogeneous coordinates: see Week 1's tutorial



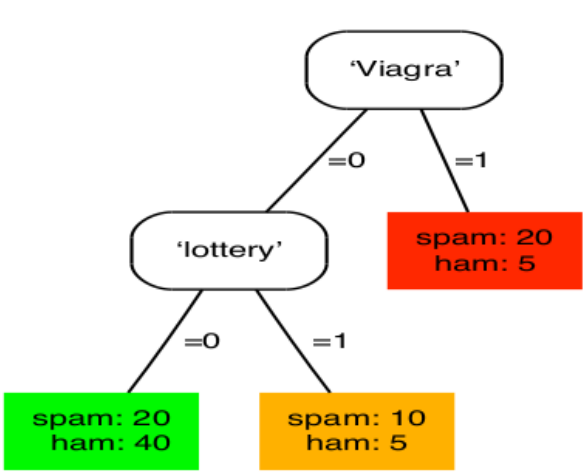
- straight line separates positives from negatives
- defined by $\mathbf{w} \cdot \mathbf{x}_i = t$
- $\mathbf{w}$ is perpendicular to decision boundary
- $\mathbf{w}$ points in direction of positives
- t is the decision threshold

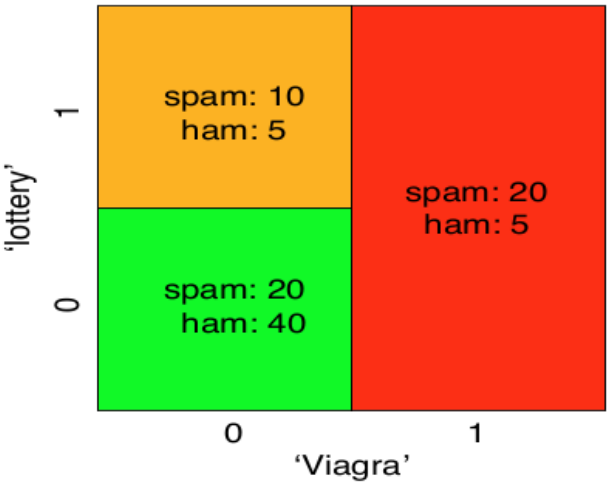
Note: $\mathbf{x}_i$ points to a point on the decision boundary. In particular, $\mathbf{x}_0$ points in the same direction as $\mathbf{w}$, from which it follows that $\mathbf{w} \cdot \mathbf{x}_0 = ||\mathbf{w}||\ ||\mathbf{x}_0|| = t$ (where $||\mathbf{x}||$ denotes the length of the vector $\mathbf{x}$).

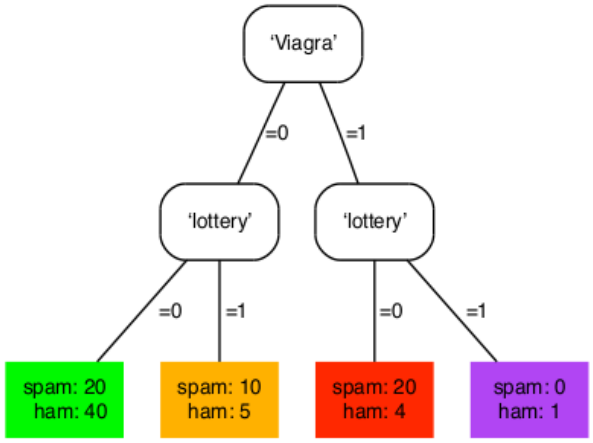
It is sometimes convenient to simplify notation further by introducing an extra constant 'variable' $x_0 = 1$, the weight of which is fixed to $w_0 = -t$.

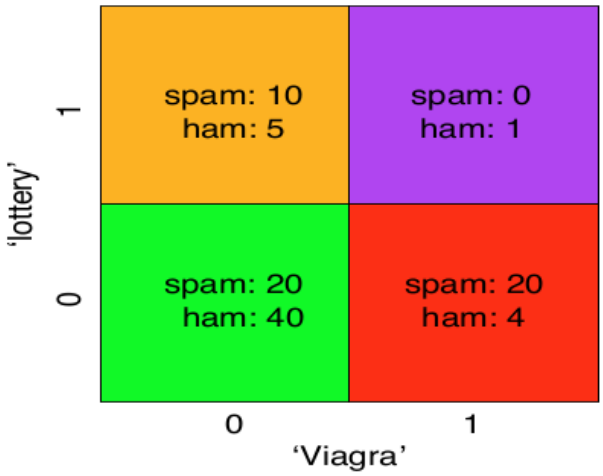
The extended data point is then $\mathbf{x}^\circ = (1, x_1, \ldots, x_n)$ and the extended weight vector is $\mathbf{w}^\circ = (-t, w_1, \ldots, w_n)$, leading to the decision rule $\mathbf{w}^\circ \cdot \mathbf{x}^\circ > 0$ and the decision boundary $\mathbf{w}^\circ \cdot \mathbf{x}^\circ = 0$.

- Bayesian classifier
- Neural Network
- SVM
    - Kernels: make data separable
- Classification tree









- Clustering

1.3 An example of probabilistic model

- Another example can be seen in Week 1's tutorial

'Viagra' and 'lottery' are two Boolean features; Y is the class variable, with values 'spam' and 'ham'. In each row the most likely class is indicated in bold.

Viagra	lottery	$P(Y = \text{spam}\|\text{Viagra}, \text{lottery})$	$P(Y = \text{ham}\|\text{Viagra}, \text{lottery})$
0	0	0.31	**0.69**
0	1	**0.65**	0.35
1	0	**0.80**	0.20
1	1	0.40	**0.60**

- If 'lottery' is in the email(lottery = 1) but we can not determine contains 'Viagra'

$$\begin{aligned} P(Y|\text{lottery}) =& P(Y|\text{Viagra} = 0, \text{lottery})P(\text{Viagra} = 0) \\ &+ P(Y|\text{Viagra} = 1, \text{lottery})P(\text{Viagra} = 1) \end{aligned}$$

- E.G. P(V = 1) = 0.10 and P(V = 0) = 0.90

$$P(Y = \text{spam}|\text{lottery} = 1) = 0.65 \cdot 0.90 + 0.40 \cdot 0.10 = 0.625$$
$$P(Y = \text{ham}|\text{lottery} = 1) = 0.35 \cdot 0.90 + 0.60 \cdot 0.10 = 0.375.$$

- Posterior odds:

$$\frac{P(Y = \text{spam}|\text{Viagra} = 0, \text{lottery} = 0)}{P(Y = \text{ham}|\text{Viagra} = 0, \text{lottery} = 0)} = \frac{0.31}{0.69} = 0.45$$
$$\frac{P(Y = \text{spam}|\text{Viagra} = 1, \text{lottery} = 1)}{P(Y = \text{ham}|\text{Viagra} = 1, \text{lottery} = 1)} = \frac{0.40}{0.60} = 0.67$$
$$\frac{P(Y = \text{spam}|\text{Viagra} = 0, \text{lottery} = 1)}{P(Y = \text{ham}|\text{Viagra} = 0, \text{lottery} = 1)} = \frac{0.65}{0.35} = 1.9$$
$$\frac{P(Y = \text{spam}|\text{Viagra} = 1, \text{lottery} = 0)}{P(Y = \text{ham}|\text{Viagra} = 1, \text{lottery} = 0)} = \frac{0.80}{0.20} = 4.0$$

1

- Likelihood ratio P(X|Y)
  - When to use: when you want to ignore the prior distribution or assume it uniform, otherwise you need to use posterior probabilities
  - We can compute likelihood ratio via marginal likelihoods
  - values in () are full posterior distribution

Y	$P(\text{Viagra} = 1\|Y)$	$P(\text{Viagra} = 0\|Y)$
spam	0.40	0.60
ham	0.12	0.88

Y	$P(\text{lottery} = 1\|Y)$	$P(\text{lottery} = 0\|Y)$
spam	0.21	0.79
ham	0.13	0.87

$$\frac{P(\text{Viagra} = 0|Y = \text{spam})}{P(\text{Viagra} = 0|Y = \text{ham})} \frac{P(\text{lottery} = 0|Y = \text{spam})}{P(\text{lottery} = 0|Y = \text{ham})} = \frac{0.60}{0.88} \frac{0.79}{0.87} = 0.62 \quad (0.45)$$
$$\frac{P(\text{Viagra} = 0|Y = \text{spam})}{P(\text{Viagra} = 0|Y = \text{ham})} \frac{P(\text{lottery} = 1|Y = \text{spam})}{P(\text{lottery} = 1|Y = \text{ham})} = \frac{0.60}{0.88} \frac{0.21}{0.13} = 1.1 \quad (1.9)$$
$$\frac{P(\text{Viagra} = 1|Y = \text{spam})}{P(\text{Viagra} = 1|Y = \text{ham})} \frac{P(\text{lottery} = 0|Y = \text{spam})}{P(\text{lottery} = 0|Y = \text{ham})} = \frac{0.40}{0.12} \frac{0.79}{0.87} = 3.0 \quad (4.0)$$
$$\frac{P(\text{Viagra} = 1|Y = \text{spam})}{P(\text{Viagra} = 1|Y = \text{ham})} \frac{P(\text{lottery} = 1|Y = \text{spam})}{P(\text{lottery} = 1|Y = \text{ham})} = \frac{0.40}{0.12} \frac{0.21}{0.13} = 5.4 \quad (0.67)$$

We see that, using a maximum likelihood decision rule, our very simple model arrives at the *Bayes-optimal* prediction in the first three cases, but not in the fourth ('Viagra' and 'lottery' both present), where the marginal likelihoods are actually very misleading.

1.4 Model Evaluation

- Cross Validation
- Contingency Table: precision / recall / accuracy

Actual Class	Predicted Class	
	Yes	No
Yes	True Positive (TP)	False Negative (FN)
No	False Positive (FP)	True Negative (TN)

- Overfitting V.S. Underfitting
- Bias V.S. Variance

1.4 Week 1's Tutorial

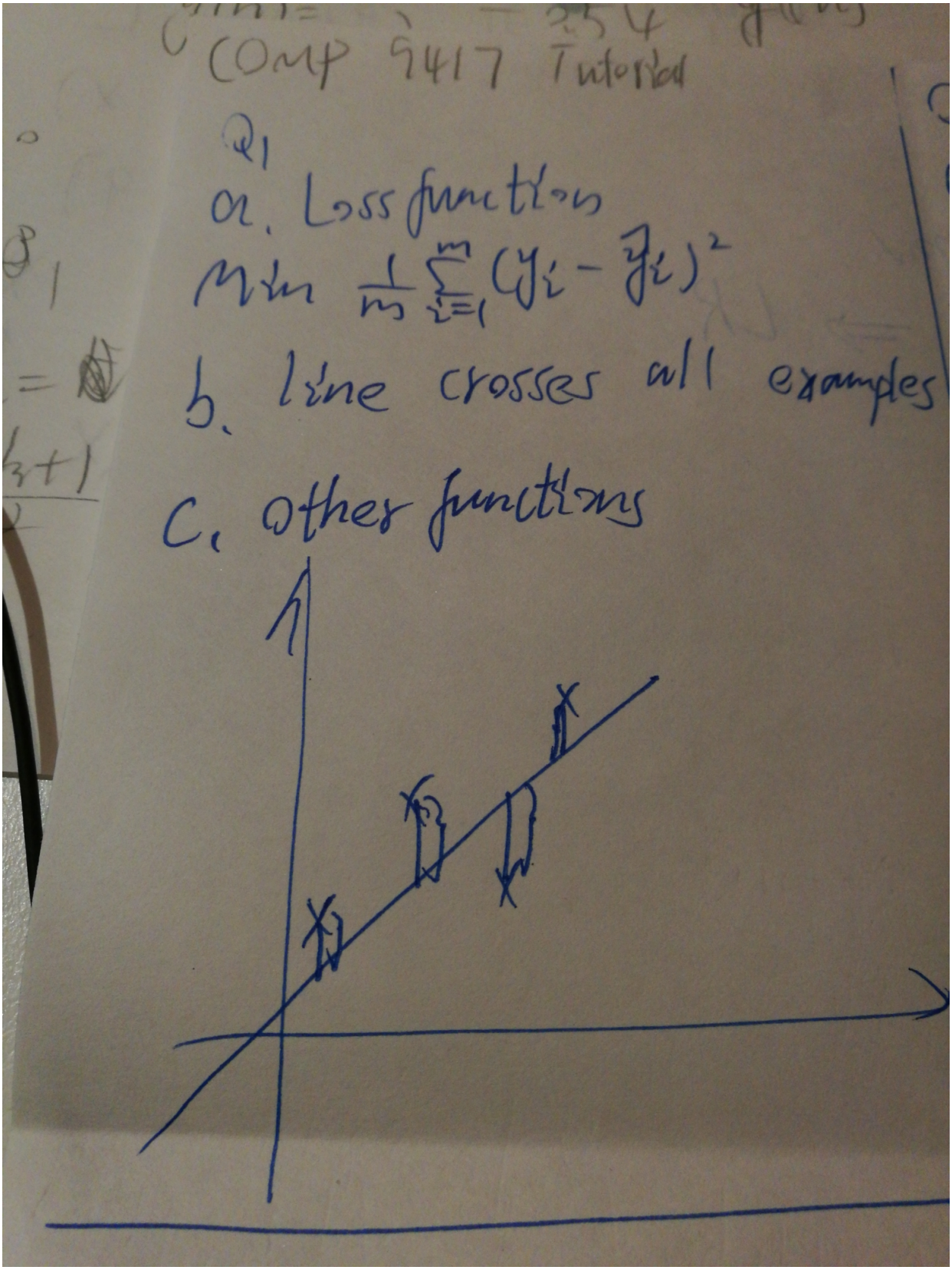
- Q1

Question 1

a) What is the function that Linear Regression is trying to minimize ?

b) Under what conditions would the value of this function be zero ?

c) Can you suggest any other properties of this function ?



- Q2

Question 2 Machine learning has a fair amount of terminology which it is important to get to know.

a) Why do we need features ?

b) What is a the difference between a “task”, a ”model” and a “learning problem” ?

c) Can different learning algorithms be applied to the same tasks and features ?

Tasks are addressed by models, whereas learning problems are solved by learning algorithms that produce models.

Machine learning is concerned with using the right features to build the right models that achieve the right tasks.

Models lend the machine learning field diversity, but tasks and features give it unity.

Does the algorithm require all training data to be present before the start of learning ? If yes, then it is categorised as **batch learning** algorithm.

If however, it can continue to learn a new data arrives, it is an **online learning** algorithm.

If the model has a fixed number of parameters it is categorised as **parametric**.

Otherwise, if the number of parameters grows with the amount of training data it is categorised as **non-parametric**.

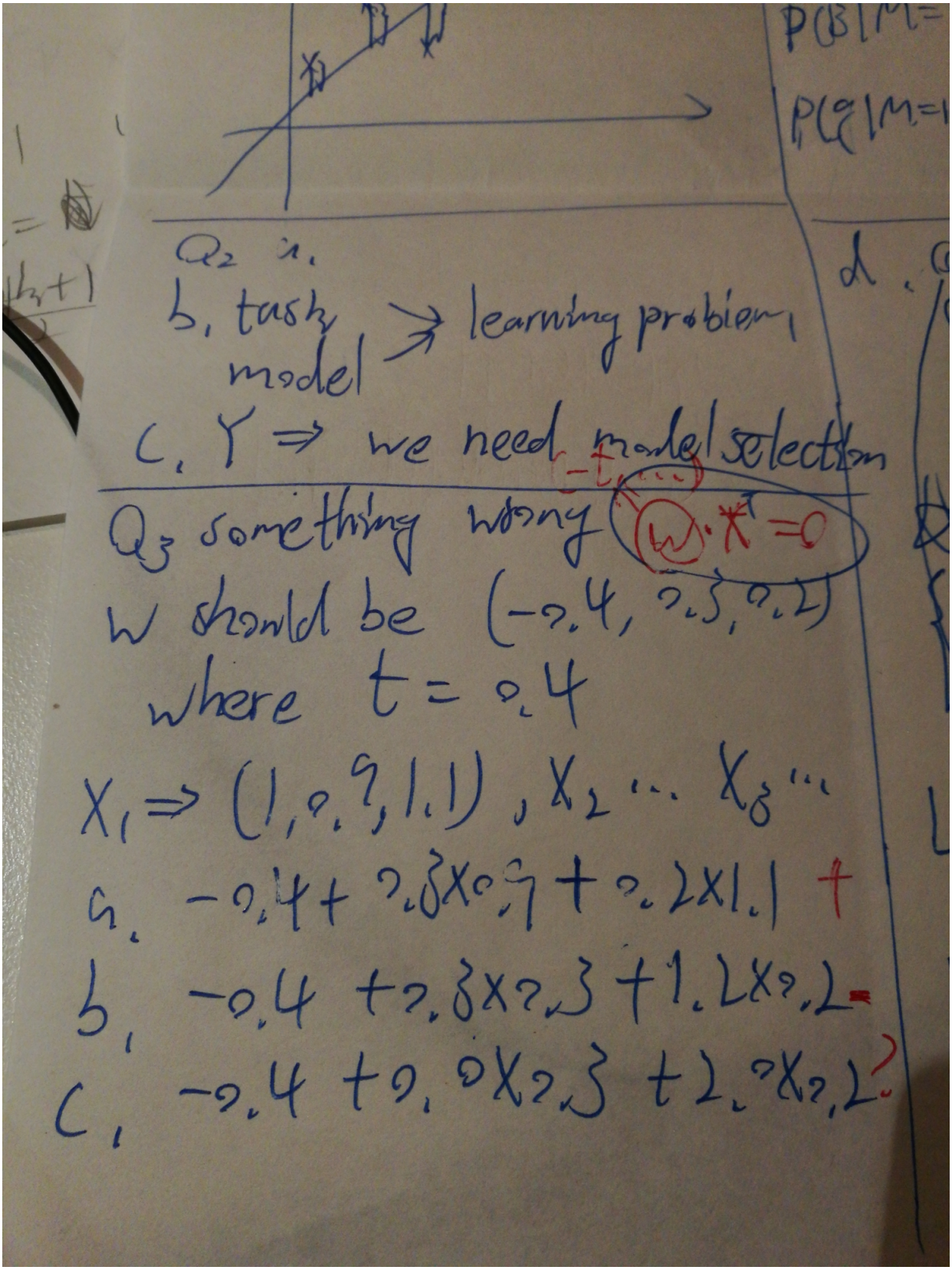
- Q3

Question 3 Suppose you run a learning algorithm that returns a basic linear classifier using the *homogeneous coordinates* representation on a set of training data and obtain the follow weight vector $\mathrm{w} = (0.4, 0.3, 0.2)$. For each of the following examples, what would the classification be using the weight vector w ?

a) $\mathrm{x}_1 = (0.9, 1.1)$?

b) $\mathrm{x}_2 = (0.3, 1.2)$?

c) $\mathrm{x}_3 = (0.0, 2.0)$?


Q2 i.
b. task
model
learning problem
c. Y ⇒ we need model selection
Q3 something wrong
w should be (-0.4, 0.5, 0.2)
where t = 0.4
X1 ⇒ (1, 0.9, 1.1), X2 ... X3 ...
a. -0.4 + 0.5X0.9 + 0.2X1.1 ≠
b. -0.4 + 0.5X0.3 + 1.2X0.2 =
c. -0.4 + 0.5X0.3 + 1.2X0.2 ?

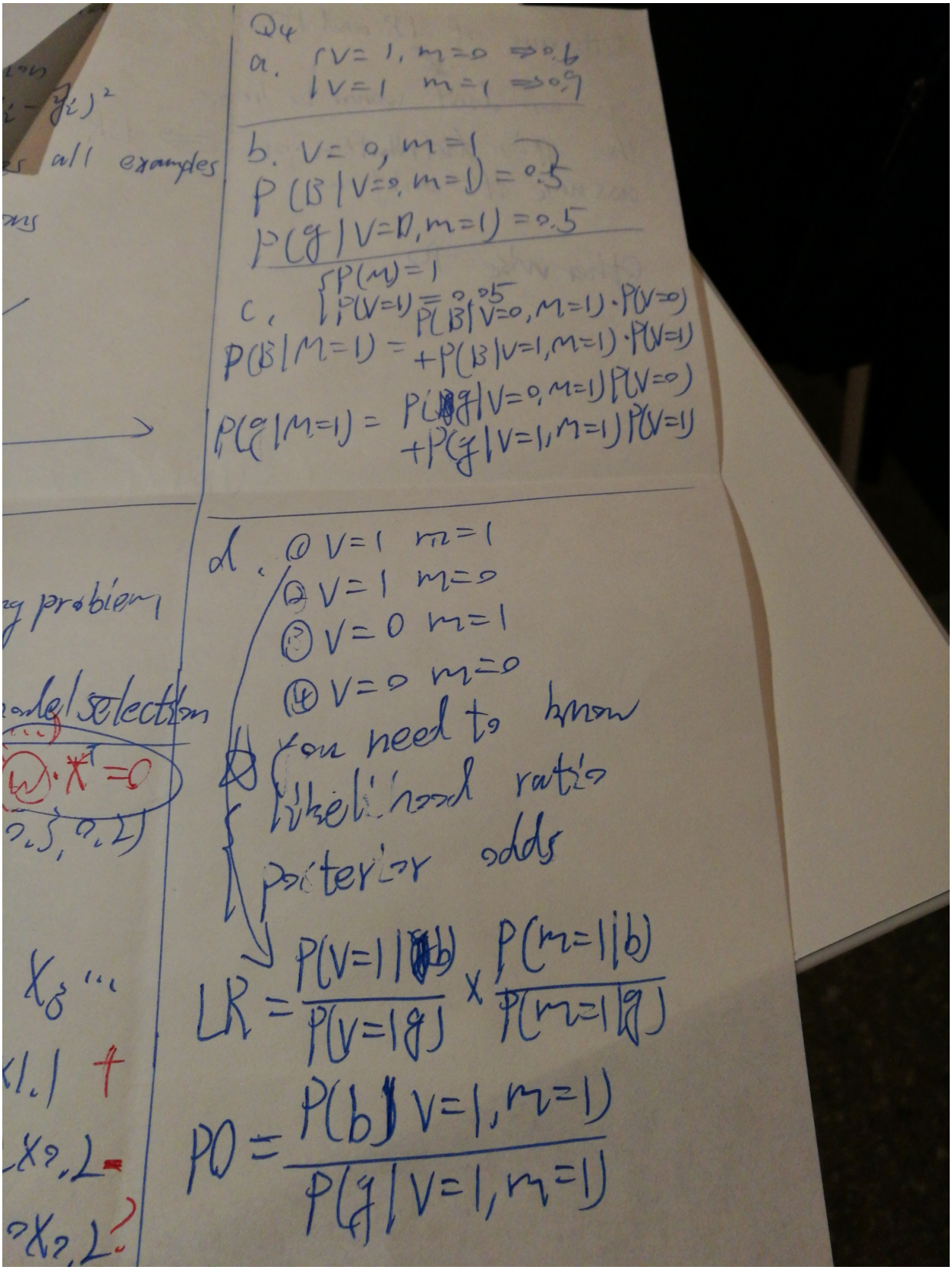
- Q4

a) using the data from Table 1, what two patterns of occurrence of keywords in a text file lead to a prediction of 'business' ?

b) what prediction should be made if we have an occurrence of 'manufacturing' but NOT 'valuation' in a text file ?

c) suppose we are given a text file to classify, and we know that 'manufacturing' occurs in the text file, but we know some words are missing from the file for some reason, and we are uncertain if 'valuation' occurred or not. However, we do know that the probability of 'valuation' occuring in any text file is 0.05. Compute the probability of each class for the given text file.

d) using the values from Table 2 compute the likelihood ratios for each of the four possible patterns of occurrence the keywords



Week 2 Supervised Learning - Regression

2.1 Some Terminologies

- hypothesis V.S. functions

For the class of *symbolic* representations, machine learning is viewed as:

searching a space of **hypotheses** ...

represented in a formal hypothesis language (trees, rules, graphs ...).

For the class of *numeric* representations, machine learning is viewed as:

"searching" a space of **functions** ...

represented as mathematical models (linear equations, neural nets, ...).
Note: in both settings, the models may be probabilistic ...

- statistics V.S. ML
  - linear regression (statistics) determining the "line of best fit" using the least squares criterion
  - linear models (machine learning) learning a predictive model from (big) data under the assumption of a linear relationship between predictor and target variables

2.2 Linear Models

- Numeric attributes and numeric prediction, i.e., regression
- Linear models, i.e. outcome is *linear* combination of attributes

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

- Weights are calculated from the training data
- **Predicted** value for first training instance $\mathbf{x}^{(1)}$ is:

$$b_0 x_0^{(1)} + b_1 x_1^{(1)} + b_2 x_2^{(1)} + \ldots + b_n x_n^{(1)} = \sum_{i=0}^{n} b_i x_i^{(1)}$$

- MSE

$n+1$ coefficients are chosen so that sum of squared error on all instances in training data is minimized

Squared error:

$$\sum_{j=1}^{m}\left(y^{(j)}-\sum_{i=0}^{n}b_i x_i^{(j)}\right)^2$$

2.3 Probability V.S. Statistics

- Probability: reasons from populations to samples
  - This is deductive reasoning, and is usually *sound* (in a logical sense of the word)
- Statistics: reasons from samples to populations
  - This is inductive reasoning, and is usually *unsound* (in a logical sense of the word)

- Statistical Analysis

- Statistical analyses usually involve one of 3 things: (1) The study of populations; (2) The study of variation; and (3) Techniques for data abstraction and data reduction
- Statistical analysis is more than statistical computation:
  1. What is the question to be answered?
  2. Can it be quantitative (i.e. can we make measurements about it)?
  3. How do we collect data?
  4. What can the data tell us?

- Estimation from a Sample

- For example, when a number of samples are drawn and the mean of each is found, then average of these means is equal to the population mean
- Such an estimator is said to be *statistically unbiased*
- More on this later

- Mean and Variance
- Bias V.S. Variance

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

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*.

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

Imagine testing the prediction of our estimator $\hat{y}$ on many samples of the same size drawn at random from the same distribution. We compute error based on the squared difference between predicted and actual values. Then the MSE can be decomposed like this:

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

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.

2.4 Correlation

$$r = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x)}\sqrt{\text{var}(y)}}$$

This is sometimes also called *Pearson's correlation coefficient*

$$\text{cov}(x, y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{n - 1}$$

- What does "covariance" mean?
  1. Case 1: $x_i > \overline{x}$, $y_i > \overline{y}$
  2. Case 2: $x_i < \overline{x}$, $y_i < \overline{y}$
  3. Case 3: $x_i < \overline{x}$, $y_i > \overline{y}$
  4. Case 4: $x_i > \overline{x}$, $y_i < \overline{y}$

  In the first two cases, x_i and y_i vary together, both being high or low relative to their means. In the other two cases, they vary in different directions

- If the positive products dominate in the calculation of $\text{cov}(x, y)$, then the value of r will be positive. If the negative products dominate, then r will be negative. If 0 products dominate, then r will be close to 0.
- You should be able to show that:

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2} \sqrt{\sum (y_i - \overline{y})^2}}$$

- Computers generally use a short-cut formula:

$$r = \frac{\sum_i x_i y_i \; - \; n\overline{x}\,\overline{y}}{n - 1}$$

- The same kinds of calculations can be done if the data were not actual values but ranks instead (i.e. ranks for the x's and the y's). This is called *Spearman's rank correlation*, but we won't do these calculations here.

  - r and sampling theory

- Suppose you have a sample of $\langle x, y \rangle$ pairs and you calculate $r = 0.3$. Is this really the case?
- Sampling theory tells us something. If: (a) the relative frequencies observed are well modelled by a special kind of mathematical function (a "Normal" or Gaussian distribution); (b) the true correlation is 0; and (c) the number of samples is large
- Then:
  - The sampling distribution of the correlation coefficient (that is, how r varies from sample to sample) is also approximately distributed according to the Normal distribution with mean 0 and s.e. of approximately $1/\sqrt{n}$
- We can use this to calculate the (approximate) probability of obtaining the sample if the assumptions were true

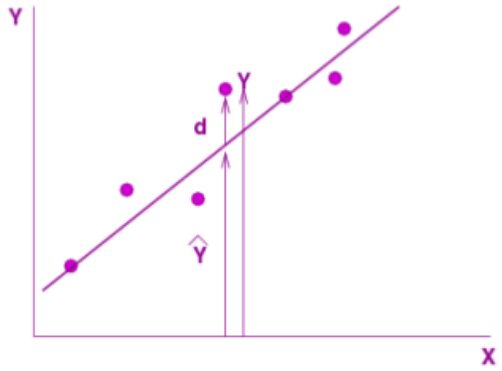
- Suppose we calculate $r = 0.3$ from the sample, and that the s.e. is 0.1 say. Then if the sample came from a population with true correlation 0, this would be quite unusual (less than 1% chance)
- We would say instead that the sample was probably from a population with correlation 0.3, with a 95% confidence interval of $\pm 2 \times 0.1$

  - what does correlation mean

- r is a quick way of checking whether there is some linear association between x and y
- The sign of the value tells you the direction of the association
- All that the numerical value tells you is about the scatter in the data
- The correlation coefficient does not model any relationship. That is, given a particular x you cannot use the r value to calculate a y value
  - It is possible for two datasets to have the same correlation, but different relationships
  - It is possible for two datasets to have the different correlations but the same relationship
- MORAL: Do not use correlations to compare datasets. All you can derive is whether there is a positive or negative relationship between x and y
- ANOTHER MORAL: Do not use correlation to imply x causes y or the other way around

2.5 Linear Relationship Between 2 Variables

- we need to get slope and intercept



GOAL: fit a line whose equation is of the form $\hat{Y} = a + bX$

HOW: minimise $\sum_i d_i^2 = \sum_i (Y_i - \hat{Y}_i)^2$ (the "least squares estimator")

The calculation for b is given by:

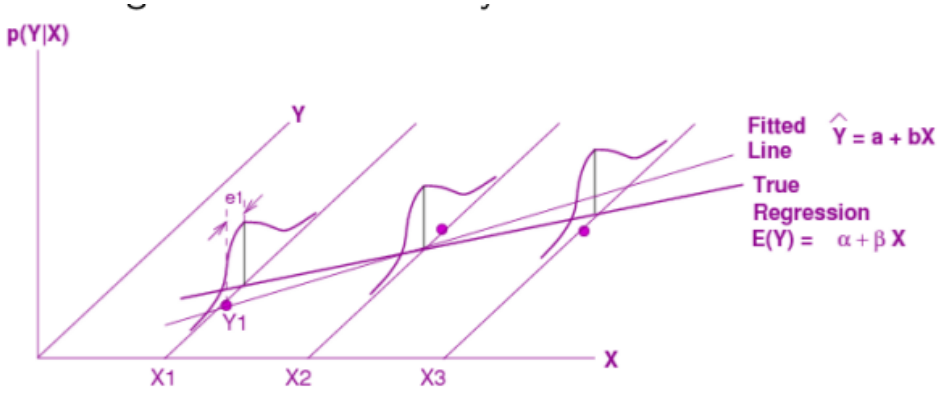
$$b = \frac{\text{cov}(x, y)}{\text{var}(x)}$$

where $\text{cov}(x, y)$ is the covariance of x and y, given by $\sum_i (x_i - \overline{x})(y_i - \overline{y})$ as before
This can be simplified to:

$$b = \sum(xy) / \sum x^2$$

where $x = (X_i - \overline{X})$ and $y = (Y_i - \overline{Y})$
$a = \overline{Y} - b\overline{X}$

- regression model



Using terminology that we will introduce later, the Y_i are identically distributed independent random variables with mean $\mu_i = \alpha + \beta X_i$ and variance σ^2

Or: $Y_i = \alpha + \beta X_i + e_i$ where the e_i are independent errors with mean 0 and variance σ^2

- univariate linear regression

Suppose we want to investigate the relationship between people's height and weight. We collect n height and weight measurements $(h_i, w_i), 1 \leq i \leq n$.

Univariate linear regression assumes a linear equation $w = a + bh$, with parameters a and b chosen such that the sum of squared residuals $\sum_{i=1}^{n} (w_i - (a + bh_i))^2$ is minimised.

$$\frac{\partial}{\partial a}\sum_{i=1}^{n}(w_i-(a+bh_i))^2=-2\sum_{i=1}^{n}(w_i-(a+bh_i))=0$$
$$\Rightarrow \hat{a}=\overline{w}-\hat{b}\overline{h}$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{n}(w_i-(a+bh_i))^2=-2\sum_{i=1}^{n}(w_i-(a+bh_i))h_i=0$$
$$\Rightarrow \hat{b}=\frac{\sum_{i=1}^{n}(h_i-\overline{h})(w_i-\overline{w})}{\sum_{i=1}^{n}(h_i-\overline{h})^2}$$

So the solution found by linear regression is $w=\hat{a}+\hat{b}h=\overline{w}+\hat{b}(h-\overline{h})$.

- linear regression: intuitions

$$\sum_{i=1}^{n}(y_i-(\hat{a}+\hat{b}x_i))=n(\overline{y}-\hat{a}-\hat{b}\overline{x})=0$$

The result follows because $\hat{a}=\overline{y}-\hat{b}\overline{x}$, as derived above.

While this property is intuitively appealing, it is worth keeping in mind that it also makes linear regression susceptible to *outliers*: points that are far removed from the regression line, often because of measurement errors.

2.6 Muitivariate LR

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

- I think the matrix form of LR is more important for that it's easier to vectorize
  - X is $m*d$ matrix $\rightarrow$ the form of $\hat{w}$ is $d*1$
  - note that X^TX should be full-rank, otherwise we need to perform regularization

2.7 Regulatization: avoid overfitting

- Ridge

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

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

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

- Lasso

An interesting alternative form of regularised regression is provided by the *lasso*, which stands for 'least absolute shrinkage and selection operator'. 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 regression favours *sparse solutions*.

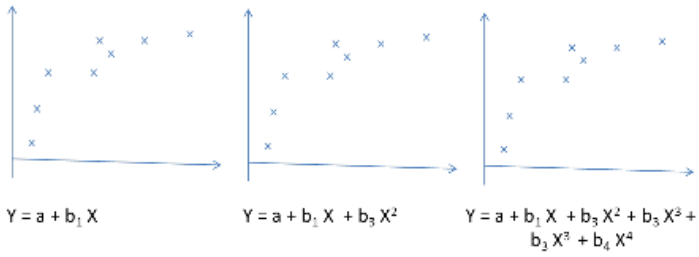
2.8 Model Selection

- How 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 is greedy:
  - Review Stanford Statistics Learning
- Parameter Estimation by Optimization: regularization

Add penalty terms to a *cost* function, forcing coefficients to shrink to zero



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

$$Cost(\theta) = \frac{1}{n}\sum_i (f_\theta(\mathbf{x_i}) - y_i)^2 + \frac{1}{n}\lambda\sum_{i=1}^{n}\theta_i$$

- Gradient descent

Using gradient descent with the penalty function will do two things: (a) we will move each θ_j in a direction that minimises the cost; and (b) each value of θ_j will also get "shrunk" on each iteration by multiplying the old value by an amount < 1

$$\theta_j{}^{(i+1)} = \alpha\theta_j{}^{(i)} - \eta\nabla_{\theta_j}$$

where $\alpha < 1$