CS2109S Tutorial 4

Decision Trees and Linear Models

(AY 24/25 Semester 2)

February 21, 2025

(Prepared by Benson)

Contents

Recap: Decision Tree Learning

- **Information content / "surprisal"**: The informational value of communicating that an event happened.

$$I(e) = \underbrace{\log}_{\text{to bits}} \left(\underbrace{\frac{1}{p}}_{\text{frequency of occurrence}} \right) = -\log p \text{ bits}$$

- **Entropy**: The expected amount of information conveyed by identifying the outcome of a random trial.

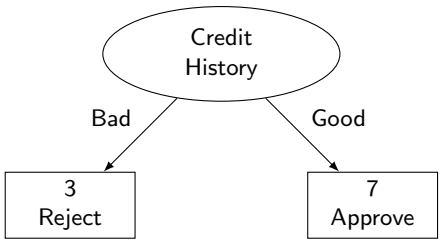
$$H(X) = \sum_{e \in E} P(e) I(e) = -\sum_{e \in E} P(e) \log P(e)$$

- **Information gain**: Difference between the entropy before the split and the expected entropy (of a sample) after the split.

$$IG(D, A) = H(D) - \underbrace{\sum_{v \in A} \frac{|D_v|}{|D|} H(D_v)}_{\text{Remainder (minimize this)}}$$

Q1. Decisions That Matter

(a) Construct the best decision tree to classify the final outcome (Decision) from the three features Income, Credit History, and Debt.



Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

Q1. Decisions That Matter

(b) Construct the best decision tree. Calculate the information gain values and remainders at each stage.

$$\begin{aligned}\text{remainder(Income)} &= \frac{3}{10}I(2,1) + \frac{7}{10}I(6,1) \\ &= \frac{3}{10}(0.9183) + \frac{7}{10}(0.5917) \\ &= 0.690\end{aligned}$$

$$\begin{aligned}\text{remainder(Credit History)} &= \frac{7}{10}I(7,0) + \frac{3}{10}I(1,2) \\ &= \frac{7}{10}(0) + \frac{3}{10}(0.9183) \\ &= 0.275\end{aligned}$$

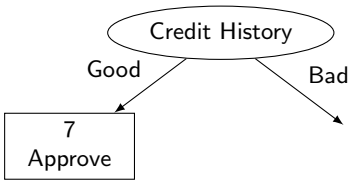
$$\begin{aligned}\text{remainder(Debt)} &= \frac{7}{10}I(6,1) + \frac{3}{10}I(1,2) \\ &= \frac{7}{10}(0.5917) + \frac{3}{10}(0.9183) \\ &= 0.690\end{aligned}$$

Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

+ \ −	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q1. Decisions That Matter

(b) Construct the best decision tree. Calculate the information gain values and remainders at each stage.



$$\text{remainder(Income)} = \frac{2}{3}I(1,1) + \frac{1}{3}I(0,1)$$
$$= \frac{2}{3}(1) + \frac{1}{3}(0) = 0.667$$

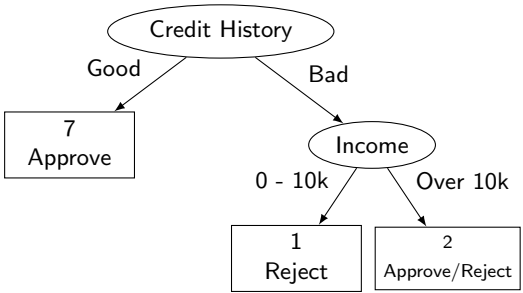
$$\text{remainder(Debt)} = \frac{2}{3}I(1,1) + \frac{1}{3}I(0,1)$$
$$= \frac{2}{3}(1) + \frac{1}{3}(0) = 0.667$$

Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

+ \ −	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q1. Decisions That Matter

(b) Construct the best decision tree. Calculate the information gain values and remainders at each stage.

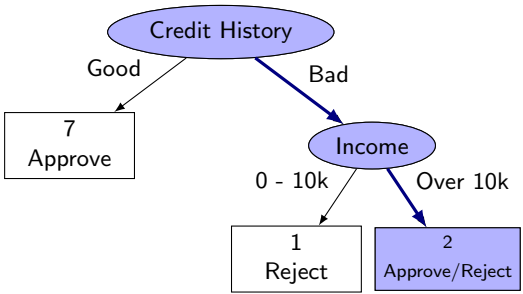


Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

+ \ −	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q1. Decisions That Matter

(c) 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**?



The decision might either reject or accept the person.

Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

+ \ −	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q1. Decisions That Matter

(d) The situation in part (c) demonstrates inconsistent data in the decision tree i.e. the attribute does not provide information to differentiate the two classes. In practice, it is usually left undecided. What are some ways to mitigate inconsistent data?

Solutions:

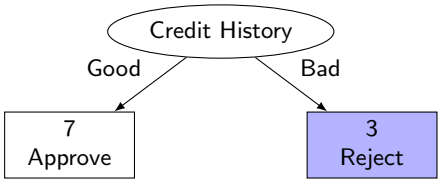
- Pruning (Min-sample / Max-depth).
- Pre-processing of data to remove outliers that create noise.
- Select only relevant features.
- Collect more data on new features to clearly differentiate the inconsistent classes.

Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

+ \ −	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q1. Decisions That Matter

(e) Let's consider a scenario where you desire a Decision Tree with each leaf node representing a minimum of 3 training data points. Derive the tree by pruning the tree you previously obtained in part (b). Which data(s) do you think are likely outlier(s)?



The first person is probably the outlier.

Income	Credit History	Debt	Decision
Over 10k	Bad	Low	Approve
Over 10k	Good	High	Approve
0 - 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
Over 10k	Good	Low	Approve
0 - 10k	Good	Low	Approve
Over 10k	Bad	Low	Reject
Over 10k	Good	High	Approve
0 - 10k	Bad	High	Reject

− / +	0	1	2	3	4	5
1	0	1	0.9183	0.8113	0.7219	0.6500
2	0	0.9183	1	0.9710	0.9183	0.8631
3	0	0.8113	0.9710	1	0.9852	0.9544
4	0	0.7219	0.9183	0.9852	1	0.9911
5	0	0.6500	0.8631	0.9544	0.9911	1
6	0	0.5917	0.8113	0.9183	0.9710	0.9940
7	0	0.5436	0.7642	0.8813	0.9457	0.9799

Q2. Linear Regression Model Fitting

(a) Apply the Normal Equation formula to obtain a linear regression model that minimizes MSE of the data points.

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

x_1	x_2	x_3	y
6	4	11	20
8	5	15	30
12	9	25	50
2	1	3	7

Solution.

$$\boldsymbol{X} = \begin{bmatrix} 1 & 6 & 4 & 11 \\ 1 & 8 & 5 & 15 \\ 1 & 12 & 9 & 25 \\ 1 & 2 & 1 & 3 \end{bmatrix}, \ \boldsymbol{y} = \begin{bmatrix} 20 \\ 30 \\ 50 \\ 7 \end{bmatrix}$$

$$\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \begin{bmatrix} 4 & -5.5 & -7 & 7 \end{bmatrix}^\top$$

$\therefore \hat{y} = 4 - 5.5x_1 - 7x_2 + 7x_3.$

Q2. Linear Regression Model Fitting

(b) Normal Equation needs the calculation of $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$. But sometimes this matrix is not invertible. When will that happen, and what should we do in that situation?

- When is $\boldsymbol{X}^\top \boldsymbol{X}$ invertible?
  - When all columns are linearly independent.
  - If some columns are linearly dependent, there are infinitely many solutions (thus it's impossible to find an "unique optimal solution").
- *Almost* linearly dependent columns create problems too...
  - A slight change in values drastically affects the result.
- **Solution**: Use gradient descent instead.
  (It is also a good practice to remove linearly dependent features.)

Q2. Linear Regression Model Fitting

Extra Slide

Lemma. $\boldsymbol{X}^\top \boldsymbol{X}$ is invertible if and only if all columns of $\boldsymbol{X}$ are linearly independent.

Proof.

$\boldsymbol{X}^\top \boldsymbol{X}$ is invertible

$\Leftrightarrow \text{rank}(\boldsymbol{X}^\top \boldsymbol{X}) = m$ ◀ since $\boldsymbol{X}^\top \boldsymbol{X}$ is an $m \times m$ matrix

$\Leftrightarrow \text{rank}(\boldsymbol{X}) = m$ ◀ $\text{rank}(\boldsymbol{X}^\top \boldsymbol{X}) = \text{rank}(\boldsymbol{X})$

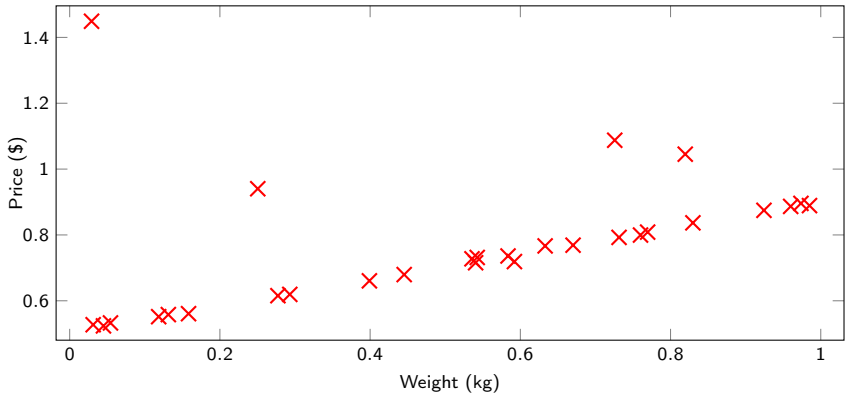
MA1522/2001 Refresher: $\text{rank}(\boldsymbol{X}^\top \boldsymbol{X}) = \text{rank}(\boldsymbol{X})$. (Prove this by showing the nullspace of $\boldsymbol{X}^\top \boldsymbol{X}$ is equal to the nullspace of $\boldsymbol{X}$. See tutorial 7 for either course.)

Q3. Examining Cost Functions

Mean Squared Error: $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$

Mean Absolute Error: $L(y, \hat{y}) = \frac{1}{2}|y - \hat{y}|$

(a) Justify your choice of cost function for this problem.

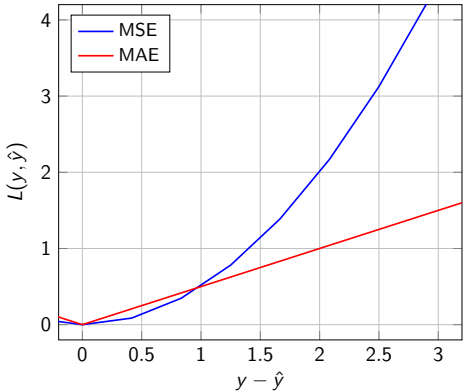


Q3. Examining Cost Functions

Mean Squared Error: $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$

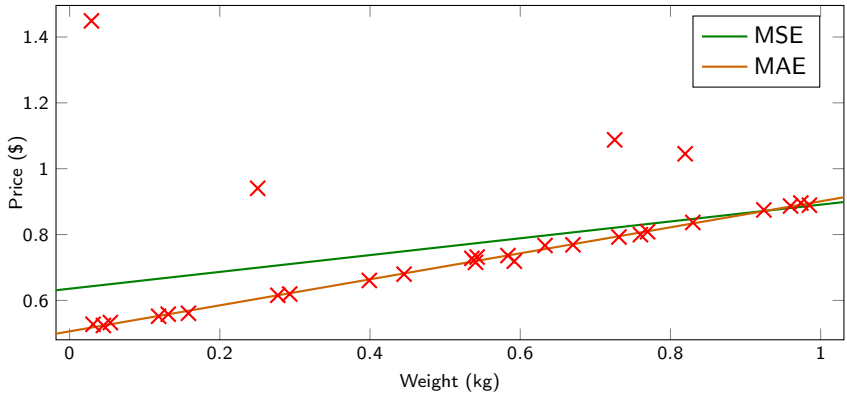
Mean Absolute Error: $L(y, \hat{y}) = \frac{1}{2}|y - \hat{y}|$

MSE penalizes large losses (caused by outliers) heavier than MAE.



Q3. Examining Cost Functions

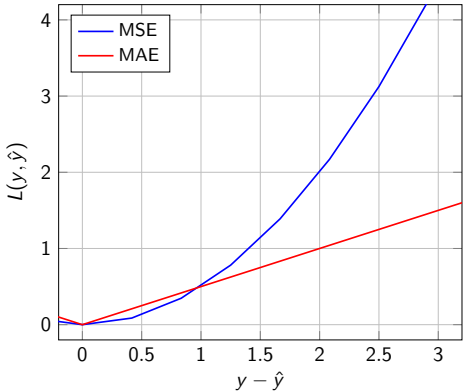
- Which line corresponds to MSE? Which line corresponds to MAE?
  - MSE peanlizes large losses heavily $\Rightarrow$ it will be more sensitive to outliers.
  - These outliers could be a result of human error, they should have a smaller impact and MAE is preferred. If we consider outliers as important, MSE is preferred.



Q3. Examining Cost Functions

True or False:

1. Consider a dataset where all y values are larger than 1. MSE penalizes the outliers more heavily than MAE.
2. Consider a dataset where all y values are between 0 and 1. MSE penalizes the outliers more heavily than MAE.



Q3. Examining Cost Functions

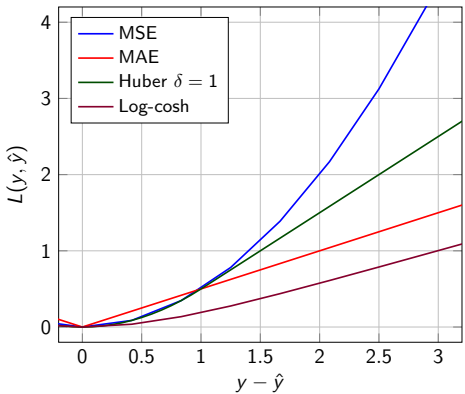
(b) Can you provide examples of cost functions that are better suited to handle outliers more effectively?

Huber loss: MSE $\rightarrow$ MAE.

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

Log-cosh loss: $\approx \frac{1}{2}x^2$ for small $x \rightarrow$ $\approx |x| - \log 2$ for large x.

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

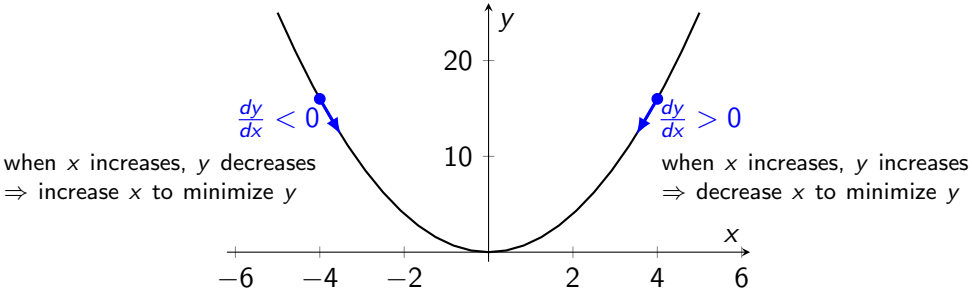


Recap: Gradient Descent

We wish to minimize the loss y by varying x.

- Parameters: Initial value x, Learning rate α. Decides this

How far do we change x?



Theorem. If α is **small enough**, gradient descent would reach a local minima. This would also be the global minima if $f(x)$ is **convex**.

For convex functions, local minima ⇒ global minima.

Q4. Choosing Learning Rates

Assume α is negative. Which of the following is the equation to perform the updates?

A. $x \leftarrow x + \alpha \frac{dy}{dx}$

B. $x \leftarrow x - \alpha \frac{dy}{dx}$

C. $y \leftarrow y + \alpha \frac{dy}{dx}$

D. $y \leftarrow y - \alpha \frac{dy}{dx}$

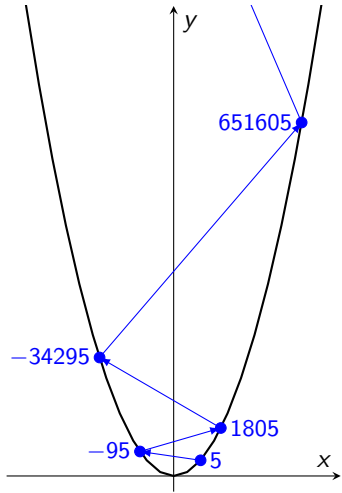
Solution. When $\frac{dy}{dx} > 0$, we need to decrease x.
Only option A successfully decreases x.

Q4. Choosing Learning Rates

(a) Trace the gradient descent algorithm at the function $y = x^2$, starting from the point $a = (5, 25)$ and using $\alpha = 10, 1, 0.1, 0.01$. Record the value of x over 5 iterations.

$\alpha = 10$:

1. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 10 \cdot 10 = -95.$
2. $\left.\frac{dy}{dx}\right|_{x=-95} = -190$
   $x = -95 - 10 \cdot (-190) = 1805.$
3. $\left.\frac{dy}{dx}\right|_{x=1805} = 3610$
   $x = 1805 - 10 \cdot 3610 = -34295.$
4. $\left.\frac{dy}{dx}\right|_{x=-34295} = -68590$
   $x = -34295 - 10 \cdot (-68590) = 651605.$
5. $\left.\frac{dy}{dx}\right|_{x=651605} = 1303210$
   $x = 651605 - 10 \cdot 1303210 = -12380495.$

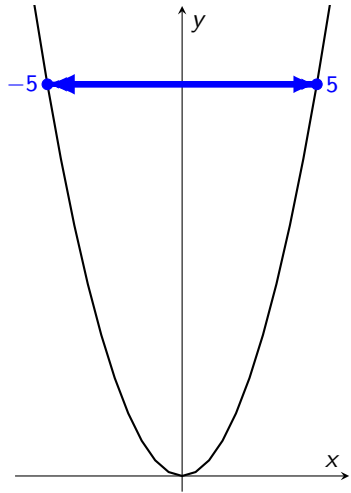


Q4. Choosing Learning Rates

(a) Trace the gradient descent algorithm at the function $y = x^2$, starting from the point $a = (5, 25)$ and using $\alpha = 10, 1, 0.1, 0.01$. Record the value of x over 5 iterations.

$\alpha = 1$:

1. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 1 \cdot 10 = -5.$
2. $\left.\frac{dy}{dx}\right|_{x=-5} = -10$
   $x = -5 - 1 \cdot (-10) = 5.$
3. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 1 \cdot 10 = -5.$
4. $\left.\frac{dy}{dx}\right|_{x=-5} = -10$
   $x = -5 - 1 \cdot (-10) = 5.$
5. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 1 \cdot 10 = -5.$

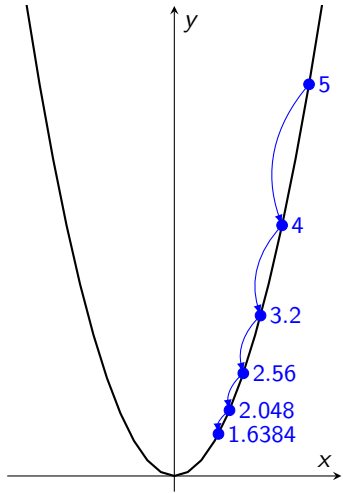


Q4. Choosing Learning Rates

(a) Trace the gradient descent algorithm at the function $y = x^2$, starting from the point $a = (5, 25)$ and using $\alpha = 10, 1, 0.1, 0.01$. Record the value of x over 5 iterations.

$\alpha = 0.1$:

1. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 0.1 \cdot 10 = 4.$
2. $\left.\frac{dy}{dx}\right|_{x=4} = 8$
   $x = 4 - 0.1 \cdot (8) = 3.2.$
3. $\left.\frac{dy}{dx}\right|_{x=3.2} = 6.4$
   $x = 3.2 - 0.1 \cdot 6.4 = 2.56.$
4. $\left.\frac{dy}{dx}\right|_{x=2.56} = 5.12$
   $x = 2.56 - 0.1 \cdot 5.12 = 2.048.$
5. $\left.\frac{dy}{dx}\right|_{x=2.048} = 4.096$
   $x = 2.048 - 0.1 \cdot 4.096 = 1.6384.$

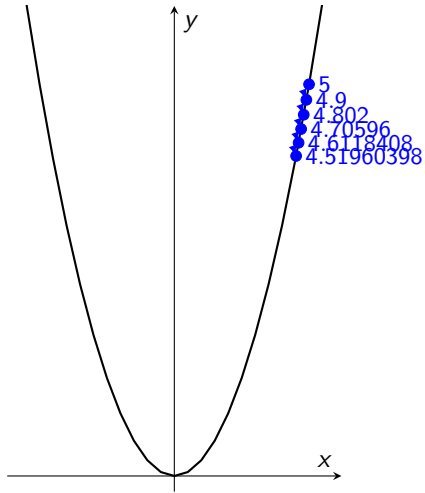


Q4. Choosing Learning Rates

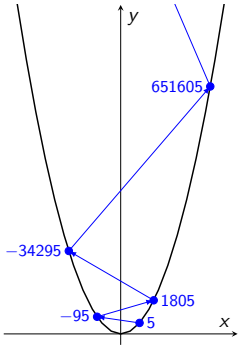
(a) Trace the gradient descent algorithm at the function $y = x^2$, starting from the point $a = (5, 25)$ and using $\alpha = 10, 1, 0.1, 0.01$. Record the value of x over 5 iterations.

$\alpha = 0.01$:

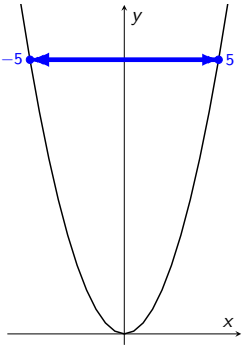
1. $\left.\frac{dy}{dx}\right|_{x=5} = 10$
   $x = 5 - 0.01 \cdot 10 = 4.9.$
2. $\left.\frac{dy}{dx}\right|_{x=4.9} = 9.8$
   $x = 4.9 - 0.01 \cdot (9.8) = 4.802.$
3. $\left.\frac{dy}{dx}\right|_{x=4.802} = 9.604$
   $x = 4.802 - 0.01 \cdot 9.604 = 4.70596.$
4. $\left.\frac{dy}{dx}\right|_{x=4.70596} = 9.41192$
   $x = 4.70596 - 0.01 \cdot 9.41192 = 4.6118408.$
5. $\left.\frac{dy}{dx}\right|_{x=4.6118408} = 9.2236816$
   $x = 4.6118408 - 0.01 \cdot 9.2236816 = 4.51960398.$



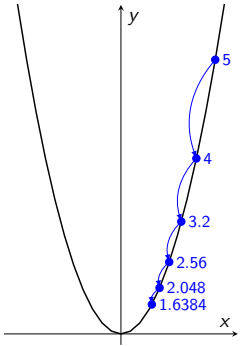
Q4. Choosing Learning Rates



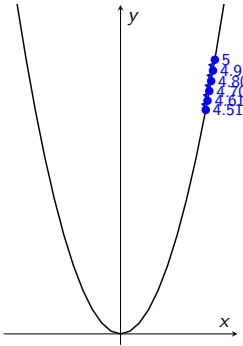
$\alpha = 10$



$\alpha = 1$



$\alpha = 0.1$



$\alpha = 0.01$

Q4. Choosing Learning Rates

(b) 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?

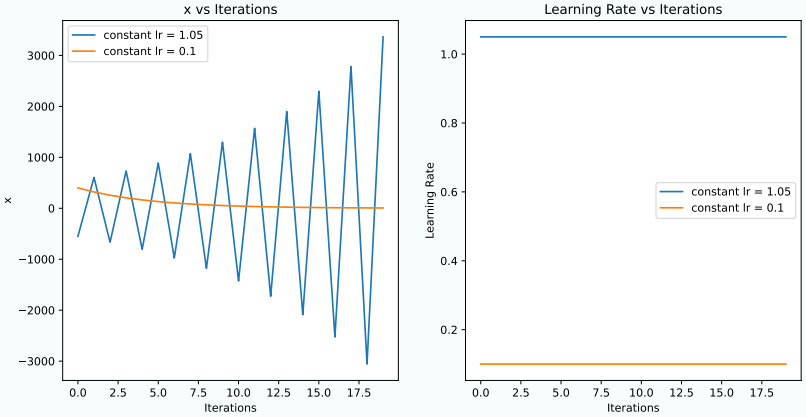
- Large α helps the model to converge faster, but might cause it to overshoot or even diverge.
- Idea: Vary α to help the model "stabilize". But how?
  - **Solution**: Decrease the learning rate α through the course of training.
  - This is the logic behind a **learning rate scheduler**.

Bonus. Learning Rate Schedulers (Practical)

Extra Slide

The given template code implements the gradient descent algorithm corresponding to Tutorial Q4. Copy the template code from the website.

Experiment with the different Pytorch learning rate schedulers and record your findings.



Bonus. Learning Rate Schedulers (Practical)

Extra Slide

Sample Solution.

```python
def optimize(lr_scheduler_class, scheduler_params={}, ...):
    ...

    scheduler = lr_scheduler_class(optimizer, **scheduler_params)

    for i in range(num_iterations):
        ...

        # Update the learning rate (i.e. take a step)
        if isinstance(scheduler, torch.optim.lr_scheduler.ReduceLROnPlateau):
            scheduler.step(metrics=loss.item())
        else:
            scheduler.step()

        x_history.append(x.item())
        lr_history.append(scheduler.get_last_lr()[0]) # Get current lr
        # optimizer.param_groups[0]['lr'] also works

    ...
```

Bonus. Learning Rate Schedulers (Practical)

Extra Slide

Sample Solution.