

Copyright @ University of Melbourne 2021. All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the author.



# Lecture 5: Introduction to Optimization

---

**COMP90049**

**Introduction to Machine Learning**

Semester 1, 2021



## Last time... Probability

- estimate the (conditional, joint) probability of observations
- Bayes rule
- Marginalization
- Probabilistic models
- Maximum likelihood estimation (taster)
- Maximum a posteriori estimation (taster)

## Last time... Probability

- estimate the (conditional, joint) probability of observations
- Bayes rule
- Marginalization
- Probabilistic models
- Maximum likelihood estimation (taster)
- Maximum a posteriori estimation (taster)

## Today... Optimization

- Curves, minima
- Gradients, derivatives
- Recipe for numerical optimization
- Maximum likelihood of the Binomial (from scratch!)



# Optimization

---

# Introduction I

We are all here to **learn** about Machine **Learning**.

- What is learning?



We are all here to **learn** about Machine **Learning**.

- What is learning?
- It probably has something to do with **change** or **mastering** or **optimizing** performance on a specific task

# Introduction I

We are all here to **learn** about Machine **Learning**.

- What is learning?
- It probably has something to do with **change** or **mastering** or **optimizing** performance on a specific task
- Machine learning typically involves to build models (like seen last time), and learning boils down to **finding model parameters that optimize some measure of performance**





# Introduction I

We are all here to **learn** about Machine **Learning**.

- What is learning?
- It probably has something to do with **change** or **mastering** or **optimizing** performance on a specific task
- Machine learning typically involves to build models (like seen last time), and learning boils down to **finding model parameters that optimize some measure of performance**

**But, how do we know what is optimal?**



# Finding Optimal Points I

Finding the **parameters** that optimize a **target**

Ex1: Estimate the **study time** which leads to the **best grade** in COMP90049.

Ex2: Find the **shoe price** which leads to **maximum profit** of our shoe shop.

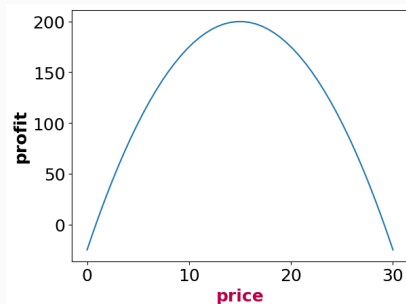


# Finding Optimal Points I

Finding the **parameters** that optimize a **target**

Ex1: Estimate the **study time** which leads to the **best grade** in COMP90049.

Ex2: Find the **shoe price** which leads to **maximum profit** of our shoe shop.



# Finding Optimal Points I

Finding the **parameters** that optimize a **target**

Ex1: Estimate the **study time** which leads to the **best grade** in COMP90049.

Ex2: Find the **shoe price** which leads to **maximum profit** of our shoe shop.

Ex3: Predicting **housing prices** from a **weighted** combination of house age and house location

Ex4: Find the **parameters  $\theta$**  of a spam classifier which lead to the **lowest error**

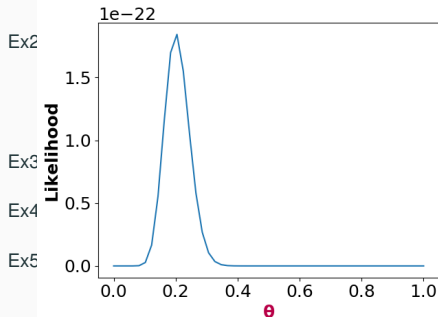
Ex5: Find the **parameters  $\theta$**  of a spam classifier which lead to the **highest data log likelihood**



# Finding Optimal Points I

Finding the **parameters** that optimize a **target**

Ex1: Estimate the **study time** which leads to the **best grade** in COMP90049.



**profit** of our shoe shop.

mbination of house age and house location

ch lead to the **lowest error**

ch lead to the **highest data log likelihood**

# Objective functions

Find parameter values  $\theta$  that maximize (or minimize) the value of a function  $f(\theta)$

- we want to find the **extreme** points of the **objective function**. Depending on our **target**, this could be
- ...the **maximum**  
E.g., the **maximum** profit of our shoe shop  
E.g., the **largest** possible (log) likelihood of the data

$$\hat{\theta} = \operatorname{argmax}_{\theta} f(\theta)$$

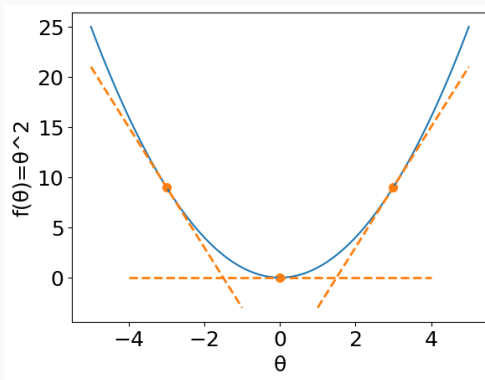
- ...or the **minimum** (in which case we often call  $f$  a **loss function**)  
E.g., the **smallest** possible classification error

$$\hat{\theta} = \operatorname{argmin}_{\theta} f(\theta)$$



# Finding extreme points of a function

- At its **extreme point**,  $f(\theta)$  is 'flat': its **slope** is equal to **zero**.
- We can measure the **slope** of a function at any point through its first **derivative** at that point
- The derivative measures the change of the output  $f(\theta)$  given a change in the input  $\theta$
- We write the derivative of  $f$  with respect to  $\theta$  as  $\frac{\partial f}{\partial \theta}$

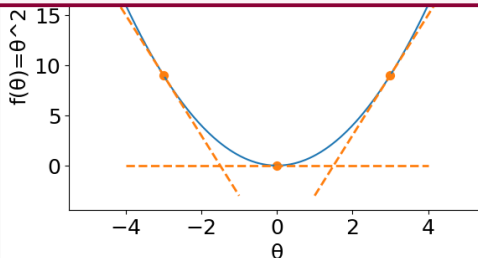


# Finding extreme points of a function

- At its **extreme point**,  $f(\theta)$  is 'flat': its **slope** is equal to **zero**.
- We can measure the **slope** of a function at any point through its first **derivative** at that point
- The derivative measures the change of the output  $f(\theta)$  given a change in the input  $\theta$
- We write the derivative of  $f$  with respect to  $\theta$  as  $\frac{\partial f}{\partial \theta}$

In order to find the parameters that maximize / minimize an objective function, we find those inputs at which the derivative of the function evaluates to zero.

That's it!





# Finding a Minimum / Maximum

## Example

- For our function, with a single 1-dimensional parameter  $\theta$

$$f(\theta) = \theta^2$$

Take the derivative

$$\frac{\partial f}{\partial \theta} = 2\theta$$

We want to find the point where this derivative is zero, so

$$2\theta = 0$$

and solve for  $\theta$

$$\theta = 0$$

# Finding a Minimum / Maximum

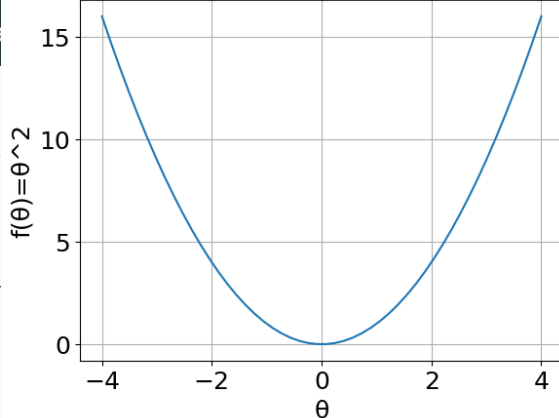
## Example

- For our function, with a

Take the derivative

We want to find the point

and solve for  $\theta$



**The global minimum of  $f(\theta) = \theta^2$  occurs at the point where  $\theta=0$ .**

# Recipe for finding Minima / Maxima

1. Define your function of interest  $f(\theta)$  (e.g., data log likelihood)
2. Compute its first derivative with respect to its input  $\theta$
3. Set the derivative equal to zero
4. Solve for  $\theta$

# Recipe for finding Minima / Maxima

1. Define your function of interest  $f(\theta)$  (e.g., data log likelihood)
2. Compute its first derivative with respect to its input  $\theta$
3. Set the derivative equal to zero
4. Solve for  $\theta$

Let's do this for a more interesting problem. Recall our binomial spam model from the last lecture?



# Maximum Likelihood Optimization of the Binomial Spam Model

## 1. Problem setup / identifying the function of interest

- Consider a data set of emails, where each email is an observation  $x$  which is labeled either as spam or not spam
- We have  $N$  observations, each with 2 possible outcomes. The data consequently follows a **binomial distribution** and the data likelihood is

$$\mathcal{L}(\theta) = p(X; N, \theta) = \frac{N!}{x!(N-x)!} \theta^x (1-\theta)^{N-x}$$

- So the parameter  $\theta = P(\text{spam})$



# Maximum Likelihood Optimization of the Binomial Spam Model

## 1. Problem setup / identifying the function of interest

- Consider a data set of emails, where each email is an observation  $x$  which is labeled either as spam or not spam
- We have  $N$  observations, each with 2 possible outcomes. The data consequently follows a **binomial distribution** and the data likelihood is

$$\mathcal{L}(\theta) = p(X; N, \theta) = \frac{N!}{x!(N-x)!} \theta^x (1-\theta)^{N-x}$$

- So the parameter  $\theta = P(\text{spam})$
- Imagine we have a data set of 100 emails: 20 are spam (and consequently 80 emails are not spam).
- In the last lecture, we agreed intuitively that  $P(\text{spam}) = \theta = 20/100 = \frac{x}{N}$ .
- We will now derive the same result mathematically, and show that  $\theta = \frac{x}{N}$  is the  $\hat{\theta}$  that maximizes the likelihood of the observed data



# Maximum Likelihood Optimization of the Binomial Spam Model

## 2. Computing its first derivative

$$\begin{aligned}\mathcal{L}(\theta) = p(X; N, \theta) &= \frac{N!}{x!(N-x)!} \theta^x (1-\theta)^{N-x} \\ &\approx \theta^x (1-\theta)^{N-x}\end{aligned}$$

Move to log space (makes our life easier)

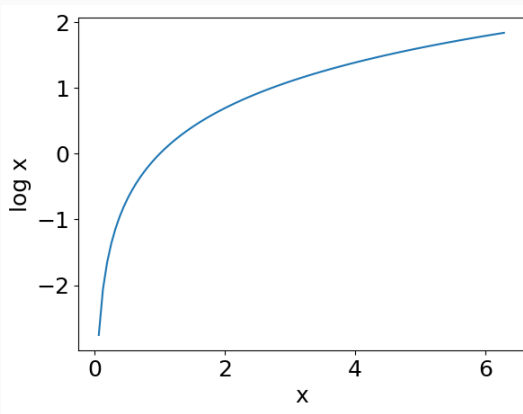
$$\log \mathcal{L}(\theta) = x \log \theta + (N-x) \log(1-\theta)$$



# Maximum Likelihood Optimization of the Binomial Spam Model

(Log transformation aside)

- Log is a monotonic transformation: The same  $\theta$  will maximize both  $p(x, y)$  and  $\log p(x, y)$
- Log values are less extreme (cf.  $x$  scale vs  $y$  scale)
- Products become sums (avoid under/overflow)





# Maximum Likelihood Optimization of the Binomial Spam Model

## 2. Computing its first derivative

$$\mathcal{L}(\theta) = p(X; N, \theta) = \frac{N!}{x!(N-x)!} \theta^x (1-\theta)^{N-x} \approx \theta^x (1-\theta)^{N-x}$$

Move to log space (makes or life easier)

$$\log \mathcal{L}(\theta) = x \log \theta + (N-x) \log(1-\theta)$$

# Maximum Likelihood Optimization of the Binomial Spam Model

## 2. Computing its first derivative

$$\mathcal{L}(\theta) = p(X; N, \theta) = \frac{N!}{x!(N-x)!} \theta^x (1-\theta)^{N-x} \approx \theta^x (1-\theta)^{N-x}$$

Move to log space (makes or life easier)

$$\log \mathcal{L}(\theta) = x \log \theta + (N-x) \log(1-\theta)$$

Take the derivative of  $\mathcal{L}$  wrt the parameters  $\theta$

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{x}{\theta} - \frac{N-x}{1-\theta}$$



# Maximum Likelihood Optimization of the Binomial Spam Model

## 3. Set the derivative to zero

$$0 = \frac{x}{\theta} - \frac{N-x}{1-\theta}$$

## 4. Solve for $\theta$

$$\frac{x}{\theta} = \frac{N-x}{1-\theta} \quad [\times(1-\theta)]$$

$$\frac{x \times (1-\theta)}{\theta} = N-x \quad [\times \frac{1}{x}]$$

$$\frac{1-\theta}{\theta} = \frac{N-x}{x} \quad [\text{rearrange}]$$

$$\frac{1}{\theta} - 1 = \frac{N}{x} - 1 \quad [+1]$$

$$\frac{1}{\theta} = \frac{N}{x} \quad [\text{flip}]$$

$$\hat{\theta} = \frac{x}{N}$$

Which corresponds to our estimate of  $\frac{x}{N} = \frac{20}{100} = 0.2$  for our spam classification problem.

# Possible Complications

Can you think of scenarios where this approach breaks down?



# Possible Complications

## Can you think of scenarios where this approach breaks down?

- Our loss function is not differentiable
- It is mathematically impossible to set the derivative to 0 and solve for the parameters  $\theta$ . “No closed-form solution”.
- Our function has multiple ‘extreme points’ where the slope equals zero. Which one is the correct one?

**to be continued...**

# Summary

- What is optimization?
- Objective function / loss function
- Gradients, derivatives, and slopes

**Next: Naive Bayes**



## **Solution subject to Constraints**

---

# Constrained Optimization

Finding the **parameters** that optimize a **target** subject to one or more constraints.

- Buy **3 pieces of fruit** which lead to the best **nutritional value**. But we only have a budget of **3\$**.
- I want to estimate the **parameters of a Categorical distribution** to maximize the **data log likelihood** and I know that **the parameters must sum to 1**.





# Constrained Optimization

It often happens that the parameters we want to learn have to obey constraints

$$\begin{aligned} & \underset{\theta}{\operatorname{argmin}} f(\theta) \\ & \text{subject to } g(\theta) = 0, \end{aligned}$$

- ideally, we would like to incorporate such constraints and still be able to follow the general recipe for optimization discussed before
- **Lagrangians** allow us to do exactly that in the case of **equality constraints** (there are also boundary constraints, which we won't cover)
- we combine our target functions with (sets of) constraints multiplied through **Lagrange multipliers**  $\lambda$

$$\mathcal{L}(\theta, \lambda) = f(\theta) - \lambda g(\theta)$$

- proceed as before: derivative, set to zero, solve for  $\theta$



## Example

- Find an optimal parameter vector  $\theta$  such that each all  $\theta_i$  sum up to a certain constant  $b$ .
- Formalize the constraint:

$$\sum_i \theta_i = b$$

- Set the constraint to zero

$$0 = \sum_i \theta_i - b = -b + \sum_i \theta_i$$

- set the constraint and write the Lagrangian

$$g_c(\theta) = -b + \sum_i \theta_i$$

$$\begin{aligned}\mathcal{L}(\theta, \lambda) &= f(\theta) - \lambda g_c(\theta) \\ &= f(\theta) - \lambda(-b + \sum_i \theta_i)\end{aligned}$$

- proceed as before: derivative, set to zero, solve for  $\theta$

Jacob Eisenstein. Introduction to Natural Language Processing, Appendix B (up to B.1)

Dan Klein. Lagrange Multipliers without Permanent Scarring.

<https://people.eecs.berkeley.edu/~klein/papers/lagrange-multipliers.pdf> . Sections 1, 2 (up to 2.4), 3.1, 3.5

