

MS4S10 Machine Learning and Decision Making

Week 2

Moizzah Asif

moizzah.asif@southwales.ac.uk

J418

1

Know thy module



Week 1 – 4
Moizzah Asif

- 27-11-2020 – Basics of Machine Learning; The machine learning process; Data collection & preprocessing
 - 04-12-2020 – Supervised Learning: classification, regression, optimisation, model selection and generalisation, parametric and non-parametric learning, Decision Trees
 - 11-12-2020 –
 - 08-01-2021 –
- Course work 1 – 50% weightage

2

Supervised Learning

Used when a certain outcome has to be predicted on a given input and prediction is done after learning from paired input and output examples.

Ex:

Suppose we have a dataset from the local surgery in Pontypridd. We'd like to know which female patients over 21 years of age have diabetes.

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 - i. Data Preparation
 - ii. Algorithm Selection

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 1 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 2 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 3 | 3 | 78 | 50 | 32 | 88 | 31.0 | 0.248 | 26 | 1 |
| 4 | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 5 | 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 |
| 6 | 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 |

3

Supervised Learning

'Outcome' is what we want to predict.

Rest of the variables are predictors/features.

So here, a supervised learning algorithm predicts whether a patient may have diabetes or not as a function of all the predictors (pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, Diabetes Pedigree Function and Age)

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 - i. Data Preparation
 - ii. Algorithm Selection

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 1 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 2 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 3 | 3 | 78 | 50 | 32 | 88 | 31.0 | 0.248 | 26 | 1 |
| 4 | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 5 | 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 |
| 6 | 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 |

4

Supervised Learning

There are two major types of Supervised Learning:

- Classification
 - Classification is used to predict target variable which only has discrete values.
- Regression
 - Regression is used to predict target variables which have continuous values.

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 - i. Data Preparation
 - ii. Algorithm Selection

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 1 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 2 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 3 | 3 | 78 | 50 | 32 | 88 | 31.0 | 0.248 | 26 | 1 |
| 4 | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 5 | 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 |
| 6 | 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 |

5

Supervised Learning

Classification example:

- Following our example, whether patient has diabetes or not. Two distinct values, i.e. classes : **yes** or **no** | **1** or **0**

Regression example:

- In the diabetes example, if we had to predict the insulin levels of the patient. The insulin level is a continuous variable as opposed to the discrete classes just seen

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 - i. Data Preparation
 - ii. Algorithm Selection

6

University of South Wales Prifysgol De Cymru

Supervised Learning

Classification

The target variable may have only two discrete classes.

Or

may have multiple discrete classes

A two class classification problem is generally referred to as '*binary classification*' problem

A multi class classification problem is generally referred to as '*multiclass classification*' problem

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

7 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

7

University of South Wales Prifysgol De Cymru

Supervised Learning

Classification

Recall last week's missing value imputation via K-NN.

K-NN can be used as a classification algorithm as well.

In it's simplest form at the value of $K = 1$, the algorithm predicts the output for new input same as the nearest neighbour's output.

Whereas for values of $K > 1$, the majority class in K nearest neighbour is predicted to be the output for new input.

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

8 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

8

University of South Wales Prifysgol De Cymru

Supervised Learning

Notations

m = # training example
 n = # test examples
 x = input variables (features)
 y = output variable (target variable)

So,

(x, y) = training example
 (x_i, y_i) = i th training example

Training Set
 Learning Algorithm
 h (hypothesis)

input → h → output

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

9 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

9

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

Remember the housing prices example from pre-processing.

```
In [142]: housing_imp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
longitude      20640 non-null float64
latitude       20640 non-null float64
housing_median_age  20640 non-null float64
total_rooms    20640 non-null float64
total_bedrooms 20640 non-null float64
population     20640 non-null float64
households     20640 non-null float64
median_income  20640 non-null float64
median_house_value 20640 non-null float64
ocean_proximity 20640 non-null float64
dtypes: float64(10)
memory usage: 1.6 MB
```

Total data points = 20460

Let's imagine we have to predict the house values (median_house_value)

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

10 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

10

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

m = # training examples
 x = input variables/ features
 y = output variable/target variable
 (x, y) = training example (row in the dataframe/table),
 so, i th training example $(x^{(i)}, y^{(i)})$
 h = hypothesis

Training set m examples → Learning Algorithm

takes input, x → hypothesis → gives output, y

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

11 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

11

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

Linear representation of hypothesis

For one feature: $h(x) = \theta_0 + \theta_1 x$
 For two features: $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$
 For three features: $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$
 For n features: $h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$

θ 's are the parameters of learning algorithm are real numbers

The learning algorithm learns the values of θ 's

Remember $y = mx + c$

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

12 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

12

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

Linear representation of hypothesis

As h is dependent on θ , it can be represented as $h_{\theta}(x)$

For completeness and more formal representation $x_0 = 1$

$$h_{\theta}(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x$$

13 Moizah Asif - Machine Learning and Decision Making © University of South Wales

13

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

So how does h learn to be more accurate?

by minimising the difference between the values of

$$h_{\theta}(x)$$

$$y$$

i.e.

$$\text{minimising} \rightarrow h_{\theta}(x) - y \rightarrow (h_{\theta}(x) - y)^2$$

formal representation:

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

14 Moizah Asif - Machine Learning and Decision Making © University of South Wales

14

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

So how does h learn to be more accurate?

Let,

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

So,

$$\min_{\theta} J(\theta)$$

$J(\theta)$ is the error function

And how does it minimise $J(\theta)$?

by calculating the value of $J(\theta)$ over different values of θ

15 Moizah Asif - Machine Learning and Decision Making © University of South Wales

15

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

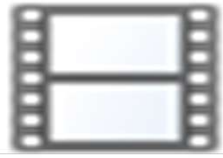
So how are different values of θ calculated?

A search algorithm can do that with in the learning algorithm

let's look at one such algorithm – **Gradient Descent**

Finds the local minimum of $J(\theta)$ over a range of values of θ

It starts from a random value of θ



16 Moizah Asif - Machine Learning and Decision Making © University of South Wales

16

University of South Wales Prifysgol De Cymru

Supervised Learning - Regression

Linear Regression

So how are different values of θ calculated?

Using optimisation algorithm

Ex: Gradient Descent

Keep changing θ to reduce $J(\theta)$

$$\theta_i := \theta_i - \alpha \frac{\partial J}{\partial \theta_i}(\theta)$$

Let's dry run this one $m = 1$, i.e. one training example only

17 Moizah Asif - Machine Learning and Decision Making © University of South Wales

17

University of South Wales Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Let's dry run this one $m = 1$, i.e. one training example only

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i} \frac{1}{2} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i} \frac{1}{2} (h_{\theta}(x) - y)^2$$

$$\frac{\partial}{\partial \theta_i} J(\theta) = 2 \cdot \frac{1}{2} (h_{\theta}(x) - y) \cdot \frac{\partial y}{\partial \theta_i} (h_{\theta}(x) - y)$$

where $h_{\theta}(x)$ is a function of n features

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

18 Moizah Asif - Machine Learning and Decision Making © University of South Wales

18

University of South Wales Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Let's dry run this one $m = 1$, i.e. one training example only

Substitute the value $h_{\theta}(x)$

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_i} (\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n - y)$$

Because the derivative is w.r.t to θ_i , all the terms will be considered constants except for $\theta_i x_i$. The derivative of $\theta_i x_i$ is x_i

So,

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_{\theta}(x) - y) \cdot x_i$$

which leads to,

$$\theta_i := \theta_i - \alpha (h_{\theta}(x) - y) \cdot x_i$$

19 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

19

University of South Wales Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

α ?

α is Learning rate
i.e. defines the pace of your descent
how large a step gradient descent takes

usually defined manually

if too small, gradient descent takes time to converge (find the local minimum)

if too large, then gradient descent can overshoot the local minimum as it is taking aggressive big steps

There has to be a decision!

20 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

20

University of South Wales Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

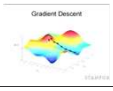
Let's see how θ_i is calculated when $m > 1$, i.e. more than one training examples y

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} \frac{1}{2} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)})^2$$

$$\theta_i := \theta_i - \alpha \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)}) \cdot x_i^{(j)}$$

Imagine finding local minimum for m training examples in n dimensions when $J(\theta) \rightarrow$



21 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

21

University of South Wales Prifysgol De Cymru


Supervised Learning - Optimisation

Gradient Descent

Let's see how θ_i is calculated when $m > 1$, i.e. more than one training examples y

Imagine finding minimum for m training examples when $J(\theta) \rightarrow$

for each value of θ_i , i.e. just one iteration:
gradient descent will have to find global minimum in n dimensions for m training examples with so many local minima



22 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

22

University of South Wales Prifysgol De Cymru

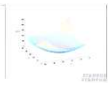
Supervised Learning - Optimisation

Gradient Descent

Let's see how θ_i is calculated when $m > 1$, i.e. more than one training examples y

Good news!


$J(\theta) \rightarrow$



A hyperbolic function with only 1 local minimum which is the only minima of the function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Ordinary Least Squares



23 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

23

University of South Wales Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Let's see how θ_i is calculated when $m > 1$, i.e. more than one training examples y

Repeat until convergence

$$\theta_i := \theta_i - \alpha \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)}) \cdot x_i^{(j)}$$

Batch Gradient Descent

For each iteration it uses **all** training examples to update parameters of the error function

24 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

24

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation


Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

As opposed to BGD, SGD uses **ONLY ONE** training sample per iteration (to update the parameters)

Preferably – randomly selected sample



Pseudo code?

25 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

25

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

Mini Batch Stochastic Gradient Descent

Mini SGD takes subset of training samples per iteration to update parameters

26 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

26

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Batch Gradient Descent

great for convex, or relatively smooth error manifolds. moves somewhat directly towards an optimum solution, either local or global

Stochastic Gradient Descent

Mini Batch Stochastic Gradient Descent

27 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

27

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

works well better than BGD for error manifolds that have lots of local maxima/minima

noisier gradient calculated using simple sample, tends to jerk the model out of local minima into a region that hopefully is more optimal

Very noisy descent

Mini Batch Stochastic Gradient Descent

28 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

28

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

Mini Batch Stochastic Gradient Descent

the amount of jerk/noise is reduced when using minibatches

good balance is struck when the minibatch size is small enough to avoid some of the poor local minima, but large enough that it doesn't avoid the global minima or better-performing local minima

A good place to find more details on optimisation algorithms:

Deep Learning, by Ian Goodfellow, Yoshua Bengio and Aaron Courville

<https://github.com/janishar/mit-deep-learning-book-pdf>

29 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

29

University of South Wales
Prifysgol De Cymru

Supervised Learning - Optimisation

Gradient Descent

Batch Gradient Descent

Stochastic Gradient Descent

Mini Batch Stochastic Gradient Descent

the amount of jerk/noise is reduced when using minibatches

good balance is struck when the minibatch size is small enough to avoid some of the poor local minima, but large enough that it doesn't avoid the global minima or better-performing local minima

A good place to find more details on optimisation algorithms:

Deep Learning, by Ian Goodfellow, Yoshua Bengio and Aaron Courville

<https://github.com/janishar/mit-deep-learning-book-pdf>

30 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

30

University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Ill-posed problem

Boolean function – two outcomes – target variable is binary

d binary inputs/features

2^d ways of writing d binary values

Ex: Let $d = 2$
all possible out comes = 2^2

| d_1 | d_2 |
|-------|-------|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

function is Boolean,
so target variable is binary

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

31

University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Ill-posed problem

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{10} h_{11} h_{12} h_{13} h_{14} h_{15}

H16??

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

32

University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Ill-posed problem

Ex: Let $d = 2$
all possible out comes = 2^2

| d_1 | d_2 |
|-------|-------|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

function is Boolean,
so target variable is binary

After all exercise we now know 2^{2^d} possible learning-functions/hypothesis

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

33

University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Ill-posed problem

What happens when $d > 5$ and inputs are non-binary (polynomial with $n > 2$)?

As n and d increase, having all possible examples gets difficult

Hence, output for only some examples is available to learn and form a function H .

Thus left with an ill-posed problem

The more training examples the more you know about underlying function

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

34

University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Inductive bias

Established – learning is ill-posed
data itself is not sufficient to find the unique solution


SO

Extra assumptions have to be made to find the solution/function of the distribution of the data sample we have

These assumptions are called *inductive bias*

HENCE

Established – learning is not possible without inductive bias



1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

35


University of South Wales Prifysgol De Cymru

Model Selection and Generalisation

Model Selection

How to choose the right/optimal bias between possible biases?

Remember



the aim of machine learning is rarely to replicate the training data but the prediction for new cases

Generalisation

How well a model trained on the training set predicts the right output for new instances is called generalisation

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

36

University of South Wales
Prifysgol De Cymru

Model Selection and Generalisation

Underfitting

For ^{good} generalization – **Try to** match the complexity of the hypothesis class H with the complexity of the function underlying the data (IDEAL)

If H is less complex than the function, we have **underfitting**.

For example, when trying to fit a line to data sampled from a third-order polynomial

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

37 Moizah Asif - Machine Learning and Decision Making © University of South Wales

37

University of South Wales
Prifysgol De Cymru

Model Selection and Generalisation

To AVOID underfitting increase the complexity of H , the training error decreases

BUT

if H gets **too complex** for the underlying function, the data may get insufficient for it and it will **overfit**

OR

if the **data is noisy** and the overcomplex H may also learn on the noise, we have **overfitting**

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

38 Moizah Asif - Machine Learning and Decision Making © University of South Wales

38

University of South Wales
Prifysgol De Cymru

Model Selection and Generalisation

Three factor trade-off

Prevails in algorithms trained from example data

1. The complexity of the hypothesis data is fit to – Capacity of the hypothesis class
2. The amount of training data
3. The generalization error on new examples

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

39 Moizah Asif - Machine Learning and Decision Making © University of South Wales

39

University of South Wales
Prifysgol De Cymru

Model Selection and Generalisation

Measurement of generalisation ability of H

i.e the quality of inductive bias

need data outside of the training set

1. Divide the dataset into two parts
2. Use one part to train H , it is called **training set**
3. The remaining part is called the **validation set**, we test the generalisation ability on it
4. The hypothesis $h \in H$ which is the most accurate, i.e has the best inductive bias is chosen

Cross - Validation

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

40 Moizah Asif - Machine Learning and Decision Making © University of South Wales

40

University of South Wales
Prifysgol De Cymru

Model Selection and Generalisation

Validation set vs Test set

How to report the expected error of the best model (the one with the best h)?

No, not the validation error

Because the validation set effectively became the part of training set when we used it to choose the best h

What now?

Need a third set, called the **test set** – it contains example unseen in the training/learning phase

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

41 Moizah Asif - Machine Learning and Decision Making © University of South Wales

41

University of South Wales
Prifysgol De Cymru

Parametric and Non-parametric Learning algorithms

Parametric Algorithms

Learning Algorithms which have fixed number of parameters which fit to the data.
Ex: Linear Regression

Non-parametric Algorithms

The number of parameters grow with the data(training examples)
Ex: locally weighted regression (loess/lowess)

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

42 Moizah Asif - Machine Learning and Decision Making © University of South Wales

42

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Non-parametric Learning algorithms

Locally weighted Regression

Makes predictions based on datapoints/training examples in the local vicinity of point x .

applies linear regression to the subset of the data (points local to x)

Formally

LWR fit θ to minimise

$$\sum_i^m w^{(i)} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

43 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

43

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Non-parametric Learning algorithms

Locally weighted Regression

LWR fit θ to minimise

$$\sum_i^m w^{(i)} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Where:
 $w^{(i)} \rightarrow$ weight

$$w^{(i)} = e^{\left(-\frac{(x^{(i)} - x)^2}{2}\right)}$$

44 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

44

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Non-parametric Learning algorithms

Locally weighted Regression

LWR fit θ to minimise

$$\sum_i^m w^{(i)} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$w^{(i)} = e^{\left(-\frac{(x^{(i)} - x)^2}{2}\right)}$$

How does the proximity of $x^{(i)}$ to x affect the weight?

If $|x^{(i)} - x|$ is small, then $w^{(i)} \cong 1$

If $|x^{(i)} - x|$ is large, then $w^{(i)} \cong 0$

45 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

45

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

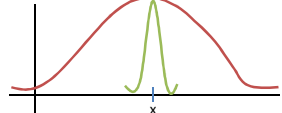
Non-parametric Learning algorithms

Locally weighted Regression

$$w^{(i)} = e^{\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)}$$

τ - the bandwidth parameter

controls how quickly the weight of a training example falls off with distance of its $x^{(i)}$ from the query point x



46 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

46

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Classification

Classification – Binary

Logistic regression

Notations

Binary classification:
 $y \in \{0,1\} \rightarrow h(x) \in [0,1]$

Bayesian Notations

$p(x|C_k)$ - Class conditional density: the probability density function for x given that it exists in class C_k

$p(x)$ - the prior probability of x

47 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

47

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Classification

Classification – Binary

Logistic regression

Estimates that x belongs to a particular class by estimating probabilities of belonging to both of the classes

Probabilistic interpretations

Assumptions:

- $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$
where $\varepsilon^{(i)}$ is error + noise
- $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ i.e. error is normally distributed (Gaussian distribution)

$$p(\varepsilon^{(i)}) = \frac{1}{2\pi\sigma} e^{-\frac{\varepsilon^{(i)2}}{2\sigma^2}}$$

48 Moizzah Asif - Machine Learning and Decision Making © University of South Wales

48

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification

Classification – Binary

Logistic regression

Assumption 2 implies:

$$p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

OR

$$y^{(i)}|x^{(i)}; \theta \sim \mathcal{N}(\theta^T x^{(i)}, \sigma^2)$$

49 Moizah Asif - Machine Learning and Decision Making © University of South Wales

49

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification

Classification – Binary

Logistic regression

Probabilistic interpretations

Assumptions:

- $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$
where $\varepsilon^{(i)}$ is error + noise
- $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ i.e. error is normally distributed (Gaussian distribution)
- $\varepsilon^{(i)}$ is IID (independently and identically distributed)

50 Moizah Asif - Machine Learning and Decision Making © University of South Wales

50

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification

Classification – Binary

Logistic regression

Probabilistic interpretations

Let

$$L(\theta) = p(y^{(i)}|x^{(i)}; \theta)$$

i.e. Likelihood of θ

$$L(\theta) = \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta)$$

$$L(\theta) = \prod_{i=1}^m \frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

So, how to choose the parameter θ ?

51 Moizah Asif - Machine Learning and Decision Making © University of South Wales

51

University of South Wales Prifysgol De Cymru


Supervised Learning – Classification

Classification – Binary

Logistic regression

Probabilistic interpretations

Maximum likelihood
choose θ to maximise the $L(\theta)$
to make the math easy



$$l(\theta) = \log L(\theta)$$

$$l(\theta) = \log \left(\prod_{i=1}^m \frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}} \right)$$

$$l(\theta) = \sum_{i=1}^m \log \left(\frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}} \right)$$

52 Moizah Asif - Machine Learning and Decision Making © University of South Wales

52

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification

Classification – Binary

Logistic regression

Probabilistic interpretations

to make the math easy

$$l(\theta) = \log L(\theta)$$

$$l(\theta) = \log \left(\prod_{i=1}^m \frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}} \right)$$

$$l(\theta) = \sum_{i=1}^m \log \left(\frac{1}{2\pi\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}} \right)$$

$$l(\theta) = m \log \frac{1}{2\pi\sigma} + \sum_{i=1}^m -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

To maximise $l(\theta)$, minimise:

$$\sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

53 Moizah Asif - Machine Learning and Decision Making © University of South Wales

53

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification

Classification – Binary

Logistic regression

Probabilistic interpretations

$$\sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2} = J(\theta)$$

54 Moizah Asif - Machine Learning and Decision Making © University of South Wales

54

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

$$y \in \{0,1\}$$

$$h_{\theta}(x) \in [0,1]$$

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid function
aka
Logistic function

55

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

55

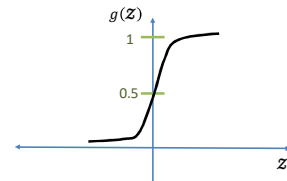
University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

$$g(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid function
aka
Logistic function



56

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

56

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

$$P(y = 1 | x; \theta) = h_{\theta}(x)$$

$$P(y = 0 | x; \theta) = 1 - h_{\theta}(x)$$

$$P(y|x; \theta) = h_{\theta}(x)^y (1 - h_{\theta}(x))^{1-y}$$

So how to fit parameter of the model?

$$l(\theta) = P(y|x; \theta) = \prod_{i=1}^m P(y^{(i)} | x^{(i)}; \theta)$$

$$l(\theta) = \prod_{i=1}^m h_{\theta}(x^{(i)})^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}$$

57

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

57

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

$$l(\theta) = P(y|x; \theta) = \prod_{i=1}^m P(y^{(i)} | x^{(i)}; \theta)$$

$$l(\theta) = \prod_{i=1}^m h_{\theta}(x^{(i)})^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}$$

Now let

$$l(\theta) = \log l(\theta)$$

$$l(\theta) = \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)}))$$

So fit the parameter by log likelihood, but how to maximise log likelihood?

58

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

58

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

$$l(\theta) = \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)}))$$

Use gradient descent to maximise the loglikelihood function

$$\theta := \theta + \alpha \nabla_{\theta} l(\theta)$$

$$\nabla_{\theta} l(\theta)$$

$$\frac{d l(\theta)}{d \theta_j} = \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

59

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

59

University of South Wales Prifysgol De Cymru

Supervised Learning – Classification
Classification – Binary
Logistic regression

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
i. Data Preparation
ii. Algorithm Selection

Use gradient descent to maximise the loglikelihood function

$$\theta := \theta + \alpha \nabla_{\theta} l(\theta)$$

$$\nabla_{\theta} l(\theta)$$

$$\frac{d l(\theta)}{d \theta_j} = \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

$$\theta := \theta + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad \text{let's go back}$$

60

Molizah Asif - Machine Learning and Decision Making

© University of South Wales

60

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

Decision trees approximate the **discrete-valued** target functions.

The learned-function is represented by a decision tree.

Can also be represented by if-else-then rules for readability

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

61 Moizah Asif - Machine Learning and Decision Making © University of South Wales

61

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

To play tennis or not?

Depends on the day

A day may have a few features based on their values we can take a decision

The features could possibly be?

1. outlook
2. Humidity
3. Wind

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

62 Moizah Asif - Machine Learning and Decision Making © University of South Wales

62

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

To play tennis or not?

Adapted from Quinlan 1986

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

63 Moizah Asif - Machine Learning and Decision Making © University of South Wales

63

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

The Idea

Classification is broken down into a set of choices about each feature in turn. Starting at the **root** (base/top) of the tree and progressing down to the **leaves**, where rests the classification decision/output.

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

64 Moizah Asif - Machine Learning and Decision Making © University of South Wales

64

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

Decision trees represent a disjunction of conjunctions of constraints on the attribute (feature) values of instances (training examples)

Tom Mitchell

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

65 Moizah Asif - Machine Learning and Decision Making © University of South Wales

65

University of South Wales
Prifysgol De Cymru

Supervised Learning – Decision Trees

Learning with Trees

Characteristics of Problems that can be learned by decision trees

1. Instances should have attribute-value pairs, i.e. fixed set of values for each attribute
2. Target function should have discrete output values
3. Disjunctive descriptions
4. The training data may contain errors as decision trees are robust to errors
5. Training data may contain missing values

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

66 Moizah Asif - Machine Learning and Decision Making © University of South Wales

66

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Learning with Trees – Basic Algorithm

Most variations employ the core algorithm i.e.,

top-down greedy search

Ex: ID3, C4.5

67 Moizah Asif - Machine Learning and Decision Making © University of South Wales

67

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

ID3

Top down greedy

Begins with question: which attribute to be tested at the root of the tree?

Answered by using a **statistical test** on each attribute of the instance to determine how well it classifies training example on its own

The best attribute is selected and used as test at the root.

Descendants is created for each possible value of that attribute.

The entire process is then repeated for each descendent node.

68 Moizah Asif - Machine Learning and Decision Making © University of South Wales

68

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

ID3

How to determine which feature is most useful for classification?

What is a good quantitative measure of the worth of an attribute for classification?

A **statistical test** of a property called **information gain**

ID3 – Information Gain

Information gain measures, how well a given attribute separates the training examples according to their target classification.

ID3 uses information gain to select among the candidate attributes at each step

69 Moizah Asif - Machine Learning and Decision Making © University of South Wales

69

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

Information Gain and Entropy

Information gain makes the selection of best attribute for classification among candidate attribute using a criteria called **Entropy**.

IG measures how much entropy of the whole training set would decrease if we choose each particular feature for the next classification step.

Entropy is considered as a measure of impurity in a collection of training example.

70 Moizah Asif - Machine Learning and Decision Making © University of South Wales

70

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

Entropy

Given a collection S , containing positive and negative examples of some target variable,

The entropy of S , relative to the Boolean classification is

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Let S be a training set of 14 instances of a boolean classification problem.

Includes 9 positive and 5 negative examples [9+, 5-]

$$Entropy([9+, 5-]) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right)$$

$$Entropy([9+, 5-]) = 0.940$$

71 Moizah Asif - Machine Learning and Decision Making © University of South Wales

71

University of South Wales Prifysgol De Cymru

Supervised Learning – Decision Trees

Entropy

Given a collection S , containing positive and negative examples of some target variable,

The entropy of S , relative to the Boolean classification is

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy graph of training examples for a binary classification problem

72 Moizah Asif - Machine Learning and Decision Making © University of South Wales

72

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

Information Gain and Entropy

Now that we understand Entropy much better, let's revisit information Gain.

Information gain is the expected reduction in entropy caused by partitioning the examples according to the attribute whose effectiveness is being measured for classification.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where

$Values(A)$ is the set of all possible values of attribute A

S_v is the subset for which attribute A has value v

73 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

73

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

Information Gain and Entropy

Weather Example

Attributes: outlook, **wind**, humidity
14 training examples
 $Values(wind) = weak, strong$

6 positives and 2 negatives have **wind = weak**
Remainder **wind = strong**

Calculate information gain due to sorting by **Wind**

$$S = \begin{matrix} [9+, 5-] \\ S_{weak} = [6+, 2-] \\ S_{strong} = [3+, 3-] \end{matrix} \quad Gain(S, wind) = Entropy(S) - \sum_{v \in \{weak, strong\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(S) - \left(\frac{5}{14} Entropy(S_{weak}) + \frac{9}{14} Entropy(S_{strong}) \right)$$

74 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

74

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

Information Gain and Entropy

Weather Example

Which attribute is the best classifier?

Gain(S, Humidity) = 0.940 - ((7/14)0.985 + (7/14)0.592) = 0.151

Gain(S, Wind) = 0.940 - ((8/14)0.811 + (6/14)1.0) = 0.048

75 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

75

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

ID3 – Inductive bias

The next feature to add into the tree is the one with highest IG

This biases the algorithm towards smaller trees.

As it tries to minimise the amount of information left.
(why? Because it chooses the to go forwards with highest IG at present)

ID3 – overfitting

Each branch of the tree grows just deep enough to perfectly classify the training examples.

This may be a reasonable strategy but:

- noise in the data
- number of training example too small or inadequate to produce a representative of the true target function

76 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

76

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

Decision trees and overfitting

Overfitting in decision trees can be avoided by:

- setting a stopping criterion: such as tree size, or using a validation set to reach certain performance threshold
- overfit the data and then post prune the tree

77 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

77

University of South Wales
Prifysgol De Cymru

1. Basics of Machine Learning (ML)
2. The Machine Learning Process
 i. Data Preparation
 ii. Algorithm Selection

Supervised Learning – Decision Trees

Decision trees, overfitting and pruning

All versions of pruning are based on computing the full tree and reducing it, evaluating the error on a validation set.

Most Naïve version runs decision tree until all features are used to the point of overfitting.

- picks each node
- replaces the subtree beneath with a leaf node, labelled with the most common classification of the replaced subtree.
- Error is evaluated on the validation set. Pruned tree is kept if error <= original tree's error, or rejected otherwise.

78 Moiz Zah Asif - Machine Learning and Decision Making © University of South Wales

78

University of
South Wales
Prifysgol
De Cymru

Supervised Learning – Decision Trees

1. Basics of Machine Learning (ML)

2. The Machine Learning Process

- i. Data Preparation
- ii. Algorithm Selection

Rule post pruning and C4.5

1. Create the tree using ID3
2. Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node (if-else rules)
3. Prune each rule by removing preconditions, if the accuracy of the rule increases without it
4. Sort the rules according to the accuracy on training set and applied in order

79
Molizah Asif - Machine Learning and Decision Making
© University of South Wales

79

University of
South Wales
Prifysgol
De Cymru

Supervised Learning – Decision Trees

1. Basics of Machine Learning (ML)

2. The Machine Learning Process

- i. Data Preparation
- ii. Algorithm Selection

Trees and Continuous variable

Simplest way to deal with them for a decision tree is to discretise the continuous variable.
split in two at a certain value

Univariate vs multivariate trees

Univariate trees pick one feature at a time and split according to that one

Multivariate trees pick combination of features.

80
Molizah Asif - Machine Learning and Decision Making
© University of South Wales

80

University of
South Wales
Prifysgol
De Cymru

Supervised Learning – Decision Trees

1. Basics of Machine Learning (ML)

2. The Machine Learning Process

- i. Data Preparation
- ii. Algorithm Selection

Classification and Regression Trees (CART)

As the name suggests can be used for both regression and classification.

How it works for regression?

A regression tree is constructed in almost the same manner as a classification tree

impurity measure that is appropriate for classification is replaced by a measure appropriate for regression.

In regression, the goodness of a split is measured by the mean square error from the estimated value

81
Molizah Asif - Machine Learning and Decision Making
© University of South Wales

81