NUS | Computing

National University of Singapore

Eric Han
eric_han@nus.edu.sg
https://eric-han.com

Computer Science

T03 – 19 Sep 2024

Tutorial 4

CS2109s TG35,36

Content I

Bonus from T3

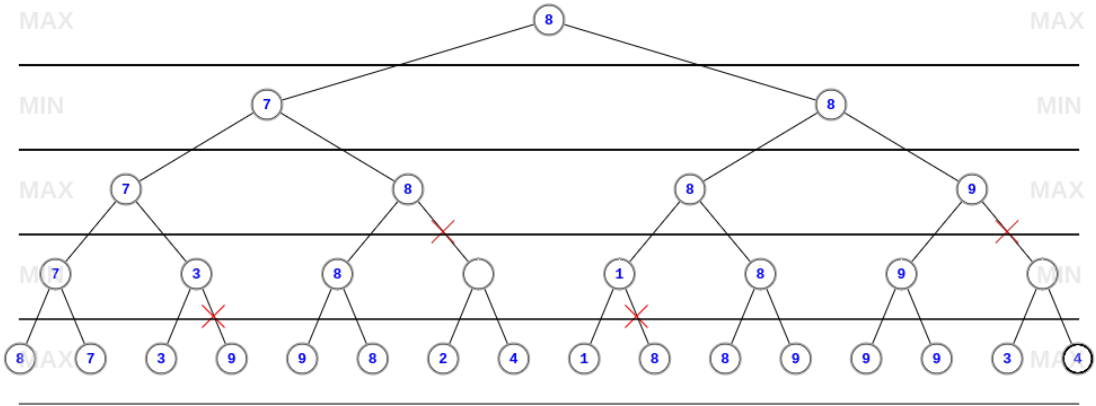

Figure 1: Alpha-Beta Answer (Credit MIT)

Section 1: **Decision Tree**

	Income	Credit History	Debt	Decision
0	Over 10k	Bad	Low	Reject
1	Over 10k	Good	High	Approve
2	0 - 10k	Good	Low	Approve
3	Over 10k	Good	Low	Approve
4	Over 10k	Good	Low	Approve
5	Over 10k	Good	Low	Approve
6	0 - 10k	Good	Low	Approve
7	Over 10k	Bad	Low	Reject
8	Over 10k	Good	High	Approve
9	0 - 10k	Bad	High	Reject

Question 1-3 [G]

1. Construct the best decision tree to classify the final outcome (Decision).
2. First row is labelled wrongly `Reject > Approve`, construct the best decision tree.
3. What is the decision made by the decision tree in part (b) for a person with an **income over 10k**, a **bad credit history**, and **low debt**?

Recap

- What is a decision tree?
- How are the features selected (*best*)?
- How to calculate information gain?

Question 1-3 [G]

1. Construct the best decision tree to classify the final outcome (Decision).
2. First row is labelled wrongly `Reject > Approve`, construct the best decision tree.
3. What is the decision made by the decision tree in part (b) for a person with an **income over 10k**, a **bad credit history**, and **low debt**?

Recap

- What is a decision tree?
- How are the features selected (*best*)?
- How to calculate information gain?
  - Suprise - 'Inverse of probability': I find this way more intuitive
  - Entropy - Expected Suprise over all events
  - Infomation Gain - Expectation of expected suprise for an attribute

$$IG(Y, X) = E(Y) - E(Y|X)$$

We Optimize: $\max_X IG(Y, X) \equiv \min_X E(Y|X)$

Answer 1

This, or the smarter way: Eyeball.

```
E(Decision) = 0.8812908992306927
IG(Decision,X) = E(Decision) - E(Decision|X)
```

	E(Decision|X)	IG(Decision|X)
Income	0.879673	0.00161775
Credit History *	0	0.881291
Debt	0.879673	0.00161775

```
Credit History
|-- [Bad] Reject:3
+-- [Good] Approve:7
```

Choice 1

`E(Decision) = 0.7219280948873623`

`IG(Decision,X) = E(Decision) - E(Decision|X)`

	E(Decision|X)	IG(Decision|X)
Income	0.68966	0.0322684
Credit History *	0.275489	0.446439
Debt	0.68966	0.0322684

Credit History is chosen.

Choice 2

`E(Decision) = 0.9182958340544894`

`IG(Decision,X) = E(Decision) - E(Decision|X)`

	E(Decision|X)	IG(Decision|X)
Income *	0.666667	0.251629
Debt *	0.666667	0.251629

Income is chosen, but **Debt** can also be chosen (see Tutorial ans).

Tree

```
Credit History
|-- [Bad] Income
|   |-- [0 - 10k] Reject:1
|   +-- [Over 10k] Debt
|       +-- [Low] Approve:1 Reject:1
+-- [Good] Approve:7
```

Cannot decide - the decision might either reject or accept the person.

- In such situations, we take expectation.
- But in this particular situation $50 - 50$, take a random guess.

4. Following prev, what are some ways to to mitigate inconsistent data?
5. Derive a Tree with each leaf node representing a minimum of 3 training data points by pruning the tree you previously obtained in part (b). Which is pruned?

Answer 4

- Regularize: Pruning to remove unnecessary branches and nodes, ie. Min-sample and Max-depth.
- Pre-processing: Remove outliers that create noise.
- Feature Selection: Select only relevant features, so less relevant features that could create inconsistencies are not part of the decision tree.
- Data Scarcity[1]: Collect more data on new features to clearly differentiate the inconsistent classes.

[1] Not always the best

Answer 5

The outlier is probably the person with an income over 10k, a bad credit history, and low debt.

	Income	Credit History	Debt	Decision
0	Over 10k	Bad	Low	Approve
7	Over 10k	Bad	Low	Reject

Section 2: **Linear Regression Model Fitting**

$$(X|Y) = \left(\begin{array}{ccc|c} 6 & 4 & 11 & 20 \\ 8 & 5 & 15 & 30 \\ 12 & 9 & 25 & 50 \\ 2 & 1 & 3 & 7 \end{array}\right)$$

1. Apply the Normal Equation formula to obtain a linear regression model that minimizes MSE of the data points.
2. Normal Equation needs the calculation of $(X^T X)^{-1}$. But sometimes this matrix is not invertible. When will that happen, and what should we do in that situation?

Recap

- What is the Normal Equation?
- What is the Gradient Descent?
- How does it compare with Gradient Descent?

Answer 1

$w = (X^T X)^{-1} X^T Y$

$$X = \begin{pmatrix} 1 & 6 & 4 & 11 \\ 1 & 8 & 5 & 15 \\ 1 & 12 & 9 & 25 \\ 1 & 2 & 1 & 3 \end{pmatrix}, \quad Y = \begin{pmatrix} 20 \\ 30 \\ 50 \\ 7 \end{pmatrix} \rightarrow \quad X^T X = \begin{pmatrix} 4 & 28 & 19 & 54 \\ 28 & 248 & 174 & 492 \\ 19 & 174 & 123 & 347 \\ 54 & 492 & 347 & 980 \end{pmatrix},$$

$$(X^T X)^{-1} = \begin{pmatrix} 5.5 & -5.75 & -4 & 4 \\ -5.75 & 7.5 & 6.5 & -5.75 \\ -4 & 6.5 & 14 & -8 \\ 4 & -5.75 & -8 & 5.5 \end{pmatrix}, \quad X^T Y = \begin{pmatrix} 107 \\ 974 \\ 687 \\ 1941 \end{pmatrix}$$

$$\rightarrow (X^T X)^{-1} X^T Y = \begin{pmatrix} 4 \\ -5.5 \\ -7 \\ 7 \end{pmatrix} \rightarrow \hat{y} = 4 - 5.5x_1 - 7x_2 + 7x_3$$

- Matrix $X^T X$ is not invertible
  - Some rows or columns in the dataset matrix are linearly dependent
    - Columns are linearly dependent $>$ Highly correlated or redundant features
    - Rows are linearly dependent $>$ collinear or duplicates
- Matrix $X^T X$ is ill-conditioned (almost singular)
  - Some rows or columns in the dataset matrix are *almost* linearly dependent
  - Not enough data points to cover the number of features

In such instances, gradient descent can be used to arrive at weights for the linear regression model that minimise the cost function.

Intuition: If invertible, there is a single solution, otherwise there are multiple solutions.

Question 3 [@]

Derive the Normal Equation from the MSE.

Derive the Normal Equation from the MSE.

Answer

From lectures:

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{n} x_{ij}w_j\right)^2 \rightarrow \frac{\partial J(w)}{\partial w_k} = -\frac{1}{m}\sum_{i=1}^{m} x_{ik}\left(y_i - \sum_{j=1}^{n} x_{ij}w_j\right)$$

$$\frac{\partial J(w)}{\partial w_k} = -\frac{1}{m}\sum_{i=1}^{m} x_{ik}\left(y_i - x_i \cdot w\right) = -\frac{1}{m}X^\top(Y - Xw)$$

$$\frac{\partial J(w)}{\partial w_k} = 0 \rightarrow 0 = -\frac{1}{m}X^\top(Y - Xw) \rightarrow 0 = X^\top Y - X^\top Xw$$

$$w = (X^\top X)^{-1}X^\top Y$$

Alternative complete derivation (RSS) - Note some notations differ.

Section 3: **Examining Cost Functions**

For Linear Regression, there are two popular cost functions, **Mean Squared Error**: $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$ and **Mean Absolute Error**: $L(y, \hat{y}) = \frac{1}{2}|y - \hat{y}|$

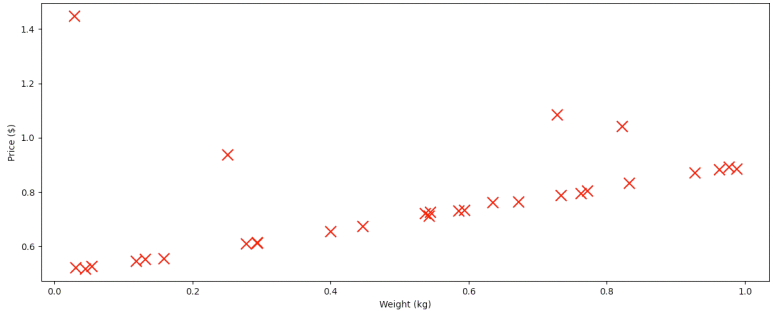

Figure 2: Scatter plot of the actual weight of meat at NTUC (x) and its price (y)

1. Justify your choice of cost function for this problem.
2. Provide examples of cost functions that are better suited to handle outliers?

- What is the difference between / when should we use MSE and MAE?
- How does it relate to L2, L1 loss?

- What is the difference between / when should we use MSE and MAE?
- How does it relate to L2, L1 loss?

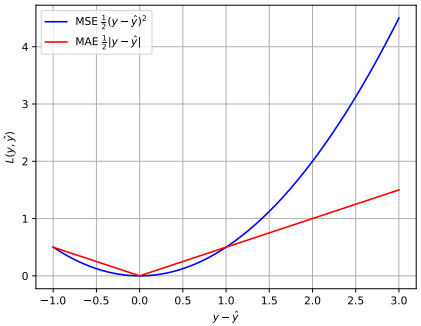

Figure 3: Cost Functions Plot

L1/L2 are technically vector lengths but are commonly interchanged with MAE/MSE.

Two examples are shown below when the cost functions are MAE and MSE respectively.

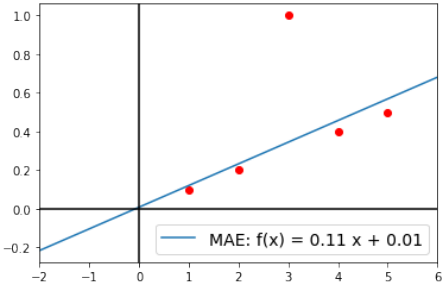

Figure 4: MAE

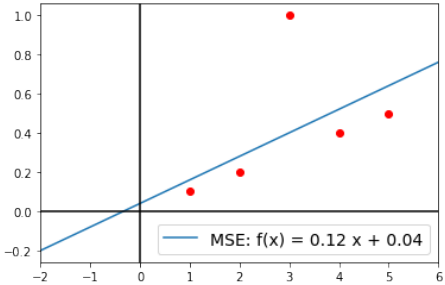

Figure 5: MSE

Note: As shown in these examples, outliers can have a greater impact on MSE than MAE, even if the y-values are between 0 and 1.

Huber loss is a combination of MSE and MAE and is defined as:

$$L(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta \cdot |y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

where δ is a transition threshold between the MSE and MAE behaviors.

Log-cosh loss is defined as:

$$L(y, \hat{y}) = \log(\cosh(y_i - \hat{y}_i))$$

For small values of x, $\log(\cosh(x)) \approx \frac{1}{2}x^2$, which is similar to MSE. For larger values of x, $\log(\cosh(x)) \approx |x| - \log(2)$, which is similar to MAE. Log cosh approximates MSE and MAE and is similar to the Huber loss function.

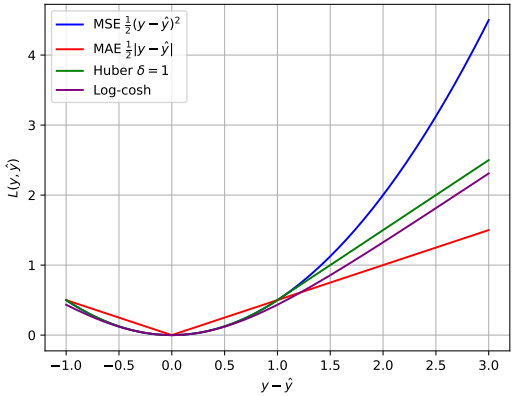

Figure 6: Cost Functions Plot

Huber and Log-cosh losses are more robust to outliers than MSE and MAE, as they assign less weight to extreme values, reducing their negative impact on model performance.

Section 4: **Choosing Learning Rates**

Question [G]

Given a simple function $y = x^2$, we know the gradient is $\frac{dy}{dx} = 2x$. As such, the minimum of this function is 0.

1. Compute over 5 iterations for $\alpha \in \{10, 1, 0.1, 0.01\}$ in tabular format.
2. During the course of training for a large number of epochs/iterations, what can be done to the value of the learning rate α to enable better convergence?

Recap

- What is gradient descent?
- Under what conditions is gradient descent guaranteed to converge to the global optimum? [@]

Question [G]

Given a simple function $y = x^2$, we know the gradient is $\frac{dy}{dx} = 2x$. As such, the minimum of this function is 0.

1. Compute over 5 iterations for $\alpha \in \{10, 1, 0.1, 0.01\}$ in tabular format.
2. During the course of training for a large number of epochs/iterations, what can be done to the value of the learning rate α to enable better convergence?

Recap

- What is gradient descent?
- Under what conditions is gradient descent guaranteed to converge to the global optimum? [@]
- What is the equation to perform the updates?
- Why do we need a learning rate α, and what is its use?

Table 6: History of x-values.

t	10.0	1.0	0.1	0.01
0	5	5	5	5
1	-95	-5	4	4.9
2	1805	5	3.2	4.802
3	-34295	-5	2.56	4.70596
4	651605	5	2.048	4.61184
5	-12380495	-5	1.6384	4.5196

Learning rate scheduler, α can be decreased through the course of training - larger values at the start, smaller values near the end:

- **Exponential Decay:** $\alpha(t) = \alpha_0 e^{-kt}$

To help you further your understanding, not compulsory; Work for Snack/EXP!

Tasks

1. Implement `create_tree(tree, df, parent=None, action = '')` to solve Decision Tree to get the following trace; `IG` implemented using pandas.
   a. Some code implemented in https://github.com/eric-vader/CS2109s-2425s1-bonus

```
Credit History
|-- [Bad] Income
|   |-- [0 - 10k] Reject:1
|   +-- [Over 10k] Debt
|       +-- [Low] Approve:1 Reject:1
+-- [Good] Approve:7
```

1. Lectures from CS3244: Differ from CS2109s, always check against syllabus
   a. Explained: Information Gain from Surprise
   b. Explained: Alternative complete derivation of Normal Equation (RSS)

References I